Universal Printer Driver Utility

Getting Started

An example driver is included which allows virtually all of the QL character set to be printed correctly on an Epson compatible dot matrix printer. To install this example ready for use:

1) Load the printer driver code

This only needs to be done once when starting up the machine (or at a later time).
With TKII installed, type

```
    LRESPR flp1_Driver
```

(or whichever device/drive is appropriate). Without TKII, you must reserve space, load, and call the code with:

```
    Base=RESPR(1188):LBYTES flp1_Driver,Base:CALL Base
```

The printer driver is now in place and will process any data sent to either serial port (SER1,SER2) through channels opened AFTER the driver is loaded. Incoming data is not affected.

2) Load the translate information

The translate information tells the driver how to process the data sent to the serial ports. Load the example to try the facilities out. Again, with TKII type

```
    LRESPR flp1_Trans_Data
```

otherwise use

```
    Base=RESPR(1124):LBYTES flp1_Trans_Data,Base:CALL Base
```

The length of the demonstration translate information is 1124 bytes. When you created your own translates, the length will be given and you will need to substitute this number into the RESPR statement.

Should you attempt to load translate information before doing 1) above, the error 'Not Implemented' will be returned.

The code loaded includes a short piece of machine code before the data to link in the information. The precise details of this are not very exciting and thus relegated to an appendix.

Try printing the non-ASCII characters to the printer; codes 127-191.

```
    OPEN ₧4,ser1                                  (ser2 if you like)
    FOR F=127 to 191:PRINT ₧4;CHR$(F),F
```

Most of these should print correctly. For the moment avoid the curly brackets () (codes 123 & 125). These have been assigned a special function (which can be assigned to any two characters or none at all).

The curly brackets allow you to include a file in your output. If your printer supports screen-dumps then try

```
    PRINT ₧4;'(flp1_Mandelbrot)'
```

and a picture should appear. The file given between the brackets (note: device MUST be included in this name) will be sent to the serial port. This allows inclusion of standard letter headings in Quill documents, and even pictures in Quill documents. Including (flp1_Mandelbrot) in a Quill document will cause the picture to be inserted into the text when it is printed. Note that 1) the (filename) should be to the far left hand side of the document (move with the MARGINS command) and 2) Quill will take no account of the length of the picture in counting lines on the page. Use the PAGE BREAK command to force new pages if necessary.

To include your own pictures, you should use a screendump utility to dump the picture to a file; with TKII,

```
SDP_DEV flp1_Pic
```

will send the screendump to the file 'Pic' on flp1_

To include your own letter headings/endings, create them with Quill and use the 'print document to file' factility (remembering that Quill will add _lis to the name)

If your printer allows downloading of weird and wonderful character sets, (typically with an ESC & NUL sequence), then you can include a file containing the data at the beginning of your letter. (Note: the standard Quill driver resets the printer at the beginning of each _lis file or letter. To use your own character sets it may be necessary to remove this from the Quill printer driver, under 'preamble code', typically ESC >

The example supplied acts the same on data regardless of wheather it is sent to SER1 or SER2. When you set up your own translate information, using the Config program supplied, you can effect SER1 only, SER2 only, both the same, or each differently.

The Config Program

The configuration program is started with LRUN flp1_Config. This program simply loads some code into the machine then executes the main program. If you wish to use this program from a device other than flp1_ just edit the Basic;

```
10 Base=RESPR(5666)
20 LBYTES flp1_RG_TK,Base
30 LBYTES flp1_TB_TK,Base+132
40 CALL Base:CALL Base+132
50 EXEC_W flp1_Config_Task
```

This program is intentionally written to 'feel familiar' if you have used the Psion 'Install' program to set up the usual Quill printer driver. Once you have selected the default device, the top small window will display options and instructions, the main window will show the bulk of the information, and the small bottom window is where what you type in will appear, together with any error messages. The help option uses its own window in the centre of the screen. On some occasions when the program requires input, it will offer a default which will appear in the lower window. This may be accepted by pressing enter, or deleted / added to / edited in the normal manner.

A set of information may be loaded with F1, the demonstration file is called flp1_QL_SET (Note: the file flp1_Trans_Data cannot be edited, in the same way as the Quill printer_dat file cannot be edited) Selecting edit (F3) should present a familiar layout. The arrow on the left points to the 'current character'. Use the up/down arrows to move around, use <CTRL> up/down to move a bit more quickly. Next to the arrow is the actual character; first the ASCII mnemonic (if any - eg ESC, LF, CR), the the character itself (if it is printable, ie 32-255), then the code of the character in decimal and hex (preceded by a $).

