

The MSX Redbook appendix

Scanned and converted to PDF by HansO, 2001

7. MACHINE CODE PROGRAMS

This chapter contains a number of machine code programs to illustrate the use of MSX system resources. Although prepared with the ZEN Assembler they are designed to run from BASIC and, if necessary, may be entered in hex form using the loader shown below. The code should then be saved on cassette before any attempt is made to run it.

```
10 CLEAR 200,&HE000
20 ADDR=&HE000
30 PRINT RIGHT$("000"+HEX$(ADDR),4);
40 INPUT D$
50 POKE ADDR,VAL("&H"+D$)
60 ADDR=ADDR+1
70 GOTO 30
```

All the programs start at address E000H and are entered at the same point. Unless stated otherwise no parameter need be passed to a program, execution may therefore be initiated with a simple DEFUSR=&HE000:?USR(0) statement.

Keyboard Matrix

This program displays the keyboard matrix on the screen so that key depressions may be directly observed. The program may be terminated by pressing the CTRL and STOP keys. Note that spurious key depressions can be produced under certain circumstances if more than three or four keys are pressed at one time. This is a characteristic of all matrix type keyboards.

```
ORG 0E000H
LOAD 0E000H

; *****
; * BIOS STANDARD ROUTINES *
; *****

INITXT:EQU 006CH
CHPUT: EQU 00A2H
SNSMAT:EQU 0141H
BREAKX:EQU 00B7H

; *****
; * WORKSPACE VARIABLES *
; *****

INTFLG:EQU 0FC9BH
```

7. MACHINE CODE PROGRAMS

```

; *****
; *   CONTROL CHARACTERS   *
; *****

LF:   EQU 10
HOME: EQU 11
CR:   EQU 13

E000 CD6C00  MATRIX:CALL INITXT      ; SCREEN 0
E003 3E0B   MX1:  LD  A,HOME        ;
E005 CDA200          CALL CHPUT      ; Home Cursor
E008 AF      XOR  A                ; A=KBD row
E009 F5      MX2:  PUSH AF          ;
E00A CD4101          CALL SNSMAT     ; Read a row
E00D 0608      LD  B,8             ; Eight cols
E00F 07      MX3:  RLCA             ; Select col
E010 F5      PUSH AF              ;
E011 E601      AND  1              ;
E013 C630      ADD  A,"0"          ; Result
E015 CDA200          CALL CHPUT      ; Display col
E018 F1      POP  AF              ;
E019 10F4      DJNZ MX3           ;
E01B 3E0D      LD  A,CR           ; Newline
E01D CDA200          CALL CHPUT      ;
E020 3E0A      LD  A,LF           ;
E022 CDA200          CALL CHPUT      ;
E025 F1      POP  AF              ; A=KBD row
E026 3C      INC  A                ; Next row
E027 FE08      CP   11             ; Finished?
E029 20DE      JR   NZ,MX2        ;
E02B CDB700          CALL BREAKX    ; CTRL-STOP
E02E 30D3      JR   NC,MX1        ; Continue
E030 AF      XOR  A                ;
E031 329BFC      LD  (INTFLG),A     ; Clear possible STOP
E034 C9      RET                   ; Back to BASIC

      END

```

40 Column Graphics Text

This program prints text on the Graphics Mode screen at forty characters per line. The string to be displayed is passed as the USR call parameter, for example A\$=USR("something"). There is no need to open a GRP file beforehand, the only requirement of the program is that the screen be in the correct mode. The heart of the program is functionally equivalent to the GRPPRT standard routine but only the first six dot columns of a given character pattern are placed on the screen instead of eight. As with GRPPRT the pattern is placed at the current graphics position and the only control character recognised is ASCII CR (13) which functions as a combined CR, LF. Unlike the GRPPRT standard routine characters printed at negative coordinates, but which overlap the screen, will be correctly displayed. The program is currently set up to perform an auto

7. MACHINE CODE PROGRAMS

linefeed after dot column 239, thus giving exactly forty characters per line. If required this may be changed, via the constant in the RMDCOL subroutine, so that the full width of the screen is usable.

```

                ORG 0E000H
                LOAD 0E000H

; *****
; * BIOS STANDARD ROUTINES *
; *****

RDSLT: EQU 000CH
CNVCHR: EQU 00ABH
MAPXYC: EQU 0111H
SETC: EQU 0120H

; *****
; * WORKSPACE VARIABLES *
; *****

FORCLR: EQU 0F3E9H
ATRBYT: EQU 0F3F2H
CGPNT: EQU 0F91FH
PATWRK: EQU 0FC40H
SCRMOD: EQU 0FCAFH
GRPACK: EQU 0FCB7H
GRPACY: EQU 0FCB9H

; *****
; * CONTROL CHARACTERS *
; *****

CR: EQU 13

E000 FE03   GFORTY: CP 3           ; String type?
E002 C0     RET NZ                ;
E003 3AAFFC LD A, (SCRMOD)        ; Mode
E006 FE02   CP 2                  ; Graphics?
E008 C0     RET NZ                ;
E009 EB     EX DE, HL             ; HL->Descriptor
E00A 46     LD B, (HL)            ; B=String len
E00B 23     INC HL                ;
E00C 5E     LD E, (HL)            ; Address LSB
E00D 23     INC HL                ;
E00E 56     LD D, (HL)            ; DE->String
E00F 04     INC B                  ;
E010 05     GF2: DEC B             ; Finished?
E011 C8     RET Z                  ;
E012 1A     LD A, (DE)            ; A=Chr from string
E013 CD19E0 CALL GPRINT             ; Print it
E016 13     INC DE                 ;
E017 18F7   JR GF2                ; Next chr

```

7. MACHINE CODE PROGRAMS

```

E019 F5          GPRINT: PUSH AF          ;
E01A C5          PUSH BC                 ;
E01B D5          PUSH DE                 ;
E01C E5          PUSH HL                 ;
E01D FDE5       PUSH IY                 ;
E01F ED4BB7FC   LD BC,(GRPACX)          ; BC=X coord
E023 ED5EB9FC   LD DE,(GRPACY)         ; DE=Y coord
E027 CD39E0     CALL GDC                 ; Decode chr
E02A ED43E7FC   LD (GRPACX),BC         ; New X coord
E02E ED53B9FC   LD (GRPACY),DE         ; New Y coord
E032 FDE1       POP IY                  ;
E034 E1         POP HL                  ;
E035 D1         POP DE                  ;
E036 C1         POP BC                  ;
E037 F1         POP AF                  ;
E038 C9         RET                     ;

E039 CDAB00     GDC:  CALL CNVCHR         ; Check graphic
E03C D0         RET NC                  ; NC=Header
E03D 2007       JR NZ,GD2              ; NZ=Converted
E03F FE0D       CP CR                   ; Carriage Return?
E041 2873       JR Z,GCRLF             ;
E043 FE20       CP 20H                  ; Other control?
E045 D8         RET C                   ; Ignore
E046 6F         GD2:  LD L,A            ;
E047 2600       LD H,0                  ; HL=Chr code
E049 29         ADD HL,HL               ;
E04A 29         ADD HL,HL               ;
E04B 29         ADD HL,HL               ; HL=Chr*8
E04C C5         PUSH BC                 ; X coord
E04D D5         PUSH DE                 ; Y coord
E04E ED5B20F9   LD DE,(CGPNT+1)        ; Character set
E052 19         ADD HL,DE               ; HL->Pattern
E053 1140FC     LD DE,PATWRK           ; DE->Buffer
E056 0608       LD B,8                  ; Eight byte pattern
E058 C5         GD3:  PUSH BC           ;
E059 D5         PUSH DE                 ;
E05A 3A1FF9     LD A,(CGPNT)           ; Slot ID
E05D CD0C00     CALL RDSLT              ; Get pattern
E060 FB         EI                      ;
E061 D1         POP DE                  ;
E062 C1         POP BC                  ;
E063 12         LD (DE),A              ; Put in buffer
E064 13         INC DE                  ;
E065 23         INC HL                  ;
E066 10F0       DJNZ GD3                ; Next
E068 D1         POP DE                  ; Y coord
E069 C1         POP BC                  ; X coord
E06A 3AE9F3     LD A,(FORCLR)          ; Current colour
E06D 32F2F3     LD (ATRBYT),A         ; Set ink
E070 FD2140FC   LD IY,PATWRK          ; IY->Patterns
E074 D5         PUSH DE                 ;
E075 2608       LD H,8                  ; Max dot rows
E077 CB7A       GD4:  BIT 7,D           ; Pos Y coord?

