VG5000

Cassette Software : GLPRT1

A) USE

GLPRT1 is Graphics LPRINT Routine version 1.

It is for use with VW0010/VW0020 printers only.

It supports separated and mosaic graphic character printing.

B) LOADING

Enter: CLOAD <RET>

The routine will load and then run. A sign-on screen (similar for all these types of routine) is output.

Control is returned to BASIC.

C) OPERATION

At load time, the routine is initialised.

To print the graphics characters, two special shift codes are LPRINTed:

&'1E'	CHR$(30)	Graphics ON
&'1F'	CHR$(31)	Graphics OFF

To print graphics, the routine must be initialised, and Graphics ON must have been LPRINTed.

Graphics is automatically turned OFF when the printer file is closed (e.g: end of program).

D) INITIALISE/DEINITIALISE

The addresses to CALL to initialise or deinitialise are given in the sign-on screen.

Initialise prints + at the cursor.

Deinitialise prints - at the cursor.

Multiple initialisation / deinitialisation is handled with no problems.

E) RELOCATION

At run time, the routine is relocated below the current "memsiz" address (unless it is already present). The memory map is then adjusted to compensate for the routine size. Multiple running of the routine will have no further effect after the first installation is complete.

Any other (eg: user) routine that uses top of memory should be loaded first.

At load time, the routines ID is checked for in "memsiz" +1 and "memsize" +2. The ID for this routine is:

a_ hex 61; hex 60 decimal 97; 96

F) HOOKS

The routine is tied to (outhk). If a user routine also uses (outhk) it must be loaded first.

The user (outhk) entry is relocated into the routine.

In any call to OUTDO, the user routine, if any, is called first.

If the user routine is to be deinitialised, deinitialise this routine first to restore (outhk).

G) ENTRY POINTS

These follow a similar pattern for all cassette routines:

Routine Base	2 byte ID (a_)
Displaced hook	Base + 2
Initialise	Base + 5
Deinitialise	Base + 8
Main entry point	Base + 11
Use entry point	Base + 14

The Main entry point calls the displaced hook.

The Use entry point omits the displaced hook call.

On the sign-on screen, only the Main entry point is given.

Programmers wishing to directly use a cassette routine should load the routine, then use the hook+1 address as the pointer "entry point". Offsets to this address would give initialise, deinitialise, etc.

For example, to deinitialise this routine from BASIC, use the following lines:

```
ADDR = (PEEK (18403) + PEEK (18404)*256) -3
IF ADDR >32767 THEN ADDR = ADDR-65536
CALL ADDR
```

H) COPYING

This information is proprietary.

First load the routine. After execution, position a fresh cassette using:

CSAVEL <RET>

Then enter:

CSAVEM"GLPRT1",&"49FC",&"458",10 <RET>

Remember to set the speed of recording if 2400 baud required.

