Printing Routine for Epson Printer#

General Information

Author: Leo G. Laporte
Language: ACTION!
Compiler/Interpreter: ACTION!
Published: 1983

EPSON/ATASCII PRINT FORMATTER

Prints listed BASIC programs and text using Epson bit mode graphics to print non-ASCII characters

With suitable changes to the Epson specific variables immediately below, this program will work with a number of other graphic printers.

```
  ;+++++++++++++++++++++++++++++++++
; EPSON/ATASCII PRINT FORMATTER
; Prints listed BASIC programs and
; text using Epson bit mode graphics
; to print non-ASCII characters
;
; With suitable changes to the Epson
; specific variables immediately
; below, this program will work with
; a number of other graphic printers.
;
; (c)1983 Leo G. Laporte
;           BOX 21248
;           San Jose, CA 95151
;           CIS PPN # 70215,1022
; Placed in public domain 12/8/83.
;+++++++++++++++++++++++++++++++++

BYTE  rts = [$60],  ; OSA+ bug fix
        bank = $D500  ; Atari DOS bug fix


MODULE

DEFINE TRUE="1",
         FALSE="0",
         BOOL="BYTE",

         KEY = "0",
         FILE = "1",
         EPSON = "2",

         MAXLINE = "55" ; max # of lines per page

; Epson specific stuff

CHAR ARRAY grmode = [4 27 75 8 0],
              ; initializes bit-mode graphics (ESC K 8 0)
              ; and tells printer that eight graphic
              ; data bytes will follow.

              italics_on = [2 27 '4], ; if you have an older Epson w/o italics
```

```
             italics_off = [2 27 '5] ; change these strings to another suitable font

CHAR         formfeed = [12]

;-----------------------------------
; PROCEDURE DECLARATIONS
;-----------------------------------

PROC grprint(CHAR chr)

; does a graphic print of non-ASCII
; characters

  BYTE ARRAY mask =[128 64 32 16 8 4 2 1], ; bit values D7 to D0
                CHARSET = $E000, ; location of character set in ROM
                grdata(8) ; character data array

  BYTE offset, ; current character data byte
         bit,    ; current bit in byte
         byt      ; graphic data byte

  BOOL bit_set, ; is bit set? flag
         inv_flag ; inverse char?

  CARD charloc ; location of character data

  ; check for inverse character

  IF (chr & 128) THEN
     inv_flag = TRUE
     chr ==& 127      ; strip off inverse bit
  ELSE
     inv_flag = FALSE
  FI

  ; find character data in ROM

  IF chr < 32 THEN
      charloc = (chr + 64) * 8
  ELSEIF chr > 31 AND chr < 96 THEN
      charloc = (chr - 32) * 8
  ELSE
      charloc = chr * 8
  FI

  ; rotate char data for Epson

  Zero(grdata, 8) ; clear character graphics data
  FOR offset = 0 TO 7 ; step through char data
     DO
        FOR bit = 0 TO 7
          DO
             bit_set =  CHARSET(charloc + offset) & mask(bit)
                IF inv_flag THEN
                  IF bit_set = FALSE THEN
                     grdata(bit) ==+  mask(offset)
                  FI
                ELSEIF bit_set THEN
                  grdata(bit) ==+  mask(offset)
```

```
                FI
           OD
      OD

  ; dump character data

  PrintD(EPSON, grmode)
  FOR byt = 0 TO 7
     DO
        IF grdata(byt) = 155 THEN ;prevent sending CR and thereby
          grdata(byt) = 151      ;cancelling graphics mode (only
        FI                              ; occurs during printing of inverse A)
        PutD(EPSON, grdata(byt))
     OD

RETURN

;-----------------------------------

PROC feed(BYTE lines)

; feeds "lines" lines

BYTE i

    FOR i = 1 TO lines
      DO
         PutDE(EPSON)
      OD

RETURN

;-------------------------------------

PROC indent(BYTE col)

; tabs to col

BYTE space = [32], i

FOR i = 1 TO col
    DO  PutD(EPSON, space)  OD

RETURN

;-------------------------------------

BYTE FUNC PrintTEXT(CHAR ARRAY line)

; prints TEXT input line

CHAR eol = [155],
      chr

BYTE cnt, col

CARD linecnt ; number of lines output

cnt = 1 ; current character in line
```

```
col  = 0  ; current printer column
linecnt = 0 ; lines printed

  chr = line(cnt)
  IF chr = eol THEN
      PutDE(EPSON)
      RETURN (1)
  FI

  WHILE chr <> eol
  DO

     IF ; printable character
     (chr > 31 AND chr < 123 AND chr <> 96) THEN
         PutD(EPSON, chr)
         col ==+ 1

     ELSE
         grprint(chr)
         col ==+ 2
     FI

     IF (col > 80) THEN
        PutDE(EPSON)
        linecnt ==+ 1
        col = 0
     FI

     cnt ==+ 1