```

7. MACHINE CODE PROGRAMS

```

E079 202A      JR    NZ,GD8      ;
E07B CDBFE0    CALL  BMDROW     ; Bottom most row?
E07E 382B      JR    C,GD9      ; C=Y too large
E080 C5        PUSH  BC        ;
E081 2E06      LD    L,6        ; Max dot cols
E083 FD7E00    LD    A,(IY+0)   ; A=Pattern row
E086 CB78      GD5:  BIT   7,B        ; Pos X coord
E088 2015      JR    NZ,GD6      ;
E08A CDC8E0    CALL  RMDCOL     ; Rightmost col?
E08D 3815      JR    C,GD7      ; C=X too large
E08F CB7F      BIT   7,A        ; Pattern bit
E091 280C      JR    Z,GD6      ; Z=0 Pixel
E093 F5        PUSH  AF        ;
E094 D5        PUSH  DE        ;
E095 E5        PUSH  HL        ;
E096 CD1101    CALL  MAPXYC     ; Map coords
E099 CD2001    CALL  SETC      ; Set pixel
E09C E1        POP   HL        ;
E09D D1        POP   DE        ;
E09E F1        POP   AF        ;
E09F 07      GD6:  RLCA      ; Shift pattern
E0A0 03        INC   BC        ; X=X+1
E0A1 2D        DEC   L        ; Finished dot cols?
E0A2 20E2      JR    NZ,GD5     ;
E0A4 C1      GD7:  POP   BC        ; Initial X coord
E0A5 FD23      GD8:  INC   IY      ; Next pattern byte
E0A7 13        INC   DE        ; Y=Y+1
E0A8 25        DEC   H        ; Finished dot rows?
E0A9 20CC      JR    NZ,GD4     ;
E0AB D1      GD9:  POP   DE        ; Initial Y coord
E0AC 210600    LD    HL,6        ; Step
E0AF 09        ADD   HL,BC      ; X=X+6
E0B0 44        LD    B,H        ;
E0B1 4D        LD    C,L        ; BC=New X coord
E0B2 CDC8E0    CALL  RMDCOL     ; Rightmost col?
E0B5 D0        RET   NC        ;

E0B6 010000    GCRLE: LD   BC,0        ; X=0
E0B9 210800    LD    HL,8        ;
E0BC 19        ADD   HL,DE      ;
E0BD EB        EX    DE,HL      ; Y=Y+8
E0BE C9        RET                    ;

E0BF E5      BMDROW: PUSH  HL      ;
E0C0 21BF00    LD    HL,191      ; Bottom dot row
E0C3 B7        OR    A        ;
E0C4 ED52      SBC   HL,DE      ; Check Y coord
E0C6 E1        POP   HL        ;
E0C7 C9        RET                    ; C=Below screen

E0C8 E5      RMDCOL: PUSH  HL      ;
E0C9 21EF00    LD    HL,239      ; Rightmost dot col
E0CC B7        OR    A        ;
E0CD ED42      SBC   HL,BC      ; Check X coord

```

7. MACHINE CODE PROGRAMS

```

EOCF E1          POP HL          ;
EOD0 C9          RET             ; C=Beyond right

                END
    
```

String Bubble Sort

This program will sort the contents of a string Array into ascending alphabetic order. The location of the Array is passed as the USR call parameter, for example V=USR(VARPTR(A\$(0))). There are no restrictions on the size of the Array or on its contents but it must only have one dimension. The program is based on the classic bubble sort algorithm where string pairs are compared and their positions swapped if the second is smaller than the first. A 250 element Array of randomly generated strings will be sorted in approximately 2.5 seconds. The equivalent BASIC program takes over six minutes.

```

                ORG 0E000H
                LOAD 0E000H

E000 FE02      SORT: CP 2          ; Integer type?
E002 C0        RET NZ            ;
E003 23        INC HL            ; HL->DAC+1
E004 23        INC HL            ; HL->DAC+2
E005 5E        LD E,(HL)         ; Address LSB
E006 23        INC HL            ; HL->DAC+3
E007 56        LD D,(HL)         ; Address MSB
E008 EB        EX DE,HL          ; HL->A$(0)
E009 E5        PUSH HL           ;
E00A DDE1      POP IX            ; IX->A$(0)
E00C DD7EF8    LD A,(IX-8)       ; Array type
E00F FE03      CP 3              ; String Array?
E011 C0        RET NZ            ;
E012 DD7BFD    LD A,(IX-3)       ; Dimension
E015 3D        DEC A             ; Single dimension?
E016 C0        RET NZ            ;
E017 DD4EFE    LD C,(IX-2)       ;
E01A DD46FF    LD B,(IX-1)       ; BC=Element count
E01D C5        SR2: PUSH BC       ;
E01E E5        PUSH HL           ; HL->Dsc(N)
E01F 46        SR3: LD B,(HL)    ; B=Len(N)
E020 23        INC HL            ;
E021 5E        LD E,(HL)         ;
E022 23        INC HL            ;
E023 E5        PUSH HL           ;
E024 56        LD D,(HL)         ; DE->String(N)
E025 23        INC HL            ; HL->Dsc(N+1)
E026 4E        LD C,(HL)         ; C=Len(N+1)
E027 23        INC HL            ;
E028 7E        LD A,(HL)         ;
E029 23        INC HL            ;
E02A E5        PUSH HL           ;
E02B 66        LD H,(HL)         ;
    
```

7. MACHINE CODE PROGRAMS

```

E02C 6F          LD    L,A          ; HL->String(N+1)
E02D EB         EX    DE,HL       ; HL->(N),DE->(N+1)
E02E 04         INC   B           ;
E02F 0C         INC   C           ;
E030 05         SR4:  DEC   B           ; Remaining len(N)
E031 2825       JR    Z,NEXT       ; Z=(N)<=(N+1)
E033 0D         DEC   C           ; Remaining len(N+1)
E034 2808       JR    Z,SWAP       ; Z=(N+1)<(N)
E036 1A         LD    A,(DE)       ; Chr from (N+1)
E037 BE         CP    (HL)       ; Chr from (N)
E038 13         INC   DE           ;
E039 23         INC   HL           ;
E03A 28F4       JR    Z,SR4        ; Same, continue
E03C 301A       JR    NC,NEXT      ; NC=(N)<(N+1)
E03E E1         SWAP: POP  HL           ; HL->Dsc(N+1)
E03F D1         POP  DE           ; DE->Dsc(N)
E040 0603       LD    B,3          ; Descriptor size
E042 1A         SW2:  LD    A,(DE)   ; Swap descriptors
E043 4E         LD    C,(HL)
E044 77         LD    (HL),A
E045 79         LD    A,C
E046 12         LD    (DE),A
E047 1B         DEC   DE
E048 2B         DEC   HL
E049 10F7       DJNZ  SW2
E04B DDE5       PUSH IX
E04D E1         POP  HL           ; HL->A$(0)
E04E B7         OR    A
E04F ED52       SBC  HL,DE       ; At Array start?
E051 3007       JR    NC,NX2      ; NC=At start
E053 1B         DEC   DE           ; Back up
E054 1B         DEC   DE
E055 BB         EX    DE,HL       ; HL->Dsc(N-1)
E056 18C7       JR    SR3          ; Go check again
E058 E1         NEXT: POP  HL           ; Lose junk
E059 E1         POP  HL
E05A E1         NX2:  POP  HL           ; HL->Dsc(N)
E05B C1         POP  BC           ; BC=Element count
E05C 23         INC   HL           ; Next descriptor
E05D 23         INC   HL
E05E 23         INC   HL
E05F 0B         DEC   BC
E060 78         LD    A,B
E061 B1         OR    C           ; Finished?
E062 20B9       JR    NZ,SR2
E064 C9         RET

```

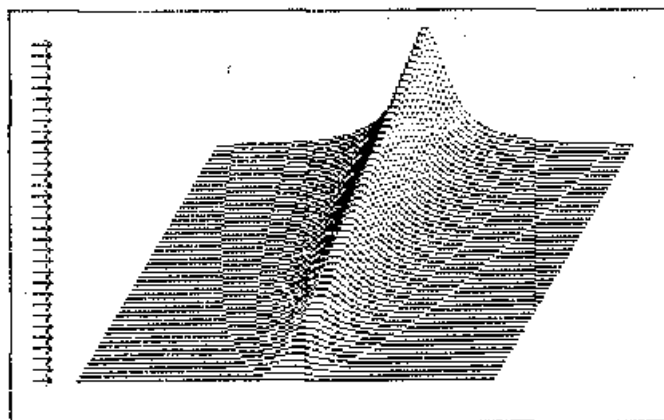
END

Graphics Screen Dump

This program will dump the screen contents, in any mode, to the printer. When first activated via a USR call the program merely patches itself into the interrupt handler keyscan hook.