TEST BEFORE THE CODE USAGE

TEST AFTER THE CODE USAGE

Graphics PRINTER Routine version 1

```
10 POKE18904,77:POKE18905,76:CLEAR
20 GOTO 50
30 BASE=PEEK(18719)+1+PEEK(18720)*256:IFBASE>32767THENBASE=BASE-65536
40 RETURN
50 PRINT
60 PRINT"Graphics LPRINT Routine version 1"
70 PRINT"-----------------------------------"
80 PRINT
90 PRINT"Hook = (outhk):   @ 18402"
100 GOSUB 30
110 IFPEEK(BASE)<>97ORPEEK(BASE+1)<>96THENBASE=BASE-253
120 PRINT"Memory size:      @";BASE
130 PRINT"Hook now:         @";BASE+2
140 PRINT"Initialise   (+): @";BASE+5
150 PRINT"Deinitialise (-): @";BASE+8
160 PRINT"Main entry point: @";BASE+11
170 CLEAR FRE(""),BASE-1
180 GOSUB 30
190 IFPEEK(BASE)<>97ORPEEK(BASE+1)<>96THENCALL(PEEK(18574)+PEEK(18575)*256)+768
200 CALLBASE+5
210 PRINT:PRINT"Installation complete":PRINT:CALL4267
```

```
VDU HARD-COPY          09:09:31

 1.  DO    DISASSEMBLY
 2.  ADDR  DATA     OPC  OPERAND
 3.  4CFC  F3       DI
 4.  4CFD  E5       PUSH HL
 5.  4CFE  DDE5     PUSH IX
 6.  4D00  2A1F49   LD   HL,(H'491F')
 7.  4D03  23       INC  HL
 8.  4D04  3661     LD   (HL),H'61'
 9.  4D06  23       INC  HL
10.  4D07  3660     LD   (HL),H'60'
11.  4D09  23       INC  HL
12.  4D0A  23       INC  HL
13.  4D0B  23       INC  HL
14.  4D0C  23       INC  HL
15.  4D0D  EB       EX   DE,HL
16.  4D0E  D5       PUSH DE
17.  4D0F  01F800   LD   BC,H'00F8'
18.  4D12  215C4D   LD   HL,H'4D5C'
19.  4D15  EDB0     LDIR
20.  4D17  3E0E     LD   A,H'0E'
21.  4D19  DD21404D LD   IX,H'4D40'
 3.  4D1D  D1       POP  DE
 4.  4D1E  D5       PUSH DE
 5.  4D1F  DD6600   LD   H,(IX+H'00')
 6.  4D22  DD6E01   LD   L,(IX+H'01')
 7.  4D25  19       ADD  HL,DE
 8.  4D26  4E       LD   C,(HL)
 9.  4D27  23       INC  HL
10.  4D28  46       LD   B,(HL)
11.  4D29  2B       DEC  HL
12.  4D2A  EB       EX   DE,HL
13.  4D2B  09       ADD  HL,BC
14.  4D2C  EB       EX   DE,HL
15.  4D2D  73       LD   (HL),E
16.  4D2E  23       INC  HL
17.  4D2F  72       LD   (HL),D
18.  4D30  3D       DEC  A
19.  4D31  2806     JR   Z,H'06'
20.  4D33  DD23     INC  IX
21.  4D35  DD23     INC  IX
 3.  4D37  18E4     JR   H'E4'
 4.  4D39  D1       POP  DE
 5.  4D3A  DDE1     POP  IX
 6.  4D3C  E1       POP  HL
 7.  4D3D  FB       EI
 8.  4D3E  C9       RET
 9. ?DO:
10. ?DO: EX 4D40
11.  DO     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
12.  4D40   00 01 00 04 00 07 00 41 00 48 00 65 00 6A 00 9D
13.  4D50   00 B2 00 BB 00 CB 00 D9 00 DF 00 E7 C3 09 00 C3
14.  DO    Function cancelled
15. ?DO:
16. ?DO:
17. ?DO:
```

