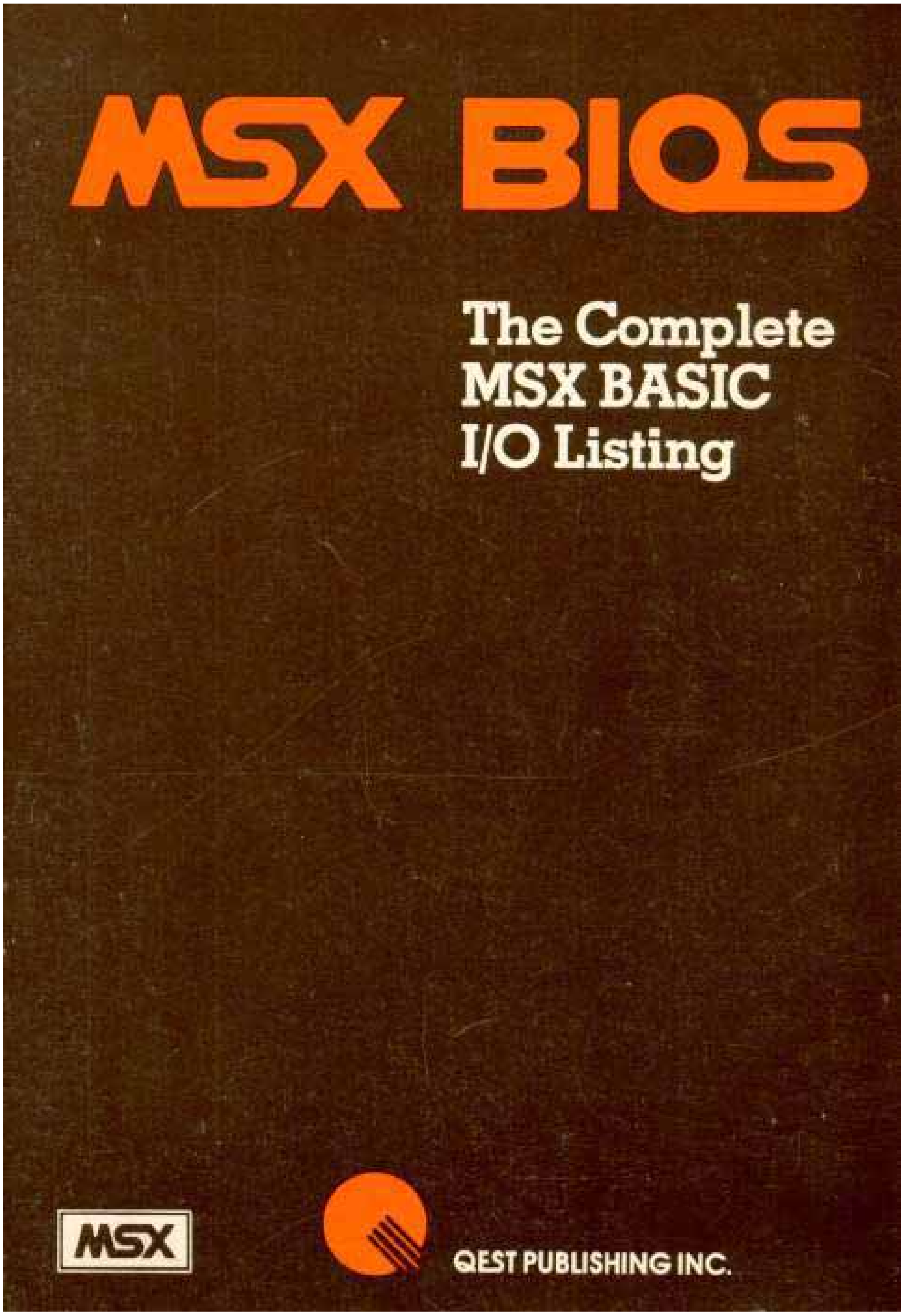


MSX BIOS

The Complete
MSX BASIC
I/O Listing

MSX

QEST PUBLISHING INC.

Edited: January 1985
by Steven M. Ting
Graphic design: Mervin Fong.

MSX® BIOS

Published by

QEST PUBLISHING INC.
39 W. 32nd Street Suite 800
New York, N. Y. 10001

(212) 564-0749
Telex: 650-190-8083 MCI

TABLE OF CONTENTS

```
                                .list
                        ;
                        ;
                        ;       (C) Copyright by ASCII Corp., 1983
                        ;       Proprietary information.  All rights reserved.
                        ;
                        ;       File:   BIOHDR.MAC
                        ;       USE:    Restart calls and ROM entries table
                        ;       Written by Jey Suzuki, Rick Yamashita
                        ;               ASCII Corporation, Japan
                        ;
                        ;       Edit:   January, 1985
                        ;       Reason: Zilog Z80 Mnemonic version and cleanup
                        ;       Edited by:      Steven M. Ting
                        ;
                        ;
                        ; Labels referenced in this listing, are the absolute locations
                        ; within the MSX ROM.  However, "ONLY" this BIOS entry point table,
                        ; and RAM variables are quaranteed to be permanent.
                        ;
                        ; All other locations in the ROM, will be changed without notice.
                        ;
                        SUBTTL -BIOS header- BIOS calls (Basic Interpreter, Slot I/O)
```

```

                               ;
                               ;  The following RST's (RST 0 thru RST 5) are reserved for BASIC
                               ;  interpreter, RST 6 for inter-slot calls, and RST 7 for
                               ;  hardware interrupt
                               ;
        0000    F3             BEGIN:  DI                          ;Fail safe
        0001    C3 02D7                JP      CHKRAM              ;Finds all connected RAM
                                                                   ;and cartridges
                               ;
                               ;
                               ;  ** Special information for the VDP. **
                               ;  Any program that accesses the VDP hardware directly
                               ;  should read the I/O port address found here, to be certain
                               ;  the software is compatible with future versions of the VDP.
                               ;
        0004    1BBF                   DW      CGTABL              ;Address of character generator table,
                                                                   ;to allow use of other character ROM.
                                                                   ;
        0006    98                     DB      98H                 ;Current port address for VDP Data read
        0007    98                     DB      98H                 ;   "     "     "     "   "   "   write
                               ;
        0008    C3 2683                JP      SYNCHR              ;Check byte following the RST 8, see
                                                                   ;if equal to the byte pointed by HL
        000B    00                     DB      0
        000C    C3 01B6                JP      RDSLT               ;Read a byte from another slot
        000F    00                     DB      0
        0010    C3 2686                JP      CHRGTR              ;Fetch next char from BASIC text
        0013    00                     DB      0
        0014    C3 01D1                JP      WRSLT               ;Write a byte to another slot
        0017    00                     DB      0
```

```
0018    C3 1B45              JP      OUTDO           ;Output a char to the Console or printer
001B    00                   DB      0
001C    C3 0217              JP      CALSLT          ;Perform Inter-slot call
001F    00                   DB      0
0020    C3 146A              JP      DCOMPR          ;Compares [HL] to [DE]
0023    00                   DB      0
0024    C3 025E              JP      ENASLT          ;Permanently enables a slot
0027    00                   DB      0
0028    C3 2689              JP      GETYPR          ;Returns the [FAC] type
002B    00                   DB      0               ;ID Byte (1)
                                                     ;Format:
                                                     ;  B7 B6 B5 B4 B3 B2 B1 B0
                                                     ;  +  +  +  +  +  +  +  +
                                                     ;  +  +  +  +  - - - - - - Type of character
                                                     ;  +  +  +  +              generator.
                                                     ;  +  +  +  +              0:Japanese
                                                     ;  +  +  +  +              1:International
                                                     ;  +  +  +  +              2:Korea
                                                     ;  +  +  - - - - - - - - - Date format
                                                     ;  +  +                    0: Y-M-D  1: M-D-Y
                                                     ;  +  +                    2: D-M-Y
                                                     ;  - - - - - - - - - - - - Interrupt frequency
                                                     ;                          0: 60 Hz  1: 50 Hz
002C    00                   DB      0               ;ID Byte (2)
                                                     ;Format:
                                                     ;  B7 B6 B5 B4 B3 B2 B1 B0
                                                     ;  +  +  +  +  +  +  +  +
                                                     ;  +  +  +  +  - - - - - - Type of Keyboard
                                                     ;  +  +  +  +              0:Japanese 2:French
                                                     ;  +  +  +  +              1:Int      3:UK
                                                     ;  +  +  +  +              4:DIN
```

```
                                                      ;  - - - - - - - - - - - -  Version of BASIC
                                                      ;                           0: Japanese
                                                      ;                           1: International
0002D   00 00 00              DB      0,0,0
0030    C3 0205               JP      CALLF           ;Performs Far-call (i.e., Inter-slot)
0033    00 00 00 00           DB      0,0,0,0,0
0037    00
                              ;
                      ;
                      ;  Following are used for I/O initialization
                      ;
0038    C3 0C3C               JP      KEYINT          ;Handlers for hardware interrupt
003B    C3 049D               JP      INITIO          ;Do device initialization
003E    C3 139D               JP      INIFNK          ;Reset all function key's text
                              ;
                      SUBTTL -BIOS header- BIOS calls (Video display processor)
```

```


                     ;
                     ;  The following entry points provides control of the
                     ;  VDP's registers, screen mode settings, and memory block
                     ;  move between DRAM and VRAM.
                     ;
0041   C3 0577              JP      DISSCR          ;Disables screen display
0044   C3 0570              JP      ENASCR          ;Enables screen display
0047   C3 057F              JP      WRTVDP          ;Write a byte to any VDP register
004A   C3 07D7              JP      RDVRM           ;Read VRAM addressed using [HL]
004D   C3 07CD              JP      WRTVRM          ;Write VRAM addressed using [HL]
0050   C3 07EC              JP      SETRD           ;Sets up VDP for read
0053   C3 07DF              JP      SETWRT          ;Sets up VDP for write
0056   C3 0815              JP      FILVRM          ;Fills VRAM with specified data
0059   C3 070F              JP      LDIRMV          ;Moves block of data from VRAM to memory
005C   C3 0744              JP      LDIRVM          ; "    "  "  "   "   memory to VRAM
005F   C3 084F              JP      CHGMOD          ;Change screen mode of VDP to [SCRMOD]
0062   C3 07F7              JP      CHGCLR          ;change Foreground, background,
                                                    ;border, color
0065   00                   DB      0
                            ;
                     ;
0066   C3 1398              JP      NMI             ;Handler for non-maskable interrupt
                     ;
0069   C3 06A8              JP      CLRSPR          ;Init sprite data
006C   C3 050E              JP      INITXT          ;Init VDP for 40 X 24 text mode (SCREEN 0)
006F   C3 0538              JP      INIT32          ; "   "   "  32 X 24 text mode (SCREEN 1)
0072   C3 05D2              JP      INIGRP          ; "   "   "  High resolution mode (SCREEN 2)
0075   C3 061F              JP      INIMLT          ; "   "   "  Multi color mode (SCREEN 3)
0078   C3 0594              JP      SETTXT          ;Sets VDP to display 40 X 24 text mode
007B   C3 05B4              JP      SETT32          ; "   "  "     "    32 X 24 text mode
```

```
007E    C3 0602              JP      SETGRP          ; "     "  "     "   High-res mode
0081    C3 0659              JP      SETMLT          ; "     "  "     "   Multi color mode
0084    C3 06E4              JP      CALPAT          ;Get address of sprite pattern table
0087    C3 06F9              JP      CALATR          ; "      "    "    "   attribute table
008A    C3 0704              JP      GSPSIZ          ;Returns current sprite size
008D    C3 1510              JP      GRPPRT          ;Print a character on the graphic screen
                             ;
                    SUBTTL -BIOS header- BIOS calls (Programmable Sound Generator control)
```

```

                              ;
                              ;  Following entry points are used for PSG initialization,
                              ;  read and write PSG registers, and PLAY statement execution.
                              ;
        0090    C3 04BD               JP      GICINI          ;Init PSG, and static data for PLAY
        0093    C3 1102               JP      WRTPSG          ;Write data to PSG
        0096    C3 110E               JP      RDPSG           ;Read data from PSG
        0099    C3 11C4               JP      STRTMS          ;Checks and start background task for PLAY
                                      ;
                              SUBTTL -BIOS header- BIOS calls (Keyboard, CRT, and Printer)
```

```

                         ;
                         ;  General INPUT and PRINT utilities.
                         ;
009C    C3 0D6A                 JP      CHSNS           ;Checks status of keyboard status
009F    C3 10CB                 JP      CHGET           ;Return char typed, with wait
00A2    C3 08BC                 JP      CHPUT           ;Output character to console
00A5    C3 085D                 JP      LPTOUT          ;  "        "     to printer, if possible
00A8    C3 0884                 JP      LPTSTT          ;Checks status of line printer
00AB    C3 089D                 JP      CNVCHR          ;Checks for graphic header byte
                                                        ;and convert code
00AE    C3 23BF                 JP      PINLIN          ;Read line from keyboard to buffer
00B1    C3 23D5                 JP      INLIN           ;Same as above, except in case of
                                                        ;AUTFLG is set
00B4    C3 23CC                 JP      QINLIN          ;Print a "?", then jump to INLIN
00B7    C3 046F                 JP      BREAKX          ;[Control-STOP] pressed??
00BA    C3 03FB                 JP      ISCNTC          ;[Shift-STOP] pressed??
00BD    C3 10F9                 JP      CKCNTC          ;Same as ISCNTC, but used by BASIC
00C0    C3 1113                 JP      BEEP            ;Buzz
00C3    C3 0848                 JP      CLS             ;Clear screen
00C6    C3 088E                 JP      POSIT           ;Place cursor at Column [H], Row [L]
00C9    C3 0B26                 JP      FNKSB           ;Display Function key, if neccessary
00CC    C3 0B15                 JP      ERAFNK          ;Stop displaying the Function keys
00CF    C3 0B2B                 JP      DSPFNK          ;Enable Function key display
00D2    C3 083B                 JP      TOTEXT          ;Force screen to text mode
                                ;
                         SUBTTL -BIOS header- BIOS calls (Game and Cassette I/O, Queue handler)
```

```

                        ;
                        ;  Following are used to read the value from Joysticks,
                        ;  Graphic pad (tablet), and Paddles.
                        ;
00D5    C3 11EE                 JP      GTSTCK          ;Return status of joystick
00D8    C3 1253                 JP      GTTRIG          ;Read joystick trigger button
00DB    C3 12AC                 JP      GTPAD           ;Returns status of graphic pad
00DE    C3 1273                 JP      GTPDL           ;Read paddle
                                ;
                        ;
                        ;  Following are used to access the cassette tape,
                        ;  data read/write, and motor on/off
                        ;
00E1    C3 1A63                 JP      TAPION          ;Turn on motor and read tape header
00E4    C3 1ABC                 JP      TAPIN           ;Read tape data
00E7    C3 19E9                 JP      TAPIOF          ;Stops reading from tape
00EA    C3 19F1                 JP      TAPOON          ;Turn on motor and write tape header
00ED    C3 1A19                 JP      TAPOUT          ;Write data to tape
00F0    C3 19DD                 JP      TAPOFF          ;Stops writing to tape
00F3    C3 1384                 JP      STMOTR          ;Start, stop cassette motor, or
                                                        ;flip motor(on to off, off to on)
                                ;
                        ;
                        ;  BASIC queues
                        ;
00F6    C3 14EB                 JP      LFTQ            ;Bytes left in queue
00F9    C3 1492                 JP      PUTQ            ;Send a byte to queue
                                ;
                        SUBTTL -BIOS header- BIOS calls (Generalized graphics)
```

```

                                 ;
                                 ;  For BASIC interpreter's GENGRP and ADVGRP modules use
        00FC    C3 16C5                 JP      RIGHTC          ;Moves one pixel right
        00FF    C3 16EE                 JP      LEFTC           ;  "    "    "   left
        0102    C3 175D                 JP      UPC             ;  "    "    "   up
        0105    C3 173C                 JP      TUPC            ;  "    "    "   "
        0108    C3 172A                 JP      DOWNC           ;  "    "    "   down
        010B    C3 170A                 JP      TDOWNC          ;  "    "    "    "
        010E    C3 1599                 JP      SCALXY          ;Scales X Y cordinates
        0111    C3 15DF                 JP      MAPXYC          ;Maps cordinates to physical address
        0114    C3 1639                 JP      FETCHC          ;Get current physical address and
                                                                ;mask pattern
        0117    C3 1640                 JP      STOREC          ;Put current physical address and
                                                                ;mask pattern
        011A    C3 1676                 JP      SETATR          ;Sets the color attribute byte
        011D    C3 1647                 JP      READC           ;Reads attribute of current pixel
        0120    C3 167E                 JP      SETC            ;Sets current pixel to specified attribute
        0123    C3 1809                 JP      NSETCX          ;Sets pixel horizontally
        0126    C3 18C7                 JP      GTASPC          ;Returns aspect ratio
        0129    C3 18CF                 JP      PNTINI          ;Do paint initialization
        012C    C3 18E4                 JP      SCANR           ;Scan pixels to the right
        012F    C3 197A                 JP      SCANL           ; "    "    "   "  left
                                        ;
                                 SUBTTL -BIOS header- BIOS calls (Misc. Entries)
```

```

                                ;
                                ;
      0132    C3 0F3D                   JP      CHGCAP          ;Turn [CAPSLOCK] light, on/off
      0135    C3 0F7A                   JP      CHGSND          ;Change status of 1 bit sound port
      0138    C3 144C                   JP      RSLREG          ;Return output of primary slot register
      013B    C3 144F                   JP      WSLREG          ;Write to primary slot register
      013E    C3 1449                   JP      RDVDP           ;Read VDP status register
      0141    C3 1452                   JP      SNSMAT          ;Read a specified row in the
                                                                ;keyboard matrix
      0144    C3 148A                   JP      PHYDIO          ;Performs operation for mass storage
                                                                ;devices (such as disks)
      0147    C3 148E                   JP      FORMAT          ;Initialize mass storage device
      014A    C3 145F                   JP      ISFLIO          ;Are we doing device I/O
      014D    C3 1B63                   JP      OUTDLP          ;Output to line printer
      0150    C3 1470                   JP      GETVCP          ;Used by Music background tasking
      0153    C3 1474                   JP      GETVC2          ; "   "    "      "        "
      0156    C3 0468                   JP      KILBUF          ;Clear the keyboard buffer
      0159    C3 01FF                   JP      CALBAS          ;Performs far-call into BASIC
      015C                              DS      005AH           ;RESERVED FOR EXPANSION
                                ;
                                SUBTTL - SLOT -  Slot handler stuff
```

```

00A8                PPI.AR  EQU    0A8h    ;A8H    read from PPI Port A
00A8                PPI.AW  EQU    0A8h    ;A8H    Write to PPI Port A
                    ;
                    ; Every cartridge located at 0000-3FFFH must contain codes in
                    ; this module which are entered via following addresses.
                    ;
                    ;    000CH RDSLT
                    ;    0014H WRSLT
                    ;    001CH CALSLT
                    ;    0024H ENASLT
                    ;
                    ;
                    ; ------------------------------ RDSLT ------------------------------
                    ;
                    ; Selects the appropriate slot according to the value given
                    ; through registers, and read the content of memory from the
                    ; slot.
                    ;
                    ; Input parameters:
                    ; A  - FxxxSSPP
                    ;       |   ||||
                    ;       |   ||++-- primary slot # (0-3)
                    ;       |   ++---- secondary slot # (0-3)
                    ;       +--------- 1 if secondary slot # specified
                    ;
                    ;                HL - address of target memory
                    ; Returned value
                    ;                A  - content of memory
                    ;
                    ;  Note: Interrupts are disabled automatically but never enabled
```

```
                          ;       by this routine.
                          ;
01B6                      RDSLT:
01B6    CD 027E                   CALL    SELPRM          ;Calculate bit pattern and mask code
01B9    FA 01C6                   JP      M,RDESLT        ;Expanded slot specified
01BC    DB A8                     IN      A,(PPI.AR)
01BE    57                        LD      D,A             ;Save current setting
01BF    A1                        AND     C               ;Cancel current setting for target address
01C0    B0                        OR      B               ;Add new setting
01C1    CD F380                   CALL    RAMLOW          ;Call read primitive routine (in system area)
01C4    7B                        LD      A,E             ;Return value via [Acc]
01C5    C9                        RET
01C6                      RDESLT:
01C6    E5                        PUSH    HL              ;Save target address
01C7    CD 02A3                   CALL    SELEXP          ;Select secondary slot
01CA    E3                        EX      (SP),HL         ;Restore target address and save [HL]
01CB    C5                        PUSH    BC
01CC    CD 01B6                   CALL    RDSLT
01CF    18 1B                     JR      WRESED          ;Restore old slot select register
                          SUBTTL  -SLOT-  Slot handler (Write slot)
```

```

                              ;
                              ; ----------------------------- WRSLT -----------------------------
                              ;
                              ; Selects the appropriate slot according to the value given
                              ; through registers, and write to the memory in the specified
                              ; slot.
