



0000'

ASEG

.Z80

TITLE VDUPRNT.MAC

.COMMENT #

```
; VDUPRNT - a program to print the content
;MicroBee/Tandy DMP 200 processor/printer combin
; Written - D K Hains, 18 October 1984
; Amended - 23 Dec 84
;
```

```
; Notes on Technique:
;
```

```
;1. The program is intended to be used with
;memory above the screen RAM (note: 1K spare in
;directly with a CP/M (or other operating system
;routine was tested this way.
;
```

```
;2. See the printer manual for details on pr
;in graphics mode, the graphics dot density (hor
;column, and the addressing method for print dot
;another type of processor, the sections of code
;may be useful.
;
```

```
;3. Since both ASCII characters and graphics
;the printer must be set up for graphics, which
;and both types of data printed as a series of 1
;For example, for characters displayed as a matr
;this will mean about three passes over each lin
;of dots used to display the data/characters - 2
;last pass of 2 dots. For example:
;
```

```
;
; +-----+
; | 1 st byte | 1st pass - offset=0, offst
; | 2         |
; |         |
; | 7         |
; | 8         | 2nd pass - offset=7, offst
; | 9         |
; |         |
; | 15        | 3rd pass - offset=2, offst
; | 16th byte |
; +-----+
```

```
;4. The MicroBee computer this program is us
;using a 6545 crt controller chip and can thus d
;alpha-numeric data. The screen data is stored i
;character, where each set bit in a byte represe
;on the screen. The start address of the 16 byte
;pattern is 16*(the contents of a screen positio
;memory, ie F000H. To print the data as graphics
;each byte have to be assembled as a character b
;printer. Each pass over the data therefore need
;to start from, and the number of bytes from whi
;The variables describing these values are OFFST
;
```

```
;5. ASCII characters are defined in a ROM, a
```

```
;obtained by 'setting' a latch. Graphics charact
;stored in RAM, however the the method of obtain
;outlined in 4. applies.
;
;6.      The program can be easily adapted to to
;the BASIC start address (BSTRT) and/or the posi
;the interpreter (xxxx bytes from the start) by
;table entries. The size of the BASIC screen for
;Each character is 16 dots high x 8 dots wide.
;
;7.      For machines running CP/M 2.2, the boot
;routine that will enable the program to be used
;In this case, changing some of the equate table
;and re-assembling would allow this approach. Th
;on the MicroBee is 80 char/line x 24 lines, wit
;dots wide x 12 dots high.
;
;8.      Different screen formats can be handled
;definitions. To use the printer routines on dif
;systems (CP/M, BASIC in ROM etc), the location
;400 bytes) should be determined and the program
;
;9.      There are 10 locations ORG'd at the star
;store local variables, defined as NOP (or DEFS
;assembly. The definitions of these locations ar
;unused space is spare.
;
;10.     To Start - make a user call (BASIC) or C
;address. A 'jump' instruction leapfrogs the wor
;On completion the program will return to the ca
;
;11.     Main Register Usage:
;Unless modified specifically for a sub-routine
;follows -
;
;         HL = screen address           B = line
;         C  = column mask              D = char
;
```

```

;12. This program was written using Wordstar
;Microsoft MACRO (tm) assembler. For those witho
;be used by omitting the initial variable equate
;code that do not apply. The following program i
;otherwise the execution file would have zeroes
;and be somewhat large. Those with a hobby Z80 a
;it anywhere that is convenient.
;

```

```

.END OF COMMENTS #

```

```

;13. Equate Table:

```

```

FFFF      TRUE      EQU      -1
0000      FALSE     EQU      NOT TRUE
FFFF      CPM22     EQU      TRUE      ;running

8000      BSTRT     EQU      8000H      ;start a
A87F      INTERP    EQU      BSTRT+287FH ;1200 Bd
E04B      RS_OUT    EQU      0E04BH     ;into th
F000      SCRNST    EQU      0F000H     ;boot RO
;                                                ;in Mbee
;                                                ;memory
;
;      ORG      0100H      ;where t
;                                                ;the wor
;                                                ;see OFF
0102      OFFSET    EQU      0102H     ;memory
0103      OFFSTA    EQU      OFFSET+1   ;memory
0104      RESULT    EQU      OFFSET+2   ;memory
;                                                ;printed
;                                                ;content
;                                                ;Result
0108      CBEGIN    EQU      OFFSET+6   ;start a

;
IF      CPM22      ;CP/M v2
0018      NUMLIN    EQU      18H       ;24 line
0050      CHARLN    EQU      50H       ;80 cara
000B      SCNLN     EQU      0BH       ;12 scan
0000      OFSTA0    EQU      0H
0007      OFSET1    EQU      7H
0007      OFSTA1    EQU      7H
0005      OFSET2    EQU      5H       ;offset2
ELSE      ;BASIC
0018      NUMLIN    EQU      10H       ;number
0050      CHARLN    EQU      40H       ;number
000B      SCNLN     EQU      10H       ;16 scan
0000      OFSTA0    EQU      0H
0007      OFSET1    EQU      7H       ;0 to 6
0007      OFSTA1    EQU      7H
0005      OFSTA2    EQU      0EH       ;7 more
0005      OFSET2    EQU      2H       ;offset2
ENDIF

0100      18 0A     ENTRY:  JR      STRT      ;always

