Print program for graphic Image files

Sep 87

The program allows a graphic Image file (e.g. a Desktop Publisher file of any sie, or a Page Designer 2 file or a 'set' of Front Page files, or a QL Screen Dump of any size) to be printed to any dot matrix printer.

The resulting printout may be scaled up or down in either dimension (or left unscaled) and rotated through 90', or left with its original orientation.

The program requires a memory expansion - its operational size is about 150k depending on the size of the file to be printed - and it uses the same Toolkit as Desktop Publisher; namely the TURBO Toolkit. (Note: if the experimental "C" version of the program is used, it does not require any toolkit).

Preparation

Before the program may be run, your printer's characteristics must have been entered on to the printer customisation file, the supplied version of which is called 'driver_dat'.

Most commonly available printers are already defined on this file, but if your's is not, the method for entering the required details is described at the end of these notes. PDQL would be interested in hearing from people who find they have to update or add to the file - so that the distributed version of the file can be amended for the benefit of others.

Also, before the program is started, the TURBO Toolkit extensions should be available in QL memory (but see note above).

Start up

The program is started in the usual way - using EXEC, EX, EXECUTE or whatever method you use for an "executable" multitasking program. e.g. exec flp1_graFix_bin

If you do not have (Turbo) toolkit loaded you may use the boot file provided instead.

Note for Taskmaster users: The program MUST NOT be loaded with the "shared code" option.

Note for QRAM users: The program MUST NOT be "hotkeyed". Select it from the 'Files' menu.

When the program has started, the screen is cleared, headings are displayed and the first 'prompt' of the start-up dialogue appears.

Dialogue

The program needs the user to supply some details about three things:

The input Image file
The printer to be used

The scaling to be used

The input file

The first few questions are about the Image file:

the name of the file to be printed,
the origin of the file,
various details depending on the file's origin
whether 90' rotation is required
whether ink and paper are to be reversed

* 'Enter device and name of file: '

Sample Responses: flp2_page_one_v4
ram1_schematic

The program expects the device/file name to be at least 6 characters long. Otherwise, it squeaks and repeats the question.

* 'File Source (DTP, PD2, FP, Scr, Again): '

Sample responses:
D - to indicate Digital's DTP
P - to indicate Sector's Page Designer 2
F - to indicate Gap's Front Page
S - to indicate a QL screen dump
A - to respecify the file name

Some supplementary questions may be asked, depending on the source of the file - see notes at end of section.

After the questions, the image dimensions are displayed, for example:

'DTP page width: ' wwww ' Height: ' hhhh

where wwww and hhhh are the selected (or default) sizes.

* 'Rotate thru 90' (n/y): '

Depending on your output requirements, it may be convenient to swap the "x" and "y" axes of the image.

* 'Invert black and white (y/n): '

In its screen image file, DTP uses 'white' as "ink" and 'black' as "paper". This will usually result in tremendous amounts of ink at print time, since it is likely that most of your image will be white. If you reply 'y' (the default) to this question, black in your screen image will be treated as 'ink' and white will be treated as 'paper'.

Supplementary questions

* Desktop Publisher

the DTP page size used,
whether the page is 'vertical' or 'sideways'
(DTP terminology)

>> 'DTP page width (960,480,800,1280): '

Sample Responses: ENTER - defaults to 960
1280

If you have taken special action in DTP to adjust the page size, then the revised size should be declared - otherwise, just press ENTER

>> 'DTP page height (800,400): '

Sample Responses: ENTER - defaults to 800

See above.

>> 'Is DTP page vertical or sideways (v/s): '

The default response - if you just press ENTER - is 'v'. In the same way as the page size, you will have selected 'sideways' as non-standard orientation when creating the page in DTP.

* Page Designer 2

No additional questions. Image dimensions are determined by examining the file. A 'compressed' file image may not be processed. The program will squeak and reject the file.

* Front Page

No additional questions. Image dimensions are fixed. It is expected that 4 files will be available on the indicated device, and that the name entered is the "root" name. For example, if the response to the file name prompt was "flp2_draft", then the files "flp2_draft1", "flp2_draft2", "flp2_draft3" and "flp2_draft4" will be expected to be present.

* Screen Dump

The usual width of a screen dump is 512 pixels. If appropriate action has been taken, the dump width may be less. This will be known to the creator of the file (?? you ??). The following question has a default response of 512.

'Dump width (<= 512): '

The image depth is computed from the file size.

The printer

The program will output to any suitable device. This may be the more obvious printer ports - ser1, ser2, par - or may be a file on disc or ramdisc etc. Some images which involve scaling and inversion and so forth may take up to 20 minutes to process (but considerably less if the "C" version is used), and if several copies are required, it may be quicker to send the output to a file and subsequently copy the file to the print device as many times as is required.