                              ;
                              ; Input parameters:
                              ; A  - FxxxSSPP
                              ;      |   ||||
                              ;      |   ||++-- primary slot # (0-3)
                              ;      |   ++---- secondary slot # (0-3)
                              ;      +--------- 1 if secondary slot # specified
                              ;
                              ;         HL - address of target memory
                              ;
                              ;         E  - value to be written
                              ;
                              ;  Note: Interrupts are disabled automatically but never enabled
                              ;        by this routine.
                              ;
        01D1                  WRSLT:
        01D1    D5                    PUSH    DE              ;Save data to be written
        01D2    CD 027E               CALL    SELPRM          ;Calculate bit pattern and mask code
        01D5    FA 01E1               JP      M,WRESLT        ;Expanded slot specified
        01D8    D1                    POP     DE              ;Restore data to be written
        01D9    DB A8                 IN      A,(PPI.AR)
        01DB    57                    LD      D,A             ;Save current setting
        01DC    A1                    AND     C               ;Cancel current setting for target address
        01DD    B0                    OR      B               ;Add new setting
```

```
01DE    C3 F385              JP      WRPRIM         ;Call write primitive routine (in system area)
01E1                 WRESLT:
01E1    E3                   EX      (SP),HL        ;Save target address, get data to be written
01E2    E5                   PUSH    HL             ;Save data to be written
01E3    CD 02A3              CALL    SELEXP         ;Select secondary slot
01E6    D1                   POP     DE             ;Restore data to be written
01E7    E3                   EX      (SP),HL        ;Restore target address and save [HL]
01E8    C5                   PUSH    BC
01E9    CD 01D1              CALL    WRSLT
01EC                 WRESED:
01EC    C1                   POP     BC
01ED    E3                   EX      (SP),HL        ;Save target address and get old [HL]
01EE    F5                   PUSH    AF             ;Save value returned by RDSLT
01EF    78                   LD      A,B            ;Get current setting
01F0    E6 3F                AND     00111111B      ;Cancel current setting for 0C000H..0FFFFH
01F2    B1                   OR      C
01F3    D3 A8                OUT     (PPI.AW),A     ;Enable 0C000H..0FFFFH of target bank
01F5    7D                   LD      A,L            ;Restore old setting of slot register
01F6    32 FFFF              LD      (0FFFFH),A
01F9    78                   LD      A,B            ;Finally restore old primary slot register
01FA    D3 A8                OUT     (PPI.AW),A
01FC    F1                   POP     AF             ;Restore value returned by RDSLT
01FD    E1                   POP     HL             ;Restore target address
01FE    C9                   RET
```

```

      01FF                     CALBAS:
      01FF    FD 2A FCC0                LD      IY,(EXPTBL-1)
      0203    18 12                     JR      CALSLT
      0205                     CALLF:
      0205    E3                        EX      (SP),HL         ;Get return address, save [HL]
      0206    F5                        PUSH    AF              ;Save working registers
      0207    D5                        PUSH    DE
      0208    7E                        LD      A,(HL)          ;Get destination slot
      0209    F5                        PUSH    AF
      020A    FD E1                     POP     IY              ;Move it to IYH
      020C    23                        INC     HL
      020D    5E                        LD      E,(HL)          ;Get destination address
      020E    23                        INC     HL
      020F    56                        LD      D,(HL)
      0210    23                        INC     HL              ;Prepare true return address
      0211    D5                        PUSH    DE
      0212    DD E1                     POP     IX              ;Move it to IX
      0214    D1                        POP     DE              ;Restore working registers
      0215    F1                        POP     AF
      0216    E3                        EX      (SP),HL         ;Resture [HL], save true return address
                               SUBTTL -SLOT-
```

```

                         ;
                         ; ------------------------------ CALSLT ------------------------------
                         ;
                         ; Performs inter-slot call to specified address.
                         ;
                         ; Input parameters:
                         ; IY  - FxxxSSPP
                         ;       |   ||||
                         ;       |   ||++-- primary slot # (0-3)
                         ;       |   ++---- secondary slot # (0-3)
                         ;       +--------- 1 if secondary slot # specified
                         ;
                         ; IX  - address to call
                         ;
                         ;  Note: Interrupts are disabled automatically but never enabled
                         ;        by this routine.
                         ;        You can never pass arguments via alternate registers
                         ;        of Z80.
                         ;
0217                     CALSLT:
0217    D9                       EXX                     ;Save environments
0218    08                       EX      AF,AF'
0219    FD E5                    PUSH    IY
021B    F1                       POP     AF              ;Get target slot information
021C    DD E5                    PUSH    IX
021E    E1                       POP     HL              ;Get target address
021F    CD 027E                  CALL    SELPRM
0222    FA 022E                  JP      M,CALESL        ;Call expanded slot
0225    DB A8                    IN      A,(PPI.AR)
0227    F5                       PUSH    AF              ;Save current value of primary slot register
```

```
0228    A1                      AND     C               ;Cancel current setting for target address
0229    B0                      OR      B               ;Add new setting
022A    D9                      EXX                     ;Restore environments except PSW
022B    C3 F38C                 JP      CLPRIM          ;Jump to primitive routine (in system area)
022E                    CALESL:
022E    CD 02A3                 CALL    SELEXP          ;Select secondary slot register
0231    F5                      PUSH    AF              ;Move primary slot # in [IYH]
0232    FD E1                   POP     IY
0234    E5                      PUSH    HL              ;Save [B,C,L]  which contain information
0235    C5                      PUSH    BC              ;for restoring slot environments
0236    4F                      LD      C,A             ;Move primary slot # to [BC]
0237    06 00                   LD      B,0
0239    7D                      LD      A,L             ;Re-calculate what is currently output
023A    A4                      AND     H               ;to expansion slot register
023B    B2                      OR      D
023C    21 FCC5                 LD      HL,SLTTBL       ;Calculate address into SLTTBL
023F    09                      ADD     HL,BC
0240    77                      LD      (HL),A          ;Set current value output to expansion
                                                        ;slot register
0241    E5                      PUSH    HL              ;Remember this address
0242    08                      EX      AF,AF'          ;Restore possible arguments passed
0243    D9                      EXX                     ;via registers
0244    CD 0217                 CALL    CALSLT          ;Call by primary slot #
0247    D9                      EXX                     :Save possible values returned via
0248    08                      EX      AF,AF'          ;registers
0249    E1                      POP     HL              ;Restore address into SLTTBL
024A    C1                      POP     BC              ;Restore information about old slots
024B    D1                      POP     DE
024C    78                      LD      A,B             ;Get current setting
024D    E6 3F                   AND     00111111B       ;Cancel current setting for 0C000H..0FFFFH
024F    B1                      OR      C
```

```
0250    F3              DI
0251    D3 A8           OUT     (PPI.AW),A      ;Enable 0C000H..0FFFFH of target bank
0253    7B              LD      A,E             ;Restore old setting of slot register
0254    32 FFFF         LD      (0FFFFH),A
0257    78              LD      A,B             ;Finally restore old primary slot register
0258    D3 A8           OUT     (PPI.AW),A
025A    73              LD      (HL),E          ;And change SLTTBL also
025B    08              EX      AF,AF'          ;Restore possible returned values
025C    D9              EXX
025D    C9              RET
```

```


                          ;
                          ; ----------------------------- ENASLT -----------------------------
                          ;
                          ; Selects the appropriate slot according to the value given
                          ; through registers, and permanently enables the slot.
                          ;
                          ; Input parameters:
                          ;
                          ; A  - FxxxSSPP
                          ;      |   ||||
                          ;      |   ||++-- primary slot # (0-3)
                          ;      |   ++---- secondary slot # (0-3)
                          ;      +--------- 1 if secondary slot # specified
                          ;
                          ; HL - address of target memory
                          ;
                          ;  Note: Interrupts are disabled automatically but never enabled
                          ;        by this routine.
                          ;
025E                      ENASLT:
025E    CD 027E                   CALL    SELPRM          ;Calculate bit pattern and mask code
0261    FA 026B                   JP      M,ENESLT        ;Expanded slot specified
0264    DB A8                     IN      A,(PPI.AR)
0266    A1                        AND     C               ;Cancel current setting for target address
0267    B0                        OR      B               ;Add new setting
0268    D3 A8                     OUT     (PPI.AW),A
026A    C9                        RET
026B                      ENESLT:
026B    E5                        PUSH    HL              ;Save target address
026C    CD 02A3                   CALL    SELEXP          ;Select secondary slot
```

```
026F    4F            LD      C,A           ;Move primary slot # to [BC]
0270    06 00         LD      B,0
0272    7D            LD      A,L           ;Re-calculate what is  currently  output
0273    A4            AND     H             ;to expansion slot register
0274    B2            OR      D
0275    21 FCC5       LD      HL,SLTTBL     ;Calculate address into SLTTBL
0278    09            ADD     HL,BC
0279    77            LD      (HL),A        ;Set current  value  output to expansion
                                            ;slot register
027A    E1            POP     HL            ;Restore target address
027B    79            LD      A,C           ;Restore primary slot # to [Acc]
027C    18 E0         JR      ENASLT        ;Enable by primary slot register
```

```

027E                SELPRM:
027E    F3                  DI
027F    F5                  PUSH    AF              ;Save slot address
0280    7C                  LD      A,H             ;Extract upper 2 bits
0281    07                  RLCA
0282    07                  RLCA
0283    E6 03               AND     00000011B
0285    5F                  LD      E,A
0286    3E C0               LD      A,0C0H          ;Format mask pat. correspond to address
0288                SLPRM1:
0288    07                  RLCA
0289    07                  RLCA
028A    1D                  DEC     E
028B    F2 0288             JP      P,SLPRM1
028E    5F                  LD      E,A             ;Save mask pattern
                                                    ;       00000011        0000-3FFF
                                                    ;       00001100        4000-7FFF
                                                    ;       00110000        8000-BFFF
                                                    ;       11000000        C000-FFFF
028F    2F                  CPL
0290    4F                  LD      C,A             ;Save mask pattern
                                                    ;       11111100        0000-3FFF
                                                    ;       11110011        4000-7FFF
                                                    ;       11001111        8000-BFFF
                                                    ;       00111111        C000-FFFF
0291    F1                  POP     AF              ;Restore slot address
0292    F5                  PUSH    AF
0293    E6 03               AND     00000011B       ;Extract primary slot #
0295    3C                  INC     A
0296    47                  LD      B,A
```

```
0297    3E AB                   LD      A,10101011B     ;Convert slot # to proper bit pattern
0299                    SLPRM2:
0299    C6 55                   ADD     A,01010101B
029B    10 FC                   DJNZ    SLPRM2
029D    57                      LD      D,A             ;Save bit pattern for primary slot #
                                                        ;       00000000        slot #0
                                                        ;       01010101        slot #1
                                                        ;       10101010        slot #2
                                                        ;       11111111        slot #3
029E    A3                      AND     E               ;Extract significant bits
029F    47                      LD      B,A             ;Set it to [B]
02A0    F1                      POP     AF              ;Expanded slot specified?
02A1    A7                      AND     A               ;Set sign flag if so
02A2    C9                      RET
02A3                    SELEXP:
02A3    F5                      PUSH    AF              ;Save target slot
02A4    7A                      LD      A,D             ;Get bit pattern for primary slot
02A5    E6 C0                   AND     11000000B       ;Extract slot # for 0C000H..0FFFFH
02A7    4F                      LD      C,A             ;Save it
02A8    F1                      POP     AF              ;Restore target slot
02A9    F5                      PUSH    AF              ;Save target slot
02AA    57                      LD      D,A             ;Load [D] with specified slot address
02AB    DB A8                   IN      A,(PPI.AR)
02AD    47                      LD      B,A             ;Save current setting
02AE    E6 3F                   AND     00111111B       ;Cancel current setting for 0C000H..0FFFFH
02B0    B1                      OR      C
02B1    D3 A8                   OUT     (PPI.AW),A      ;Enable 0C000H..0FFFFH or target bank
02B3    7A                      LD      A,D             ;Load slot information
02B4    0F                      RRCA
02B5    0F                      RRCA
02B6    E6 03                   AND     00000011B       ;Extract secondary slot #
```

```
02B8    57                      LD      D,A
02B9    3E AB                   LD      A,10101011B     ;Convert secondary   slot  #  to  proper
02BB                    SLEXP1:
02BB    C6 55                   ADD     A,01010101B     ;bit pattern
02BD    15                      DEC     D
02BE    F2 02BB                 JP      P,SLEXP1        ;       00000000        slot #0
                                                        ;       01010101        slot #1
                                                        ;       10101010        slot #2
                                                        ;       11111111        slot #3
02C1    A3                      AND     E               ;Make bit pattern to be added
02C2    57                      LD      D,A             ;Save this
02C3    7B                      LD      A,E             ;Make bit pattern to strip off old value
02C4    2F                      CPL
02C5    67                      LD      H,A             ;Save this
02C6    3A FFFF                 LD      A,(0FFFFH)      ;Read expanded slot register
02C9    2F                      CPL
02CA    6F                      LD      L,A             ;Save current setting
02CB    A4                      AND     H               ;Strip off old bits
02CC    B2                      OR      D               ;And set new bits
02CD    32 FFFF                 LD      (0FFFFH),A      ;Set secondary slot register
02D0    78                      LD      A,B
02D1    D3 A8                   OUT     (PPI.AW),A      ;Restore original primary port
02D3    F1                      POP     AF              ;Restore target slot
02D4    E6 03                   AND     00000011B       ;Fake read from primary slot
02D6    C9                      RET
                        SUBTTL - MSXIO -  I/O Module
```

```

                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                         ;                          ;
                         ;       Port definition    ;
                         ;                          ;
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                         ;
                         ;          VDP address definition
                         ;
        0098             VDP.DRW EQU     10011000B       ;98H    Read/write data VDP
        0099             VDP.CW  EQU     10011001B       ;99H    write command to VDP
        0099             VDP.SR  EQU     10011001B       ;99H    read status from VDP
                         ;
        0007             V.COLR  EQU     7               ;In text mode, foreground and background color
                                                         ;Otherwise background color
                         ;
                         ;          PSG address definition
                         ;
        00A0             PSG.LW  EQU     10100000B       ;A0H    latch address for PSG
        00A1             PSG.DW  EQU     10100001B       ;A1H    write data to PSG
        00A2             PSG.DR  EQU     10100010B       ;A2H    read data from PSG
                         ;
        000E             PSG.PA  EQU     14               ;Port A of PSG
        000F             PSG.PB  EQU     15               ;Port B of PSG
                         ;
                         ;          PPI address definition
                         ;
        00A8             PPI.AR  EQU     10101000B       ;A8H    read from PPI Port A
        00A9             PPI.BR  EQU     10101001B       ;A9H    read from PPI Port B
        00AA             PPI.CR  EQU     10101010B       ;AAH    read from PPI Port C
        00A8             PPI.AW  EQU     10101000B       ;A8H    Write to PPI Port A
```

```
00AA            PPI.CW  EQU     10101010B       ;AAH    write to PPI Port C
00AB            PPI.CM  EQU     10101011B       ;ABH    write to PPI command register
                ;
                ;       Printer port definition
                ;
0091            LPT.DW  EQU     10010001B       ;Data port
0090            LPT.SB  EQU     10010000B       ;Strobe output
0090            LPT.ST  EQU     10010000B       ;Printer status
                ;
                ;       Text mode (40*24)                       SCREEN 0
                ;
                ;               TXTNAM,TXTCGP
                ;
                ;       Text mode (graphics 1)                  SCREEN 1
                ;
                ;               T32NAM,T32COL,T32CGP,T32ATR,T32PAT
                ;
                ;       Hires mode                              SCREEN 2
                ;
                ;               GRPNAM,GRPCOL,GRPCGP,GRPATR,GRPPAT
                ;
                ;       Multi-color mode                        SCREEN 3
                ;
                ;               MLTNAM,MLTCGP,MLTATR,MLTPAT
                ;
                ;       Screen size
                ;
                ;               LINLEN,CRTCNT,LINL32,LINL40
                ;
                ;       External constants
                ;
```

```
                        ;               CGTABL          Character generator table
                        ;
                        ;       External variables
                        ;
                        ;               FORCLR          Foreground color
                        ;               BAKCLR          Background color
                        ;               BDRCLR          Border color for PAINT
                        ;               SCRMOD          Current screen mode
                                                        ;       0 - 40*24 text
                                                        ;       1 - 32*24 text
                                                        ;       2 - hiresolution graphics
                                                        ;       3 - Multicolor graphics
                        ;               OLDSCR
                        ;               NAMBAS          Base of current name table
                        ;               CGPBAS          Base of current cgen table
                        ;               PATBAS          Base of current sprite pattern table
                        ;               ATRBAS          Base of current sprite attribute table
                        ;               JIFFY           Jiffy count
                        ;               CLIKSW          Click switch
                        ;               CLIKFL          Click flag to suppress multiple key clicks
                        ;               RG0SAV          VDP register #0 save area
                        ;               RG1SAV          VDP register #1 save area
                        ;               STATFL          VDP status register
                        ;               PATWRK          Work area for pattern converter
                        ;
                        ;       External routines
                        ;
                        ;               GETQ
                        ;               PUTQ
                        ;               INITQ
                        SUBTTL - MSXIO -  Find available RAM
```

```
        02D7                    CHKRAM:
                                ;
                                ; ------------------------------ CHKRAM ------------------------------
                                ;
                                ; Look into every slot from 0FFFFH to C000H, and set system work
                                ; area. Note that we cannot use RAM as work area nor perform
                                ; subroutine calls 'cause we do not yet know where the available
                                ; RAM exits. Everything has to be done inside ROM and CPU's
                                ; register until the RAM is found.
                                ;
        02D7    3E 82                   LD      A,82H           ;Port A - output (mode 0)
        02D9    D3 AB                   OUT     (PPI.CM),A      ;Port B - input (mode 0)
        02DB    AF                      XOR     A               ;Port C - output (mode 0)
        02DC    D3 A8                   OUT     (PPI.AW),A      ;Select slot 0 for all addresses
        02DE    3E 50                   LD      A,'P'           ;Disable all cassette related outputs
        02E0    D3 AA                   OUT     (PPI.CW),A      ;Motor off
                                ;
                                ; Start searching
                                ;
                                ; Register usage:
                                ; B - non 0 if we're now checking secondary slot
                                ; SPH - slot # of the biggest RAM block
                                ; SPL - secondary slot # of the biggest RAM block (if any)
                                ; DE - lowest address of the biggest RAM block ever found
                                ; C - 'slot-expanded' flag
                                ;
                                ; 0000xxxx
                                ;     ||||
                                ;     |||+- slot #3 expanded
                                ;     ||+-- slot #2 expanded
```

```
                             ;      |+--- slot #1 expanded
                             ;      +---- slot #0 expanded
                             ;
        02E2     11 FFFF             LD      DE,0FFFFH       ;Initialize lowest address ever found
        02E5     AF                  XOR     A               ;Start from slot #0
        02E6     4F                  LD      C,A             ;Clear bit pattern
        02E7                 CKRM05:
        02E7     D3 A8               OUT     (PPI.AW),A      ;Select the slot
        02E9     CB 21               SLA     C               ;Shift bit pattern
        02EB     06 00               LD      B,0             ;Assume this slot is not expanded
        02ED     21 FFFF             LD      HL,0FFFFH       ;Read from possible expansion slot register
        02F0     36 F0               LD      (HL),0F0H       ;Write  a binary 11110000
        02F2     7E                  LD      A,(HL)
        02F3     D6 0F               SUB     0FH             ;Read back as 00001111?
        02F5     20 0B               JR      NZ,CKRM15       ;Nop, this is not an expanded slot
        02F7     77                  LD      (HL),A          ;Write 00000000
        02F8     7E                  LD      A,(HL)
        02F9     3C                  INC     A               ;Read back as 11111111?