To the right of this are two pieces of information; the translate sequence for this character, if any (ie the sequence of characters sent IN PLACE of this character) in white, and a comment field in black. The comment can contain text to tell you what the translate actually does, as this is frequently not obvious unless you look up all the control codes. The translate sequence can contain a list of characters to send. These can be written as decimal codes, hex (preceeded by a $), strings in 'quotes' or "quotes" (Note: unlike the Psion program, many characters may be in one string, and the closing quote is necessary), or ASCII mnemonics (in lower or upper case). Each item in the list should be separated by a comma or a space. Press F1 to edit the translate sequence for the current character. Only valid sequences will be accepted - if an error is detected, the cursor will appear near to the mistake. The translate sequence may go off the edge of the screen - this is allowed. Press F2 to edit the comment - this will be truncated to fit onto the screen.

F3 is used to set the 'include file' characters. These may be left null, or any two characters used (apart from ASCII code 0, NUL). Pressing F3 again will set them to the curly brackets.

It is recomended that, with the possible exceptions of LF (linefeed) code 10, CR (carriage return) code 13, and FF (formfeed) code 12, the control characters 0-31 should not be translated. But feel free to experiment...!

Quit edit by pressing ESC. Remember to save this with F2 if you are likely to want to edit it in the future - better still, save it anyway.

Install this set of translate sequences with F4. You will be asked which port to install it for - SER1, SER2, or both. If you only want ONE port affected, or BOTH with the SAME sequences, then you can use F5 'Save Install' immediately.

If you want a different set of translate sequences for the other port, now is the time to set these up. Load a new file or edit the current one, and select F4 again followed by the OTHER port.

Now select F5 which will save the data onto the device of your choice under the name Trans_Data.

Appendix - Details of the Trans_Data file

You will not need to read this unless you really dislike the Config program so much that you wish to write your own!

The first part of the file is machine code to link in the data to the printer driver program. This code is 110 bytes long, and may be obtained by taking the first 110 bytes of a Trans_Data file, or by typing in the assembler listing below (you'll have to look up the QL mnemonics yourself though!). The first two bytes are a short branch, the next six contain information as follows: a long word equal to the length of the SER1 translate information (zero if none) followed by a byte which can be -1,0, or 1

-1 if SER2 is to use the same translate information as SER1
0 if SER2 is to be unaffected
+1 if SER2 has its own translate information

```
* Universal Printer Driver, Link code      R Goodwin 1988

START      BRA.S      START2
SER1_LEN   DC.L       0
SER2_FLG   DC.W       0

START2     MOVEQ      #MT.INF,D0
           TRAP       #1
           LEA        SV.DRLST(A0),A0
LOOP       MOVE.L     (A0),A0
           CLR.L      D0
           CMPA.L     D0,A0
           BEQ.S      EXIT
           MOVE.L     -12(A0),D0
           CMPI.L     #$4AFB0101,D0
           BNE.S      LOOP
FOUND      LEA        DATA,A1
           LEA        SER1_LEN,A2
           MOVE.L     (A2),D0
           TST.L      D0
           BNE.S      IS_SER1
           MOVE.L     D0,-8(A0)
           BRA.S      DO_SER2
IS_SER1    MOVE.L     A1,D1
           ADDQ       #2,D1      Skip file ON/OFF characters
           MOVE.L     D1,-8(A0)
DO_SER2    CLR.L      D0
           MOVE.B     4(A2),D0
           TST.B      D0
           BNE.S      IS_SER2
           MOVE.L     D0,-4(A0)
           RTS
IS_SER2    SUBQ       #1,D0
           TST.B      D0
           BNE.S      SAME_SER1
           ADDA.L     (A2),A1
           ADDQ       #2,A1      Skip file ON/OFF characters
           MOVE.L     A1,-4(A0)
RET        CLR.L      D0
           RTS
SAME_SER1  MOVE.L     -8(A0),-4(A0)
           BRA.S      RET

EXIT       MOVEQ      #ERR.NI,D0  !Printer driver not loaded!
           RTS
```

```
DATA        DC.W        0               Data starts here
                                        OVERWRITING this word

            END
```

After this code comes the information for SER1 followed directly by the information for SER2. The format of this information is as follows;

```
    Byte - file include 'ON' character, or zero for none
    Byte - file include 'OFF' chraracter, or zero for none
Index;
    256 x Word - offset (from start of Index) for the
                 translate sequence for code 0, then code 1
                 etc, or zero if there is no sequence
Information;
    Byte - length of translate sequence for this chr (max
           length 80)
    n x byte - translate sequence
```

This last part is repeated until all the defined translate sequences have been included. If this entire section has an odd number of bytes, it must be padded with an extra '0' byte at the end.