```

```

;workspa
0102          ;
                DEFS      10          ;work sp
                ;assembl

;
;Start Up.
;
010C  F5      STRT:  PUSH    AF          ;save th
010D  E5      PUSH    HL
010E  D5      PUSH    DE
010F  C5      PUSH    BC
0110  3E 0C   LD      A,0CH          ;form fe
0112  CD 015B CALL    PRINT
0115  3E 0D   LD      A,0DH          ;print h
0117  CD 015B CALL    PRINT
011A  3E 1B   LD      A,1BH          ;set up
011C  CD 015B CALL    PRINT
011F  3E 14   LD      A,14H
0121  CD 015B CALL    PRINT
0124  3E 12   LD      A,12H          ;select
0126  CD 015B CALL    PRINT
0129  3E 00   LD      A,OFSTA0        ;set up
012B  32 0103 LD      (OFFSTA),A
012E  3E 07   LD      A,OFSET1
0130  32 0102 LD      (OFFSET),A

;
;Screen Control Routine
;
0133  21 F000 LD      HL,SCRNST        ;commenc
0136  06 18   LD      B,NUMLIN        ;set lin
0138  E5      LCNT:  PUSH    HL          ;save st
0139  C5      PUSH    BC          ;save B
013A  06 50   LD      B,CHARLN        ;set cha
013C  CD 019A CCNT:  CALL    MULT        ;compute
013F  CD 01A7 CALL    CGEN            ;where w
0142  CD 01BE CALL    ASMBLE         ;assembl
0145  23      INC     HL          ;on to t
0146  10 F4   DJNZ   CCNT          ;do this

```

```

0148      C1                POP      BC                ;back to
0149      E1                POP      HL                ;back to
014A      CD 015F          CALL     EOLFLG            ;chck th
014D      FE 00            CP        0                ;is offs
014F      20 E7            JR        NZ,LCNT          ;no - ke
0151      16 00            LD        D,0                ;16 bit
0153      1E 50            LD        E,CHARLN          ;yes, ne
0155      19                ADD     HL,DE                ;HL now
0156      10 E0            RPT:    DJNZ   LCNT          ;loop if
0158      C3 01F8          JP        FIN                ;exit at
;
;Print Routine - all print is routed thru here,
;      only be made once, here.
;
015B      CD E04B          PRINT:  CALL     RS_OUT            ;boot RO
015E      C9                RET
;
;End of line Flag: ckeck & reset offset and offs
;
015F      3A 0103          EOLFLG: LD      A,(OFFSTA)        ;1st, ge
0162      FE 00            CP        00                ;if 1st
0164      28 0E            JR        Z,PASS2
IF CPM22
0166      CD 0182          CALL     SLINE
0169      3E 07            LD      A,OFSET1            ;set up
016B      32 0102          LD      (OFFSET),A
016E      3E 00            LD      A,OFSTA0
0170      32 0103          LD      (OFFSTA),A
0173      C9                RET
0174      CD 0182          PASS2:  CALL     SLINE            ;set up
0177      3E 05            LD      A,OFSET2
0179      32 0102          LD      (OFFSET),A
017C      3E 07            LD      A,OFSTA1
017E      32 0103          LD      (OFFSTA),A
0181      C9                RET
ELSE
;BASIC
CP        7
JR        Z,PASS3
CALL     SLINE                ;end of
;on the
LD      A,OFSET1
LD      (OFFSET),A
LD      A,OFSTA0
LD      (OFFSTA),A
RET
PASS2:  CALL     SLINE
LD      A,OFSET1            ;for the
LD      (OFFSET),A
LD      A,OFSTA1
LD      (OFFSTA),A
RET
PASS3   CALL     SLINE
LD      A,OFSET2            ;for 3rd
LD      (OFFSET),A        ;and off
LD      A,OFSTA2
LD      (OFFSTA),A

