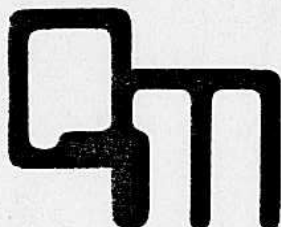


Quantum Micros



QUANTUM MICROS MZ80K HI-RES
=====

Thank you for purchasing our MZ80K graphics system. If you have any problems or suggestions as to how we can improve this product please let us know.

In view of continuing improvement we reserve the right to change the specifications without prior notice.

The hardware and software associated with this product are
Copyright (C) QUANTUM MICROS

CONTENTS

=====

PAGE No.

| | |
|-------|---|
| 2 | <i>An introduction to the software</i> |
| 3-4 | <i>How the system works</i> |
| 5-6 | <i>Operating the editor</i> |
| 7-9 | <i>Editor listing</i> |
| 10-12 | <i>GRAPHICS BASIC 4</i> |
| 13 | <i>Some useful POKE's and addresses</i> |
| 14 | <i>Example program to modify the pixel graphics</i> |
| 14 | <i>Example program to give reverse field characters</i> |
| 15 | <i>Example program to copy the character set into PCG ram</i> |



COMPUTER PRODUCTS LTD

The Leeds Computer Centre
60-62 The Balcony Merrion Centre Leeds LS2 8NG
Tel: (0532) 458877

SOFTWARE

=====

The enclosed cassette contains two programs on side A. They are as follows.

SIDE A

P.C.G. EDITOR
HEX

Written to run under SHARP
BASIC and self executing.

SIDE B Graphics BASIC and demonstration

P.C.G. EDITOR

=====

The QUANTUM MZ80K PCG editor has been written to make the programming of special characters very simple. It also allows you to save special character sets on tape and to reload them for further modification or for use in a program.

The editor has been written in SHARP BASIC which must of course be loaded first.

HEX

===

This self executing BASIC program has been included to assist you in creating your own graphics programs. It will convert a hexadecimal number with up to four digits into decimal.

GRAPHICS BASIC

=====

Quantum graphics BASIC 4 has all the facilities offered by Sharp BASIC SP-5025 plus additional commands for controlling the high resolution screen. A demonstration program has been included to help explain the use of these extra commands.

THE NEW MEMORY MAP

=====

The memory map of a SHARP fitted with the QUANTUM system functions perfectly normally, i.e. Nothing is missing from the memory map.

There are however the following additions:

- 1) A 2k block of read/write RAM at 55296 decimal (D800hex)
- 2) The character generator now sits on the memory map at 59392 decimal (E800hex) and its contents can be read.
- 3) There are now 3 new control locations on page E to control the graphics system.

HOW THE SYSTEM WORKS

=====

The character generator of the MZ80K is in fact a 2k EPROM the data in this rom is clocked out in parallel 8 bit data form and is fed into a serial shift register to form the patterns of dots required to form the alphanumeric and graphic characters.

Fig 1. may help to clarify how these characters are programmed into this EPROM.

The first half of this EPROM contains the upper case characters and some graphics, the upper half contains lower case and graphics.

QUANTUM MICROS have taken this character generator EPROM and placed it in a spare location on the memory map. This means that the data making the characters can now be read. This can be done without effecting its use as a character generator and in fact at the same time as characters are being displayed.

Also placed on the memory map in spare locations are 2 static RAM chips. These are extra memory, they can be read from and written to, a machine code sub routine could be written and executed in them. Most important of all though is the fact that under software control they can be switched in to replace the character generator EPROM. They can be switched to overlay all of or only the upper or lower locations of the character generator.

This is of course very important because if all the EPROM space were overlayed at once the alpha numeric characters would be lost and it would be impossible to read anything on the screen. The contents of the EPROM can be copied over into the RAM and then the RAM can be turned on with no obvious effect because of course its contents are identical to those of the EPROM. Characters can then be selectively modified at will. An important example of this is the fact that the pixel squares can now be made to join.