7. MACHINE CODE PROGRAMS

Once the program has installed itself it effectively becomes an extension of the interrupt handler and a screen dump may then be initiated from any part of the system simply by pressing the ESC key. If necessary the dump can be terminated by pressing the CTRL and STOP keys. An example of a Graphics Mode screen, in which all thirty-two sprites are active, is shown below:



The simplest method of generating a screen dump is to copy all the character codes from the Name Table to the printer. However this would only work in the two text modes, the sprites could not be displayed and the result would reflect the printer's internal character set rather than the VDP character set. The program therefore reproduces the screen as a 240/256x192 bit image on the printer in all modes, each point in the image being derived from the colour code of the corresponding point on the screen. No dot for colours 0 to 7 and a dot for colours 8 to 15.

The colour code for a given point is obtained by first examining the thirty-two sprites in sequence to determine whether any one overlaps it. If every sprite is transparent at the point then the character plane is examined. This is done by using the point coordinates to locate the corresponding entry in the Name Table then, via the character code, to isolate the relevant bit in the associated pattern. If the bit's colour code is found to be transparent the background plane colour is returned.

Note that the control code sequences used in the program are for the Epson FX80 printer. These are marked in the listings in case another printer is to be used. One sequence is used to enter bit image mode at the start of a 240/256 byte line (each byte defines eight vertical dots) and one sequence is used to

7. MACHINE CODE PROGRAMS

initiate a paper feed at the end of the line. The program is generally optimised for speed, rather than for minimal code, and takes about five seconds plus printer time to produce the 46,080/49,152 dots in the image.

```

                ORG 0E000H
                LOAD 0E000H

; *****
; *   BIOS STANDARD ROUTINES   *
; *****

RDVRM: EQU 004AH
CALATR: EQU 0087H
LPTOUT: EQU 00A5H

; *****
; *   WORKSPACE VARIABLES     *
; *****

T32COL: EQU 0F3BFH
GRPNAM: EQU 0F3C7H
GRPCOL: EQU 0F3C9H
GRPCGP: EQU 0F3CBH
MLTNAM: EQU 0F3D1H
MLTCGP: EQU 0F3D5H
RG1SAV: EQU 0F3E0H
RG7SAV: EQU 0F3E6H
NAMBAS: EQU 0F922H
CGPBAS: EQU 0F924H
PATBAS: EQU 0F926H
ATRBAS: EQU 0F928H
SCRMOD: EQU 0FCAFH
HKEYC: EQU 0FDCCH

; *****
; *   CONTROL CHARACTERS     *
; *****

CR: EQU 13
ESC: EQU 27

E000 3ACCFD ENTRY: LD A,(HKEYC) ; Hook
E003 FEC9 CP 0C9H ; Free to use?
E005 C0 RET NZ ;
E006 2112E0 LD HL,DUMP ; Where to go
E009 22CDFD LD (HKEYC+1),HL ; Redirect hook
E00C 3ECD LD A,0CDH ; CALL
E00E 32CCFD LD (HKEYC),A ;
E011 C9 RET ;

E012 FE3A DUMP: CP 3AH ; ESC key number?
E014 C0 RET NZ ;
E015 F5 PUSH AF ;

```

7. MACHINE CODE PROGRAMS

```

E016 C5          PUSH BC          ;
E017 D5          PUSH DE          ;
E018 E5          PUSH HL          ;
E019 ED734FE2    LD (BRKSTK),SP    ; For CTRL-STOP
E01D 0E00        LD C,0          ; C=Row
E01F 3AAFFC      DU1: LD A,(SCRMOD) ; Mode
E022 B7          OR A            ;
E023 21F000      LD HL,240        ; T40 Dots per row
E026 112806      LD DE,6*256+40    ;
E029 2806        JR Z,DU2        ;
E02B 210001      LD HL,256        ; T32,GRP,MLT Dots
E02E 112008      LD DE,8*256+32    ;
E031 3E1B        DU2: LD A,ESC     ; ***** FX80 *****
E033 CD8DE0      CALL PRINT      ; * *
E036 3E4B        LD A,"K"        ; * Bit mode *
E038 CD8DE0      CALL PRINT      ; * *
E03B 7D          LD A,L          ; * Bytes LSB *
E03C CD8DE0      CALL PRINT      ; * *
E03F 7C          LD A,H          ; * Bytes MSB *
E040 CD8DE0      CALL PRINT      ; *****
E043 0600        LD B,0          ; B=Column
E045 CD97E0      DU3: CALL CELL    ; Do an 8x8 cell
E048 D5          PUSH DE          ;
E049 C5          PUSH BC          ;
E04A 2151E2      LD HL,CBUFF     ; HL->Colours
E04D 42          LD B,D          ; B=Dot cols (6 or 8)
E04E 110800      LD DE,8        ; CBUFF offset
E051 C5          DU4: PUSH BC      ;
E052 E5          PUSH HL          ;
E053 0608        LD B,8          ; B=Dot rows
E055 7E          DU5: LD A,(HL)    ; A=Colour code
E056 FE08        CP 8            ; Dark or light?
E058 3F          CCF            ; Light=Print dot
E059 CB11        RL C            ; Build result
E05B 19          ADD HL,DE        ; Next dot row
E05C 10F7        DJNZ DU5        ;
E05E 79          LD A,C          ; 8 Vertical dots
E05F CD8DE0      CALL PRINT      ;
E062 E1          POP HL          ;
E063 C1          POP BC          ;
E064 23          INC HL          ; Next dot col
E065 10EA        DJNZ DU4        ;
E067 C1          POP BC          ;
E068 D1          POP DE          ;
E069 04          INC B            ; Next column
E06A 78          LD A,B          ;
E06B BB          CP E            ; End of row?
E06C 20D7        JR NZ,DU3       ;
E06E 3E0D        LD A,CR        ; Head left
E070 CD8DE0      CALL PRINT      ;
E073 3E1B        LD A,ESC     ; ***** FX80 *****
E075 CD8DE0      CALL PRINT      ; * *
E078 3E4A        LD A,"J"        ; * Paper feed *
E07A CD8DE0      CALL PRINT      ; * *