        02FA     20 06               JR      NZ,CKRM15       ;Nop, not expanded slot
        02FC     04                  INC     B               ;We're checking expanded slot
        02FD     CB C1               SET     0,C             ;Say this slot is expanded
        02FF                 CKRM10:
                             ;
                             ;Start from expansion slot #0
                             ;
        02FF     32 FFFF             LD      (0FFFFH),A      ;Select the expanded slot
        0302                 CKRM15:
        0302     21 BF00             LD      HL,0BF00H       ;Start checking from 0BF00H to 8000H
        0305                 CKRM20:
        0305     7E                  LD      A,(HL)
        0306     2F                  CPL
```

```
0307    77                      LD      (HL),A
0308    BE                      CP      (HL)
0309    2F                      CPL
030A    77                      LD      (HL),A
030B    20 07                   JR      NZ,CKRM25       ;RAM not equipped in this page
030D    2C                      INC     L               ;Make sure it's not a coincidence
030E    20 F5                   JR      NZ,CKRM20       ;Check more
0310    25                      DEC     H
0311    FA 0305                 JP      M,CKRM20        ;Check next page
0314                    CKRM25:
0314    2E 00                   LD      L,0
0316    24                      INC     H
0317    7D                      LD      A,L             ;Below the one ever found
0318    93                      SUB     E
0319    7C                      LD      A,H
031A    9A                      SBC     A,D
031B    30 0A                   JR      NC,CKRM30       ;No
031D    EB                      EX      DE,HL           ;Register this address as the lowest
031E    3A FFFF                 LD      A,(0FFFFH)      ;Set possible secondary slot #
0321    2F                      CPL
0322    6F                      LD      L,A
0323    DB A8                   IN      A,(PPI.AR)      ;Set primary slot #
0325    67                      LD      H,A
0326    F9                      LD      SP,HL           ;Register these slot #'s
0327                    CKRM30:
0327    78                      LD      A,B
0328    A7                      AND     A               ;Are we checking secondary slot
0329    28 0A                   JR      Z,CKRM35        ;No
032B    3A FFFF                 LD      A,(0FFFFH)
032E    2F                      CPL
032F    C6 10                   ADD     A,10H           ;Prepare to select next secondary slot
```

```
  773     0331    FE 40                   CP      01000000B
  774     0333    38 CA                   JR      C,CKRM10        ;Continue if more secondary slots remain
  775     0335                    CKRM35:
  776     0335    DB A8                   IN      A,(PPI.AR)
  777     0337    C6 50                   ADD     A,01010000B     ;Prepare to select next slot
  778     0339    30 AC                   JR      NC,CKRM05       ;Continue if more primary slots remain
```

```

                          ;
                          ; Check is done, select the biggest one
                          ;
0338   21 0000                    LD      HL,0
033E   39                         ADD     HL,SP
033F   7C                         LD      A,H
0340   D3 A8                      OUT     (PPI.AW),A       ;Set primary slot register
0342   7D                         LD      A,L
0343   32 FFFF                    LD      (0FFFFH),A       ;Set possible secondary slot register
                          ;
                          ; Next, check 0C000H..0FFFFH
                          ;
0346   79                         LD      A,C
0347   07                         RLCA
0348   07                         RLCA
0349   07                         RLCA
034A   07                         RLCA
034B   4F                         LD      C,A
034C   11 FFFF                    LD      DE,0FFFFH        ;Initialize lowest address ever found
034F   DB A8                      IN      A,(PPI.AR)       ;Start from slot #0
0351   E6 3F                      AND     00111111B
0353                      CKRM50:
0353   D3 A8                      OUT     (PPI.AW),A       ;Select the slot
0355   06 00                      LD      B,0              ;Assume this slot is not expanded
0357   CB 01                      RLC     C                ;Shift bit pattern
0359   30 0A                      JR      NC,CKRM60        ;This slot is not expanded
035B   04                         INC     B                ;We're checking expanded slot
035C   3A FFFF                    LD      A,(0FFFFH)
035F   2F                         CPL
0360   E6 3F                      AND     00111111B
```

```
0362                  CKRM55:
0362      32 FFFF              LD      (0FFFFH),A       ;Select the expanded slot
0365                  CKRM60:
0365      21 FE00              LD      HL,0FE00H        ;Start checking from 0FE00H to 0C000H
0368                  CKRM65:
0368      7E                   LD      A,(HL)
0369      2F                   CPL
036A      77                   LD      (HL),A
036B      BE                   CP      (HL)
036C      2F                   CPL
036D      77                   LD      (HL),A
036E      20 09                JR      NZ,CKRM70        ;RAM not equipped in this page
0370      2C                   INC     L                ;Make sure it's not a coincidence
0371      20 F5                JR      NZ,CKRM65        ;Check more
0373      25                   DEC     H
0374      7C                   LD      A,H
0375      FE C0                CP      0C0H
0377      30 EF                JR      NC,CKRM65        ;Check next page
0379                  CKRM70:
0379      2E 00                LD      L,0
037B      24                   INC     H
037C      7D                   LD      A,L              ;Below the one ever found
037D      93                   SUB     E
037E      7C                   LD      A,H
037F      9A                   SBC     A,D
0380      30 0A                JR      NC,CKRM75        ;No
0382      EB                   EX      DE,HL            ;Register this address as the lowest
0383      3A FFFF              LD      A,(0FFFFH)       ;Set possible secondary slot #
0386      2F                   CPL
0387      6F                   LD      L,A
0388      DB A8                IN      A,(PPI.AR)       ;Set primary slot #
```

```
038A    67                    LD      H,A
038B    F9                    LD      SP,HL           ;Register these slot #'s
038C                CKRM75:
038C    78                    LD      A,B
038D    A7                    AND     A               ;Are we checking secondary slot
038E    28 08                 JR      Z,CKRM80        ;No
0390    3A FFFF               LD      A,(0FFFFH)
0393    2F                    CPL
0394    C6 40                 ADD     A,01000000B     ;Prepare to select next secondary slot
0396    30 CA                 JR      NC,CKRM55       ;Continue if more secondary slots remain
0398                CKRM80:
0398    DB A8                 IN      A,(PPI.AR)
039A    C6 40                 ADD     A,01000000B     ;Prepare to select next slot
039C    30 B5                 JR      NC,CKRM50       ;Continue if more primary slots remain
                    SUBTTL - MSXIO -  Slot attribute setup
```

```

                              ;
                              ; Check is done, select the biggest one
                              ;
 039E    21 0000                      LD      HL,0
 03A1    39                           ADD     HL,SP
 03A2    7C                           LD      A,H
 03A3    D3 A8                        OUT     (PPI.AW),A      ;Set primary slot register
 03A5    7D                           LD      A,L
 03A6    32 FFFF                      LD      (0FFFFH),A      ;Set possible secondary slot register
 03A9    79                           LD      A,C             ;Set 'slot expanded' flag
                              ;
                              ; Clear work area with zero
                              ;
 03AA    01 0C49                      LD      BC,0C49H        ;length of work area
 03AD    11 F381                      LD      DE,RAMLOW+1
 03B0    21 F380                      LD      HL,RAMLOW       ;beginning of work
 03B3    36 00                        LD      (HL),0          ;init first byte
 03B5    ED B0                        LDIR                    ;transfer it to rest of area
                              ;
                              ; Set EXPTBL
                              ;
 03B7    4F                           LD      C,A             ;Get 'slot-expanded' flag
 03B8    06 04                        LD      B,4             ;Loop 4 times
 03BA    21 FCC4                      LD      HL,EXPTBL+3
 03BD                         SSLTLP:
 03BD    CB 19                        RR      C               ;Set carry if LSB is set
 03BF    9F                           SBC     A,A             ;[Acc]=255 if expanded, 0 if not expanded
 03C0    E6 80                        AND     80H             ;Affects only MSB
 03C2    77                           LD      (HL),A          ;Set table for each slot
 03C3    2B                           DEC     HL
```

```
03C4    10 F7                   DJNZ    SSLTLP
                        ;
                        ; Set SLTTBL
                        ;
03C6    DB A8                   IN      A,(PPI.AR)      ;Remember primary slot register's content
03C8    4F                      LD      C,A
03C9    AF                      XOR     A               ;Read from slot #0
03CA    D3 A8                   OUT     (PPI.AW),A
03CC    3A FFFF                 LD      A,(0FFFFH)
03CF    2F                      CPL
03D0    6F                      LD      L,A
03D1    3E 40                   LD      A,01000000B     ;Read from slot #1
03D3    D3 A8                   OUT     (PPI.AW),A
03D5    3A FFFF                 LD      A,(0FFFFH)
03D8    2F                      CPL
03D9    67                      LD      H,A
03DA    3E 80                   LD      A,80H           ;Read from slot #2
03DC    D3 A8                   OUT     (PPI.AW),A
03DE    3A FFFF                 LD      A,(0FFFFH)
03E1    2F                      CPL
03E2    5F                      LD      E,A
03E3    3E C0                   LD      A,0C0H          ;Read from slot #3
03E5    D3 A8                   OUT     (PPI.AW),A
03E7    3A FFFF                 LD      A,(0FFFFH)
03EA    2F                      CPL
03EB    57                      LD      D,A
03EC    79                      LD      A,C             ;Restore primary slot register
03ED    D3 A8                   OUT     (PPI.AW),A
03EF    22 FCC5                 LD      (SLTTBL),HL     ;Set SLTTBL
03F2    EB                      EX      DE,HL
03F3    22 FCC7                 LD      (SLTTBL+2),HL
```

```
 918     03F6    ED 56                   IM      1               ;IM 1
 919     03F8    C3 2680                 JP      INIT
 920                             SUBTTL - MSXIO -  Control-[C] processing
```

```

03FB                    ISCNTC:
03FB    3A FBB1                 LD      A,(BASROM)      ;Is BASIC text in ROM
03FE    A7                      AND     A
03FF    C0                      RET     NZ              ;Yes
0400    E5                      PUSH    HL
0401    21 FC9B                 LD      HL,INTFLG       ;Seen any interesting key
0404    F3                      DI
0405    7E                      LD      A,(HL)
0406    36 00                   LD      (HL),0
0408    E1                      POP     HL
0409    FB                      EI
040A    A7                      AND     A
040B    C8                      RET     Z               ;No
040C    FE 03                   CP      3               ;Is it ctrl-stop?
040E    28 1C                   JR      Z,EXCABO        ;Yes, execution aborted
                        ;
                        ; Pause until next STOP is pressed
                        ;
0410    E5                      PUSH    HL              ;STOP pressed (pause)
0411    D5                      PUSH    DE
0412    C5                      PUSH    BC
0413    CD 09DA                 CALL    CKDPC0          ;Display cursor if disabled
0416    21 FC9B                 LD      HL,INTFLG       ;Wait for next interesting key
0419                    WATINT:
0419    F3                      DI
041A    7E                      LD      A,(HL)
041B    36 00                   LD      (HL),0
041D    FB                      EI                      ;Wait for character if SELECT pressed
041E    A7                      AND     A               ;Seen?
041F    28 F8                   JR      Z,WATINT        ;Not yet
```

```
0421    F5                  PUSH    AF
0422    CD 0A27             CALL    CKERC0          ;Erase cursor if disabled
0425    F1                  POP     AF
0426    C1                  POP     BC
0427    D1                  POP     DE
0428    E1                  POP     HL
0429    FE 03               CP      3               ;Abort?
042B    C0                  RET     NZ              ;No
042C              EXCABO:
042C    E5                  PUSH    HL              ;Save text pointer
042D    CD 0468             CALL    KILBUF          ;Cancel any input
0430    CD 0454             CALL    CKSTTP          ;Is STOP trap ON
0433    30 0A               JR      NC,EXABO1       ;No, accept this break
0435    21 FC6A             LD      HL,REQSTP       ;Request STOP trap
0438    F3                  DI                      ;Since REQTRP does not change interrupt mask,
0439    CD 0EF1             CALL    REQTRP          ;this must be enclosed by 'DI' and 'EI'
043C    FB                  EI
043D    E1                  POP     HL              ;Restore text pointer
043E    C9                  RET
043F              EXABO1:
                  ;
043F    CD 083B             CALL    TOTEXT          ;Make sure we're in text mode
0442    3A FCC1             LD      A,(EXPTBL)      ;Make sure BASIC is enabled
0445    26 40               LD      H,01000000B
0447    CD 025E             CALL    ENASLT
044A    E1                  POP     HL              ;Restore text pointer
044B    AF                  XOR     A               ;Must return with carry cleared, zero set
044C    ED 7B F6B1          LD      SP,(SAVSTK)     ;LSPD
0450    C5                  PUSH    BC
0451    C3 63E6             JP      STOP
```

```
0454                    CKSTTP:
                        ;
                        ; Check for STOP trap
                        ;
                        ;
0454    3A FC6A                 LD      A,(REQSTP)      ;Is STOP trap ON
0457    0F                      RRCA
0458    D0                      RET     NC              ;No, accept this break
0459    2A FC6B                 LD      HL,(REQSTP+1)   ;Is STOP trap specified
045C    7C                      LD      A,H
045D    B5                      OR      L
045E    C8                      RET     Z               ;No, accept this break
045F    2A F41C                 LD      HL,(CURLIN)     ;Are we in direct mode
0462    23                      INC     HL
0463    7C                      LD      A,H
0464    B5                      OR      L
0465    C8                      RET     Z               ;Yes, treat as break
0466    37                      SCF                     ;Set flag to indicate STOP trap active
0467    C9                      RET
0468                    KILBUF:
                        ;
0468    2A F3F8                 LD      HL,(PUTPNT)     ;Empties ring buffer
046B    22 F3FA                 LD      (GETPNT),HL
046E    C9                      RET
```

```

046F                      BREAKX:
                          ;
                          ; Check if stop key pressed. If pressed, return with carry set.
                          ;
046F    DB AA                     IN      A,(PPI.CR)
0471    E6 F0                     AND     0F0H            ;Leave others unaffected
0473    F6 07                     OR      7               ;Select 6th row
0475    D3 AA                     OUT     (PPI.CW),A
0477    DB A9                     IN      A,(PPI.BR)
0479    E6 10                     AND     10H             ;STOP key is assigned to bit 4
047B    C0                        RET     NZ              ;0 when pressed
047C    DB AA                     IN      A,(PPI.CR)
047E    3D                        DEC     A
047F    D3 AA                     OUT     (PPI.CW),A
0481    DB A9                     IN      A,(PPI.BR)
0483    E6 02                     AND     2
0485    C0                        RET     NZ
0486    E5                        PUSH    HL
0487    2A F3F8                   LD      HL,(PUTPNT)     ;Cancel any input
048A    22 F3FA                   LD      (GETPNT),HL
048D    E1                        POP     HL
048E    3A FBE1                   LD      A,(OLDKEY+7)    ;STOP pressed, mark as pressed to prevent
0491    E6 EF                     AND     0EFH            ; to be doubly recognized
0493    32 FBE1                   LD      (OLDKEY+7),A
0496    3E 0D                     LD      A,0DH
0498    32 F3F7                   LD      (REPCNT),A
049B    37                        SCF
049C    C9                        RET
                          SUBTTL - MSXIO -  PSG Initialization
```

```
049D                    INITIO:
                        ;
                        ; Initialize I O
                        ;
049D    3E 07                   LD      A,7
049F    1E 80                   LD      E,80H
04A1    CD 1102                 CALL    WRTPSG          ;Set Port A to input mode
04A4    3E 0F                   LD      A,0FH           ;Port B to output mode
04A6    1E CF                   LD      E,0CFH
04A8    CD 1102                 CALL    WRTPSG
04AB    3E 0B                   LD      A,0BH           ;Dummy write cycle to wake up the PSG
04AD    5F                      LD      E,A             ;envelope register
04AE    CD 1102                 CALL    WRTPSG          ;Any value is OK!
04B1    CD 110C                 CALL    INGI
04B4    E6 40                   AND     01000000B
04B6    32 FCAD                 LD      (KANAMD),A
04B9    3E FF                   LD      A,0FFH
04BB    D3 90                   OUT     (LPT.SB),A
04BD                    GICINI:
                        ;
                        ; Initialize GI sound chip, queues, and static data.
                        ;
                        ; Entry - Interrupts must be disabled
                        ; Exit - All registers preserved.