Fig. 1
=====

| Address | | | Decimal value of bit | | | | | | | |
|---------|------|------|----------------------|-----|-----|-----|-----|-----|-----|---|
| Dec. | Hex. | Byte | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 59392 | E800 | 0 | | | | | | | | |
| 3 | E801 | 0 | | | | | | | | |
| 4 | E802 | 0 | | | | | | | | |
| 5 | E803 | 0 | | | | | | | | |
| 6 | E804 | 0 | | | | | | | | |
| 7 | E805 | 0 | | | | | | | | |
| 8 | E806 | 0 | | | | | | | | |
| 9 | E807 | 0 | | | | | | | | |
| 59400 | E808 | 24 | | | | /// | | | | |
| 1 | E809 | 36 | | | /// | | /// | | | |
| 2 | E80A | 66 | /// | | | | | /// | | |
| 3 | E80B | 126 | /// | /// | /// | /// | /// | /// | /// | |
| 4 | E80C | 66 | /// | | | | | /// | | |
| 5 | E80D | 66 | /// | | | | | /// | | |
| 6 | E80E | 66 | /// | | | | | /// | | |
| 7 | E80F | 0 | | | | | | | | |
| 8 | E810 | 124 | /// | /// | /// | /// | /// | /// | /// | |
| 9 | E811 | 34 | | | /// | | | | /// | |
| 59410 | E812 | 34 | | | /// | | | | /// | |
| 1 | E813 | 60 | | | /// | /// | /// | /// | /// | |
| 2 | E814 | 34 | | | /// | | | | /// | |
| 3 | E815 | 34 | | | /// | | | | /// | |
| 4 | E816 | 124 | /// | /// | /// | /// | /// | /// | /// | |
| 5 | E817 | 0 | | | | | | | | |
| 6 | E818 | 28 | | | | /// | /// | /// | /// | |
| 7 | E819 | 34 | | | /// | | | | /// | |
| 8 | E81A | 64 | /// | | | | | | | |
| 9 | E81B | 64 | /// | | | | | | | |

SHARP PCG EDITOR

=====

STEP 1

Place the rewound Editor tape into the MZ80K cassette machine, type LOAD then press carriage return, the editor will then LOAD and RUN automatically.

STEP 2

The Editor is now at command level at this time you can enter one of the following commands.

R = Record/Replay goto tape routine (see step 4)
 J = Join, this routine will join the pixel graphics
 T = Transfer from ROM to RAM
 C = Clear will clear the high res RAM
 O = Off turns high res off
 I = will turn part 1 Hi-res on
 P = will turn part 2 Hi res on

After each of these commands has been executed the Editor will return to command mode.

O-255 = Will cause the dot pattern in the high res RAM to be displayed on screen and take you to STEP 3.

STEP 3

At this point you will have a large pattern on the right hand side of the screen, 8 keys on the lower right of the keyboard are now used to give cursor movement within the 8 x 8 matrix. See Fig. 2.

In addition to this the keys shown below have the following functions.

S = Set the square under the cursor
 R = Reset the square under the cursor
 C = Clear the square under the cursor
 N = Negate (display the negative of the cell)
 L = Left by 90 degrees (rotate the cell)
 O = Off turn the hi-res off
 I = turn part 1 hi-res on
 P = turn part two hi-res on
 V = Value. display the decimal values of the bytes making up the cell
 U = Update the PCG RAM from the display
 M = Move the cell on screen to some other cell
 Q = Quit current cell and return to command level (STEP 2)

STEP 4

RECORD/REPLAY

On entry the routine will ask SAVE or LOAD

If you enter S then a file name will be requested. This name can be upto 8 letters long. The character set is then saved as a 2k type 4 file.

If you enter L then the program will load the character set on tape into the hi-res RAM.