```

7. MACHINE CODE PROGRAMS

```

E07D 3E18          LD  A,24          ; * 24/216= 1/9" *
E07F CD8DE0       CALL PRINT        ; *****
E082 0C           INC  C           ; Next row
E083 79          LD  A,C          ;
E084 FE18        CP   24          ; Finished screen?
E086 2097        JR   NZ,DU1       ;
E088 E1          DU6: POP  HL          ;
E089 D1          POP  DE          ;
E08A C1          POP  BC          ;
E08B F1          POP  AF          ;
E08C C9          RET           ;

E08D CDA500      PRINT: CALL LPTOUT ; To printer
E090 D0          RET  NC          ; CTRL-STOP?
E091 ED7B4FE2   LD   SP,(BRKSTK) ; Restore stack
E095 18F1        JR   DU6        ; Terminate program

E097 C5          CELL: PUSH BC      ;
E098 D5          PUSH DE          ;
E099 E5          PUSH HL          ;
E09A FDE5       PUSH IY          ;
E09C 2151E2     LD   HL,CBUFF     ; For results
E09F 3E40       LD   A,64         ;
EOA1 3600       CL1: LD   (HL),0   ; Transparent
EOA3 23         INC  HL           ;
EOA4 3D         DEC  A           ; Fill
EOA5 20FA       JR   NZ,CL1       ;
EOA7 3AAFFC     LD   A,(SCRMOD)    ; Mode
EOAA B7         OR   A           ; T40?
EOAB F5         PUSH AF          ;
EOAC C5         PUSH BC          ;
EOAD C469E1     CALL NZ,SPRTES ; Sprites first
EOB0 C1         POP  BC          ;
EOB1 69         LD   L,C         ;
EOB2 2600       LD   H,0         ; HL=Row
EOB4 29         ADD  HL,HL        ;
EOB5 29         ADD  HL,HL        ;
EOB6 29         ADD  HL,HL        ; HL=Row*8
EOB7 5D         LD   E,L         ;
EOB8 54         LD   D,H         ; DE=Row*8
EOB9 29         ADD  HL,HL        ;
EOBA 29         ADD  HL,HL        ; HL=Row*32
EOBB F1         POP  AF          ; Mode
EOBC F5         PUSH AF          ;
EOBD 2001       JR   NZ,CL2       ; T40?
EOBF 19         ADD  HL,DE        ; HL=Row*40
BOC0 58         CL2: LD   E,B      ; DE=Column
EOC1 19         ADD  HL,DE        ;
EOC2 EB         EX   DE,HL        ; DE=NAMTAB offset
EOC3 D602       SUB  2           ; Mode
EOC5 79         LD   A,C         ; A=Row
EOC6 010000     LD   BC,0         ; BC=CGPTAB offset
EOC9 2A24F9     LD   HL,(CGPBAS) ;
EOCC E5         PUSH HL          ;

```

7. MACHINE CODE PROGRAMS

```

E0CD 2A22F9      LD   HL,(NAMBAS)      ;
E0D0 3819        JR   C,CL4           ; C=T40 or T32
E0D2 200C        JR   NZ,CL3         ; NZ=MLT
E0D4 2ACBF3      LD   HL,(GRPCGP)     ; Else GRP
E0D7 E3          EX   (SP),HL      ;
E0D8 2AC7F3      LD   HL,(GRPNAM)    ;
E0DB E618        AND  18H          ; Row MSBs
E0DD 47          LD   B,A          ; 1/3=2kB CGP offset
E0DE 180B        JR   CL4           ;
E0E0 2AD5F3      CL3: LD   HL,(MLTCGP)   ;
E0E3 E3          EX   (SP),HL      ;
E0E4 2AD1F3      LD   HL,(MLTNAM)   ;
E0E7 07          RLCA          ; Row*2
E0E8 E606        AND  6            ;
E0EA 4F          LD   C,A          ; 1/6=2B CGP offset
E0EB 19          CL4: ADD  HL,DE      ; HL->NAMTAB
E0EC CD4A00      CALL RDVRM      ; Get chr code
E0EF 6F          LD   L,A          ;
E0F0 2600        LD   H,0          ; HL=Chr code
E0F2 29          ADD  HL,HL      ;
E0F3 29          ADD  HL,HL      ;
E0F4 29          ADD  HL,HL      ; HL=Chr*8
E0F5 09          ADD  HL,BC      ; GRP,MLT offsets
E0F6 EB          EX   DE,HL      ; DE=CGPTAB offset
E0F7 FDE1        POP  IY          ; IY=CGPTAB base
E0F9 FD19        ADD  IY,DE      ; IY->Pattern
E0FB 2AC9F3      LD   HL,(GRPCOL)    ;
E0FE 19          ADD  HL,DE      ; HL->GRP colours
E0FF 0F          RRCA          ;
E100 0F          RRCA          ;
E101 0F          RRCA          ; Chr code/8
E102 E61F        AND  1FH          ;
E104 4F          LD   C,A          ;
E105 0600        LD   B,0          ;
E107 3AE6F3      LD   A,(RG7SAV)     ; T40 Colours
E10A 57          LD   D,A          ; D=T40 Colours
E10B E60F        AND  0FH          ;
E10D 5F          LD   E,A          ; E=Background colour
E10E F1          POP  AF          ; Mode
E10F E5          PUSH HL      ; STK->GRP Colours
E110 3D          DEC  A          ;
E111 2008        JR   NZ,CL5         ; Z=T32
E113 2ABFF3      LD   HL,(T32COL)    ;
E116 09          ADD  HL,BC      ; HL->T32 Colours
E117 CD4A00      CALL RDVRM      ; Get T32 Colours
E11A 57          LD   D,A          ; D=T32 Colours
E11B 2151E2      CL5: LD   HL,CBUFF    ; Results
E11E 0608        LD   B,8          ; Dot rows
E120 FDE5        CL6: PUSH IY      ;
E122 E3          EX   (SP),HL      ; HL->Pattern
E123 CD4A00      CALL RDVRM      ; Get pattern
E126 4F          LD   C,A          ; C=Pattern
E127 E1          POP  HL          ;
E128 FD23        INC  IY          ; Next dot row

```

7. MACHINE CODE PROGRAMS

```

E12A 3AAFFC      LD  A,(SCRMOD)      ; Mode
E12D D602        SUB  2              ;
E12F 3815        JR   C,CL8       ; C=T40 or T32
E131 280C        JR   Z,CL7       ; Z=GRP
E133 51          LD  D,C          ; MLT Colours=Pattern
E134 0EF0        LD  C,0F0H       ; Dummy MLT pattern
E136 78          LD  A,B          ; Dot row
E137 FE05        CP   5           ; Cell halfway mark?
E139 280B        JR   Z,CL8       ;
E13B FD2B        DEC  IY          ; Back up pattern
E13D 1807        JR   CL8         ;
E13F E3          CL7: EX  (SP),HL   ; HL->GRP Colours
E140 CD4A00      CALL RDVRM       ; Get colours
E143 57          LD  D,A          ; D=GRP Colours
E144 23          INC  HL           ; Next dot row
E145 E3          EX  (SP),HL     ; STR->GRP Colours
E146 C5          CL8: PUSH BC      ;
E147 0608        LD  B,8         ; Dot cols
E149 CB11        CL9: RL  C        ; Dot from pattern
E14B 34          INC  (HL)        ;
E14C 35          DEC  (HL)        ; Check CBUFF clear
E14D 200D        JR   NZ,CL12     ; NZ=Sprite above
E14F 7A          LD  A,D          ; A=Colours
E150 3004        JR   NC,CL10     ; NC=0 Pixel
E152 0F          RRCA            ;
E153 0F          RRCA            ;
E154 0F          RRCA            ;
E155 0F          RRCA            ; Select 1 colour
E156 E60F        CL10: AND  0FH    ;
E158 2001        JR   NZ,CL11     ; Z=Transparent
E15A 7B          LD  A,E          ; Use background
E15B 77          CL11: LD  (HL),A  ; Colour in CBUFF
E15C 23          CL12: INC  HL     ;
E15D 10EA        DJNZ CL9         ; Next dot col
E15F C1          POP  BC          ;
E160 10BE        DJNZ CL6         ; Next dot row
E162 E1          POP  HL          ;
E163 FDE1        POP  IY         ;
E165 E1          POP  HL          ;
E166 D1          POP  DE          ;
E167 C1          POP  BC          ;
E168 C9          RET             ;

E169 78          SPRTES:LD  A,B    ; A=Column
E16A 07          RLCA            ;
E16B 07          RLCA            ;
E16C 07          RLCA            ; A=X coord
E16D C607        ADD  A,7         ; RH edge of cell
E16F 47          LD  B,A          ; B=X coord
E170 79          LD  A,C          ; A=Row
E171 07          RLCA            ;
E172 07          RLCA            ;
E173 07          RLCA            ; A=Y coord
E174 C607        ADD  A,7         ; Bottom of cell