                        ;
04BD    E5                      PUSH    HL              ;save caller's registers
04BE    D5                      PUSH    DE
04BF    C5                      PUSH    BC
04C0    F5                      PUSH    AF
                        ;
```

```
                                ; First, clear all static data
                                ;
        04C1    21 FB3F                 LD      HL,MUSICF
        04C4    06 71                   LD      B,71H           ;=VCBC + VCBSIZ + MUSCIF
        04C6    AF                      XOR     A
        04C7                    MUSCLL:
        04C7    77                      LD      (HL),A
        04C8    23                      INC     HL
        04C9    10 FC                   DJNZ    MUSCLL
                                ;
                                ; Then clear music dynamic queue
                                ;
        04CB    11 F975                 LD      DE,VOICAQ       ;Address of music queue
        04CE    06 7F                   LD      B,7FH           ;Mask pattern, 7F = Music queue len - 1
        04D0    21 0080                 LD      HL,80H          ;Queue length
        04D3                    GICIN1:
        04D3    E5                      PUSH    HL              ;Save length of queue
        04D4    D5                      PUSH    DE              ;Save address of queue
        04D5    C5                      PUSH    BC              ;Save mask pattern
        04D6    F5                      PUSH    AF              ;Save queue ID
        04D7    CD 14DA                 CALL    INITQ           ;Initialize a queue by [Acc],[B],[DE]
        04DA    F1                      POP     AF
        04DB    C6 08                   ADD     A,8             ;write to regs 8,9,10
        04DD    1E 00                   LD      E,0
        04DF    CD 1102                 CALL    WRTPSG          ;0 out amplitude (turn voice off)
        04E2    D6 08                   SUB     8               ;Restore [Acc]
        04E4    F5                      PUSH    AF              ;Save queue ID
        04E5    2E 0F                   LD      L,0FH           ;OctaveX
        04E7    CD 1477                 CALL    GETVC1          ;[HL] points to octave for voice [A]
        04EA    EB                      EX      DE,HL
        04EB    21 0508                 LD      HL,MUSITB       ;[HL] points to default value table
```

```
04EE    01 0006             LD      BC,6            ;EMSITB - MUSITB
04F1    ED B0               LDIR                    ;default variables for this voice
04F3    F1                  POP     AF              ;Restore queue ID
04F4    C1                  POP     BC              ;Restore mask
04F5    E1                  POP     HL              ;Restore queue address
04F6    D1                  POP     DE              ;Restore queue length
04F7    19                  ADD     HL,DE           ;Update queue address
04F8    EB                  EX      DE,HL
04F9    3C                  INC     A               ;Next channel
04FA    FE 03               CP      3
04FC    38 D5               JR      C,GICIN1        ;Loop till done all three voices
04FE    3E 07               LD      A,7             ;write to reg 7 mixer control
0500    1E B8               LD      E,0B8H          ;input port A, output port B
0502    CD 1102             CALL    WRTPSG          ;disable noise, enable all 3 tones
0505    C3 08DA             JP      POPALL          ;Restore environments
0508                MUSITB:
                    ;
                    ; table of default values for music variables
                    ;
0508    04                  DB      04H             ;default octave
0509    04                  DB      04H             ;default note length
050A    78                  DB      78H             ;default tempo
050B    88                  DB      88H             ;default volume
050C    FF                  DB      0FFH            ;default envelope period
050D    00                  DB      00H
050E                EMSITB:                         ;end of music table
                    SUBTTL - MSXIO -  Utility routines for VDP
```

```

      050E                INITXT:
                          ;
                          ; Initialize VDP for text mode (40 by 24)
                          ;
      050E    CD 0577             CALL    DISSCR
      0511    AF                  XOR     A
      0512    32 FCAF             LD      (SCRMOD),A
      0515    32 FCB0             LD      (OLDSCR),A
      0518    3A F3AE             LD      A,(LINL40)
      051B    32 F3B0             LD      (LINLEN),A
      051E    2A F3B3             LD      HL,(TXTNAM)
      0521    22 F922             LD      (NAMBAS),HL
      0524    2A F3B7             LD      HL,(TXTCGP)
      0527    22 F924             LD      (CGPBAS),HL
      052A    CD 07F7             CALL    CHGCLR          ;Set border/foreground/background color
      052D    CD 077E             CALL    CLRTXT
      0530    CD 071E             CALL    INIPAT          ;Initialize character pattern
      0533    CD 0594             CALL    SETTXT          ;Actually set VDP registers
      0536    18 38               JR      ENASCR
      0538                INIT32:
                          ;
                          ; Initialize VDP for text mode (graphics 1)
                          ;
      0538    CD 0577             CALL    DISSCR
      053B    3E 01               LD      A,1
      053D    32 FCAF             LD      (SCRMOD),A
      0540    32 FCB0             LD      (OLDSCR),A
      0543    3A F3AF             LD      A,(LINL32)
      0546    32 F3B0             LD      (LINLEN),A
      0549    2A F3BD             LD      HL,(T32NAM)
```

```
054C    22 F922                 LD      (NAMBAS),HL
054F    2A F3C1                 LD      HL,(T32CGP)
0552    22 F924                 LD      (CGPBAS),HL
0555    2A F3C5                 LD      HL,(T32PAT)
0558    22 F926                 LD      (PATBAS),HL
055B    2A F3C3                 LD      HL,(T32ATR)
055E    22 F928                 LD      (ATRBAS),HL
0561    CD 07F7                 CALL    CHGCLR          ;Set border foreground background color
0564    CD 077E                 CALL    CLRTXT
0567    CD 071E                 CALL    INIPAT          ;Initialize character pattern
056A    CD 06BB                 CALL    ERASPR          ;Clear sprites
056D    CD 05B4                 CALL    SETT32          ;Actually set VDP registers
0570                    ENASCR:
                        ;
                        ; Enable screen display
                        ;
0570    3A F3E0                 LD      A,(RG1SAV)
0573    F6 40                   OR      01000000B
0575    18 05                   JR      DISSC1
0577                    DISSCR:
                        ;
                        ; Disable screen display
                        ;
0577    3A F3E0                 LD      A,(RG1SAV)
057A    E6 BF                   AND     0BFH
057C                    DISSC1:
057C    47                      LD      B,A
057D    0E 01                   LD      C,1
```

```

057F                WRTVDP:
                    ;
                    ; Write data to VDP
                    ;
                    ; C = register #
                    ; B = value to be set
                    ;
                    ; Register save area for the register is properly set
                    ;
057F    78                  LD      A,B             ;Get data to set
0580    F3                  DI
0581    D3 99               OUT     (VDP.CW),A
0583    79                  LD      A,C             ;Get register #
0584    F6 80               OR      80H
0586    D3 99               OUT     (VDP.CW),A
0588    FB                  EI
0589    E5                  PUSH    HL
058A    78                  LD      A,B             ;Remember this value 'cause this is
058B    06 00               LD      B,0             ;a write-only register
058D    21 F3DF             LD      HL,RG0SAV
0590    09                  ADD     HL,BC
0591    77                  LD      (HL),A
0592    E1                  POP     HL
0593    C9                  RET
0594                SETTXT:
                    ;
                    ; Set VDP for text mode (40 by 32)
                    ;
0594    3A F3DF             LD      A,(RG0SAV)      ;Set register #0
0597    E6 01               AND     1
```

```
0599    47                      LD      B,A
059A    0E 00                   LD      C,0
059C    CD 057F                 CALL    WRTVDP
059F    3A F3E0                 LD      A,(RG1SAV)      ;Set register #1
05A2    E6 E7                   AND     0E7H
05A4    F6 10                   OR      10H
05A6    47                      LD      B,A
05A7    0C                      INC     C
05A8    CD 057F                 CALL    WRTVDP
05AB    21 F3B3                 LD      HL,TXTNAM
05AE    11 0000                 LD      DE;0            ;Set mask pattern
05B1    C3 0677                 JP      SETSCM          ;Set screen mode
05B4                    SETT32:
                        ;
                        ; Set VDP for text mode (graphics 1)
                        ;
05B4    3A F3DF                 LD      A,(RG0SAV)      ;Set register #0
05B7    E6 01                   AND     1
05B9    47                      LD      B,A
05BA    0E 00                   LD      C,0
05BC    CD 057F                 CALL    WRTVDP
05BF    3A F3E0                 LD      A,(RG1SAV)      ;Set register #1
05C2    E6 E7                   AND     0E7H
05C4    47                      LD      B,A
05C5    0C                      INC     C
05C6    CD 057F                 CALL    WRTVDP
05C9    21 F3BD                 LD      HL,T32NAM
05CC    11 0000                 LD      DE,0            ;Set mask pattern
05CF    C3 0677                 JP      SETSCM          ;Set screen mode
05D2                    INIGRP:
                        ;
```

```
                          ; Initialize VDP for graphics mode
                          ;
05D2    CD 0577                   CALL    DISSCR
05D5    3E 02                     LD      A,2
05D7    32 FCAF                   LD      (SCRMOD),A
05DA    2A F3CF                   LD      HL,(GRPPAT)
05DD    22 F926                   LD      (PATBAS),HL
05E0    2A F3CD                   LD      HL,(GRPATR)
05E3    22 F928                   LD      (ATRBAS),HL
05E6    2A F3C7                   LD      HL,(GRPNAM)     ;Initialize name table
05E9    CD 07DF                   CALL    SETWRT
05EC    AF                        XOR     A
05ED    06 03                     LD      B,3
05EF                      INIGR1:
05EF    D3 98                     OUT     (VDP.DRW),A
05F1    3C                        INC     A
05F2    20 FB                     JR      NZ,INIGR1
05F4    10 F9                     DJNZ    INIGR1
05F6    CD 07A1                   CALL    CLSHRS          ;Clear pattern and color table
05F9    CD 06BB                   CALL    ERASPR
05FC    CD 0602                   CALL    SETGRP          ;Actually set VDP mode
05FF    C3 0570                   JP      ENASCR
0602                      SETGRP:
                          ;
                          ; Set VDP for graphics mode (graphics 2)
                          ;
0602    3A F3DF                   LD      A,(RG0SAV)      ;Set register #0
0605    F6 02                     OR      2
0607    47                        LD      B,A
0608    0E 00                     LD      C,0
060A    CD 057F                   CALL    WRTVDP
```

```
060D    3A F3E0              LD      A,(RG1SAV)      ;Set register #1
0610    E6 E7                AND     0E7H
0612    47                   LD      B,A
0613    0C                   INC     C
0614    CD 057F              CALL    WRTVDP
0617    21 F3C7              LD      HL,GRPNAM
061A    11 7F03              LD      DE,7F03H
061D    18 58                JR      SETSCM
061F                 INIMLT:
                     ;
                     ; Initialize VDP for multi-color mode
                     ;
061F    CD 0577              CALL    DISSCR
0622    3E 03                LD      A,3
0624    32 FCAF              LD      (SCRMOD),A
0627    2A F3D9              LD      HL,(MLTPAT)
062A    22 F926              LD      (PATBAS),HL
062D    2A F3D7              LD      HL,(MLTATR)
0630    22 F928              LD      (ATRBAS),HL
0633    2A F3D1              LD      HL,(MLTNAM)     ;Initialize name table
0636    CD 07DF              CALL    SETWRT
0639    11 0006              LD      DE,6
063C                 INIML1:
063C    0E 04                LD      C,4
063E                 INIML2:
063E    7A                   LD      A,D
063F    06 20                LD      B,' '
0641                 INIML3:
0641    D3 98                OUT     (VDP.DRW),A
0643    3C                   INC     A
0644    10 FB                DJNZ    INIML3
```

```
0646    0D                      DEC     C
0647    20 F5                   JR      NZ,INIML2
0649    57                      LD      D,A
064A    1D                      DEC     E
064B    20 EF                   JR      NZ,INIML1
064D    CD 07B9                 CALL    CLSMLT          ;Clear pattern table
0650    CD 06BB                 CALL    ERASPR
0653    CD 0659                 CALL    SETMLT          ;Actually set VDP mode
0656    C3 0570                 JP      ENASCR
0659                    SETMLT:
                        ;
                        ; Set VDP for multicolor mode
                        ;
0659    3A F3DF                 LD      A,(RG0SAV)      ;Set register #0
065C    E6 01                   AND     1
065E    47                      LD      B,A
065F    0E 00                   LD      C,0
0661    CD 057F                 CALL    WRTVDP
0664    3A F3E0                 LD      A,(RG1SAV)      ;Set register #1
0667    E6 E7                   AND     0E7H
0669    F6 08                   OR      8
066B    47                      LD      B,A
066C    0E 01                   LD      C,1
066E    CD 057F                 CALL    WRTVDP
0671    21 F3D1                 LD      HL,MLTNAM
0674    11 0000                 LD      DE,0            ;Set mask pattern
0677                    SETSCM:
0677    01 0602                 LD      BC,SETGRP
067A    CD 0690                 CALL    SETREG          ;Set name table
067D    06 0A                   LD      B,0AH
067F    7A                      LD      A,D
```

```
0680    CD 0691                 CALL    SETRGl          ;Set color table
0683    06 05                   LD      B,5
0685    7B                      LD      A,E
0686    CD 0691                 CALL    SETRGl          ;Set pattern table
0689    06 09                   LD      B,9
068B    CD 0690                 CALL    SETREG          ;Set sprite attribute table
068E    06 05                   LD      B,5             ;Set sprite pattern table
0690                    SETREG:
0690    AF                      XOR     A
0691                    SETRGl:
0691    E5                      PUSH    HL
0692    F5                      PUSH    AF
0693    7E                      LD      A,(HL)
0694    23                      INC     HL
0695    66                      LD      H,(HL)
0696    6F                      LD      L,A
0697    AF                      XOR     A
0698                    SETRG2:
0698    29                      ADD     HL,HL
0699    8F                      ADC     A,A
069A    10 FC                   DJNZ    SETRG2
069C    6F                      LD      L,A
069D    Fl                      POP     AF
069E    B5                      OR      L
069F    47                      LD      B,A
06A0    CD 057F                 CALL    WRTVDP
06A3    El                      POP     HL
06A4    23                      INC     HL
06A5    23                      INC     HL
06A6    0C                      INC     C
06A7    C9                      RET
```

```

06A8                    CLRSPR:
                        ;
                        ; Clear all sprites
                        ;
06A8    3A F3E0                 LD      A,(RG1SAV)      ;Set register #1
06AB    47                      LD      B,A
06AC    0E 01                   LD      C,1
06AE    CD 057F                 CALL    WRTVDP
06B1    2A F926                 LD      HL,(PATBAS)     ;Clear sprite pattern table
06B4    01 0800                 LD      BC,0800H        ;Length of sprite pattern table
06B7    AF                      XOR     A
06B8    CD 0815                 CALL    FILVRM
06BB                    ERASPR:
06BB    3A F3E9                 LD      A,(FORCLR)      ;Load foreground color (default) to [E]
06BE    5F                      LD      E,A
06BF    2A F928                 LD      HL,(ATRBAS)
06C2    01 2000                 LD      BC,2000H        ;Set number of sprite plane to [B]
06C5                    CLSPR2:
                        ; default sprite name to [C]
                        ;
06C5    3E D1                   LD      A,0D1H          ;Erase code (i.e. vertical position)
06C7    CD 07CD                 CALL    WRTVRM          ;Set vertical position
06CA    23                      INC     HL
06CB    23                      INC     HL
06CC    79                      LD      A,C             ;Load default sprite name
06CD    CD 07CD                 CALL    WRTVRM
06D0    23                      INC     HL
06D1    0C                      INC     C               ;Prepare for next
06D2    3A F3E0                 LD      A,(RG1SAV)
06D5    0F                      RRCA
```

```
06D6    0F                      RRCA                    ;16*16?
06D7    30 03                   JR      NC,CLSPR3       ;No
06D9    0C                      INC     C               ;Yes, C=C+4
06DA    0C                      INC     C
06DB    0C                      INC     C
06DC                    CLSPR3:
06DC    7B                      LD      A,E             ;Load default color
06DD    CD 07CD                 CALL    WRTVRM
06E0    23                      INC     HL
06E1    10 E2                   DJNZ    CLSPR2
06E3    C9                      RET
06E4                    CALPAT:
                        ;
06E4    6F                      LD      L,A
06E5    26 00                   LD      H,0
06E7    29                      ADD     HL,HL           ;Assume 8 byte long
06E8    29                      ADD     HL,HL
06E9    29                      ADD     HL,HL
06EA    CD 0704                 CALL    GSPSIZ          ;Check size of sprite
06ED    FE 08                   CP      8
06EF    28 02                   JR      Z,GSPAD1        ;Good assumption
06F1    29                      ADD     HL,HL           ;32 byte long sprite
06F2    29                      ADD     HL,HL
06F3                    GSPAD1:
06F3    EB                      EX      DE,HL
06F4    2A F926                 LD      HL,(PATBAS)     ;Get base address of sprite pattern table
06F7    19                      ADD     HL,DE           ;Form destination/source address
06F8    C9                      RET
06F9                    CALATR:
                        ;
06F9    6F                      LD      L,A             ;Get plane number to [L]
```

```
06FA   26 00                  LD      H,0
06FC   29                     ADD     HL,HL          ;Sprite attribute consists of 4 bytes
06FD   29                     ADD     HL,HL
06FE   EB                     EX      DE,HL
06FF   2A F928                LD      HL,(ATRBAS)    ;Load base address
0702   19                     ADD     HL,DE          ;Calculate target address
0703   C9                     RET
0704                  GSPSIZ:
                      ;
                      ; Get sprite size
                      ;
0704   3A F3E0                LD      A,(RG1SAV)
0707   0F                     RRCA
0708   0F                     RRCA
0709   3E 08                  LD      A,8            ;Assume 8 byte long
070B   D0                     RET     NC             ;Good assumption
070C   3E 20                  LD      A,32           ;32 byte long sprite
070E   C9                     RET
```

```

070F                   LDIRMV:
                       ;
070F    CD 07EC                 CALL    SETRD
0712    E3                      EX      (SP),HL
0713    E3                      EX      (SP),HL
0714                   LDIMV1:
0714    DB 98                   IN      A,(VDP.DRW)
0716    12                      LD      (DE),A
0717    13                      INC     DE
0718    0B                      DEC     BC
0719    79                      LD      A,C
071A    B0                      OR      B
071B    20 F7                   JR      NZ,LDIMV1
071D    C9                      RET
071E                   INIPAT:
                       ;
                       ; Set default character pattern
                       ;
071E    CD FDC7                 CALL    H.INIP
0721    2A F924                 LD      HL,(CGPBAS)     ;Get target address of VRAM
0724    CD 07DF                 CALL    SETWRT          ;Set VDP for write operation
0727    3A F91F                 LD      A,(CGPNT)       ;Get slot # of character genarator table
072A    2A F920                 LD      HL,(CGPNT+1)    ;Get address of character genarator table
072D    01 0800                 LD      BC,0800H        ;Load total length
0730    F5                      PUSH    AF              ;Save source slot
0731                   INIPT1:
0731    F1                      POP     AF              ;Restore source slot
0732    F5                      PUSH    AF              ;Save source slot
0733    C5                      PUSH    BC              ;Save counter
0734    F3                      DI
```

```
0735    CD 01B6                 CALL    RDSLT           ;Read from specified slot
0738    FB                      EI
0739    C1                      POP     BC              ;Restore counter
073A    D3 98                   OUT     (VDP.DRW),A
073C    23                      INC     HL              ;Bump character source pointer
073D    0B                      DEC     BC
073E    79                      LD      A,C
073F    B0                      OR      B
0740    20 EF                   JR      NZ,INIPT1
0742    F1                      POP     AF              ;Discard stack
0743    C9                      RET
0744                    LDIRVM:
                        ;
0744    EB                      EX      DE,HL
0745    CD 07DF                 CALL    SETWRT
0748                    LDIVM1:
0748    1A                      LD      A,(DE)
0749    D3 98                   OUT     (VDP.DRW),A
074B    13                      INC     DE
074C    0B                      DEC     BC
074D    79                      LD      A,C
074E    B0                      OR      B
074F    20 F7                   JR      NZ,LDIVM1
0751    C9                      RET
0752                    GETPAT:
                        ;
                        ; Get pattern corresponding to ASCII code in [A]
                        ;
                        ; Pattern is returned to 8 byte work area (PATWRK). Entered
                        ; by GRPPRT (print a character to graphic screen) subroutine.