Fig. 2

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|--|--|---|---|---|
| | | | | | | | | | | | | | | | |
| Q | | | R | T | | U | I | O | P | | | | | | |
| | S | | | | | J | | L | | | | | ↖ | ↑ | ↗ |
| | | C | V | | N | M | | | | | | | ↙ | | → |
| | | | | | | | | | | | | | ↘ | ↓ | ↘ |

```

10 LIMIT 48151:REM MAKE SPACE FOR
11 REM          DUMP TO M/C TAPE
70 MP=55296:REM HI-RES POINTER 3900
80 MD=59392      6000
100 DIMTS(7),LC(7),CL(63)
110 REM
111 REM
112 REM
113 REM
120 FORX=0TD7:READA:LC(X)=A:NEXT
130 DATA 128,64,32,16,8,4,2,1
160 SP=53248:REM SCREEN POINTER 3000
200 PRINT"          QUANTUM MICRO8 GRAPHICS EDITOR"
210 Z=0:FORX=1TD16:FORY=1TD16:POKESP+(Y+3)*40+X,Z:Z=Z+1:NEXTY,X
220 PRINT""
229 INPUT"PLEASE ENTER DISPLAY CODE NUMBER ";DC#
230 DC=VAL(DC#):IF DC>0 THEN 240
231 IF LEFT$(DC#,1)="R" THEN 3000
232 IF LEFT$(DC#,1)="J" THEN 2000
233 IF LEFT$(DC#,1)="Q" THEN 240
234 IF LEFT$(DC#,1)="C" THEN 1700
235 IF LEFT$(DC#,1)="T" THEN 1300
236 IF LEFT$(DC#,1)="O" THENPOKE57364,1
237 IF LEFT$(DC#,1)="I" THENPOKE57368,1
238 IF LEFT$(DC#,1)="P" THENPOKE57372,1
239 GOTO 220
240 IF (DC<0)+(DC<>INT(DC))+(DC>255) THEN 260
250 GOTO 280
260 PRINT"
270 GOTO 220
280 PRINT"";
290 FORX=0TD7:TS(X)=PEEK(MP+(8*(DC))+X)
300 FOR Y=0TD7
310 IFTS(X)>=LC(Y) THEN 330
320 PRINT" ";GOTO 340
330 PRINT"█";TS(X)=TS(X)-LC(Y)
340 NEXT:PRINT
350 PRINT"";:NEXT
360 REM
370 PS=SP+(8*40)+66:REM LARGE EDIT.
380 XX=0:YY=0:REM TOP LEFT POSITION
390 TS=PEEK(PS+XX+YY)
400 ZZ=PS+XX+YY
410 USR(3494):POKEZZ,0
420 REM
430 REM
440 REM
450 USR(3494):POKEZZ,67
460 GET A#:IF A#="" THEN 410
490 KS=0
500 IF A#="" THEN GOSUB 700:XX=XX-40:IF XX<0 THEN XX=280:REM UP -
501 IF A#="↑" THEN GOSUB 700:XX=XX-40:IFXX<0 THEN XX=280
502 IF A#="↓" THEN YY=YY+1:IF YY>7 THEN YY=0
503 IF A#="←" THEN GOSUB 700:XX=XX-40:IFXX<0 THEN XX=280
504 IF A#="→" THEN YY=YY-1:IF YY<0 THEN YY=7
505 IF A#=" " THEN GOSUB 700:XX=XX+40:IFXX>280 THEN XX=0
506 IF A#=" " THEN YY=YY-1:IF YY<0 THEN YY=7
507 IF A#="," THEN GOSUB 700:XX=XX+40:IF XX>280 THEN XX=0
508 IF A#="." THEN YY=YY+1:IF YY>7 THEN YY=0
510 IF A#="," THEN GOSUB 700:XX=XX+40:IF XX>280 THEN XX=0:REM DEL.