```

7. MACHINE CODE PROGRAMS

```

E176 4F          LD    C,A          ; C=Y coord
E177 AF          XOR    A          ; Sprite number
E178 CD8700     SS1:  CALL CALATR       ; HL->Attributes
E17B 57          LD    D,A          ; D=Sprite number
E17C CD4A00     CALL RDVRM       ; Get Sprite Y
E17F FED0       CP    208         ; Terminator?
E181 C8         RET    Z          ;
E182 D5         PUSH DE         ;
E183 C5         PUSH BC         ;
E184 CD8FE1     CALL SPRITE      ; Do a sprite
E187 C1         POP    BC         ;
E188 F1         POP    AF         ;
E189 3C         INC    A          ; Next sprite number
E18A FE20       CP    32         ; Done all?
E18C 20EA       JR    NZ,SS1     ;
E18E C9         RET             ;

E18F 91         SPRITE:SUB  C      ; (SY-Y)
E190 2F         CPL             ; Make (Y-SY)
E191 FE27       CP    39         ; Possible overlap?
E193 D0         RET    NC         ;
E194 4F         LD    C,A          ; C=(Y-SY)
E195 23         INC    HL         ;
E196 CD4A00     CALL RDVRM       ; Get Sprite X
E199 5F         LD    E,A          ;
E19A 78         LD    A,B          ; A=X coord
E19B 93         SUB    E          ;
E19C 5F         LD    E,A          ; E=(X-SX)
E19D 9F         SBC    A,A        ; Make 16 bit
E19E 57         LD    D,A          ; DE=(X-SX)
E19F 23         INC    HL         ;
E1A0 CD4A00     CALL RDVRM       ; Get pattern#
E1A3 47         LD    B,A          ;
E1A4 23         INC    HL         ;
E1A5 CD4A00     CALL RDVRM       ; Get BC & Colour
E1A6 CB7F       BIT    7,A        ; Early clock?
E1AA 2805       JR    Z,SP1      ;
E1AC 212000     LD    HL,32       ;
E1AF 19         ADD    HL,DE      ; Increase (X-SX)
E1B0 EB         EX    DE,HL      ;
E1B1 14         SP1:  INC    D      ;
E1B2 15         DEC    D          ; (X-SX)>255 or neg?
E1B3 C0         RET    NZ        ; NZ=Outside cell
E1B4 E60F       AND    0FH        ; Colour
E1B6 C8         RET    Z          ; Z=Transparent
E1B7 57         LD    D,A          ; D=Colour
E1B8 3AE0F3     LD    A,(RGLSAV)    ; Flags
E1BB CB4F       BIT    1,A        ; SIZE
E1BD 0F         RRCA           ; MAG
E1BE 3E08       LD    A,8         ; Minimum size
E1C0 3001       JR    NC,SP2     ;
E1C2 87         ADD    A,A        ; Double for MAG
E1C3 2805       SP2:  JR    Z,SP3  ;
E1C5 CB80       RES    0,B        ; Change pattern#

```

7. MACHINE CODE PROGRAMS

```

E1C7 CB88      RES 1,B      ;
E1C9 87        ADD A,A      ; Double for SIZE
E1CA 6F        SP3: LD L,A    ; L=Sprite size
E1CB C606      ADD A,6     ; Allow cell size
E1CD B9        CP C        ;
E1CE D8        RET C       ; Sprite above
E1CF BB        CP E        ;
E1D0 D8        RET C       ; Sprite to left
E1D1 79        LD A,C      ;
E1D2 D607      SUB 7       ; (Y-SY) from top
E1D4 4F        LD C,A      ;
E1D5 7D        LD A,L      ; A=Sprite size
E1D6 2608      LD H,8      ; Max dot rows
E1D8 3808      JR C,SP5    ; C=Below cell top
E1DA 91        SUB C       ; A=Dot row overlap
E1DB FE09      CP 9        ;
E1DD 3802      JR C,SP4    ;
E1DF 3E08      LD A,8      ;
E1E1 67        SP4: LD H,A   ; H=Row overlap
E1E2 7B        SP5: LD A,E   ;
E1E3 D607      SUB 7       ; (X-SX) from cell LH
E1E5 5F        LD E,A      ;
E1E6 7D        LD A,L      ; A=Sprite size
E1E7 2E08      LD L,8      ; Max dot cols
E1E9 3808      JR C,SP7    ; C=Past cell LH
E1EB 93        SUB E       ; A=Dot col overlap
E1EC FE09      CP 9        ;
E1EE 3802      JR C,SP6    ;
E1F0 3E08      LD A,8      ;
E1F2 6F        SP6: LD L,A   ; L=Col overlap
E1F3 FD2151E2 SP7: LD IY,CBUFF ; Results
E1F7 D5        SP8: PUSH DE  ;
E1F8 CB79      BIT 7,C     ; Reached sprite?
E1FA 2048      JF NZ,SP15  ;
E1FC E5        PUSH HL    ;
E1FD FDE5      PUSH IY    ;
E1FF CB7B      SP9: BIT 7,E  ; Reached sprite?
E201 2038      JR NZ,SP14  ;
E203 FD7E00    LD A,(IY+0) ; CBUFF
E206 B7        OR A       ; Transparent?
E207 2032      JR NZ,SP14  ;
E209 C5        PUSH BC    ;
E20A D5        PUSH DE    ;
E20B E5        PUSH HL    ;
E20C 3AE0F3    LD A,(RGISAV) ; Flags
E20F 0F        RRCA       ; MAG
E210 3004      JR NC,SP10  ;
E212 CB39      SRL C      ; (Y-SY)/2
E214 CB3B      SPL E       ; (X-SX)/2
E216 CB5B      SP10: BIT 3,E ; (X-SX)>7?
E218 2804      JR Z,SP11  ;
E21A CB9B      RES 3,E     ; (X-SX)-8
E21C CBE1      SET 4,C     ; (Y-SY)+16
E21E 68        SP11: LD L,B ;

```


7. MACHINE CODE PROGRAMS

```

E21F 2600          LD  H,0           ; HL=Pattern#
E221 44           LD  B,H           ; BC=Y offset
E222 29           ADD  HL,HL         ;
E223 29           ADD  HL,HL         ;
E224 29           ADD  HL,HL         ; HL=Pattern*8
E225 09           ADD  HL,BC         ; Select dot row
E226 ED4B26F9    LD  BC,(PATBAS)      ;
E22A 09           ADD  HL,BC         ; HL->Pattern
E22B CD4A00      CALL RDVRM        ; Get dot row
E22E 1C           INC  E             ;
E22F 07          SP12: RLCA         ; Select dot col
E230 1D           DEC  E             ;
E231 20FC        JR  NZ,SP12        ;
E233 3003        JR  NC,SP13        ; NC=0 Pixel
E235 FD7200      LD  (IY+0),D       ; Colour in CBUFF
E238 E1          SP13: POP  HL         ;
E239 D1          POP  DE         ;
E23A C1          POP  BC         ;
E23B FD23        SP14: INC  IY         ;
E23D 1C          INC  E             ; Right a dot col
E23E 2D          DEC  L             ; Finished cols?
E23F 20BE        JR  NZ,SP9        ;
E241 FDE1        POP  IY         ;
E243 E1          POP  HL         ;
E244 110800      SP15: LD  DE,8       ;
E247 FD19        ADD  IY,DE         ;
E249 D1          POP  DE         ;
E24A 0C          INC  C             ; Down a dot row
E24B 25          DEC  H             ; Finished?
E24C 20A9        JR  NZ,SP8        ;
E24E C9          RET              ;
E24F 0000      BRKSTK:DEFW 0       ; Break stack

```

```

; *****
; * This buffer holds the 64 *
; * colour codes produced by *
; * a cell scan: *
; * *
; * CCCCCCCC Bytes 00-07 *
; * CCCCCCCC Bytes 08-15 *
; * CCCCCCCC Bytes 16-23 *
; * CCCCCCCC Bytes 24-31 *
; * CCCCCCCC Bytes 32-39 *
; * CCCCCCCC Bytes 40-47 *
; * CCCCCCCC Bytes 48-55 *
; * CCCCCCCC Bytes 56-63 *
; * *
; *****