VDU HARD-COPY 09:01:58

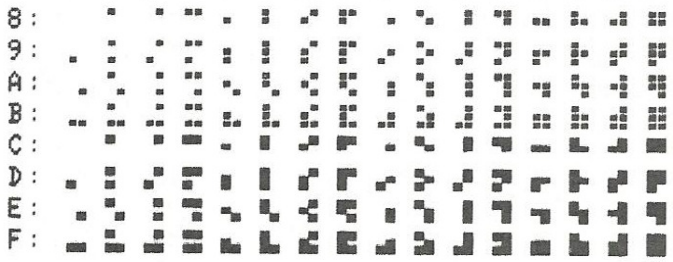
```
1. ?DO:   )))))))))) Installed Graphics LPRINT Routine version 1 ((((((((((
2.
3. ?DO: MEM 7F00 FFFF DIS
4.  DO    DISASSEMBLY
5.  ADDR  DATA      OPC  OPERAND
3.  7F00  61        LD   H,C
4.  7F01  60        LD   H,B
5.  7F02  C9        RET
6.  7F03  C9        RET
7.  7F04  C9        RET
8.  7F05  C39B7F    JP   H'7F9B'
9.  7F08  C3C97F    JP   H'7FC9'
10. 7F0B  CD027F    CALL H'7F02'
11. 7F0E  F5        PUSH AF
15. 7F0F  3A6F48    LD   A,(H'486F')
16. 7F12  3D        DEC  A
17. 7F13  2802      JR   Z,H'02'
18. 7F15  F1        POP  AF
19. 7F16  C9        RET
20. 7F17  3A7348    LD   A,(H'4873')
21. 7F1A  B7        OR   A
22. 7F1B  20F8      JR   NZ,H'F8'
 2. 7F1D  F1        POP  AF
 3. 7F1E  F5        PUSH AF
 7. 7F1F  FE1E      CP   H'1E'
 8. 7F21  2007      JR   NZ,H'07'
 9. 7F23  3A7448    LD   A,(H'4874')
10. 7F26  CBE7      SET  4,A
11. 7F28  1809      JR   H'09'
12. 7F2A  FE1F      CP   H'1F'
13. 7F2C  200A      JR   NZ,H'0A'
14. 7F2E  3A7448    LD   A,(H'4874')
15. 7F31  CBA7      RES  4,A
16. 7F33  327448    LD   (H'4874'),A
17. 7F36  1838      JR   H'38'
18. 7F38  3A7448    LD   A,(H'4874')
19. 7F3B  CB67      BIT  4,A
20. 7F3D  28D6      JR   Z,H'D6'
21. 7F3F  F1        POP  AF
 1. 7F40  CB7F      BIT  7,A
 2. 7F42  C8        RET  Z
 3. 7F43  E5        PUSH HL
 4. 7F44  F5        PUSH AF
 8. 7F45  CD797F    CALL H'7F79'
 9. 7F48  67        LD   H,A
10. 7F49  F1        POP  AF
11. 7F4A  F5        PUSH AF
12. 7F4B  1F        RRA
13. 7F4C  CD797F    CALL H'7F79'
14. 7F4F  6F        LD   L,A
15. 7F50  3EFF      LD   A,H'FF'
16. 7F52  327348    LD   (H'4873'),A
17. 7F55  3E1B      LD   A,H'1B'
18. 7F57  DF        RST  H'18'
19. 7F58  3E53      LD   A,H'53'
20. 7F5A  DF        RST  H'18'
21. 7F5B  3E30      LD   A,H'30'
22. 7F5D  DF        RST  H'18'
 1. 7F5E  DF        RST  H'18'
 2. 7F5F  DF        RST  H'18'
 3. 7F60  3E38      LD   A,H'38'
 4. 7F62  DF        RST  H'18'
 8. 7F63  F1        POP  AF
 9. 7F64  F5        PUSH AF
10. 7F65  CB77      BIT  6,A
11. 7F67  F5        PUSH AF
12. 7F68  7C        LD   A,H
13. 7F69  CD8D7F    CALL H'7F8D'
14. 7F6C  F1        POP  AF
15. 7F6D  7D        LD   A,L
16. 7F6E  CD8D7F    CALL H'7F8D'
17. 7F71  F1        POP  AF
18. 7F72  E1        POP  HL
19. 7F73  F1        POP  AF
20. 7F74  AF        XOR  A