```

```

520 IF A$="=" THEN GOSUB 700 : YY=YY-1 : IF YY<0 THEN YY=7 : REM LEFT
530 IF A$=">" THEN GOSUB 700 : YY=YY+1 : IF YY>7 THEN YY=0 : REM RIGHT
540 IF A$="S" THEN TS=67 : GOSUB 700
550 IF A$="R" THEN TS=0 : GOSUB 700
560 IF A$="N" THEN GOSUB 700:GOTO 800
570 IF A$="C" THEN GOSUB 700:GOTO 900
580 IF A$="L" THEN GOSUB 700:GOTO 1000
590 IF A$="U" THEN GOSUB 700 :GOTO1100
591 IF A$="M" THEN GOSUB 700 :GOTO1400
592 IF A$="Q" THEN GOSUB 700 :GOTO220
593 IF A$="V" THEN GOSUB 700:GOTO1600
595 IF A$="T" THEN GOSUB 700 :GOTO1300
596 IF A$="O" THEN POKE 57364,1:KS=1 EOL4
597 IF A$="I" THEN POKE 57368,1:KS=1 EOL4
598 IF A$="P" THEN POKE 57372,1:KS=1 EOL4
599 IF KS=0 THEN 410
600 GOTO 390
700 KS=1:POKE PS+XX+YY,TS:RETURN
720 PRINT TS;:NEXT:PRINT"
800 REM INVERT CELL CONTENTS ✓
810 LC=53514 EOL4
820 FOR X=0 TO 7
830 FOR Y=0 TO 7
840 TS=PEEK(LC+X+(Y*40))
850 IF TS=67 THEN TS=0 : GOTO 870
860 TS=67
870 POKE LC+X+(Y*40),TS
880 NEXT Y,X
890 GOTO 390
900 REM CLEAR CELL CONTENTS ✓
910 LC=53514
920 FOR X=0 TO 7
930 FOR Y=0 TO 7
940 POKE LC+X+(Y*40),0
950 NEXT Y,X
960 GOTO 390
1000 REM ROTATE CELL CONTENTS LEFT ✓
1010 Z=0 : LC=53514
1020 FORX=0TO7:FORY=0TO7:CL(Z)=PEEK(LC+X+(Y*40)):Z=Z+1:NEXT Y,X
1030 Z=0:FORY=7TO0STEP-1:FORX=0TO7:POKE(LC+X+(Y*40)),CL(Z):Z=Z+1:NEXTX,Y
1040 GOTO 390
1100 REM UPDATE RAM FROM SCREEN DISPLAY ✓
1110 LC=53514
1120 FOR Y=0 TO 7 : TS=0 : FOR X=0 TO 7
1130 IF PEEK(LC+X+(Y*40))=67 THEN TS=TS+LC(X)
1140 NEXT
1150 POKE MP+(B*(DC))+Y,TS:NEXT
1160 GOTO 1500
1300 REM
1310 FOR X=0 TO 2047
1320 POKE MP+X,PEEK(RD+X)
1330 NEXT
1340 GOTO 220
1400 REM COPY CHARACTER CELL ROUTINE
1410 REM MODIFY DC THEN GOTO NORMAL
1420 REM CREATE SMALL CHARACTER...
1430 PRINT"
1440 PRINT "Copy from ";DC
1441 INPUT " to ?";DC
1442 IF (DC>255)+(DC<0)+(DC<>INT(DC)) THEN 1441
1450 GOTO 1100
1500 REM PRINT VALUE ✓
1510 LC=53514
1520 PRINT":