```

```

E251          CBUFF: DEFS 64       ; Cell buffer

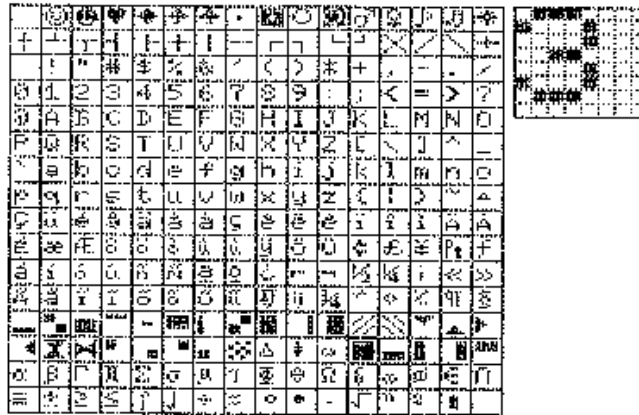
```

END

7. MACHINE CODE PROGRAMS

Character Editor

This program allows the MSX character patterns to be modified. When the program is first entered it copies the 2KB character set from its present location (usually the MSX ROM) to the CHRTAB buffer (E2A3H to EAA2H) and sets up the screen as shown below:



The program has two levels of operation, command and edit, with the RETURN key being used to toggle between them. In command mode the four arrow keys are used to select the character for editing. This is marked by a large cursor and is also displayed in magnified form on the right hand side of the screen. The "Q" key will quit the program and return to BASIC. The "A" key is used to adopt the character set, that is, to make it the system character set. When the character set is adopted it is copied to the highest part of memory (EB80H to F37FH) and its Slot ID and address placed in CGPNT.

In edit mode the four arrow keys are used to select the dot for editing, this is marked by a small cursor. The SPACE key will erase the current dot and the "." key set it. As the pattern is modified the character menu on the left hand side of the screen is updated.

The character set in CHRTAB may be saved on cassette using a "BSAVE" statement and later re-loaded with a "BLOAD" statement. The ADOPT subroutine should be saved with the patterns and executed upon re-loading so that the system adopts the new character set. Alternatively the character set alone can be saved and its Slot ID and address placed in CGPNT upon re-loading using BASIC statements. Note that altering the character patterns does not affect the operation of the MSX system in the slightest.

7. MACHINE CODE PROGRAMS

ORG 0E000H
LOAD 0E000H

```
; *****
; * BIOS STANDARD ROUTINES *
; *****
```

```
RDSLT: EQU 000CH
RDVRM: EQU 004AH
WRTVRM: EQU 004DH
FILVRM: EQU 0056H
INIGRP: EQU 0072H
CHSNS: EQU 009CH
CHGET: EQU 009FH
MAPXYC: EQU 0111H
FETCHC: EQU 0114H
RSLREG: EQU 0138H
```

```
; *****
; * WORKSPACE VARIABLES *
; *****
```

```
GRPCOL: EQU 0F3C9H
FORCLR: EQU 0F3E9H
BAKCLR: EQU 0F3EAH
CGPNT: EQU 0F91FH
EXPTBL: EQU 0FC11H
SLTTBL: EQU 0FCC5H
```

```
; *****
; * CONTROL CHARACTERS *
; *****
```

```
CR: EQU 13
RIGHT: EQU 28
LEFT: EQU 29
UP: EQU 30
DOWN: EQU 31
```

```
E000 CDF6E0 CHEDIT: CALL INIT ; Cold start
E003 CDBDE0 CH1: CALL CHRMAG ; Magnify chr
E006 CDFEE1 CALL CHRXY ; Chr coords
E009 1608 LD D,8 ; Cursor size
E00B CD2FE2 CALL GETKEY ; Get command
E00E FE51 CP "Q" ; Quit
E010 C8 RET Z ;
E011 2103E0 LD HL,CH1 ; Set up return
E014 E5 PUSH HL ;
E015 EE41 CP "A" ; Adopt
E017 CA6EE2 JP Z,ADOPT ;
E01A FE0D CP CR ; Edit
E01C 281F JR Z,EDIT ;
E01E 0E01 LD C,1 ; C=Offset
E020 FE1C CP RIGHT ; Right
```

7. MACHINE CODE PROGRAMS

```

E022 2811          JR    Z,CH2          ;
E024 0EFF          LD    C,0FFH         ;
E026 FE1D          CP    LEFT          ; Left
E028 280E          JR    Z,CH2          ;
E02A 0EF0          LD    C,0F0H         ;
E02C FE1E          CP    UP            ; Up
E02E 2805          JR    Z,CH2          ;
E030 0E10          LD    C,16          ;
E032 FE1F          CP    DOWN         ; Down
E034 C0            RET    NZ           ;
E035 3AA1E2        CH2:  LD    A,(CHNUM)    ; Current chr
E038 81            ADD    A,C          ; Add offset
E039 32A1E2        LD    (CHNUM),A    ; New chr
E03C C9            RET

E03D CDE6E1        EDIT: CALL DOTXY     ; Dot coords
E040 1602          LD    D,2          ; Cursor size
E042 CD2FE2        CALL GETKEY    ; Get command
E045 FE0D          CP    CR            ; Quit
E047 C8            RET    Z            ;
E048 213DE0        LD    HL,EDIT     ; Set up return
E04B E5            PUSH HL          ;
E04C 0100FE        LD    BC,0FE00H   ; AND/OR masks
E04F FE20          CP    " "          ; Space
E051 2824          JR    Z,ED3          ;
E053 0C            INC    C            ; New OR mask
E054 FE2E          CP    "."          ; Dot
E056 281F          JR    Z,ED3          ;
E058 FE1C          CP    RIGHT         ; Right
E05A 2811          JR    Z,ED2          ;
E05C 0EFF          LD    C,0FFH         ; C=Offset
E05E FE1D          CP    LEFT          ; Left
E060 280B          JR    Z,ED2          ;
E062 0EF8          LD    C,0F8H        ;
E064 FE1E          CP    UP            ; Up
E066 2805          JR    Z,ED2          ;
E068 0E08          LD    C,8          ;
E06A FE1F          CP    DOWN         ; Down
E06C C0            RET    NZ           ;
E06D 3AA2E2        ED2:  LD    A,(DOTNUM)    ; Current dot
E070 81            ADD    A,C          ; Add offset
E071 E63F          AND    63          ; Wrap round
E073 32A2E2        LD    (DOTNUM),A  ; New dot
E076 C9            RET

E077 CD1EE2        ED3:  CALL PATPOS    ; IY->Pattern
E07A 3AA2E2        LD    A,(DOTNUM)  ; Current dot
E07D F5            PUSH AF          ;
E07E 0F            RRCA           ;
E07F 0F            RRCA           ;
E080 0F            RRCA           ;
E081 E607          AND    7           ; A=Row
E083 5F            LD    E,A         ;
E084 1600          LD    D,0         ; DE=Row
E086 FD19          ADD    IY,DE      ; IY->Row