21. 7F75  327348    LD   (H'4873'),A
22. 7F78  C9        RET
```

```
 1.  7F79  6F       LD   L,A
 2.  7F7A  AF       XOR  A
 3.  7F7B  CB45     BIT  0,L
 4.  7F7D  2802     JR   Z,H'02'
 8.  7F7F  C607     ADD  A,H'07'
 9.  7F81  CB55     BIT  2,L
10.  7F83  2802     JR   Z,H'02'
11.  7F85  C618     ADD  A,H'18'
12.  7F87  CB65     BIT  4,L
13.  7F89  C8       RET  Z
14.  7F8A  C6E0     ADD  A,H'E0'
15.  7F8C  C9       RET


16.  7F8D  2007     JR   NZ,H'07'
17.  7F8F  E6DB     AND  H'DB'
18.  7F91  F5       PUSH AF
19.  7F92  AF       XOR  A
20.  7F93  DF       RST  H'18'
 1.  7F94  F1       POP  AF
 2.  7F95  26DF     LD   H,H'DF'
 3.  7F97  DF       RST  H'18'
 4.  7F98  DF       RST  H'18'
 5.  7F99  DF       RST  H'18'
 6.  7F9A  C9       RET


 8.  D0    DISASSEMBLY
 9.  ADDR  DATA     OPC  OPERAND
10.  7F9B  F5       PUSH AF
11.  7F9C  E5       PUSH HL
 3.  7FCB  D5       PUSH DE
 4.  7FCC  3AE247   LD   A,(H'47E2')
 5.  7FCF  210B7F   LD   HL,H'7F0B'
 6.  7FD2  ED5BE347 LD   DE,(H'47E3')
16.  7FA8  FEC9     CP   H'C9'
17.  7FAA  2807     JR   Z,H'07'
18.  7FAC  FEC3     CP   H'C3'
19.  7FAE  2003     JR   NZ,H'03'
20.  7FB0  E7       RST  H'20'
 1.  7FB1  2812     JR   Z,H'12'
 2.  7FB3  3AE247   LD   A,(H'47E2')
 3.  7FB6  32027F   LD   (H'7F02'),A
 4.  7FB9  3EC3     LD   A,H'C3'
 5.  7FBB  32E247   LD   (H'47E2'),A
 6.  7FBE  ED53037F LD   (H'7F03'),DE
 7.  7FC2  22E347   LD   (H'47E3'),HL
 8.  7FC5  3E2B     LD   A.H'2B'
12.  7FC7  1827     JR   H'27'


13.  7FC9  F5       PUSH AF
14.  7FCA  E5       PUSH HL
16.  7F9D  D5       PUSH DE
17.  7F9E  3AE247   LD   A,(H'47E2')
18.  7FA1  210B7F   LD   HL,H'7F0B'
19.  7FA4  ED5BE347 LD   DE,(H'47E3')
19.  7FD6  FEC3     CP   H'C3'
20.  7FD8  2014     JR   NZ,H'14'
21.  7FDA  E7       RST  H'20'
 1.  7FDB  2011     JR   NZ,H'11'
 2.  7FDD  3A027F   LD   A,(H'7F02')
 3.  7FE0  32E247   LD   (H'47E2'),A
 4.  7FE3  2A037F   LD   HL,(H'7F03')
 5.  7FE6  22E347   LD   (H'47E3'),HL
 6.  7FE9  3EC9     LD   A,H'C9'
 7.  7FEB  32027F   LD   (H'7F02'),A
 8.  7FEE  3E2D     LD   A,H'2D'
 9.  7FF0  2A0548   LD   HL,(H'4805')
13.  7FF3  F5       PUSH AF
14.  7FF4  CD9102   CALL H'0291'
15.  7FF7  F1       POP  AF
16.  7FF8  77       LD   (HL),A
17.  7FF9  D1       POP  DE
18.  7FFA  E1       POP  HL
19.  7FFB  F1       POP  AF
20.  7FFC  C9       RET
```