                        ;
                        ; All registers are completely destroyed
                         ;
0752    26 00                   LD      H,0             ;Prepare for calculation
0754    6F                      LD      L,A
0755    29                      ADD     HL,HL
0756    29                      ADD     HL,HL
0757    29                      ADD     HL,HL
0758    EB                      EX      DE,HL
0759    2A F920                 LD      HL,(CGPNT+1)
075C    19                      ADD     HL,DE           ;[HL]:=source address
075D    11 FC40                 LD      DE,PATWRK       ;Load destination address
0760    06 08                   LD      B,8             ;Load total length
0762    3A F91F                 LD      A,(CGPNT)       ;Get slot # of character genarator table
0765                    GTPAT1:
0765    F5                      PUSH    AF              ;Save source slot
0766    E5                      PUSH    HL              ;Save source address
0767    D5                      PUSH    DE              ;Save destination address
0768    C5                      PUSH    BC              ;Save counter
0769    CD 01B6                 CALL    RDSLT           ;Read from specified slot
076C    FB                      EI
076D    C1                      POP     BC              ;Restore counter
076E    D1                      POP     DE              ;Restore destination address
076F    E1                      POP     HL              ;Restore source address
0770    12                      LD      (DE),A
0771    13                      INC     DE              ;Bump destination pointer
0772    23                      INC     HL              ;Bump character source pointer
0773    F1                      POP     AF              ;Restore source slot
0774    10 EF                   DJNZ    GTPAT1
0776    C9                      RET
0777                    CLSSUB:
                        ;
```

```
0777    CD 0B9F                 CALL    CHKSCR          ;Check current screen mode
077A    28 25                   JR      Z,CLSHRS        ;Hires
077C    30 3B                   JR      NC,CLSMLT       ;Multi-color
077E                    CLRTXT:
                        ;
                        ; Clear screen (text mode)
                        ;
077E    3A FCAF                 LD      A,(SCRMOD)
0781    A7                      AND     A
0782    2A F922                 LD      HL,(NAMBAS)     ;Set address for write
0785    01 03C0                 LD      BC,03C0H        ;40 * 24
0788    28 03                   JR      Z,CLRTX1
078A    01 0300                 LD      BC,0300H        ;32 * 24
078D                    CLRTX1:
078D    3E 20                   LD      A,' '           ;Fill space character code
078F    CD 0815                 CALL    FILVRM
0792    CD 0A7F                 CALL    CSHOME          ;Set cursor at home position
0795    21 FBB2                 LD      HL,LINTTB       ;Say all lines are terminated
0798    06 18                   LD      B,18H
079A                    CLRTX2:
079A    70                      LD      (HL),B          ;Load non 0 value
079B    23                      INC     HL
079C    10 FC                   DJNZ    CLRTX2
079E    C3 0B26                 JP      FNKSB
07A1                    CLSHRS:
                        ;
07A1    CD 0832                 CALL    CHGBDR          ;Set border color
07A4    01 1800                 LD      BC,1800H        ;Initialize color
07A7    C5                      PUSH    BC              ;Save this for future use
07A8    2A F3C9                 LD      HL,(GRPCOL)
07AB    3A F3EA                 LD      A,(BAKCLR)      ;Load background color
```

```
07AE    CD 0815                 CALL    FILVRM
07B1    2A F3CB                 LD      HL,(GRPCGP)
07B4    C1                      POP     BC              ;Load 6144
07B5    AF                      XOR     A
07B6                    JFLVRM:
07B6    C3 0815                 JP      FILVRM
07B9                    CLSMLT:
                        ;
07B9    CD 0832                 CALL    CHGBDR          ;Set border color
07BC    21 F3EA                 LD      HL,BAKCLR       ;Set all pixels to background color
07BF    7E                      LD      A,(HL)
07C0    87                      ADD     A,A
07C1    87                      ADD     A,A
07C2    87                      ADD     A,A
07C3    87                      ADD     A,A
07C4    B6                      OR      (HL)
07C5    2A F3D5                 LD      HL,(MLTCGP)     ;Set up address for write
07C8    01 0600                 LD      BC,0600H
07CB    18 E9                   JR      JFLVRM          ;Clear sprites (except sprite pattern)
```

```

07CD                WRTVRM:
                    ;
                    ; Write a byte to VRAM
                    ;
07CD    F5                  PUSH    AF              ;Save data to be written
07CE    CD 07DF             CALL    SETWRT
07D1    E3                  EX      (SP),HL
07D2    E3                  EX      (SP),HL
07D3    F1                  POP     AF
07D4    D3 98               OUT     (VDP.DRW),A
07D6    C9                  RET
07D7                RDVRM:
                    ;
                    ; Read a byte from VRAM
                    ;
07D7    CD 07EC             CALL    SETRD
07DA    E3                  EX      (SP),HL
07DB    E3                  EX      (SP),HL
07DC    DB 98               IN      A,(VDP.DRW)
07DE    C9                  RET
07DF                SETWRT:
                    ;
                    ; Set address for write to VDP
                    ;
                    ; Address is passed to HL
                    ;
07DF    7D                  LD      A,L
07E0    F3                  DI
07E1    D3 99               OUT     (VDP.CW),A
07E3    7C                  LD      A,H
```

```
07E4     E6 3F                    AND     00111111B
07E6     F6 40                    OR      01000000B       ;For write, set bit 6 high
07E8     D3 99                    OUT     (VDP.CW),A
07EA     FB                       EI
07EB     C9                       RET
07EC                      SETRD:
                          ;
                          ; Set address for read from VDP
                          ;
                          ; Address is passed to HL
                          ;
07EC     7D                       LD      A,L
07ED     F3                       DI
07EE     D3 99                    OUT     (VDP.CW),A
07F0     7C                       LD      A,H
07F1     E6 3F                    AND     00111111B
07F3     D3 99                    OUT     (VDP.CW),A
07F5     FB                       EI
07F6     C9                       RET
07F7                      CHGCLR:
                          ;
                          ; CHGCLR - changes foreground, background, and border color
                          ;
07F7     3A FCAF                  LD      A,(SCRMOD)      ;Are we in text mode
07FA     3D                       DEC     A
07FB     FA 0824                  JP      M,CHCLTX        ;Yes, change color in 40*24 text mode
07FE     F5                       PUSH    AF
07FF     CD 0832                  CALL    CHGBDR          ;Change border color for all
0802     F1                       POP     AF
0803     C0                       RET     NZ              ;No
0804     3A F3E9                  LD      A,(FORCLR)      ;We're in 32*24 text mode
```

```
0807    87                      ADD     A,A
0808    87                      ADD     A,A
0809    87                      ADD     A,A
080A    87                      ADD     A,A
080B    21 F3EA                 LD      HL,BAKCLR
080E    B6                      OR      (HL)
080F    2A F3BF                 LD      HL,(T32COL)
0812    01 0020                 LD      BC,20H
0815                    FILVRM:
0815    F5                      PUSH    AF
0816    CD 07DF                 CALL    SETWRT
0819                    FLVRM1:
0819    F1                      POP     AF
081A    D3 98                   OUT     (VDP.DRW),A
081C    F5                      PUSH    AF
081D    0B                      DEC     BC
081E    79                      LD      A,C
081F    B0                      OR      B
0820    20 F7                   JR      NZ,FLVRM1
0822    F1                      POP     AF
0823    C9                      RET
0824                    CHCLTX:
                        ;
0824    3A F3E9                 LD      A,(FORCLR)
0827    87                      ADD     A,A
0828    87                      ADD     A,A
0829    87                      ADD     A,A
082A    87                      ADD     A,A
082B    21 F3EA                 LD      HL,BAKCLR
082E    B6                      OR      (HL)
082F    47                      LD      B,A
```

```
0830    18 03                   JR      CHGBD1
0832                    CHGBDR:
                        ;
0832    3A F3EB                 LD      A,(BDRCLR)      ;Get border color
0835                    CHGBD1:
0835    47                      LD      B,A
0836    0E 07                   LD      C,7
0838    C3 057F                 JP      WRTVDP
```

```

083B                 TOTEXT:
                     ;
                     ; TOTEXT - Force screen to text mode
                     ;
083B    CD 0B9F              CALL    CHKSCR          ;Check current screen mode
083E    D8                   RET     C               ;We're in text mode
083F    3A FCB0              LD      A,(OLDSCR)
0842    CD FDBD              CALL    H.TOTE
0845    C3 084F              JP      CHGMOD          ;No, change to text mode then
0848                 CLS:
                     ;
                     ; CLS - clears screen
                     ;
0848    C0                   RET     NZ              ;Statement not ending
0849    E5                   PUSH    HL              ;Save text pointer
084A    CD 0777              CALL    CLSSUB
084D    E1                   POP     HL              ;Restore text pointer
084E    C9                   RET
084F                 CHGMOD:
                     ;
                     ; CHGMOD - changes mode of screen
                     ;
084F    3D                   DEC     A               ;Change to what mode
0850    FA 050E              JP      M,INITXT        ;To text mode
0853    CA 0538              JP      Z,INIT32
0856    3D                   DEC     A
0857    CA 05D2              JP      Z,INIGRP        ;To hires mode
085A    C3 061F              JP      INIMLT          ;To multicolor mode
                     SUBTTL - MSXIO -  Some entry points
```

```

085D                LPTOUT:
                    ;
                    ; Output a character to printer
                    ;
085D    CD FFB6             CALL    H.LPTO
0860    F5                  PUSH    AF              ;Save character to output
0861                CHPLP1:
0861    CD 046F             CALL    BREAKX          ;Check if aborted
0864    38 12               JR      C,LPTABO
0866    CD 0884             CALL    LPTSTT
0869    28 F6               JR      Z,CHPLP1        ;No
086B    F1                  POP     AF              ;Restore character
086C                CHPLP2:
086C    F5                  PUSH    AF              ;Save it again
086D    D3 91               OUT     (LPT.DW),A      ;Send to output port
086F    AF                  XOR     A               ;Generate strobe
0870    D3 90               OUT     (LPT.SB),A
0872    3D                  DEC     A
0873    D3 90               OUT     (LPT.SB),A
0875    F1                  POP     AF              ;Restore data output
0876    A7                  AND     A
0877    C9                  RET
0878                LPTABO:
                    ;
0878    AF                  XOR     A               ;Reset carriage position
0879    32 F415             LD      (LPTPOS),A
087C    3E 0D               LD      A,0DH           ;Send CR even if LPT not active
087E    CD 086C             CALL    CHPLP2
0881    F1                  POP     AF
0882    37                  SCF
```

```
0883    C9                      RET
0884                    LPTSTT:
                        ;
0884    CD FFBB                 CALL    H.LPTS
0887    DB 90                   IN      A,(90H)         ;LSB is 0 if ready
0889    0F                      RRCA
088A    0F                      RRCA
088B    3F                      CCF
088C    9F                      SBC     A,A
088D    C9                      RET                     ;No
088E                    POSIT:
                        ;
                        ; Position cursor to specified position
                        ;
088E    3E 1B                   LD      A,1BH
0890    DF                      RST     18H             ;OUTCHR
0891    3E 59                   LD      A,'Y'
0893    DF                      RST     18H
0894    7D                      LD      A,L
0895    C6 1F                   ADD     A,1FH           ;= ' ' - 1
0897    DF                      RST     18H
0898    7C                      LD      A,H
0899    C6 1F                   ADD     A,1FH
089B    DF                      RST     18H
089C    C9                      RET
089D                    CNVCHR:
                        ;
                        ; Convert character code
                        ;
089D    E5                      PUSH    HL
089E    F5                      PUSH    AF
```

```
089F    21 FCA6                 LD      HL,GRPHED       ;Preceeded by a header byte
08A2    AF                      XOR     A
08A3    BE                      CP      (HL)
08A4    77                      LD      (HL),A          ;Clear this since seen
08A5    28 0D                   JR      Z,CNVCH3        ;No
08A7    F1                      POP     AF
08A8    D6 40                   SUB     01000000B       ;Get rid of offset
08AA    FE 20                   CP      ' '             ;Valid range
08AC    38 04                   JR      C,CNVCH2        ;Yes
08AE    C6 40                   ADD     A,01000000B     ;Compensate value
08B0                    CNVCH1:
08B0    BF                      CP      A               ;Set Z flag
08B1    37                      SCF                     ;Make sure carry is cleared
08B2                    CNVCH2:
08B2    E1                      POP     HL
08B3    C9                      RET
08B4                    CNVCH3:
                        ;
08B4    F1                      POP     AF
08B5    FE 01                   CP      1               ;Graphic header
08B7    20 F7                   JR      NZ,CNVCH1       ;No, do not modify
08B9    77                      LD      (HL),A          ;Set GRPHED flag
08BA    E1                      POP     HL              ;Carry is clear indicating one more byte is
08BB    C9                      RET                     ;required
                        SUBTTL - MSXIO -  Output a character to CRT
```

```

     08BC                CHPUT:
                         ;
     08BC     E5                   PUSH    HL
     08BD     D5                   PUSH    DE
     08BE     C5                   PUSH    BC
     08BF     F5                   PUSH    AF
     08C0     CD FDA4              CALL    H.CHPU
     08C3     CD 0B9F              CALL    CHKSCR           ;Are we in text mode
     08C6     30 12                JR      NC,POPALL        ;No, ignore this
     08C8     CD 0A2E              CALL    CKERCS           ;Erase old cursor if cursor enabled
     08CB     F1                   POP     AF
     08CC     F5                   PUSH    AF
     08CD     CD 08DF              CALL    CHPUT1
     08D0     CD 09E1              CALL    CKDPCS           ;Display new cursor if cursor enabled
     08D3     3A F3DD              LD      A,(CSRX)
     08D6     3D                   DEC     A
     08D7     32 F661              LD      (TTYPOS),A
     08DA                POPALL:
     08DA     F1                   POP     AF
     08DB                PBDHRT:
     08DB     C1                   POP     BC
     08DC     D1                   POP     DE
     08DD     E1                   POP     HL
     08DE     C9                   RET
     08DF                CHPUT1:
                         ;
     08DF     CD 089D              CALL    CNVCHR           ;Convert character code
     08E2     D0                   RET     NC               ;Was a graphic header, wait for next
     08E3     4F                   LD      C,A              ;Save character code in [C]
     08E4     20 0D                JR      NZ,CHPUT3        ;Converted code, send as is
```

```
08E6    21 FCA7                 LD      HL,ESCCNT
08E9    7E                      LD      A,(HL)          ;Are we executing escape sequence
08EA    A7                      AND     A               ;
08EB    C2 098F                 JP      NZ,INESC        ;Yes
08EE    79                      LD      A,C             ;Restore character
08EF    FE 20                   CP      ' '             ;Control code
08F1    38 21                   JR      C,CNTPUT        ;Yes
08F3                    CHPUT3:
08F3    2A F3DC                 LD      HL,(CSRY)
08F6    FE 7F                   CP      7FH             ;Rubout
08F8    CA 0AE3                 JP      Z,RUBOUT        ;Yes
08FB    CD 0BE6                 CALL    PUTVRM          ;Convert to raw code and write to VRAM
08FE    CD 0A44                 CALL    RIGHT           ;Advance cursor
0901    C0                      RET     NZ              ;All done if not wrapped to next line
0902    AF                      XOR     A
0903    CD 0C2B                 CALL    SETTRM          ;Unterminate this line
0906    26 01                   LD      H,1             ;Go to start of the next line
0908                    LF:
                        ;
                        ; Line feed
                        ;
0908    CD 0A61                 CALL    DOWN            ;Down cursor
090B    C0                      RET     NZ              ;Exit if not at bottom
090C    CD 0A69                 CALL    STOCSR
090F    2E 01                   LD      L,1             ;L:=window top line
0911    C3 0A88                 JP      DELLN0          ;Scroll up by deleting the first line
0914                    CNTPUT:
                        ;
                        ; Following control codes are supported
                        ;
                        ; 7 Bell
```

```
                                ; 8 Back space
                                ; 9 Tab
                                ; 10 Line feed
                                ; 11 Cursor home
                                ; 12 Clear screen
                                ; 13 Carriage return
                                ;
                                ; 27 Enter escape sequence
                                ; 28 Cursor right
                                ; 29 Cursor left
                                ; 30 Cursor up
                                ; 31 Cursor down
                                ;
0914    21 092D                         LD      HL,JMPBC
0917    0E 0C                           LD      C,0CH
0919                            INDJMP:
0919    23                              INC     HL
091A    23                              INC     HL
091B    A7                              AND     A               ;Make sure carry is cleared
091C    0D                              DEC     C
091D    F8                              RET     M               ;Undefined function
091E    BE                              CP      (HL)            ;Found?
091F    23                              INC     HL
0920    20 F7                           JR      NZ,INDJMP       ;No
0922    4E                              LD      C,(HL)          ;Get routine address in BC
0923    23                              INC     HL              ;
0924    46                              LD      B,(HL)          ;
0925    2A F3DC                         LD      HL,(CSRY)       ;Jump to each routine with cursor pos
0928    CD 092D                         CALL    JMPBC
092B    AF                              XOR     A               ;Tell screen editor not to echo this character
092C    C9                              RET
```

```
092D                  JMPBC:
                      ;
092D    C5                    PUSH    BC
092E    C9                    RET
                      ;
                      ;          Function dispatch table
                      ;
092F                  CNTTBL:
092F    07                    DB      7               ;Beep
0930    1113                  DW      BEEP
0932    08                    DB      8               ;Back space
0933    0A4C                  DW      BS
0935    09                    DB      9               ;Tabulation
0936    0A71                  DW      TAB
0938    0A                    DB      10              ;Line feed
0939    0908                  DW      LF
093B    0B                    DB      11              ;Home
093C    0A7F                  DW      CSHOME
093E    0C                    DB      12              ;Clear
093F    077E                  DW      CLRTXT
0941    0D                    DB      13              ;Carriage return
0942    0A81                  DW      CR
0944    1B                    DB      27              ;Enter escape sequence
0945    0989                  DW      ENTESC
0947    1C                    DB      28              ;Cursor right
0948    0A5B                  DW      ADVCUR
094A    1D                    DB      29              ;Cursor left
094B    0A4C                  DW      BS
094D    1E                    DB      30              ;Cursor up
094E    0A57                  DW      UP
0950    1F                    DB      31              ;Cursor down
```

```
 1936     0951    0A61                    DW      DOWN
 1937                               SUBTTL - MSXIO -  Escape sequence handler
```

```

0953                    ESCTBL:
0953    6A                      DB      "j"             ;Clear screen
0954    077E                    DW      CLRTXT
0956    45                      DB      "E"             ;Clear screen
0957    077E                    DW      CLRTXT          ; To maintain compatibility with VT52
0959    4B                      DB      "K"             ;Erase to end-of-line
095A    0AEE                    DW      EOL
095C    4A                      DB      "J"             ;Erase to end-of-page
095D    0B05                    DW      EOP
095F    6C                      DB      "l"             ;Erase entire line
0960    0AEC                    DW      ELN
0962    4C                      DB      "L"             ;Insert a line
0963    0AB4                    DW      ILN
0965    4D                      DB      "M"             ;Delete a line
0966    0A85                    DW      DLN
0968    59                      DB      "Y"             ;Locate cursor
0969    0986                    DW      LOC
096B    41                      DB      "A"             ;Cursor up
096C    0A57                    DW      UP
096E    42                      DB      "B"             ;Cursor down
096F    0A61                    DW      DOWN
0971    43                      DB      "C"             ;Cursor right
0972    0A44                    DW      RIGHT
0974    44                      DB      "D"             ;Cursor left
0975    0A55                    DW      LEFT
0977    48                      DB      "H"             ;Cursor home
0978    0A7F                    DW      CSHOME
097A    78                      DB      "x"             ;Set modes
097B    0980                    DW      SETMOD
097D    79                      DB      "y"             ;Reset modes
```

```
097E    0983                DW      RSTMOD
0980                SETMOD:
                    ;
                    ; Function dispatch table
                    ;
0980    3E 01               LD      A,1
0982    01                  DB      1
0983                RSTMOD:
0983    3E 02               LD      A,2
0985    01                  DB      1
0986                LOC:
0986    3E 04               LD      A,4             ;Say row is expected next
0988    01                  DB      1               ;'LXI B' instruction
0989                ENTESC:
0989    3E FF               LD      A,0FFH          ;Tell him we're in escape sequence
098B    32 FCA7             LD      (ESCCNT),A
098E    C9                  RET
```

```

098F                INESC:
                    ;
098F    F2 099D             JP      P,INESC1        ;Arguments expected
0992    36 00               LD      (HL),0          ;Exit from escape sequence
0994    79                  LD      A,C             ;Restore character
0995    21 0951             LD      HL,ESCTBL-2
0998    0E 0F               LD      C,0FH           ;Number of ESC handler entries
099A    C3 0919             JP      INDJMP
099D                INESC1:
                    ;
099D    3D                  DEC     A               ;Set modes?
099E    28 1E               JR      Z,GOSET         ;Yes
09A0    3D                  DEC     A               ;Reset modes?
09A1    28 25               JR      Z,GORSET
09A3    3D                  DEC     A
09A4    77                  LD      (HL),A          ;Update ESCCNT
09A5    3A F3B0             LD      A,(LINLEN)      ;Assume column expected
09A8    11 F3DD             LD      DE,CSRX         ;
09AB    28 06               JR      Z,INESC2        ;Column expected
09AD    36 03               LD      (HL),3
09AF    CD 0C32             CALL    GETLEN          ;Row expected
09B2    1B                  DEC     DE              ;Point CSRY
09B3                INESC2:
09B3    47                  LD      B,A             ;Get max limit in B
09B4    79                  LD      A,C             ;Restore character
09B5    D6 20               SUB     ' '             ;0-xx
09B7    B8                  CP      B
09B8    3C                  INC     A
09B9    12                  LD      (DE),A
09BA    D8                  RET     C               ;Legal value
```

```
09BB    78                      LD      A,B             ;Substitute by possible largest value
09BC    12                      LD      (DE),A
09BD    C9                      RET
09BE                    GOSET:
                        ;
                        ; Set various modes
                        ;
09BE    77                      LD      (HL),A          ;Exit from escape sequence
09BF    79                      LD      A,C             ;Restore character
09C0    D6 34                   SUB     '4'             ;Block cursor?
09C2    28 0B                   JR      Z,STSTYL        ;Yes
09C4    3D                      DEC     A               ;Cursor off?
09C5    28 0F                   JR      Z,STCSSW        ;Yes, reset cursor-enable switch
09C7    C9                      RET                     ;Unimplemented feature
09C8                    GORSET:
                        ;
                        ; Reset various modes
                        ;
09C8    77                      LD      (HL),A          ;Exit from escape sequence
09C9    79                      LD      A,C             ;Restore character
09CA    D6 34                   SUB     '4'             ;Underscore cursor?
09CC    20 05                   JR      NZ,RSET10       ;No, try next
09CE    3C                      INC     A
09CF                    STSTYL:
09CF    32 FCAA                 LD      (CSTYLE),A
09D2    C9                      RET
09D3                    RSET10:
                        ;
09D3    3D                      DEC     A               ;Cursor on?