```

```

1530 PRINT "HI-RES RAM ADDRESS=";STR$(MP+(8*(DC))) ✓
1540 FOR Y=0TO7:TS=0:FORX=0TO7
1550 IFPEEK(LC+X+(40*Y))=67THENTB=TS+LC(X)
1560 NEXT
1570 PRINT TB;:NEXT:PRINT" "
1580 GOTO 220
1600 REM PRINT VALUE
1610 LC=53514
1620 PRINT"";
1630 PRINT "HI-RES RAM ADDRESS=";STR$(MP+(8*(DC)))
1640 FOR Y=0TO7:TS=0:FORX=0TO7
1650 IFPEEK(LC+X+(40*Y))=67THENTB=TS+LC(X)
1660 NEXT
1670 PRINT TB;:NEXT:PRINT" "
1680 GOTO 390
1700 PRINT" "
1710 INPUT "CLEAR-ALL,PART 1 or 2 ";DC# ✓
1720 IF DC#="ALL" THEN 1730
1730 IF DC#="1" THEN D1=0:D2=1023
1740 IF DC#="2" THEN D1=1024:D2=2047
1745 GOTO 1770
1750 PRINT" "
1751 PRINT " ARE YOU SURE Y/N"
1752 FOR TD=1 TO 100:NEXT
1753 GET A# : IF A#="" THEN 1753
1754 IF A#="Y" THEN D1=0:D2=2047
1770 FOR X=D1 TO D2
1780 POKE MP+X,0:NEXT
1790 GOTO 220
2000 REM JOIN SET/RESEY ROUTINE ✓
2010 REM
2100 FOR X=57340 TO 57220 STEP-4
2110 FOR Y=3 TO 1 STEP-1
2120 Z=PEEK(X+Y)
2130 IF Z=7 THEN Z=15
2140 IF Z=112 THEN Z=240
2150 IF Z=119 THEN Z=255
2160 POKE (X+Y),Z:W=(X+Y)-1
2170 NEXT
2180 POKE W,Z
2190 NEXT:GOTO 220
3000 REM SAVE/LOAD ROUTINES ✓
3010 PRINT""
3020 PRINT "SAVE OR LOAD"
3030 Z#="" : GET Z# : IF Z#="" THEN 3030
3040 IF Z#="S" THEN 3060
3050 IF Z#="L" THEN 3160
3059 GOTO 3030
3060 REM SAVE ROUTINE
3070 INPUT "FILE NAME PLEASE ";Z#
3080 Z#=LEFT$(Z#,8)+" chr/set"
3081 FOR X=0 TO 2047
3084 POKE 48151+X,PEEK(55296+X)
3085 NEXT
3086 POKE 17810,4 : REM FILE TYPE
3090 FORX=1TOLEN(Z#):POKE(4336+X),ASC(MID$(Z#,X,1)):NEXT:POKE(4336+X),13
3091 POKE 4354,00
3092 POKE 4355,08
3093 POKE 4356,00
3094 POKE 4357,192
3095 USR(33):USR(36):RUN
3160 USR(39):USR(42):RUN

```

GRAPHICS BASIC 4

=====

This is a modified version of SHARP BASIC SP5025 with the following additional commands.

GRAPH this is a software switch and is used to control the high resolution graphics unit. It provides the following functions:

GRAPH0 = RESET HIGH RESOLUTION

GRAPH1 = ENABLE HIGH RESOLUTION

GRAPH2 = RETURN TO LOW RESOLUTION

GRAPH3 = SCREEN FLASH SUPPRESSION ON

GRAPH4 = SCREEN FLASH SUPPRESSION OFF

GRAPH5 = REVERSE VIDEO (TOGGLE ACTION)

*GRAPH6 = HIGH RESOLUTION SCREEN DUMP (ADDITIONAL COMMAND)

*GRAPH7 = SHAPE TABLE (ADDITIONAL COMMAND)

LINE A,B,C,D This command draws a line between points A,B and C,D

WIPE A,B,C,D This command erases a line between points A,B and C,D

In addition to the above the following commands are changed:

SET and RESET now operate on a matrix 320 by 200 when the High resolution unit has been enabled.

LIST automatically resets the system to low resolution mode

PEEK and POKE now do not cause screen flash

A demonstration program has been provided to give a full explanation of the use of GRAPHICS BASIC.

* Denotes optional commands available at extra cost. See page 12

DEMONSTRATION PROGRAM

=====

```
10 GRAPH1:REM ENABLE HIGH RESOLUTION
20 GRAPH0:REM RESET HIGH RESOLUTION
30 GRAPH3:REM FLASH SUPPRESSION OFF
40 GOTO 60
50 GRAPH4:REM FLASH SUPPRESSION ON
60 REM DRAW BORDER
70 LINE 0,0,319,0
80 LINE 319,0,319,199
90 LINE 319,199,0,199
100 LINE 0,199,0,0
110 REM DRAW SINE WAVE
120 FOR X=1 TO 318 STEP 2
130 Y=SIN(X*66)*100+100
140 SET X,Y
150 NEXT X
160 FOR TD=1 TO 1000:NEXT TD
170 GRAPH5:REM REVERSE SCREEN
180 FOR TD=1 TO 1000:NEXT TD
190 GRAPH5:REM REVERSE SCREEN
200 Z=Z+0.5
210 GRAPH0:REM RESET
220 IF Z=INT(Z) THEN 30
230 GOTO 50
```

SCREEN DUMP(High resolution)

=====

This extra command (GRAPH6) can be added to GRAPHICS BASIC to give a high resolution screen dump. Overlay programs are available for the following printers:

- 1)EPSON MX80 TYPE 2 VIA SP4 INTERFACE*
- 2)EPSON MX80 TYPE 1 VIA SP4 INTERFACE and SHARP ROM'S*
- 3)SEIKOSHA GPI00A and QUANTUM INTERFACE*

A modification guide to enable interfacing to other printers is also available.