```

7. MACHINE CODE PROGRAMS

```

E088 F1          POP AF          ;
E089 E607        AND 7          ; A=Column
E08B 3C          INC A          ;
E08C CB08        ED4: RRC E      ; AND mask
E08E CB09        RRC C          ; OR mask
E090 3D          DEC A          ; Count columns
E091 20F9        JR NZ,ED4      ;
E093 FD7E00      LD A,(IY+0)    ; A=Pattern
E096 A0          AND B          ; Strip old bit
E097 B1          OR C           ; New bit
E098 FD7700      LD (IY+0),A    ; New pattern
E09B CDBDE0      CALL CHR MAG   ; Update magnified

E09E CD1EE2      CHROUT:CALL PATPOS ; IY->Pattern
E0A1 CDFEE1      CALL CHRXY     ; Get coords
E0A4 CDA3E1      CALL MAP       ; Map
E0A7 0608        LD B,8         ; Dot rows
E0A9 D5          COL:  PUSH DE   ;
E0AA E5          PUSH HL        ;
E0AB 3E08        LD A,8         ; Dot cols
E0AD FD5E00      LD E,(IY+0)    ; E=Pattern
E0B0 CDC4E1      CALL SETROW    ; Set row
E0B3 E1          POP HL         ; HL=CLOC
E0B4 D1          POP DE         ; D=CMASK
E0B5 CDB8E1      CALL DOWNP     ; Down a pixel
E0B8 FD23        INC IY         ;
E0BA 10ED        DJNZ COL       ;
E0BC C9          RET            ;

E0BD CD1EE2      CHRMAG:CALL PATPOS ; IY->Pattern
E0C0 0EBF        LD C,191       ; Start X
E0C2 1E07        LD E,7         ; Start Y
E0C4 CDA3E1      CALL MAP       ; Map
E0C7 0608        LD B,8         ; Dot rows
E0C9 0E05        CM1:  LD C,5    ; Row mag
E0CB C5          CM2:  PUSH BC   ;
E0CC D5          PUSH DE        ;
E0CD E5          PUSH HL        ;
E0CE 0608        LD B,8         ; Dot columns
E0D0 FD7E00      LD A,(IY+0)    ; A=Pattern
E0D3 07          CM3:  RLCA      ; Test bit
E0D4 F5          PUSH AF        ;
E0D5 9F          SBC A,A        ; 0=00H,1=FFH
E0D6 5F          LD E,A         ; E=Mag pattern
E0D7 3E05        LD A,5         ; Column mag
E0D9 CBC4E1      CALL SETROW    ; Set row
E0DC CDAEE1      CALL RIGHTP    ; Right a pixel
E0DF CDAEE1      CALL RIGHTP    ; Skip grid
E0E2 F1          POP AF         ;
E0E3 10EE        DJNZ CM3      ;
E0E5 E1          POP HL         ; HL=CLOC
E0E6 D1          POP DE         ; D=CMASK
E0E7 C1          POP BC         ;
E0E8 CDB8E1      CALL DOWNP     ; Down a pixel

```

7. MACHINE CODE PROGRAMS

```

E0EB 0D          DEC  C          ;
E0EC 20DD       JR   NZ,CM2     ;
E0EE CDB8E1     CALL DOWNP   ; Skip grid
E0F1 FD23       INC  IY         ;
E0F3 10D4       DJNZ CML       ;
E0F5 C9         RET            ;

E0F6 010008     INIT: LD  BC,2048   ; Size
E0F9 11A3E2     LD  DE,CHRTAB  ; Destination
E0FC 2A20F9     LD  HL,(CGPNT+1) ; Source
E0FF C5         INI:  PUSH BC      ;
E100 D5         PUSH DE      ;
E101 3A1FF9     LD  A,(CGPNT)   ; Slot ID
E104 CD0C00     CALL RDSLT   ; Read chr pattern
E107 FB         EI            ;
E108 D1         POP  DE        ;
E109 C1         POP  BC        ;
E10A 12         LD  (DE),A     ; Put in buffer
E10B 13         INC  DE        ;
E10C 23         INC  HL        ;
E10D 0B         DEC  BC        ;
E10E 78         LD  A,B        ;
E10F B1         OR   C         ;
E110 20ED       JR   NZ,INI     ;
E112 CD7200     CALL INIGRP  ; SCREEN 2
E115 3AE9F3     LD  A,(FORCLR) ; Colour 1
E118 07         RLCA          ;
E119 07         RLCA          ;
E11A 07         RLCA          ;
E11B 07         RLCA          ;
E11C 4F         LD  C,A        ; C=Colour 1
E11D 3AEAF3     LD  A,(BAKCLR) ; Colour 0
E120 B1         OR   C         ; Mix
E121 010018     LD  BC,6144    ; Colour table size
E124 2AC9F3     LD  HL,(GPCOL)  ; Colour table
E127 CD5600     CALL FILVRM  ; Fill colours
E12A 210BB1     LD  HL,177*256+11 ;
E12D 010AFF     LD  BC,0FFH*256+10 ;
E130 1E06       LD  E,6        ;
E132 3E11       LD  A,17       ;
E134 CD62E1     CALL GRID    ; Draw chr grid
E137 210631     LD  HL,49*256+6   ;
E13A 01BEAA     LD  BC,0AAH*256+190 ;
E13D 1E06       LD  E,6        ;
E13F 3E09       LD  A,9        ;
E141 CD62E1     CALL GRID    ; Draw mag grid
E144 213031     LD  HL,49*256+48  ;
E147 01BEFF     LD  BC,0FFH*256+190 ;
E14A 1E06       LD  E,6        ;
E14C 3E02       LD  A,2        ;
E14E CD62E1     CALL GRID    ; Draw mag box
E151 AF         XOR  A         ;
E152 32A2E2     LD  (DOTNUM),A ; Current dot
E155 21A1E2     LD  HL,CHNUM   ;

```

7. MACHINE CODE PROGRAMS

```

E158 77          LD   (HL),A          ; Current chr
E159 E5          IN2:  PUSH HL          ;
E15A CD9EE0      CALL CHR0UT        ; Display chr
E15D E1          POP   HL          ;
E15E 34          INC   (HL)         ; Next chr
E15F 20F8        JR    NZ,IN2       ; Do 256
E161 C9          RET                ;

E162 F5          GRID:  PUSH AF        ;
E163 C5          PUSH BC          ;
E164 E5          PUSH HL          ;
E165 CDA3E1      CALL MAP          ; Map
E168 C1          POP   BC          ; B=Len,C=Step
E169 F1          POP   AF          ;
E16A 5F          LD    E,A         ; E=Pattern
E16B F1          POP   AF          ; A=Count
E16C F5          PUSH AF          ;
E16D D5          PUSH DE          ;
E16E E5          PUSH HL          ;
E16F F5          GR1:  PUSH AF        ;
E170 C5          PUSH BC          ;
E171 D5          PUSH DE          ;
E172 E5          PUSH HL          ;
E173 78          LD    A,B         ; A=Len
E174 CDC4E1      CALL SETROW        ; Horizontal line
E177 E1          POP   HL          ; HL=CLOC
E178 D1          POP   DE          ; D=CMASK
E179 CDB8E1      GR3:  CALL DOWNP    ; Down a pixel
E17C 0D          DEC   C           ; Done step?
E17D 20FA        JR    NZ,GR3      ;
E17F C1          POP   BC          ;
E180 F1          POP   AF          ; A=Count
E181 3D          DEC   A           ; Done lines?
E182 20EB        JR    NZ,GR1      ;
E184 E1          POP   HL          ; HL=Initial CLOC
E185 D1          POP   DE          ; D=Initial CMASK
E186 F1          POP   AF          ; A=Count
E187 F5          GR4:  PUSH AF        ;
E188 C5          PUSH BC          ;
E189 D5          PUSH DE          ;
E18A E5          PUSH HL          ;
E18B 3E01        GR5:  LD    A,1     ; Line width
E18D CDC4E1      CALL SETROW        ; Thin line
E190 CDB8E1      CALL DOWNP    ; Down a pixel
E193 1DF6        DJNZ GR5         ; Vertical len
E195 E1          POP   HL          ; HL=CLOC
E196 D1          POP   DE          ; D=CMASK
E197 CDAEE1      GR6:  CALL RIGHTP   ; Right a pixel
E19A 0D          DEC   C           ; Done step?
E19B 20FA        JR    NZ,GR6      ;
E19D C1          POP   BC          ;
E19E F1          POP   AF          ; A=Count
E19F 3D          DEC   A           ; Done lines?
E1A0 20E5        JR    NZ,GR4      ;