09D4    C0                      RET     NZ              ;No, unimplemented feature
09D5    3C                      INC     A
```

```
09D6                    STCSSW:
09D6    32 FCA9                 LD      (CSRSW),A
09D9    C9                      RET
09DA                    CKDPC0:
                        ;
                        ; Display cursor if disabled
                        ;
09DA    3A FCA9                 LD      A,(CSRSW)
09DD    A7                      AND     A
09DE    C0                      RET     NZ
09DF    18 05                   JR      DSPCSR
09E1                    CKDPCS:
                        ;
                        ; Display cursor if enabled
                        ;
09E1    3A FCA9                 LD      A,(CSRSW)
09E4    A7                      AND     A
09E5    C8                      RET     Z
09E6                    DSPCSR:
                        ;
                        ; Display a cursor
                        ;
09E6    CD FDA9                 CALL    H.DSPC
09E9    CD 0B9F                 CALL    CHKSCR
09EC    D0                      RET     NC
09ED    2A F3DC                 LD      HL,(CSRY)       ;Get current cursor position
09F0    E5                      PUSH    HL              ;Save it for future use
09F1    CD 0BD8                 CALL    GETVRM          ;Get a raw character at cursor
09F4    32 FBCC                 LD      (CODSAV),A      ;Remember this code
09F7    6F                      LD      L,A             ;Then read pattern for this code
09F8    26 00                   LD      H,0
```

```
09FA    29                      ADD     HL,HL           ; [A] * 8
09FB    29                      ADD     HL,HL
09FC    29                      ADD     HL,HL
09FD    EB                      EX      DE,HL
09FE    2A F924                 LD      HL,(CGPBAS)
0A01    E5                      PUSH    HL
0A02    19                      ADD     HL,DE
0A03    CD 0BA5                 CALL    GET8B
0A06    21 FC1F                 LD      HL,BUFEND+7     ;Make a complement of this pattern
0A09    06 08                   LD      B,8             ;Assume full reverse cursor
0A0B    3A FCAA                 LD      A,(CSTYLE)
0A0E    A7                      AND     A
0A0F    28 02                   JR      Z,DSPCS1        ;Good assumption
0A11    06 03                   LD      B,3             ;No, reverse bottom 3 lines only
0A13                    DSPCS1:
0A13    7E                      LD      A,(HL)
0A14    2F                      CPL
0A15    77                      LD      (HL),A
0A16    2B                      DEC     HL
0A17    10 FA                   DJNZ    DSPCS1
0A19    E1                      POP     HL              ;Assign this pattern to 255
0A1A    01 07F8                 LD      BC,07F8H
0A1D    09                      ADD     HL,BC
0A1E    CD 0BBE                 CALL    PUT8B
0A21    E1                      POP     HL              ;Restore cursor position
0A22    0E FF                   LD      C,0FFH          ;Get code for cursor
0A24    C3 0BE6                 JP      PUTVRM          ;Set it at cursor position
0A27                    CKERC0:
                        ;
                        ; Erase cursor if disabled
                        ;
```

```
0A27    3A FCA9                 LD      A,(CSRSW)
0A2A    A7                      AND     A
0A2B    C0                      RET     NZ
0A2C    18 05                   JR      ERACSR
0A2E                    CKERCS:
                        ;
                        ; Erase a cursor if enabled
                        ;
0A2E    3A FCA9                 LD      A,(CSRSW)
0A31    A7                      AND     A
0A32    C8                      RET     Z
0A33                    ERACSR:
                        ;
                        ; Erase cursor
                        ;
0A33    CD FDAE                 CALL    H.ERAC
0A36    CD 0B9F                 CALL    CHKSCR
0A39    D0                      RET     NC
0A3A    2A F3DC                 LD      HL,(CSRY)
0A3D    3A FBCC                 LD      A,(CODSAV)      ;Get old code
0A40    4F                      LD      C,A
0A41    C3 0BE6                 JP      PUTVRM          ;Restore old code
                        ;
                        SUBTTL - MSXIO -  Cursor movements
```

```
0A44                RIGHT:
                    ;
                    ; Cursor right
                    ;
0A44    3A F3B0             LD      A,(LINLEN)
0A47    BC                  CP      H               ;Are we at the right-end of line?
0A48    C8                  RET     Z               ;Yes, return with Z flag
0A49    24                  INC     H               ;Go to next column
0A4A    18 1D               JR      STOCSR
0A4C                BS:
                    ;
                    ; Back space
                    ;
0A4C    CD 0A55             CALL    LEFT
0A4F    C0                  RET     NZ              ;Not at left-end
0A50    3A F3B0             LD      A,(LINLEN)
0A53    67                  LD      H,A
0A54    11                  DB      11H             ;'LXI D,' instruction
0A55                LEFT:
                    ;
                    ; Cursor left
                    ;
0A55    25                  DEC     H               ;Are we at the left-end of line?
0A56    3E                  DB      3EH             ;'MVI A,' instruction
0A57                UP:
                    ;
                    ; Cursor up
                    ;
0A57    2D                  DEC     L               ;Are we at the top of any window?
0A58    C8                  RET     Z               ;Yes, return with Z flag
```

```
0A59    18 0E                   JR      STOCSR
0A5B                    ADVCUR:
                        ;
                        ; Advance cursor
                        ;
0A5B    CD 0A44                 CALL    RIGHT
0A5E    C0                      RET     NZ
0A5F    26 01                   LD      H,1
0A61                    DOWN:
                        ;
                        ; Cursor down
                        ;
0A61    CD 0C32                 CALL    GETLEN          ;Get an actual bottom of screen
0A64    BD                      CP      L               ;Are we at the bottom of screen?
0A65    C8                      RET     Z               ;Yes, return with Z flag
0A66    38 05                   JR      C,DOWN1         ;We're below screen bottom
0A68    2C                      INC     L               ;Go to next line
0A69                    STOCSR:
0A69    22 F3DC                 LD      (CSRY),HL
0A6C    C9                      RET
0A6D                    DOWN1:
                        ;
0A6D    2D                      DEC     L
0A6E    AF                      XOR     A
0A6F    18 F8                   JR      STOCSR
0A71                    TAB:
                        ;
                        ; Tabulation
                        ;
0A71    3E 20                   LD      A,' '
0A73    CD 08DF                 CALL    CHPUT1
```

```
0A76     3A F3DD                 LD      A,(CSRX)
0A79     3D                      DEC     A
0A7A     E6 07                   AND     7
0A7C     20 F3                   JR      NZ,TAB
0A7E     C9                      RET
0A7F                     CSHOME:
                         ;
                         ; Cursor home
                         ;
0A7F     2E 01                   LD      L,1
0A81                     CR:
                         ;
                         ; Carriage return
                         ;
0A81     26 01                   LD      H,1             ;CR only, not new-line
0A83     18 E4                   JR      STOCSR
                         ;
                         SUBTTL - MSXIO -  Line insert and delete of CRT
```

```

0A85                DLN:
                    ;
                    ; Delete a line specified by [L]
                    ;
                    ; Cursor should be set at the top of line
                    ;
0A85  CD 0A81               CALL    CR
0A88                DELLN0:
0A88  CD 0C32               CALL    GETLEN          ;Get an actual height of screen
0A8B  95                    SUB     L
0A8C  D8                    RET     C               ;Something is wrong
0A8D  CA 0AEC               JP      Z,ELN           ;Delete the bottom line only
0A90  E5                    PUSH    HL              ;Save row
0A91  F5                    PUSH    AF              ;Save counter (# of lines to be moved upward)
0A92  4F                    LD      C,A
0A93  06 00                 LD      B,0
0A95  CD 0C1D               CALL    GETTRM          ;Get address of [LINTTB] in [DE]
0A98  6B                    LD      L,E
0A99  62                    LD      H,D
0A9A  23                    INC     HL
0A9B  ED B0                 LDIR
0A9D  21 FBCA               LD      HL,FSTPOS
0AA0  35                    DEC     (HL)
0AA1  F1                    POP     AF
0AA2  E1                    POP     HL
0AA3                DELLN1:
0AA3  F5                    PUSH    AF              ;Save counter
0AA4  2C                    INC     L
0AA5  CD 0BAA               CALL    GET1LN          ;Get 1 line specified by L
0AA8  2D                    DEC     L
```

```
0AA9    CD 0BC3                 CALL    PUT1LN          ;Put 1 line specified by L
0AAC    2C                      INC     L
0AAD    F1                      POP     AF              ;Restore counter
0AAE    3D                      DEC     A
0AAF    20 F2                   JR      NZ,DELLN1
0AB1    C3 0AEC                 JP      ELN             ;Blank bottom line
0AB4                    ILN:
                        ;
                        ; Insert a line
                        ;
                        ; Cursor should be set at the top of line
                        ;
0AB4    CD 0A81                 CALL    CR
0AB7                    INSLN0:
0AB7    CD 0C32                 CALL    GETLEN          ;Get an actual height of screen
0ABA    67                      LD      H,A
0ABB    95                      SUB     L
0ABC    D8                      RET     C               ;Something is wrong!!
0ABD    CA 0AEC                 JP      Z,ELN
0AC0    6C                      LD      L,H
0AC1    E5                      PUSH    HL              ;Save row to be inserted
0AC2    F5                      PUSH    AF              ;Save # of lines to be moved downward
0AC3    4F                      LD      C,A
0AC4    06 00                   LD      B,0
0AC6    CD 0C1D                 CALL    GETTRM
0AC9    6B                      LD      L,E
0ACA    62                      LD      H,D
0ACB    E5                      PUSH    HL              ;Save pointer to [LINTTB] for the bottom line
0ACC    2B                      DEC     HL              ;Form source address
0ACD    ED B8                   LDDR
0ACF    E1                      POP     HL
```

```
0AD0    74                      LD      (HL),H          ;Make sure the bottom line is terminated
0AD1    F1                      POP     AF
0AD2    E1                      POP     HL
0AD3                    INSLN1:
0AD3    F5                      PUSH    AF              ;Save counter
0AD4    2D                      DEC     L
0AD5    CD 0BAA                 CALL    GET1LN
0AD8    2C                      INC     L
0AD9    CD 0BC3                 CALL    PUT1LN
0ADC    2D                      DEC     L
0ADD    F1                      POP     AF              ;Restore counter
0ADE    3D                      DEC     A
0ADF    20 F2                   JR      NZ,INSLN1
0AE1    18 09                   JR      ELN
                        ;
                        SUBTTL - MSXIO -  Character(s) erase
```

```
         0AE3                   RUBOUT:
                                ;
                                ; Erase previous character
                                ;
         0AE3    CD 0A4C                CALL    BS              ;Back space
         0AE6    C8                     RET     Z               ;We're at the top of screen
         0AE7    0E 20                  LD      C,' '           ;Overstrike with a space
         0AE9    C3 0BE6                JP      PUTVRM
         0AEC                   ELN:
                                ;
                                ; Erase entire line
                                ;
                                ; Cursor should remain unchanged
                                ;
         0AEC    26 01                  LD      H,1
         0AEE                   EOL:
                                ;
                                ; Erase to end-of-line
                                ;
                                ; Cursor should remain unchanged
                                ;
         0AEE    CD 0C29                CALL    TERMIN
         0AF1    E5                     PUSH    HL              ;Save current position (column)
         0AF2    CD 0BF2                CALL    VADDR
         0AF5    CD 07DF                CALL    SETWRT
         0AF8    E1                     POP     HL              ;Restore current position
         0AF9                   EREOL1:
         0AF9    3E 20                  LD      A,' '           ;Overstrike with a space
         0AFB    D3 98                  OUT     (VDP.DRW),A
         0AFD    24                     INC     H
```

```
        0AFE    3A F3B0                 LD      A,(LINLEN)
        0B01    BC                      CP      H
        0B02    30 F5                   JR      NC,EREOL1
        0B04    C9                      RET
        0B05                    EOP:
                                ;
                                ; Erase to end-of-page
                                ;
                                ; Cursor should remain unchanged
                                ;
        0B05    E5                      PUSH    HL              ;Save current position
        0B06    CD 0AEE                 CALL    EOL             ;Erase to end-of-line
        0B09    E1                      POP     HL              ;Restore current position
        0B0A    CD 0C32                 CALL    GETLEN          ;Get an actual height of CRT
        0B0D    BD                      CP      L
        0B0E    D8                      RET     C               ;Something is wrong
        0B0F    C8                      RET     Z               ;All done
        0B10    26 01                   LD      H,1
        0B12    2C                      INC     L
        0B13    18 F0                   JR      EOP
                                ;
                                SUBTTL - MSXIO -  Function keys display/erase.
```

```

0B15                    ERAFNK:
                        ;
                        ; Erase function key
                        ;
0B15    CD FDB8                 CALL    H.ERAF
0B18    AF                      XOR     A               ;Say no function key is displayed
0B19    CD 0B9C                 CALL    SETCHK
0B1C    D0                      RET     NC              ;We're not in text mode, just set flag
0B1D    E5                      PUSH    HL              ;Save possible text pointer
0B1E    2A F3B1                 LD      HL,(CRTCNT)     ;Erase last line
0B21    CD 0AEC                 CALL    ELN
0B24    E1                      POP     HL              ;Restore possible text pointer
0B25    C9                      RET
0B26                    FNKSB:
                        ;
                        ; Display function key if enabled
                        ;
0B26    3A F3DE                 LD      A,(CNSDFG)      ;Now being displayed?
0B29    A7                      AND     A
0B2A    C8                      RET     Z               ;No
0B2B                    DSPFNK:
                        ;
                        ; Display function key
                        ;
0B2B    CD FDB3                 CALL    H.DSPF
0B2E    3E FF                   LD      A,0FFH          ;Say function key is displayed
0B30    CD 0B9C                 CALL    SETCHK
0B33    D0                      RET     NC              ;We're not in text mode, just set flag
0B34    E5                      PUSH    HL              ;Save possible text pointer
0B35    3A F3DC                 LD      A,(CSRY)
```

```
0B38    21 F3B1                 LD      HL,CRTCNT
0B3B    BE                      CP      (HL)
0B3C    3E 0A                   LD      A,0AH           ;Scroll up if we're at the bottom of screen
0B3E    20 01                   JR      NZ,NTBOTM
0B40    DF                      RST     18H
0B41                    NTBOTM:
0B41    3A FBEB                 LD      A,(SFTKEY)      ;Get current shift status
0B44    0F                      RRCA
0B45    21 F87F                 LD      HL,FNKSTR       ;Assume shift not pressed
0B48    3E 01                   LD      A,1
0B4A    38 04                   JR      C,DSPFK1        ;Good assumption
0B4C    21 F8CF                 LD      HL,FNKSTR+80    ;Shift is being pressed
0B4F    AF                      XOR     A
0B50                    DSPFK1:
0B50    32 FBCD                 LD      (FNKSWI),A      ;Mark which part of function key is displayed
0B53    11 FC18                 LD      DE,BUFEND       ;Set temporary destination
0B56    D5                      PUSH    DE
0B57    06 28                   LD      B,'('           ;=40
0B59    3E 20                   LD      A,' '
0B5B                    DSFKCL:
0B5B    12                      LD      (DE),A
0B5C    13                      INC     DE
0B5D    10 FC                   DJNZ    DSFKCL
0B5F    D1                      POP     DE              ;Restore temporary destination in [DE]
0B60    0E 05                   LD      C,5             ;Total number of keys
0B62    3A F3B0                 LD      A,(LINLEN)      ;Calculate (LINLEN-4) / 5
0B65    D6 04                   SUB     4
0B67    38 2B                   JR      C,DSPFKE        ;Not enough room for function keys
0B69    06 FF                   LD      B,0FFH
0B6B                    DSPFK4:
0B6B    04                      INC     B
```

```
0B6C    D6 05                   SUB     5
0B6E    30 FB                   JR      NC,DSPFK4
0B70    78                      LD      A,B
0B71    A7                      AND     A
0B72    28 20                   JR      Z,DSPFKE        ;No enough room
0B74    3E                      DB      3EH             ;Skip next byte
0B75                    DSPFK2:
0B75    13                      INC     DE              ;Put separator space
0B76    C5                      PUSH    BC              ;Save key counter
0B77    0E 00                   LD      C,0             ;Reset # of characters actually fetched
0B79                    DSPFK5:
0B79    7E                      LD      A,(HL)          ;Get from function key string
0B7A    23                      INC     HL              ;Prepare for next fetch
0B7B    0C                      INC     C
0B7C    CD 089D                 CALL    CNVCHR
0B7F    30 F8                   JR      NC,DSPFK5       ;This is a graphic header, fetch more
0B81    20 04                   JR      NZ,DSPFK8       ;Converted graphics character, store this
0B83    FE 20                   CP      ' '             ;Printable?
0B85    38 01                   JR      C,DSPFK6        ;No, ignore this
0B87                    DSPFK8:
0B87    12                      LD      (DE),A
0B88                    DSPFK6:
0B88    13                      INC     DE
0B89    10 EE                   DJNZ    DSPFK5
0B8B    3E 10                   LD      A,10H
0B8D    91                      SUB     C
0B8E    4F                      LD      C,A             ;Skip rest
0B8F    09                      ADD     HL,BC
0B90    C1                      POP     BC              ;Restore counter
0B91    0D                      DEC     C
0B92    20 E1                   JR      NZ,DSPFK2
```

```
0B94                  DSPFKE:
0B94     2A F3B1              LD      HL,(CRTCNT)     ;Display at the lowest line
0B97     CD 0BC3              CALL    PUT1LN
0B9A     E1                   POP     HL              ;Restore possible text pointer
0B9B     C9                   RET
                      ;
                      SUBTTL - MSXIO -  Low level routines
```

```

0B9C                   SETCHK:
                       ;
                       ; Set CNSDFG and check current screen mode
                       ;
0B9C    32 F3DE                LD      (CNSDFG),A
0B9F                   CHKSCR:
                       ;
                       ; Check current screen mode
                       ;
0B9F    3A FCAF                LD      A,(SCRMOD)
0BA2    FE 02                  CP      2
0BA4    C9                     RET                     ;Return with the status
0BA5                   GET8B:
                       ;
                       ; Get 8 bytes from HL
                       ;
0BA5    E5                     PUSH    HL
0BA6    0E 08                  LD      C,8
0BA8    18 0A                  JR      GET1L1
0BAA                   GET1LN:
                       ;
                       ; Get character and attribute of position specified by H,L
                       ;
                       ; Character returned in C
                       ;
0BAA    E5                     PUSH    HL
0BAB    26 01                  LD      H,1
0BAD    CD 0BF2                CALL    VADDR
0BB0    3A F3B0                LD      A,(LINLEN)
0BB3    4F                     LD      C,A
```

```
0BB4                    GET1L1:
0BB4    06 00                   LD      B,0
0BB6    11 FC18                 LD      DE,BUFEND       ;Storage for 1 line
0BB9    CD 070F                 CALL    LDIRMV
0BBC    E1                      POP     HL
0BBD    C9                      RET
0BBE                    PUT8B:
                        ;
0BBE    E5                      PUSH    HL
0BBF    0E 08                   LD      C,8
0BC1    18 0A                   JR      PUT1L1
0BC3                    PUT1LN:
                        ;
0BC3    E5                      PUSH    HL
0BC4    26 01                   LD      H,1
0BC6    CD 0BF2                 CALL    VADDR
0BC9    3A F3B                  LD      A,(LINLEN)
0BCC    4F                      LD      C,A
0BCD                    PUT1L1:
0BCD    06 00                   LD      B,0
0BCF    EB                      EX      DE,HL
0BD0    21 FC18                 LD      HL,BUFEND
0BD3    CD 0744                 CALL    LDIRVM
0BD6    E1                      POP     HL
0BD7    C9                      RET
0BD8                    GETVRM:
                        ;
0BD8    E5                      PUSH    HL              ;Save coordinate
0BD9    CD 0BF2                 CALL    VADDR           ;Calculate VRAM address
0BDC    CD 07EC                 CALL    SETRD           ;Set up VDP for read
0BDF    E3                      EX      (SP),HL
```

```
0BE0    E3                       EX      (SP),HL
0BE1    DB 98                    IN      A,(VDP.DRW)     ;Get character code in C
0BE3    4F                       LD      C,A
0BE4    E1                       POP     HL              ;Restore coordinate
0BE5    C9                       RET
0BE6                     PUTVRM:
                         ;
0BE6    E5                       PUSH    HL
0BE7    CD 0BF2                  CALL    VADDR
0BEA    CD 07DF                  CALL    SETWRT
0BED    79                       LD      A,C
0BEE    D3 98                    OUT     (VDP.DRW),A
0BF0    E1                       POP     HL
0BF1    C9                       RET
0BF2                     VADDR:
                         ;
                         ; Calculate buffer address out of H,L (column,row)
                         ;
                         ; address returned in HL
                         ;
0BF2    C5                       PUSH    BC
0BF3    5C                       LD      E,H             ;Get column in L
0BF4    26 00                    LD      H,0
0BF6    54                       LD      D,H
0BF7    2D                       DEC     L
0BF8    29                       ADD     HL,HL
0BF9    29                       ADD     HL,HL
0BFA    29                       ADD     HL,HL
0BFB    4D                       LD      C,L
0BFC    44                       LD      B,H
0BFD    29                       ADD     HL,HL
```

```
0BFE    29                      ADD     HL,HL
0BFF    19                      ADD     HL,DE
0C00    3A FCAF                 LD      A,(SCRMOD)
0C03    A7                      AND     A
0C04    3A F3B0                 LD      A,(LINLEN)
0C07    28 04                   JR      Z,VADDR1
0C09    D6 22                   SUB     '"'
0C0B    18 03                   JR      VADDR2
0C0D                    VADDR1:
                        ;
0C0D    09                      ADD     HL,BC
0C0E    D6 2A                   SUB     41+1
0C10                    VADDR2:
0C10    2F                      CPL
0C11    A7                      AND     A
0C12    1F                      RRA
0C13    5F                      LD      E,A
0C14    19                      ADD     HL,DE
0C15    EB                      EX      DE,HL
0C16    2A F922                 LD      HL,(NAMBAS)
0C19    19                      ADD     HL,DE
0C1A    2B                      DEC     HL
0C1B    C1                      POP     BC
0C1C    C9                      RET
0C1D                    GETTRM:
                        ;
                        ; Get value of line-terminator-table and affect flags
                        ;
                        ; Entry: L has the line #
                        ; Exit: DE has the address of corresponding terminator byte.