* 'Enter print device (default "ser1"): '

Sample Responses: ENTER - default ser1
flp2_grafdump

* 'Passes: '

```
            Sample Responses:  ENTER - default 1 pass
                               2 - two passes
```

Each 'pass' of the print head may be repeated, if necessary with a suitable vertical offset - assuming the appropriate parameters have been defined for the printer to be used. For printers with low vertical resolution, this technique may result in a more 'solid' image being produced.

Whether or not printing is actually going to take place, it is necessary for the program to know which type of printer will ultimately be used. The printer must be fully described to the program, and the details are required to be on the 'driver' file.

* 'Driver filename (default "driver_dat"): '

```
            Sample Responses:  ENTER - default filename
                               flp2_myprinter_driver
```

The format of the parameters on the driver file is described at the end of these notes.

For each set of parameters on the driver file, the program displays the printer name and an associated 'id number' from 1 upwards, and when all names have been listed the following message is displayed:

* 'Enter id number of required printer type: '

If the name of the printer that you want is on view, then key in the id number. If it is not, then just press ENTER. If there are more names on file, these will then be displayed, otherwise the program will HALT with the message:

'Update Graphics Driver file with required parameters and restart'

When the required printer has been identified, the selection dialogue is cleared from the screen, and replaced by the message:

'Selected printer: ' printer_name

Scaling
=======

The program is already aware of the dimensions and 'dot density' of the QL screen - so it knows what it takes to make a circle look like a circle on the QL.

Having been told what printer is in use, it also knows what it takes to make a circle look like a circle on that printer. So, the program does the necessary calculations to work out in what way the incoming data has to be transformed (rescaled) to preserve the "aspect ratio" so that the image on screen can be reproduced on paper.

The results of these calculations are displayed, as follows:

```
                                             Preserving aspect ratio
 The resulting print image will be as follows:  Width      Height
                                    (inches):    wwww       hhhh
      The scaling factors are (input : output):  Horiz.     Vertical
                                                  a:b        c:d
```

The program has worked out what is the minimum adjustment to preserve the aspect ratio, and also at that adjustment what the image size on the paper will be. One of a:b or c:d will be 1:1, so this tells you what the vertical/horizontal scale ratio should be to preserve the aspect ratio. If the program computes a horizontal scaling that will result in a line longer than your printer can handle, the program will squeak and scale down both the figures accordingly.

The scaling factors might be displayed as:

```
1:1.032     1:1
```

This means that for every input pixel on a 'row', the program must generate 1.032 pixels, or more properly, the program will "invent" 32 pixels in the print output for every 1000 pixels in the image file row of pixels. No vertical adjustment will be made. Another case might be:

```
1:1          1.022:1
```

This would mean that each 'row' would be left unscaled, but that out of every 1022 input rows, only 1000 would get printed.

You are then provided with the chance to override these figures - if for example the printed image will be too small, you may want to scale up the image, or you may be prepared for the aspect ratio to be changed or whatever.

* 'Do you want to change the scaling (n/r/y): '

If the response is 'r', the program will restart the dialogue - perhaps you wish to use a different printer resolution, or decide that the image would "fit" better if rotated, or whatever.

Only if you reply 'y' will the next question appear:

* 'Enter scaling as numbers (horiz, vert): '

```
Sample Responses:  1.33 (ENTER) 1.777 (ENTER)
                   1 (ENTER)     .978 (ENTER)
```

The numbers you are required to enter are the OUTPUT scaling numbers, so you can assume that you are entering 'x' and 'y' in the implied:

```
1:x  1:y
```

If you want to increase the image in size, your number should be greater than 1. If you want to reduce in size, the number should be less than 1. If you change the horizontal scale, the program will check that the number of pixels implied does not exceed the maximum width of your printer. If it does, the program will squeak and ask for the numbers again.

After the new scaling has been entered and accepted, the program redisplays the scaling 'panel' showing the revised stats:

```
                                              Manually scaled
The resulting print image will be as follows:  Width     Height
                                   (inches):   wwww      hhhh
   The scaling factors are (input : output):  Horiz.    Vertical
                                               a:b        c:d
```

Interpolation

If either of the scaling figures is in excess of 1 - i.e. the image is to be scaled up vertically or horizontally, or both - the program will ask if interpolation is required.

What this means is that if the program "invents" pixels or rows, and these are left as "paper" then the resulting image could become sparse. With interpolation switched 'on', the program will have a look at the context of the invented pixel or row and make a determination as to whether ink or paper is more appropriate to preserve the image.

```
*       'Do you want horiz. interpolation (n/y): '
*       'Do you want vert. interpolation (n/y): '
```

Note also that if vertical scale reduction is in effect, the program attempts to preserve the image integrity. Pixel rows that are to be 'discarded' are examined and structurally significant information is extracted before the row is discarded. This information is incorporated on the next row to be printed.