5-6-84

*> GRAPHICS WITH: 3-3-2 Ratio for SEPARATED
3-2-3 Ratio for MOSAIC

GP-500PH VGP GRAPHICS (VG5000)

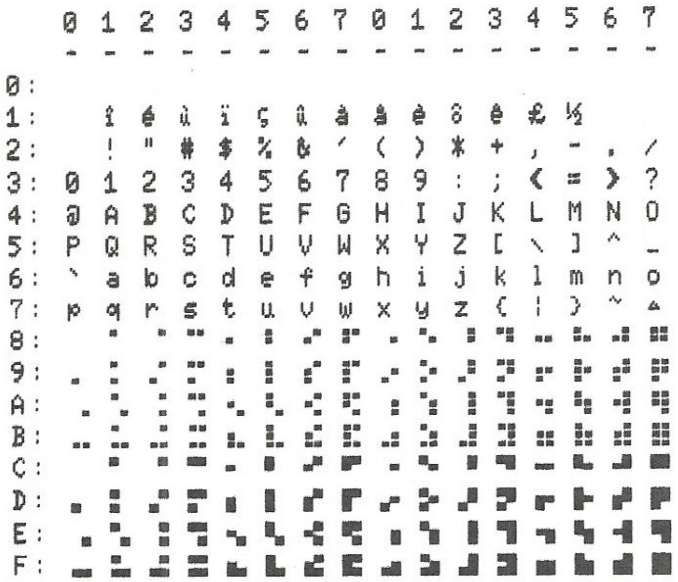



```
10 REM ****************************************
20 REM *                                      *
30 REM *  GP-500PH  VGP GRAPHICS (VG5000)     *
40 REM *                                      *
50 REM ****************************************
60 :
70 A=18402:POKE A,195:POKE A+1,0:POKE A+2,62:REM Hook up the graphics driver
80 LET SO=&"1E":SI=&"1F"
90 LPRINT:LPRINT CHR$(SO)
100 LPRINT"  GP-500PH  VGP GRAPHICS (VG5000)"
110 LPRINT"  ================================":LPRINT
120 RESTORE:FOR COUNT=0 TO 16:READ CODE$
130 LPRINT CODE$;" ";:NEXT COUNT:LPRINT:LPRINT"   ";
140 FOR COUNT=0 TO 15:LPRINT"-";" ";:NEXT COUNT::LPRINT
150 RESTORE 230:FOR LOOP=8 TO 15:READ CODE$:LPRINT CODE$;": ";
160 FOR COUNT=0 TO 15
170 LPRINT CHR$(LOOP*16+COUNT);" ";
180 NEXT COUNT:LPRINT
190 NEXT LOOP
200 LPRINT CHR$(SI)
210 LPRINT:LPRINT:LPRINT:LLIST
220 DATA"  ",0,1,2,3,4,5,6,7
230 DATA8,9,A,B,C,D,E,F
```

```
      VW0020   VGP  GRAPHICS  (VG5000)
      ==========================================

      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
      -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
0 :
1 :      ¡  ¢  £  ¤  ¥  ¦  §  ¨  ©  ª  «  ¬  ½
2 :      !  "  #  $  %  &  '  (  )  *  +  ,  -  .  /
3 :   0  1  2  3  4  5  6  7  8  9  :  ;  <  =  >  ?
4 :   @  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O
5 :   P  Q  R  S  T  U  V  W  X  Y  Z  [  \  ]  ^  _
6 :   `  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o
7 :   p  q  r  s  t  u  v  w  x  y  z  {  |  }  ~
8 :
9 :
A :
B :
C :
D :
E :
F :
```

```
10 REM ****************************************
20 REM *                                      *
30 REM *   VW0020  VGP GRAPHICS (VG5000)      *
40 REM *                                      *
50 REM ****************************************
60 :
70 A=18402:POKE A,195:POKE A+1,0:POKE A+2,62:REM Hook up the graphics driver
80 LET WIDE=&"0E":NARROW=&"0F":SO=&"1E":SI=&"1F"
90 LPRINT:LPRINT CHR$(WIDE);CHR$(SO)
100 LPRINT"   VW0020  VGP GRAPHICS (VG5000)"
110 LPRINT"   ============================":LPRINT
120 RESTORE:FOR COUNT=0 TO 16:READ CODE$
130 LPRINT CODE$;" ";:NEXT COUNT:LPRINT:LPRINT"   ";
140 FOR COUNT=0 TO 15:LPRINT"-";" ";:NEXT COUNT::LPRINT
150 RESTORE 260:FOR LOOP=0 TO 15:READ CODE$:LPRINT CODE$;": ";
160 FOR COUNT=0 TO 15
170 IF LOOP>1 THEN 200
180 IF LOOP=0 THEN LPRINT"  ";:GOTO 210
190 IF COUNT=0 OR COUNT=>14 THEN LPRINT"  ";:GOTO 210
200 LPRINT CHR$(LOOP*16+COUNT);" ";
210 NEXT COUNT:LPRINT
220 NEXT LOOP
230 LPRINT CHR$(NARROW);CHR$(SI)
240 LPRINT:LPRINT:LPRINT:LLIST
250 DATA"  "
260 DATA0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
```