                        ; Z flag is affected.
                        ;
0C1D    E5                      PUSH    HL              ;Save HL
0C1E    11 FBB1                 LD      DE,BASROM
0C21    26 00                   LD      H,0
0C23    19                      ADD     HL,DE           ;Get address of table
0C24    7E                      LD      A,(HL)
0C25    EB                      EX      DE,HL           ;Move address to DE
0C26    E1                      POP     HL              ;Restore HL
0C27    A7                      AND     A               ;Affect flags
0C28    C9                      RET
0C29                    TERMIN:
                        ;
0C29    3E                      DB      3EH             ;Load non 0 value in Acc
0C2A                    UNTERM:
0C2A    AF                      XOR     A
0C2B                    SETTRM:
0C2B    F5                      PUSH    AF
0C2C    CD 0C1D                 CALL    GETTRM          ;Get address of terminator byte in DE
0C2F    F1                      POP     AF
0C30    12                      LD      (DE),A          ;Change table
0C31    C9                      RET
0C32                    GETLEN:
                        ;
                        ; Get an actual height of screen
                        ;
0C32    3A F3DE                 LD      A,(CNSDFG)      ;0 or -1
0C35    E5                      PUSH    HL
0C36    21 F3B1                 LD      HL,CRTCNT
0C39    86                      ADD     A,(HL)
0C3A    E1                      POP     HL
0C3B    C9                      RET
```

```
                        ;
                        SUBTTL - MSXIO -  Keyboard encoding routines
```

```
        0C3C                    KEYINT:
                                ;
                                ; Encode keyboard
                                ;
                                ; Timer interrupt routine
                                ;
        0C3C    E5                      PUSH    HL              ;Save all registers
        0C3D    D5                      PUSH    DE
        0C3E    C5                      PUSH    BC
        0C3F    F5                      PUSH    AF
        0C40    D9                      EXX
        0C41    08                      EX      AF,AF'
        0C42    E5                      PUSH    HL
        0C43    D5                      PUSH    DE
        0C44    C5                      PUSH    BC
        0C45    F5                      PUSH    AF
        0C46    FD E5                   PUSH    IY
        0C48    DD E5                   PUSH    IX
        0C4A    CD FD9A                 CALL    H.KEYI          ;To allow other interrupts than 60Hz timer
        0C4D    DB 99                   IN      A,(VDP.SR)      ;Clear possible interrupt request
        0C4F    A7                      AND     A               ;Interrupt requested by VDP?
        0C50    F2 0D02                 JP      P,INTRET        ;No, skip the rest
        0C53    CD FD9F                 CALL    H.TIMI          ;To allow timer interrupt to be
                                                                ;used elsewhere.
        0C56    FB                      EI                      ;Now that it became obvious that VDP
                                                                ;generated the interrupt, we re-enable
                                                                ;interrupt here to allow RS232C's
                                                                ;interrupt or something like that.
        0C57    32 F3E7                 LD      (STATFL),A      ;Store this new status
        0C5A    E6 20                   AND     ' '             ;Collision detected?
```

```
0C5C    21 FC6D                 LD      HL,TRPTBL+33    ;Assume so
0C5F    C4 0EF1                 CALL    NZ,REQTRP       ;Request trap if so
                        ;
                        ; Check interval trap
                        ;
0C62    2A FCA2                 LD      HL,(INTCNT)     ;Count down interval count
0C65    2B                      DEC     HL
0C66    7C                      LD      A,H
0C67    B5                      OR      L
0C68    20 09                   JR      NZ,NTINTT       ;Not yet reached 0
0C6A    21 FC7F                 LD      HL,TRPTBL+3*17  ;Request trap
0C6D    CD 0EF1                 CALL    REQTRP
0C70    2A FCA0                 LD      HL,(INTVAL)     ;Load initial value
0C73                    NTINTT:
0C73    22 FCA2                 LD      (INTCNT),HL     ;Update interval count
                        ;
                        ; Increment jiffy count
                        ;
0C76    2A FC9E                 LD      HL,(JIFFY)
0C79    23                      INC     HL
0C7A    22 FC9E                 LD      (JIFFY),HL
                        ;
                        ; Check music queue
                        ;
0C7D    3A FB3F                 LD      A,(MUSICF)      ;Check music flag
0C80    4F                      LD      C,A
0C81    AF                      XOR     A               ;Start with queue 0
0C82                    MUSINT:
0C82    CB 19                   RR      C               ;C7=carry, carry=C0, [C]=[C]/2
0C84    F5                      PUSH    AF              ;Save queue ID
0C85    C5                      PUSH    BC              ;Save MUSICF
```

```
0C86    DC 113B                 CALL    C,ACTION
0C89    C1                      POP     BC
0C8A    F1                      POP     AF
0C8B    3C                      INC     A               ;Next queue
0C8C    FE 03                   CP      3               ;All done?
0C8E    38 F2                   JR      C,MUSINT        ;Not yet
0C90    21 F3F6                 LD      HL,SCNCNT
0C93    35                      DEC     (HL)            ;Need to scan?
0C94    20 6C                   JR      NZ,INTRET       ;No, return soon
0C96    36 03                   LD      (HL),3          ;Time delay of first repeat
                        ;
                        ; Check trigger button of joy sticks
                        ;
0C98    AF                      XOR     A
0C99    CD 120C                 CALL    SLSTCK          ;Read joystick A
0C9C    E6 30                   AND     00110000B
0C9E    F5                      PUSH    AF
0C9F    3E 01                   LD      A,1
0CA1    CD 120C                 CALL    SLSTCK
0CA4    E6 30                   AND     '0'
0CA6    07                      RLCA
0CA7    07                      RLCA
0CA8    C1                      POP     BC
0CA9    B0                      OR      B
0CAA    F5                      PUSH    AF
0CAB    CD 1226                 CALL    GTROW8
0CAE    E6 01                   AND     1
0CB0    C1                      POP     BC
0CB1    B0                      OR      B
0CB2    4F                      LD      C,A             ;Save this
0CB3    21 F3E8                 LD      HL,TRGFLG
```

```
0CB6    AE                      XOR     (HL)            ;Any transition?
0CB7    A6                      AND     (HL)            ;Is this transition negative
0CB8    71                      LD      (HL),C          ;Update trigger status
0CB9    4F                      LD      C,A
0CBA    0F                      RRCA                    ;Check space key trigger
0CBB    21 FC70                 LD      HL,TRPTBL+3*12
0CBE    DC 0EF1                 CALL    C,REQTRP
0CC1    CB 11                   RL      C               ;Check trigger 4
0CC3    21 FC7C                 LD      HL,TRPTBL+3*16
0CC6    DC 0EF1                 CALL    C,REQTRP
0CC9    CB 11                   RL      C               ;Check trigger 2
0CCB    21 FC76                 LD      HL,TRPTBL+3*14
0CCE    DC 0EF1                 CALL    C,REQTRP
0CD1    CB 11                   RL      C               ;Check trigger 3
0CD3    21 FC79                 LD      HL,TRPTBL+3*15
0CD6    DC 0EF1                 CALL    C,REQTRP
0CD9    CB 11                   RL      C               ;Check trigger 1
0CDB    21 FC73                 LD      HL,TRPTBL+3*13
0CDE    DC 0EF1                 CALL    C,REQTRP
                        ;
                        ; Scan keyboard
                        ;
0CE1    AF                      XOR     A               ;Enable first key click
0CE2    32 FBD9                 LD      (CLIKFL),A
0CE5    CD 0D12                 CALL    KEYCHK          ;Detect valid key transition and check buffer
0CE8    20 18                   JR      NZ,INTRET       ;Some characters still remain, don't repeat
0CEA    21 F3F7                 LD      HL,REPCNT
0CED    35                      DEC     (HL)            ;Need to enter repeat mode
0CEE    20 12                   JR      NZ,INTRET       ;No
0CF0    36 01                   LD      (HL),1          ;Set short time repeat
0CF2    21 FBDA                 LD      HL,OLDKEY       ;Clear OLDKEY status
```

```
0CF5   11 FBDB                LD      DE,OLDKEY+1
0CF8   01 000A                LD      BC,0AH
0CFB   36 FF                  LD      (HL),0FFH
0CFD   ED B0                  LDIR
0CFF   CD 0D4E                CALL    KEYCK4          ;Check if currently pressed key is valid
0D02                  INTRET:
0D02   DD E1                  POP     IX              ;Restore all registers
0D04   FD E1                  POP     IY
0D06   F1                     POP     AF
0D07   C1                     POP     BC
0D08   D1                     POP     DE
0D09   E1                     POP     HL
0D0A   08                     EX      AF,AF'
0D0B   D9                     EXX
0D0C   F1                     POP     AF
0D0D   C1                     POP     BC
0D0E   D1                     POP     DE
0D0F   E1                     POP     HL
0D10   FB                     EI
0D11   C9                     RET
0D12                  KEYCHK:
                      ;
0D12   DB AA                  IN      A,(PPI CR)      ;Get what is currently output to Port C
0D14   E6 F0                  AND     0F0H            ;Leave higher 4 bits unaffected
0D16   4F                     LD      C,A
0D17   06 0B                  LD      B,0BH
0D19   21 FBE5                LD      HL,NEWKEY       ;Move current key status to NEWKEY
0D1C                  KEYCK1:
0D1C   79                     LD      A,C
0D1D   D3 AA                  OUT     (PPI.CW),A      ;Select row
0D1F   DB A9                  IN      A,(PPI.BR)      ;Get column information of selected row
```

```
0D21    77                      LD      (HL),A          ;Move it
0D22    0C                      INC     C               ;Select next row
0D23    23                      INC     HL
0D24    10 F6                   DJNZ    KEYCK1          ;Loop until all rows are sensed
0D26    3A FBB0                 LD      A,(ENSTOP)      ;Warm start enabled?
0D29    A7                      AND     A
0D2A    28 0E                   JR      Z,NOSTOP        ;No
0D2C    3A FBEB                 LD      A,(SFTKEY)      ;Get current status of the 6th row
0D2F    FE E8                   CP      0E8H            ;Check if KANA, GRAPH, CTRL and SHIFT
0D31    20 07                   JR      NZ,NOSTOP       ;are pressed simultaneously
0D33    DD 21 409B              LD      IX,READYR
0D37    C3 01FF                 JP      CALBAS
0D3A                    NOSTOP:
                        ;
0D3A    11 FBE5                 LD      DE,NEWKEY       ;[OLDKEY] + 11
0D3D    06 0B                   LD      B,0BH
0D3F                    KEYCK2:
0D3F    1B                      DEC     DE
0D40    2B                      DEC     HL
0D41    1A                      LD      A,(DE)          ;Get OLDKEY status
0D42    BE                      CP      (HL)            ;Compare with NEWKEY status
0D43    20 04                   JR      NZ,KEYCK3       ;Changed, set long repeat interval
0D45    10 F8                   DJNZ    KEYCK2
0D47    18 05                   JR      KEYCK4          ;No change
0D49                    KEYCK3:
                        ;
0D49    3E 0D                   LD      A,0DH
0D4B    32 F3F7                 LD      (REPCNT),A
0D4E                    KEYCK4:
0D4E    06 0B                   LD      B,0BH           ;Set number of rows
0D50    21 FBDA                 LD      HL,OLDKEY
```

```
0D53    11 FBE5                 LD      DE,NEWKEY
0D56                    KEYCK5:
0D56    1A                      LD      A,(DE)          ;Get current key status
0D57    4F                      LD      C,A
0D58    AE                      XOR     (HL)            ;See if any bit changed
0D59    A6                      AND     (HL)            ;See if this change is negative transition
0D5A    71                      LD      (HL),C          ;Update old status
0D5B    C4 0D89                 CALL    NZ,KEYANY       ;Active transition, go find it
0D5E    13                      INC     DE
0D5F    23                      INC     HL
0D60    10 F4                   DJNZ    KEYCK5
0D62                    CHKBUF:
                        ;
                        ; Check if buffer is empty or not
                        ;
0D62    2A F3FA                 LD      HL,(GETPNT)     ;Load GETPNT
0D65    3A F3F8                 LD      A,(PUTPNT)      ;Load lower 8 bit of PUTPNT
0D68    95                      SUB     L               ;Check if same
0D69    C9                      RET
0D6A                    CHSNS:
                        ;
0D6A    FB                      EI                      ;Make sure interrupts are enabled
0D6B    E5                      PUSH    HL              ;Save environments
0D6C    D5                      PUSH    DE
0D6D    C5                      PUSH    BC
0D6E    CD 0B9F                 CALL    CHKSCR          ;Are we in text mode?
0D71    30 0F                   JR      NC,CHSNS1       ;No, do not flip function keys
0D73    3A FBCD                 LD      A,(FNKSWI)      ;Get current shift status
0D76    21 FBEB                 LD      HL,SFTKEY       ;Get current function key display
0D79    AE                      XOR     (HL)            ;Are they different
0D7A    21 F3DE                 LD      HL,CNSDFG       ;Function key displayed at all?
```

```
0D7D     A6                   AND     (HL)
0D7E     0F                   RRCA
0D7F     DC 0B2B              CALL    C,DSPFNK        ;Update display
0D82                  CHSNS1:
0D82     CD 0D62              CALL    CHKBUF
0D85     C1                   POP     BC              ;Restore environments
0D86     D1                   POP     DE
0D87     E1                   POP     HL
0D88     C9                   RET
0D89                  KEYANY:
                      ;
                      ; [[[ SUBROUTINE 'KEYANY' ]]]
                      ;
0D89     E5                   PUSH    HL              ;Save environments
0D8A     D5                   PUSH    DE
0D8B     C5                   PUSH    BC
0D8C     F5                   PUSH    AF              ;Save pressed bit
0D8D     3E 0B                LD      A,0BH
0D8F     90                   SUB     B               ;Calculate base code
0D90     87                   ADD     A,A
0D91     87                   ADD     A,A
0D92     87                   ADD     A,A
0D93     4F                   LD      C,A
0D94     06 08                LD      B,8             ;Set up counter for 8 bit
0D96     F1                   POP     AF              ;Restore pressed bit
0D97                  KYANY1:
0D97     1F                   RRA
0D98     C5                   PUSH    BC
0D99     F5                   PUSH    AF
0D9A     DC 0E3B              CALL    C,KEYCOD        ;If pressed bit, call key coder.
0D9D     F1                   POP     AF
```

```
0D9E    C1                      POP     BC
0D9F    0C                      INC     C               ;Try next code
0DA0    10 F5                   DJNZ    KYANY1          ;Loop until all bits are checked
0DA2    C3 08DB                 JP      PBDHRT          ;Restore environments
                        ;
                        ;          [[[ SUBROUTINE 'KEYCOD' ]]
                        ;
                        ;          Return key-code in buffer if valid
                        ;
0DA5                    KYJTAB:
0DA5    0A                      DB      10
0DA6    0E67                    DW      KYNUM           ;0..9
0DA8    16                      DB      22
0DA9    0EA1                    DW      KYCOD1
0DAB    30                      DB      48
0DAC    0E7E                    DW      KYALP           ;A..Z
0DAE    33                      DB      51
0DAF    0F10                    DW      KYEASY
0DB1    34                      DB      52
0DB2    0F36                    DW      KYLOCK          ;Capital lock
0DB4    35                      DB      53
0DB5    0F1F                    DW      KYKLOK          ;Kana lock
0DB7    3A                      DB      58
0DB8    0EBB                    DW      KYFUNC          ;Function key
0DBA    3C                      DB      60
0DBB    0F10                    DW      KYEASY
0DBD    3D                      DB      61
0DBE    0F46                    DW      KYSTOP          ;Stop key
0DC0    41                      DB      65
0DC1    0F10                    DW      KYEASY
0DC3    42                      DB      66
```

```
0DC4    0F06                    DW      KYCLS           ;CLS/HOME key
0DC6    FF                      DB      255
0DC7    0F10                    DW      KYEASY
                        ;
0DC9                    NMSFTB:
0DC9    FF                      DB      255
0DCA    21                      DB      "!"
0DCB    22                      DB      34              ;Double quote
0DCC    23 24 25 26             DB      "#$%&'()"
0DD0    27 28 29
                        ;
0DD3                    ALPJMP:
0DD3    0F55                    DW      PUTCHR          ;CTRL+shift
0DD5    0F55                    DW      PUTCHR          ;CTRL
0DD7    0E93                    DW      KEYSFT          ;     SHIFT
0DD9    0E95                    DW      KEYNOM          ;
                        ;
0DDB                    KYC1TB:
0DDB    0DFD                    DW      KY1SFC-10       ;CTRL+SHIFT
0DDD    0DF1                    DW      KY1CNT-10       ;CTRL
0DDF    0DE5                    DW      KY1SFT-10       ;     SHIFT
0DE1    0DD9                    DW      KY1NOM-10       ;
0DE3                    KY1NOM:
0DE3    2D 5E 5C 40             DB      "_^\@[;:],./"
0DE7    5B 3B 3A 5D
0DEB    2C 2E 2F
0DEE    FF                      DB      255
0DEF                    KY1SFT:
0DEF    3D 7E 7C 60             DB      "=~|`{+*}"
0DF3    7B 2B 2A 7D
0DF7    3C                      DB      00111100B       ;Less than sign
```

```
0DF8    3E                      DB      00111110B       ;Greater than sign
0DF9    3F 5F                   DB      "?_"
0DFB                    KY1CNT:
0DFB    2D                      DB      "-"
0DFC    1E                      DB      "^"-"@"
0DFD    1C                      DB      "\"-"@"
0DFE    00                      DB      "@"-"@"
0DFF    1B                      DB      "["-"@"
0E00    3B 3A                   DB      ";:"
0E02    1D                      DB      "]"-"@"
0E03    2C 2E 2F                DB      ",./"
0E06    FF                      DB      255
0E07                    KY1SFC:
0E07    3D                      DB      "="
0E08    1E                      DB      "^"-"@"
0E09    1C                      DB      "\"-"@"
0E0A    00                      DB      "@"-"@"
0E0B    1B                      DB      "["-"@"
0E0C    2B 2A                   DB      "+*"
0E0E    1D                      DB      "]"-"@"
0E0F    3C                      DB      00111100B       ;Less than sign
0E10    3E                      DB      00111110B       ;Greater than sign
0E11    3F                      DB      "?"
0E12    1F                      DB      "_"-"@"
                        ;
0E13                    EASYTB:
0E13    00                      DB      0               ;Shift          (48)
0E14    00                      DB      0               ;Control        (49)
0E15    00                      DB      0               ;Graph          (50)
0E16    00                      DB      0               ;Cap lock       (51)
0E17    00                      DB      0               ;Kana lock      (52)
```

```
0E18    00                  DB      0               ;F1             (53)
0E19    00                  DB      0               ;F2             (54)
0E1A    00                  DB      0               ;F3             (55)
0E1B    00                  DB      0               ;F4             (56)
0E1C    00                  DB      0               ;F5             (57)
0E1D    1B                  DB      27              ;Escape         (58)
0E1E    09                  DB      9               ;Tab            (59)
0E1F    00                  DB      0               ;Stop           (60)
0E20    08                  DB      8               ;Back space     (61)
0E21    18                  DB      "X"-"@"         ;Select         (62)
0E22    0D                  DB      13              ;Enter          (63)
0E23    20                  DB      32              ;Space          (64)
0E24    0C                  DB      12              ;Clear          (65)
0E25    12                  DB      "R"-"@"         ;Insert         (66)
0E26    7F                  DB      127             ;Rubout         (67)
0E27    1D                  DB      29              ;Left           (68)
0E28    1E                  DB      30              ;Up             (69)
0E29    1F                  DB      31              ;Down           (70)
0E2A    1C                  DB      28              ;Right          (71)
                      ;
                      ;        For additional key matrix
                      ;
0E2B    01                  DB      "A"-"@"         ;               (72)
0E2C    04                  DB      "D"-"@"         ;               (73)
0E2D    0F                  DB      "O"-"@"         ;               (74)
0E2E    10                  DB      "P"-"@"         ;               (75)
0E2F    11                  DB      "Q"-"@"         ;               (76)
0E30    12                  DB      "R"-"@"         ;               (77)
0E31    13                  DB      "S"-"@"         ;               (78)
0E32    14                  DB      "T"-"@"         ;               (79)
0E33    00                  DB      0               ;               (80)
```

```
 2974      0E34     00                       DB      0                 ;                   (81)
 2975      0E35     00                       DB      0                 ;                   (82)
 2976      0E36     00                       DB      0                 ;                   (83)
 2977      0E37     00                       DB      0                 ;                   (84)
 2978      0E38     00                       DB      0                 ;                   (85)
 2979      0E39     00                       DB      0                 ;                   (86)
 2980      0E3A     00                       DB      0                 ;                   (87)
```

```

                              ;
0E3B                          KEYCOD:
                              ;
                              ; [[[ SUBROUTINE 'KEYCOD' ]]]
                              ;
                              ; Return key-code in buffer if valid
                              ;
0E3B    79                            LD      A,C             ;Get raw code
0E3C    FE FF                         CP      0FFH            ;Just for fail safe
0E3E    C8                            RET     Z
0E3F    21 0DA5                       LD      HL,KYJTAB
0E42    CD FDCC                       CALL    H.KEYC
0E45    FE 30                         CP      48              ;Possibly a KANA or graphic character
0E47    30 13                         JR      NC,KYCLAS       ;No
0E49    3A FBEB                       LD      A,(SFTKEY)      ;Get shift key status
0E4C    0F                            RRCA                    ;Control pressed?