```

7. MACHINE CODE PROGRAMS

```

E1A2 C9          RET          ;
E1A3 0600      MAP:  LD   B,0    ; X MSB
E1A5 50        LD   D,B        ; Y MSB
E1A6 CD1101    CALL  MAPXYC    ; Map coords
E1A9 CD1401    CALL  FETCHC    ; HL=CLOC
E1AC 57        LD   D,A        ; D=CMASK
E1AD C9        RET          ;

E1AE CB0A      RIGHTP:RRC  D    ; Shift CMASK
E1B0 D0        RET  NC        ; NC=Same cell
E1B1 C5        RP1:  PUSH BC    ;
E1B2 010800    LD   BC,8      ; Offset
E1B5 09        ADD  HL,BC     ; HL=Next cell
E1B6 C1        POP  BC        ;
E1B7 C9        RET          ;

E1B8 23        DOWNP: INC  HL    ; CLOC down
E1B9 7D        LD   A,L      ;
E1BA E607      AND  7        ; Select pixel row
E1BC C0        RET  NZ        ; NZ=Same cell
E1BD C5        PUSH BC      ;
E1BE 01F800    LD   BC,00F8H   ; Offset
E1C1 09        ADD  HL,BC     ; HL=Next cell
E1C2 C1        POP  BC        ;
E1C3 C9        RET          ;

E1C4 C5        SETROW: PUSH BC ;
E1C5 47        LD   B,A      ; B=Count
E1C6 CD4A00    SE1:  CALL  RDVRM ; Get old pattern
E1C9 4F        SE2:  LD   C,A ; C=Old
E1CA 7A        LD   A,D      ; A=CMASK
E1CB 2F        CPL          ; AND mask
E1CC A1        AND  C        ; Strip old bit
E1CD CB03      RLC  E        ; Shift pattern
E1CF 3001      JR   NC,SE3    ; NC=0 Pixel
E1D1 B2        OR   D        ; Set 1 Pixel
E1D2 05        SE3:  DEC  B    ; Finished?
E1D3 280C      JR   Z,SE4    ;
E1D5 CB0A      RRC  D        ; CMASK right
E1D7 30F0      JR   NC,SE2    ; NC=Same cell
E1D9 CD4D00    CALL  WRTVRM   ; Update cell
E1DC CDB1E1    CALL  RP1     ; Next cell
E1DF 18E5      JR   SE1      ; Start again
E1E1 CD4D00    SE4:  CALL  WRTVRM ; Update cell
E1E4 C1        POP  BC        ;
E1E5 C9        RET          ;

E1E6 3AA2E2    DOTXY: LD   A,(DOTNUM) ; Current dot
E1E9 F5        PUSH AF      ;
E1EA E607      AND  7        ; Column
E1EC 07        RLCA        ;
E1ED 4F        LD   C,A      ; C=Col*2
E1EE 07        RLCA        ; A=Col*4

```


7. MACHINE CODE PROGRAMS

```

E1EF 81          ADD  A,C           ; A=Col*6
E1F0 C6BF       ADD  A,191        ; Grid start
E1F2 4F         LD   C,A         ; C=X coord
E1F3 F1        POP  AF          ;
E1F4 E638       AND  38H        ; Row*8
E1F6 0F        RRCA           ;
E1F7 5F        LD   E,A         ; E=Row*4
E1F8 0F        RRCA           ; A=Row*2
E1F9 83        ADD  A,E         ; A=Row*6
E1FA C607       ADD  A,7         ; Grid start
E1FC 5F        LD   E,A         ; E=Y coord
E1FD C9        RET             ;

E1FE 3AA1E2     CHRXY: LD  A,(CHNUM) ; Current chr
E201 F5        PUSH AF          ;
E202 CD14E2     CALL MULT11      ; Column*11
E205 C60C       ADD  A,12        ; Grid start
E207 4F        LD   C,A         ; C=X coord
E208 F1        POP  AF          ;
E209 0F        RRCA           ;
E20A 0F        RRCA           ;
E20B 0F        RRCA           ;
E20C 0F        RRCA           ;
E20D CD14E2     CALL MULT11      ; Row*11
E210 C608       ADD  A,8         ; Grid start
E212 5F        LD   E,A         ; E=Y coord
E213 C9        RET             ;

E214 E60F     MULT11:AND  0FH      ;
E216 57        LD   D,A         ; D=N
E217 07        RLCA           ;
E218 47        LD   B,A         ; B=N*2
E219 07        RLCA           ;
E21A 07        RLCA           ; A=N*8
E21B 80        ADD  A,B         ;
E21C 82        ADD  A,D         ; A=N*11
E21D C9        RET             ;

E21E 3AA1E2     PATPOS:LD  A,(CHNUM) ; Current chr
E221 6F        LD   L,A         ;
E222 2600      LD   H,0         ; HL=Chr
E224 29        ADD  HL,HL       ;
E225 29        ADD  HL,HL       ;
E226 29        ADD  HL,HL       ; HL=Chr*8
E227 EB        EX   DE,HL       ; DE=Chr*8
E228 FD21A3E2  LD   IY,CHRTAB    ; Patterns
E22C FD19      ADD  IY,DE       ; IY->Pattern
E22E C9        RET             ;

E22F 0600     GETKEY:LD  B,0      ; Cursor flag
E231 C5        GE1:  PUSH BC      ; C=X coord
E232 D5        PUSH DE          ; E=Y coord
E233 CD50E2     CALL INVERT     ; Flip cursor
E236 D1        POP  DE          ;

```

7. MACHINE CODE PROGRAMS

```

E237 C1          POP  BC          ;
E238 04          INC  B           ; Flip flag
E239 21401F     LD   HL,8000      ; Blink rate
E23C CD9C00     GE2: CALL CHSNS    ; Check KEYBUF
E23F 2007       JR   NZ,GE3      ; NZ=Got key
E241 2B         DEC  HL          ;
E242 7C         LD   A,H         ;
E243 B5         OR   L           ;
E244 20F6       JR   NZ,GE2      ;
E246 18E9       JR   GE1         ; Time for cursor
E248 CB40       GE3: BIT  0,B     ; Cursor state
E24A C450E2     CALL NZ,INVERT    ; Remove cursor
E24D C39F00     JP   CHGET       ; Collect character

E250 D5         INVERT:PUSH DE   ;
E251 CDA3E1     CALL MAP         ; Map coords
E254 F1         POP  AF         ; A=Cursor size
E255 47         LD   B,A         ; B=Rows
E256 5F         LD   E,A         ; E=Cols
E257 D5         IV1:  PUSH DE    ;
E258 E5         PUSH HL         ;
E259 CD4A00     IV2:  CALL RDVRM  ; Old pattern
E25C AA         XOR  D           ; Flip a bit
E25D CD4D00     CALL WRTVRM    ; Put it back
E260 CDAEE1     CALL RIGHTP    ; Right a pixel
E263 1D         DEC  E           ;
E264 20F3       JR   NZ,IV2     ;
E266 E1         POP  HL         ; HL=CLOC
E267 D1         POP  DE         ; D=CMASK
E268 CDB8E1     CALL DOWNP     ; Down a pixel
E26B 10EA       DJNZ IV1       ;
E26D C9         RET            ;

E26E 010008     ADOPT: LD  BC,2048 ; Size
E271 1180EB     LD  DE,0EB80H    ; Destination
E274 ED5320F9   LD  (CGPNT+1),DE        ;
E278 21A3E2     LD  HL,CHRTAB   ; Source
E27B EDB0       LDIR          ; Copy up high
E27D CD3801     CALL RSLREG    ; Read PSLOT reg
E280 07         RLCA          ;
E281 07         RLCA          ;
E282 E603       AND  3         ; Select Page 3
E284 4F         LD   C,A         ;
E285 0600       LD   B,0         ; BC=Page 3 PSLOT#
E287 21C1FC     LD   HL,EXPTBL  ; Expanders
E28A 09         ADD  HL,BC       ;
E28B CB7E       BIT  7,(HL)     ; PSLOT expanded?
E28D 280E       JR   Z,AD1      ; Z=Normal
E28F 21C5FC     LD   HL,SLTTBL  ; Secondary regs
E292 09         ADD  HL,BC       ;
E293 7E         LD   A,(HL)     ; A=Secondary reg
E294 07         RLCA          ;
E295 07         RLCA          ;
E296 07         RLCA          ;

```

7. MACHINE CODE PROGRAMS

```
E297 07          RLCA          ;
E298 E60C        AND  0CH       ; A=Page 3 SSLOT#
E29A B1          OR   C         ; Mix Page 3 PSLOT#
E29B CBFF        SET  7,A       ; A=Slot ID
E29D 321FF9      AD1:  LD  (CGPNT),A ;
E2A0 C9          RET           ;

E2A1 00          CHRNUM:DEFB 0   ; Current chr
E2A2 00          DOTNUM:DEFB 0   ; Current dot
E2A3             CHRTAB:DEFS 2048 ; Patterns to EAA2H
```

END