```

```

                                RET
ENDIF
;
;Special Line Feed for graphics
;
0182    C5                      SLINE:  PUSH    BC                      ;special
0183    3A 0102                 LD      A, (OFFSET)          ;repeat
0186    47                      LD      B,A                      ;have be
0187    3E 1B                   SL1:   LD      A,1BH
0189    CD 015B                 CALL   PRINT
018C    3E 32                   LD      A,32H
018E    CD 015B                 CALL   PRINT
0191    10 F4                   DJNZ   SL1
0193    3E 0D                   LD      A,0DH                      ;CR to p
0195    CD 015B                 CALL   PRINT
0198    C1                      POP     BC
0199    C9                      RET

;
;Multiply Routine - a cut down version which mul
;
019A    E5                      MULT:  PUSH    HL                      ;save cu
019B    6E                      LD      L, (HL)                    ;get con
019C    26 00                   LD      H,0                        ;extend
019E    29                      ADD     HL,HL                       ;shifts
019F    29                      ADD     HL,HL                       ;4x
01A0    29                      ADD     HL,HL                       ;8x
01A1    29                      ADD     HL,HL                       ;16x
01A2    22 0104                 LD      (RESULT),HL              ;store 1
01A5    E1                      POP     HL
01A6    C9                      RET

;
;Compute Character Generator start address for t
;
01A7    E5                      CGEN:  PUSH    HL
01A8    D5                      PUSH    DE
01A9    21 F000                 LD      HL,SCRNST                  ;where w
01AC    ED 5B 0104             LD      DE,(RESULT)                ;and the
01B0    19                      ADD     HL,DE                       ;togethe
01B1    16 00                   LD      D,0                        ;0 since

```

```

01B3      3A 0103          LD      A, (OFFSTA)
01B6      5F              LD      E,A
01B7      19              ADD     HL,DE                      ;we begi
01B8      22 0108        LD      (CBEGIN),HL
01BB      D1              POP     DE
01BC      E1              POP     HL
01BD      C9              RET

;
;Assemble the dot columns for one character/grap
;and print them, 1 column at a time.
;
01BE      C5              ASMBLE: PUSH   BC
01BF      E5              PUSH   HL
01C0      3E 01          LD      A,1                      ;set the
;to the
01C2      D3 0B          OUT     (0BH),A
01C4      0E 80          BIT8:  LD      C,80H              ;set up
01C6      2A 0108        GADDR: LD      HL, (CBEGIN)      ;start o
01C9      E5              COLN1: PUSH   HL              ;start 1
01CA      16 00          LD      D,0                      ;clear t
01CC      3A 0102        LD      A, (OFFSET)            ;loop on
01CF      47              LD      B,A
01D0      CB 3A          TOTEM:  SRL     D                      ;output
01D2      7E              LD      A, (HL)              ;next by
01D3      A1              AND     C                      ;test fo
01D4      28 0D          JR      Z,MARKIT              ;if 0, p
01D6      3A 0102        LD      A, (OFFSET)            ;else, t
;and set
IF      CPM22
01D9      FE 05          CP      5                      ;5 dots
01DB      20 04          JR      NZ,SIX
01DD      CB E2          FOUR:  SET     4,D              ;bit 4 i
01DF      18 02          JR      MARKIT
ELSE
CP      2                      ;2 dots
JR      NZ,SIX
ONE:    SET     1,D              ;bit 1 i
JR      MARKIT
ENDIF
01E1      CB F2          SIX:   SET     6,D              ;else, s
01E3      23              MARKIT: INC     HL              ;next by
01E4      10 EA          DJNZ   TOTEM              ;loop un
01E6      7A              LD      A,D              ;move th
01E7      C6 80          ADD     A,80H              ;boost t
01E9      CD 015B        CALL   PRINT
01EC      E1              POP     HL                      ;back to
;case we
01ED      CB 39          SRL     C                      ;shift t
01EF      30 D8          JR      NC,COLN1            ;8 colum
01F1      E1              POP     HL                      ;yes, re
01F2      C1              POP     BC
01F3      3E 00          LD      A,0              ;reset R
01F5      D3 0B          OUT     (0BH),A
01F7      C9              RET

;
;Exit Routine
;

```



```
01F8 3E 1E          FIN: LD      A,1EH      ;end gra
01FA CD 015B       CALL   PRINT
01FD 3E 0C       LD      A,0CH      ;form fe
01FF CD 015B       CALL   PRINT
0202 3E 1B       LD      A,1BH      ;return
0204 CD 015B       CALL   PRINT
0207 3E 17       LD      A,17H
0209 CD 015B       CALL   PRINT
020C C1            POP    BC          ;recover
020D D1            POP    DE
020E E1            POP    HL
020F F1            POP    AF
0210 C9            RET
;
END
```

Macros:

Symbols:

ASMBLE	01BE	BIT8	01C4	BSTRT	8000	CBEGIN	0108
CCNT	013C	CGEN	01A7	CHARLN	0050	COLN1	01C9
CPM22	FFFF	ENTRY	0100	EOLFLG	015F	FALSE	0000
FIN	01F8	FOUR	01DD	GADDR	01C6	INTERP	A87F
LCNT	0138	MARKIT	01E3	MULT	019A	NUMLIN	0018
OFFSET	0102	OFFSTA	0103	OFSET1	0007	OFSET2	0005
OFSTA0	0000	OFSTA1	0007	PASS2	0174	PRINT	015B
RPT	0156	RESULT	0104	RS_OUT	E04B	SCNLN	000B
SCRNST	F000	SIX	01E1	SL1	0187	SLINE	0182
STRT	010C	TOTEM	01D0	TRUE	FFFF		

No Fatal error(s)

→:

ASMBLE 01BE