VW0020 VGP GRAPHICS (VG5000)



```
10 REM ***************************
20 REM *                         *
30 REM *   SPACE INVADER HEADS   *
40 REM *                         *
50 REM ***************************
60 :
70 :
80 GOTO 160
90 :
100 :
110 LPRINT CHR$(27);"S";"0008";
120 FOR I=1 TO 8:READ BITS:LPRINT CHR$(BITS);:NEXT I
130 RETURN
140 :
150 :
160 POKE 18547,255:LPRINT CHR$(27);CHR$(66)
170 FOR ZAP=1 TO 2
180 LPRINT:LPRINT
190 LPRINT:FOR LINE=1 TO 11
200 LPRINT"      ";
210 RESTORE 310:GOSUB 110:GOSUB 110
220 NEXT LINE:LPRINT:FOR LINE=1 TO 11
230 LPRINT"      ";
240 RESTORE 330:GOSUB 110:GOSUB 110
250 NEXT LINE:NEXT ZAP
260 LPRINT CHR$(27);CHR$(65):POKE 18547,0
270 LPRINT:LPRINT:LPRINT:LLIST:END
280 LPRINT:LPRINT:LPRINT
290 :
300 :
310 DATA&"00",&"60",&"F0",&"98",&"0C",&"0E",&"9E",&"FE"
320 DATA&"FE",&"9E",&"0E",&"0C",&"98",&"F0",&"60",&"00"
330 DATA&"00",&"00",&"01",&"03",&"07",&"3F",&"0F",&"3F"
340 DATA&"3F",&"0F",&"3F",&"07",&"03",&"01",&"00",&"00"
```

VDU HARD-COPY 01:43:09

))))))))))))))))))))) accessing ROM slices ((((((((((((((((((((

The slice required is written into R6 and R7. The format of these is:

		B7	B6	B5	B4	B3	B2	B1	B0
	R6:	x	x	b6	c6	c5	c4	c3	c2
	R7:	b4	b5	t3	t2	t1	t0	c1	c0

x indicates "don't care" bits.
b6 b5 b4 select the character set of the ROM:

b6	b5	b4	Character Set	
0	0	x	128 alphanum	G00
0	1	0	128 mosaics	G10
0	0	1	32 strokes	G11
1	0	x	accentuated	G20
1	1	x	lower case	G21

c6 - c0 is the character code (c6 is "x" for G11 set).
t3 - t0 is the slice read (slices are reversed as B0 B1 B2 ... B7).

VDU HARD-COPY 00:20:44

```
1.  ))))))))))))))))))) accessing ROM slices ((((((((((((((((((((
2.  D0    DISASSEMBLY
3.  ADDR  DATA      OPC   OPERAND
4.  6000  CD3060    CALL  H'6030'          Wait for access
5.  6003  3E26      LD    A,H'26'         Address R6 -
6.  6005  D38F      OUT   (H'8F'),A
7.  6007  3E12      LD    A,H'12'          - and write parameter
8.  6009  D3CF      OUT   (H'CF'),A
9.  600B  3E27      LD    A,H'27'         Address R7 -
10. 600D  D38F      OUT   (H'8F'),A
11. 600F  3E00      LD    A,H'00'          - and write parameter
12. 6011  D3CF      OUT   (H'CF'),A
13. 6013  3E28      LD    A,H'28'         Address command register -
14. 6015  D38F      OUT   (H'8F'),A
15. 6017  3E88      LD    A,H'88'          - and read ROM slice
16. 6019  D3CF      OUT   (H'CF'),A
17. 601B  CD3060    CALL  H'6030'         Wait for access
18. 601E  3E21      LD    A,H'21'         Address R1 -
19. 6020  D38F      OUT   (H'8F'),A
20. 6022  0ECF      LD    C,H'CF'          - and read slice into D
21. 6024  ED50      IN    D,(C)
22. 6026  76        HALT                  Break on A=6026
23.
24. DEBUG   D0    Z80                         pos: 25  drives:BB--          00:20
```

VDU HARD-COPY 00:27:24

```
1. ))))))))))))))))))))) accessing ROM slices ((((((((((((((((((((
2.  DO    DISASSEMBLY
3.  ADDR  DATA     OPC  OPERAND
4.  6030  0E8F     LD   C,H'8F'      Wait for access permit
5.  6032  1620     LD   D,H'20'
6.  6034  ED51     OUT  (C),D
7.  6036  0ECF     LD   C,H'CF'
8.  6038  ED50     IN   D,(C)
9.  603A  FA3860   JP   M,H'6038'    Loop till status good
10. 603D  C9       RET
11. ?DO:
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24. DEBUG   DO    Z80                    pos: 55  drives:BB--          00:27
```