The screen dump command is available from your QUANTUM HI-RES stockist for £5.50

SHAPE TABLE

=====

This extra command (GRAPH7) can be added to GRAPHICS BASIC to give shape manipulation.

Both the above programs are supplied on cassette with demonstration programs.

USEFUL POKES AND ADDRESSES (required to operate the Hi-res system)
 =====

POKE 10167,1 = remove PEEK protect (this allows you to access the new memory locations).

POKE 57364,1 = High res off (E014hex)

POKE 57368,1 = High res 1 on (E018hex)

POKE 57372,1 = High res 2 on (E01Chex)

SCREEN START 53248 (D000hex)

RAM START 55296 (D800hex)

ROM START 59392 (E800hex)

HIGHCELL POINTER 15956

LOWCELL POINTER 15960

USEFUL NOTE
 =====

If during program development a system crash occurs leaving you with Hi-res only on the screen, take the following action.

Enter CLR (to clear the screen) then type LIST followed by CR.

SOME USEFUL POKES
 =====

POKE 10167,1 Remove PEEK protect from BASIC

POKE 57507,0 Turn screen off

POKE 57507,1 Turn screen on

POKE 4465,X (where X=1 to 40) moves cursor to position X on line

POKE 4466,Y (where Y=1 to 24) moves cursor to line Y

POKE 57347,4 Make LED red

POKE 57347,5 Make LED green

POKE 57364,1 Turn hi-res off

POKE 57368,1 Turn hi-res part 1 on

POKE 57372,1 Turn hi-res part 2 on

POKE 4464,1 Make keyboard entry lower case

POKE 4464,0 Make keyboard entry upper case

POKE 10682,1 Doing this before saving a program will make it auto run when loaded.

```

1 REM This program will modify the
2 REM set and reset pixels in the
3 REM graphics RAM so that they
4 REM join.....
5 REM Note. The characters should
6 REM have first been copied from
7 REM the ROM.....
8 REM
9 POKE 57507,0:REM Screen off
10 FOR X=57340 TO 57220 STEP-4
11 FOR Y=3 TO 1 STEP-1
12 Z=PEEK(X+Y)
13 IF Z=7 THEN Z=15
14 IF Z=112 THEN Z=240
15 IF Z=119 THEN Z=255
16 POKE (X+Y),Z:W=(X+Y)-1
17 NEXT:POKE W,Z:NEXT
18 POKE 57507,1:REM Screen on

```

```

10 REM This program will convert the
20 REM RAM based character set so that
30 REM the characters produced by the
40 REM shift key will be the reverse
50 REM of their non shifted form.....
60 REM
70 REM In short shifted A is reverse
80 REM field A.
90 REM
100 POKE 57507,0 : REM Turn screen off
110 F=55304 : REM From address
120 T=55816 : REM To address
130 FOR X=0 TO 207
140 POKE T+X,255-PEEK(F+X)
150 NEXT
160 POKE 57507,1 : REM Turn screen on

```

NOTE

====

In graphics BASIC 4 this routine can be replaced by GRAPH0.

GRAPH0 is a composite command which performs the following tasks:

- 1) CLEAR SCREEN
- 2) RESET HI-RES POINTERS
- 3) COPY ROM CHARACTER SET TO RAM CHARACTER SET
- 4) SWITCH SYSTEM TO HI-RES

```
10 REM This program will copy the
20 REM full character set from ROM
30 REM to RAM.....
40 REM
50 REM
60 F=59392:REM ROM address (from)
70 T=55296:REM RAM address (to)
80 POKE 57507,0:REM Blank screen
90 REM
100 FOR X=0 TO 2047:REM 2K Bytes
110 REM
120 POKE T+X,PEEK (F+X)
130 REM
140 NEXT X
150 POKE 57507,1:REM Restore screen
```