Execution

That finishes the dialogue. The program goes on to load the whole of the Image file (it has to have it all if sideways images are to be processed) - if insufficient memory is available, the program will HALT displaying the message "Out of memory".

Processing then commences. Factors that influence the process time are - in decreasing significance:

(a) Sideways orientation of input file
(b) Horizontal scaling up
(c) Width of a pixel row (longer row = longer time)
(d) Vertical scaling up
(e) Horizontal interpolation
(f) Vertical interpolation
(g) Inversion of ink and paper

A file that scores badly on all of these points may take up to 20 minutes to print.

There are 13 (or 16) lines required for each printer (or printer mode). The first line is the printer name, followed by up to 15 parameter lines. Each parameter is distinguished by a 4 character identifier.

```
Name, Identifying name of printer
GPRE, Preamble for graphics - e.g. set linefeed height
GPOS, Postamble after graphics - e.g. reset lf height
GEOL, End of line code - single pass printing
GFST, End of line code - first pass of print **
GLST, End of line code - last pass of print **
GINT, End of line code - intermediate print passes **
GPFX, Graphics line prefix
GCTL, Length of graphics string controller (chars)
GCTT, Type of controller
              0 = characters
              1 = binary; m.s. byte first
              2 = binary; m.s. byte last
GHZD, Horiz. dots per inch
GVTD, Vert. dots per inch
GHZM, Max. Horiz dots per line
GFBH, First bit of byte is High pin(1) or Low pin(0)
GPIN, Number of pins in print head
GSPD, Transmission speed (BAUD)
```

** These parameters are mandatory only if multi-pass printing is to be used.

Some of the information required for the parameters contains some strange characters, as you will see from your printer manual.

Rather than have you searching around for some weird key combinations on the QL, we have set up the string conventions so that information in your printer manual may be entered directly into the driver file.

Parameter Strings

Characters in parameter strings may be expressed either as a letter or sequence of letters - which will be used exactly as they appear (e.g. UPPER CASE or lower case) - or as a decimal number representing the ASCII code for the (single) character to be used.

Combinations of letters and decimal numbers are allowed. A comma must be used as a separator. Leading spaces in front of each element of the string are ignored. A decimal number must be prefixed by ₧. If the ₧ character is required as a letter, it must occur as ₧₧. Neither a comma nor a semicolon character may used in a string of letters. If either is needed, instead use ₧44 or ₧59 respectively. Equally, a string may not BEGIN with a space - use ₧32 if the requirement arises.

So, for example, to set up a string as the Graphics preamble

```
ESC "E" ESC "T14" ESC ";" ESC " AB,"
```

the parameter line would look like:

```
;GPRE, ₧27, E, ₧27, T14, ₧27, ₧59, ₧27, ₧32, AB, ₧44
```

An example parameter set

```
;Name,CITOH M7500 - Double Density
;GPRE,₧27,E,₧27,T14
. select 12 cpi; set lf to 14/144 of an inch
;GPOS,₧27,A
. reset to 6 lines per inch
;GEOL,₧10,₧13
. LF CR
;GPFX,₧27,S
. ESC S in front of each graphics line (double den.)
;GCTL,4
. 4 byte string controller
;GCTT,0
. controller is ASCII characters
;GHZD,120
. printing at 120 horiz dots per inch
;GVTD,82
. printing at 82 vert. dots per inch
;GHZM,960
. platten width 8 inches => 960 dots max
;GFBH,0
. leading bit of graphics byte is low pin, not high pin
;GPIN,8
. 8 pins fired in print head
;GSPD,4800
. baud rate 4800
```

Another parameter set - Epson standard

```
;Name, Star LC-10 - Double Den
;GPRE, ₧27, 1, ₧13
.  set linefeed at 7/72", do carriage return
;GPOS, ₧27, 2, ₧13
.  set linefeed at 1/6", do carriage return
;GEOL, ₧13, ₧10
.  end of line code is CR, LF - single pass printing
;GFST, ₧13, ₧27, J, ₧1
.  set CR and 1/216" linefeed after 1st pass
;GLST, ₧13, ₧27, J, ₧20
.  set CR and 20/216" linefeed after last pass
;GINT, ₧13
.  set CR and no linefeed on intermediate passes
;GPFX, ₧27, L
.  double density prefix is ESC, L
;GCTL, 2
.  controller length is 2 bytes
;GCTT, 2
.  controller type is binary with m.s. byte 2nd
;GHZD, 120
.  double density ==>  120 dpi
;GVTD, 82
.  8 pins per 7/72" ==> 82 dots per inch vertical
;GHZM, 960
.  platten width 8", times 120 dpi
;GFBH, 1
.  first bit of byte maps to 'top' pin in print head
;GPIN, 8
.  8 usable pins in print head
;GOFF, 0
.  Byte offset not relevant
;GSPD, 9600
.  Standard baud rate
```