     chr = line(cnt) ; get next char
  OD

PutDE(EPSON)
linecnt ==+ 1

RETURN (linecnt)

;-----------------------------------

BYTE FUNC PrintBASIC(CHAR ARRAY line)

; prints BASIC input line

CHAR space = [32],
      invsp = [160],
      colon = [58],
      semic = [59],
      eol   = [155],
      quote = [34],
      comma = [44],
      chr

BYTE cnt, col, tab

BOOL inquotes

CARD linecnt ; number of lines output

cnt = 1 ; current character in line
```

```
col  = 0  ; current printer column
linecnt = 0 ; lines printed
inquotes = FALSE

  chr = line(cnt)
  IF chr = eol THEN RETURN (0) FI

  ; drop leading spaces...

  WHILE chr = space
     DO
        cnt ==+ 1
        chr = line(cnt)
     OD

  ; print line number...

  WHILE (chr >= '0 AND chr <= '9)
     DO
        PutD(EPSON, chr)
        col ==+ 1
        cnt ==+ 1
        chr = line(cnt)
     OD

  ; output a space...

  IF chr = space THEN
     PutD(EPSON, chr)
     cnt ==+ 1
     chr = line(cnt)
  ELSE PutD(EPSON, space)
  FI

  col ==+ 1

  ; set tab...

  tab = col

  ; now print rest of line...

  WHILE chr <> eol
  DO
     IF chr = quote THEN
        IF inquotes THEN
          inquotes = FALSE
        ELSE
          inquotes = TRUE
        FI
     FI

     IF ; printable character
     (chr > 31 AND chr < 123 AND chr <> 96) THEN
         PutD(EPSON, chr)
         col ==+ 1

     ELSE
         grprint(chr)
```

```
        col ==+ 2
    FI

    ; should we break line?...

    IF
    (col > 65 AND  ; close to R margin
        (chr = space OR ; break line
         chr = invsp OR ; at a logical
         chr = comma OR ; spot if
         chr = semic))  ; possible
    OR
    (chr = colon AND inquotes = FALSE) ; separate BASIC commands
    OR
    (col > 80) ; unconditional line break

    ; yes...

    THEN
        PutDE(EPSON)
        linecnt ==+ 1
        indent(tab)
        col = tab

    ; no...

    FI

    cnt ==+ 1
    chr = line(cnt) ; get next char
  OD

feed(2) ; end of input line
linecnt ==+ 2

RETURN (linecnt)

;-------------------------------------

PROC main()

  CHAR ARRAY source(20),
               title1(75),
               title2(75),
               choice(10),
               line(255)

  BYTE consol = $D01F, ; start key
       invflg = $2B6,  ; inverse off
       shflok = $2BE,  ; shift lock
       crsinh = $2F0,  ; cursor off
       linecnt, linetot

  CARD page = [1] ; pages printed

  BOOL basic

  Bank = 0
```

```
Put(125) ; clear screen
Setcolor(2,12,2)
PutE()
PrintE("  EPSON/ATASCII PRETTY PRINTER")
PrintE("      (c)1983 Leo G. Laporte")
PutE()
PrintE("  File must be LISTed BASIC or TEXT")
PutE()
Print("Enter source file > ")

invflg = 0    ; inverse off
shflok = 64  ; caps lock
InputS(source)

PutE()
PrintE("Enter header line #1 (max 75 chars)")
Print(">")
InputMD(KEY, title1, 75)

PutE()
PrintE("Enter header line #2")
Print(">")
InputMD(KEY, title2, 75)

PutE()
Print("Is this a (B)ASIC or (T)ext file? ")

invflg = 0    ; inverse off
shflok = 64  ; caps lock
InputMD(KEY, choice, 10)

IF choice(1) = 'T THEN
   basic = FALSE
ELSE
   basic = TRUE
FI

Position(2,22)
crsinh = 1
Print("  PRESS -START- TO BEGIN PRINTING")
Position(2, 17)

consol = 8
WHILE consol <> 6  ; wait for start
   DO
      consol = 0
   OD

      Close(FILE) Close(EPSON)

      Open(FILE, source, 4)
      Open(EPSON, "P:", 8)

      PutDE(EPSON)
      PrintD(EPSON, italics_on)
      PrintDE(EPSON, title1)
      PrintDE(EPSON, title2)
      PrintD(EPSON,"Page ") PrintCDE(EPSON, page)
      PrintD(EPSON, italics_off)
```

```
        feed(2)
        linetot = 6

        InputMD(FILE, line, 255)
        WHILE EOF(FILE) = FALSE
          DO
             IF basic THEN
                linecnt = PrintBASIC(line)
             ELSE
                linecnt = PrintTEXT(line)
             FI

             linetot ==+ linecnt
             IF linetot >= MAXLINE THEN ; next page
                PutD(EPSON, formfeed)
                page ==+ 1
                PrintD(EPSON, italics_on)
                PrintDE(EPSON, title1)
                PrintDE(EPSON, title2)
                PrintD(EPSON,"Page ") PrintCDE(EPSON, page)
                PrintD(EPSON, italics_off)
                feed(2)
                linetot = 6
             FI
             Print(".")
             InputMD(FILE, line, 255)
          OD

  PutDE(EPSON)  ;flush buffer
  Close(FILE) Close(EPSON)

  crsinh = 0
  Graphics(0)
RETURN
```