0E4D    0F                            RRCA
0E4E    30 0B                         JR      NC,KYCLA0       ;Yes, this supersedes everything
0E50    0F                            RRCA                    ;How about graphic shift
0E51    D2 107D                       JP      NC,KYGRAP       ;Yes, this has the 2nd priority
0E54    3A FCAC                       LD      A,(KANAST)      ;KANA lock active
0E57    A7                            AND     A
0E58    C2 0F83                       JP      NZ,KYKANA       ;Yes
0E5B                          KYCLA0:
0E5B    79                            LD      A,C
0E5C                          KYCLAS:
0E5C    BE                            CP      (HL)            ;Compare range
0E5D    23                            INC     HL
0E5E    5E                            LD      E,(HL)          ;Get jump address in [DE]
0E5F    23                            INC     HL
```

```
0E60   56                      LD      D,(HL)
0E61   23                      INC     HL
0E62   D5                      PUSH    DE              ;Assume matched
0E63   D8                      RET     C               ;Good assumption
0E64   D1                      POP     DE              ;Discard stack
0E65   18 F5                   JR      KYCLAS          ;Check next possibility
0E67                   KYNUM:
                       ;
0E67   C6 30                   ADD     A,'0'           ;Assume no shift
0E69   47                      LD      B,A             ;Save code
0E6A   3A FBEB                 LD      A,(SFTKEY)      ;Check shift status
0E6D   0F                      RRCA
0E6E   78                      LD      A,B             ;Restore code
0E6F   38 0A                   JR      C,JPUTCH        ;Good assumption
0E71   06 00                   LD      B,0
0E73   21 0DC9                 LD      HL,NMSFTB
0E76   09                      ADD     HL,BC           ;This must not be 'DADF'
0E77   7E                      LD      A,(HL)          ;Get code for shift-number
0E78   FE FF                   CP      0FFH            ;Shift '0'?
0E7A   C8                      RET     Z               ;Yes, ignore this
0E7B                   JPUTCH:
0E7B   C3 0F55                 JP      PUTCHR          ;Put this in buffer
0E7E                   KYALP:
                       ;
0E7E   3A FBEB                 LD      A,(SFTKEY)
0E81   E6 03                   AND     3
0E83   87                      ADD     A,A
0E84   5F                      LD      E,A
0E85   16 00                   LD      D,0
0E87   21 0DD3                 LD      HL,ALPJMP
0E8A   19                      ADD     HL,DE
```

```
0E8B   7E                  LD      A,(HL)          ;Get jump address
0E8C   23                  INC     HL
0E8D   66                  LD      H,(HL)
0E8E   6F                  LD      L,A
0E8F   79                  LD      A,C             ;Get code
0E90   D6 15               SUB     15H             ;Make it a control character (1 - 26)
0E92   E9                  JP      (HL)
0E93             KEYSFT:
                 ;
0E93   C6 20               ADD     A,' '
0E95             KEYNOM:
0E95   47                  LD      B,A             ;Save code
0E96   3A FCAB             LD      A,(CAPST)
0E99   2F                  CPL
0E9A   E6 20               AND     00100000B       ;Bit 5 is on if CAP lock not active
0E9C   A8                  XOR     B
0E9D   C6 40               ADD     A,01000000B
0E9F   18 DA               JR      JPUTCH
0EA1             KYCOD1:
                 ;
0EA1   21 0DDB             LD      HL,KYC1TB
0EA4   3A FBEB             LD      A,(SFTKEY)
0EA7   E6 03               AND     3               ;Extract shift and control status
0EA9   87                  ADD     A,A
0EAA   5F                  LD      E,A
0EAB   16 00               LD      D,0
0EAD   19                  ADD     HL,DE
0EAE   7E                  LD      A,(HL)
0EAF   23                  INC     HL
0EB0   66                  LD      H,(HL)
0EB1   6F                  LD      L,A
```

```
0EB2    59                      LD      E,C
0EB3    19                      ADD     HL,DE
0EB4    7E                      LD      A,(HL)
0EB5    FE FF                   CP      0FFH            ;Should generate some code?
0EB7    C2 0F55                 JP      NZ,PUTCHR       ;Yes
0EBA    C9                      RET                     ;No code should be generated
0EBB                    KYFUNC:
                        ;
                        ; Function keys
                        ;
0EBB    3A FBEB                 LD      A,(SFTKEY)      ;Is shift pressed?
0EBE    0F                      RRCA
0EBF    38 04                   JR      C,KYFNC1        ;No
0EC1    79                      LD      A,C
0EC2    C6 05                   ADD     A,5
0EC4    4F                      LD      C,A
0EC5                    KYFNC1:
0EC5    59                      LD      E,C             ;[DE] is (56..65)
0EC6    16 00                   LD      D,0
0EC8    21 FB99                 LD      HL,FNKFLG-53    ;Check if this function key is an event device
0ECB    19                      ADD     HL,DE
0ECC    7E                      LD      A,(HL)
0ECD    A7                      AND     A
0ECE    20 13                   JR      NZ,FNKINT       ;Request trap if not in direct mode
0ED0                    KYFNC2:
0ED0    EB                      EX      DE,HL
0ED1    29                      ADD     HL,HL
0ED2    29                      ADD     HL,HL
0ED3    29                      ADD     HL,HL
0ED4    29                      ADD     HL,HL
0ED5    11 F52F                 LD      DE,FNKSTR-53*16
```

```
0ED8    19                ADD     HL,DE           ;Get function key string address
0ED9    EB                EX      DE,HL           ;Move address to DE
0EDA              KYFNC3:
0EDA    1A                LD      A,(DE)          ;Get from function key string
0EDB    A7                AND     A               ;End of string
0EDC    C8                RET     Z               ;Yes
0EDD    CD 0F55           CALL    PUTCHR          ;Put this character in buffer
0EE0    13                INC     DE              ;Check next character
0EE1    18 F7             JR      KYFNC3
0EE3              FNKINT:
                  ;
0EE3    2A F41C           LD      HL,(CURLIN)     ;Are we in direct mode (CURLIN=65535)
0EE6    23                INC     HL
0EE7    7C                LD      A,H
0EE8    B5                OR      L
0EE9    28 E5             JR      Z,KYFNC2        ;Yes, treat as normal function key
0EEB    21 FBAD           LD      HL,TRPTBL-53*3
0EEE    19                ADD     HL,DE
0EEF    19                ADD     HL,DE
0EF0    19                ADD     HL,DE
```

```

                                  ;
0EF1                              REQTRP:
                                  ;
                                  ; Request trap (called to request trap for event devices)
                                  ;
                                  ;
                                  ; Since REQTRP is mostly called from within an interrupt routine,
                                  ; don't touch the interrupt mask through DI or EI.
                                  ;
0EF1    7E                                LD      A,(HL)
0EF2    E6 01                             AND     1               ;Trap on?
0EF4    C8                                RET     Z               ;TRAP NOT ON
0EF5    7E                                LD      A,(HL)
0EF6    F6 04                             OR      4               ;Trap request
0EF8    BE                                CP      (HL)
0EF9    C8                                RET     Z               ;No change
0EFA    77                                LD      (HL),A
0EFB    EE 05                             XOR     5               ;Trap on + Trap request
0EFD    C0                                RET     NZ
0EFE    3A FBD8                           LD      A,(ONGSBF)
0F01    3C                                INC     A
0F02    32 FBD8                           LD      (ONGSBF),A
0F05    C9                                RET
                                  ;
0F06                              KYCLS:
0F06    3A FBEB                           LD      A,(SFTKEY)      ;Set carry if shift not pressed
0F09    0F                                RRCA
0F0A    3E 0C                             LD      A,0CH           ;Load code for CLS
0F0C    DE 00                             SBC     A,0             ;Change to HOME if shift not pressed
0F0E    18 45                             JR      PUTCHR
```

```
0F10                   KYEASY:
                       ;
                       ; Easily converted keys
                       ;
0F10    CD FDD1                CALL    H.KYEA          ;For CCP (Cut, copy, paste) editor rom
0F13    5F                     LD      E,A             ;These character are simply taken from table
0F14    16 00                  LD      D,0
0F16    21 0DE3                LD      HL,EASYTB-48
0F19    19                     ADD     HL,DE
0F1A    7E                     LD      A,(HL)
0F1B    A7                     AND     A               ;Should this key generate some code
0F1C    C8                     RET     Z               ;No
0F1D    18 36                  JR      PUTCHR          ;Yes
0F1F                   KYKLOK:
                       ;
                       ; Kana lock key
                       ;
0F1F    21 FCAC                LD      HL,KANAST
0F22    7E                     LD      A,(HL)
0F23    2F                     CPL
0F24    77                     LD      (HL),A
0F25    3E 0F                  LD      A,0FH
0F27    D3 A0                  OUT     (PSG.LW),A
0F29    DB A2                  IN      A,(PSG.DR)
0F2B    E6 7F                  AND     7FH
0F2D    47                     LD      B,A
0F2E    7E                     LD      A,(HL)
0F2F    2F                     CPL
0F30    E6 80                  AND     80H
0F32    B0                     OR      B
0F33    D3 A1                  OUT     (PSG.DW),A
```

```
0F35                    NOKEY:
0F35    C9                      RET
0F36                    KYLOCK:
                        ;
                        ; Capital lock key
                        ;
0F36    21 FCAB                 LD      HL,CAPST
0F39    7E                      LD      A,(HL)          ;Toggle capital status
0F3A    2F                      CPL
0F3B    77                      LD      (HL),A          ;Update capital status
0F3C    2F                      CPL
0F3D                    CHGCAP:
0F3D    A7                      AND     A
0F3E    3E 0C                   LD      A,0CH           ;Assume 'turn off'
0F40    28 01                   JR      Z,CGCAP1        ;Good assumption
0F42    3C                      INC     A               ;Change to 'turn on'
0F43                    CGCAP1:
0F43    D3 AB                   OUT     (PPI.CM),A
0F45    C9                      RET
0F46                    KYSTOP:
                        ;
                        ; STOP key
                        ;
0F46    3A FBEB                 LD      A,(SFTKEY)
0F49    0F                      RRCA                    ;Move CTRL status to carry
0F4A    0F                      RRCA
0F4B    3E 03                   LD      A,3             ;Assume CTRL pressed also
0F4D    30 01                   JR      NC,KYSTP1       ;Good assumption
0F4F    3C                      INC     A               ;CTRL not pressed, just treat as pause
0F50                    KYSTP1:
0F50    32 FC9B                 LD      (INTFLG),A
```

```
0F53    38 0F                   JR      C,GENCLK        ;Only generate click if pause
0F55                    PUTCHR:
                        ;
                        ; Put one character in key buffer.
                        ;
0F55    2A F3F8                 LD      HL,(PUTPNT)     ;Load PUTPNT in [HL]
0F58    77                      LD      (HL),A          ;Save the character to buffer
0F59    CD 10C2                 CALL    UPDATE          ;Increment PUTPNT
0F5C    3A F3FA                 LD      A,(GETPNT)      ;Load lower 8bit of GETPNT
0F5F    BD                      CP      L               ;Compare it with new PUTPNT
0F60    C8                      RET     Z               ;If same skip next step
0F61    22 F3F8                 LD      (PUTPNT),HL     ;Save HL in PUTPNT
0F64                    GENCLK:
0F64    3A F3DB                 LD      A,(CLIKSW)      ;Key click enabled?
0F67    A7                      AND     A
0F68    C8                      RET     Z               ;No
0F69    3A FBD9                 LD      A,(CLIKFL)      ;Already generated?
0F6C    A7                      AND     A
0F6D    C0                      RET     NZ              ;Yes, don't click any more
0F6E    3E 0F                   LD      A,0FH
0F70    32 FBD9                 LD      (CLIKFL),A      ;Set flag to disable more clicks
0F73    D3 AB                   OUT     (PPI.CM),A
0F75    3E 0A                   LD      A,0AH
0F77                    CLICKW:
0F77    3D                      DEC     A
0F78    20 FD                   JR      NZ,CLICKW
0F7A                    CHGSND:
0F7A    A7                      AND     A
0F7B    3E 0E                   LD      A,0EH           ;Assume 'turn off'
0F7D    28 01                   JR      Z,CGSND1        ;Good assumption
0F7F    3C                      INC     A               ;Change to 'turn on'
```

```
0F80                    CGSND1:
0F80    D3 AB                   OUT     (PPI.CM),A
0F82    C9                      RET
0F83                    KYKANA:
                        ;
                        ; KANA key pressed while KANA lock is active
                        ;
0F83    3A FCAD                 LD      A,(KANAMD)      ;JIS or AIUEO?
0F86    A7                      AND     A               ;Affect Z flag
0F87    3A FBEB                 LD      A,(SFTKEY)      ;Check shift key
0F8A    0F                      RRCA                    ;Affect Carry flag
0F8B    28 0A                   JR      Z,KAIUEO        ;AIUEO order
0F8D    21 101D                 LD      HL,KANJNO
0F90    38 0D                   JR      C,KYKAN1
0F92    21 104D                 LD      HL,KANJSF
0F95    18 08                   JR      KYKAN1
0F97                    KAIUEO:
                        ;
0F97    21 0FBD                 LD      HL,KANANO       ;Assume shift not pressed
0F9A    38 03                   JR      C,KYKAN1        ;Good assumption
0F9C    21 0FED                 LD      HL,KANASF
0F9F                    KYKAN1:
0F9F    06 00                   LD      B,0
0FA1    09                      ADD     HL,BC
0FA2    01 0F55                 LD      BC,PUTCHR       ;Push jump address
0FA5    C5                      PUSH    BC
0FA6    3A FCAB                 LD      A,(CAPST)       ;Capital lock (katakana) active?
0FA9    A7                      AND     A
0FAA    7E                      LD      A,(HL)
0FAB    C0                      RET     NZ              ;active
0FAC    FE A6                   CP      165+1           ;Special characters?
```

```
0FAE    D8                          RET     C               ;Yes, no conversion necessary
0FAF    FE B0                       CP      0B0H
0FB1    C8                          RET     Z
0FB2    FE DE                       CP      0DEH
0FB4    D0                          RET     NC
0FB5    D6 20                       SUB     ' '             ;Assume first half
0FB7    FE A0                       CP      191-32+1        ;Really first half
0FB9    D8                          RET     C               ;Good assumption
0FBA    C6 40                       ADD     A,32+32         ;Compensate
0FBC    C9                          RET
0FBD                        KANANO:
                            ;       Kana table (AIUEO order, un-shifted
                            ;
0FBD    C9 B1 B2 B3                 DB      0C9H,0B1H,0B2H,0B3H,0B4H,0B5H,0C5H
0FC1    B4 B5 C5
0FC4    C6 C7 C8 D7                 DB      0C6H,0C7H,0C8H,0D7H,0D8H,0D9H,0DAH
0FC8    D8 D9 DA
0FCB    DB D3 DE DF                 DB      0DBH,0D3H,0DEH,0DFH,0D6H,0DCH,0A6H
0FCF    D6 DC A6
0FD2    DD BB C4 C2                 DB      0DDH,0BBH,0C4H,0C2H,0BDH,0B8H,0BEH
0FD6    BD B8 BE
0FD9    BF CF CC D0                 DB      0BFH,0CFH,0CCH,0D0H,0D1H,0D2H,0D5H
0FDD    D1 D2 D5
0FE0    D4 CD CE B6                 DB      0D4H,0CDH,0CEH,0B6H,0B9H,0BCH,0BAH
0FE4    B9 BC BA
0FE7    CB C3 B7 C1                 DB      0CBH,0C3H,0B7H,0C1H,0CAH,0C0H
0FEB    CA C0
0FED                        KANASF:
                            ;       Shifted
                            ;
0FED    C9 A7 A8 A9                 DB      0C9H,0A7H,0A8H,0A9H,0AAH,0ABH,0C5H
```

```
0FF1    AA AB C5
0FF4    C6 C7 C8 D7             DB      0C6H,0C7H,0C8H,0D7H,0D8H,0D9H,0DAH
0FF8    D8 D9 DA
0FFB    A2 D3 B0 A3             DB      0A2H,0D3H,0B0H,0A3H,0AEH,0A4H,0A1H
0FFF    AE A4 A1
1002    A5 BB C4 AF             DB      0A5H,0BBH,0C4H,0AFH,0BDH,0B8H,0BEH
1006    BD B8 BE
1009    BF CF CC D0             DB      0BFH,0CFH,0CCH,0D0H,0D1H,0D2H,0ADH
100D    D1 D2 AD
1010    AC CD CE B6             DB      0ACH,0CDH,0CEH,0B6H,0B9H,0BCH,0BAH
1014    B9 BC BA
1017    CB C3 B7 C1             DB      0CBH,0C3H,0B7H,0C1H,0CAH,0C0H
101B    CA C0
101D                    KANJNO:
                        ;       Kana table JIS order, un-shifted
                        ;
101D    DC C7 CC B1             DB      0DCH,0C7H,0CCH,0B1H,0B3H,0B4H,0B5H
1021    B3 B4 B5
1024    D4 D5 D6 CE             DB      0D4H,0D5H,0D6H,0CEH,0CDH,0B0H,0DEH
1028    CD B0 DE
102B    DF DA B9 D1             DB      0DFH,0DAH,0B9H,0D1H,0C8H,0D9H,0D2H
102F    C8 D9 D2
1032    DB C1 BA BF             DB      0DBH,0C1H,0BAH,0BFH,0BCH,0B2H,0CAH
1036    BC B2 CA
1039    B7 B8 C6 CF             DB      0B7H,0B8H,0C6H,0CFH,0C9H,0D8H,0D3H
103D    C9 D8 D3
1040    D0 D7 BE C0             DB      0D0H,0D7H,0BEH,0C0H,0BDH,0C4H,0B6H
1044    BD C4 B6
1047    C5 CB C3 BB             DB      0C5H,0CBH,0C3H,0BBH,0DDH,0C2H
104B    DD C2
104D                    KANJSF:
```

```
                        ;       Shifted
                        ;
1049    A6 C7 CC A7             DB      0A6H,0C7H,0CCH,0A7H,0A9H,0AAH,0ABH
1051    A9 AA AB
1054    AC AD AE CE             DB      0ACH,0ADH,0AEH,0CEH,0CDH,0B0H,0DEH
1058    CD B0 DE
105B    A2 DA B9 A3             DB      0A2H,0DAH,0B9H,0A3H,0A4H,0A1H,0A5H
105F    A4 A1 A5
1062    DB C1 BA BF             DB      0DBH,0C1H,0BAH,0BFH,0BCH,0A8H,0CAH
1066    BC A8 CA
1069    B7 B8 C6 CF             DB      0B7H,0B8H,0C6H,0CFH,0C9H,0D8H,0D3H
106D    C9 D8 D3
1070    D0 D7 BE C0             DB      0D0H,0D7H,0BEH,0C0H,0BDH,0C4H,0B6H
1074    BD C4 B6
1077    C5 CB C3 BB             DB      0C5H,0CBH,0C3H,0BBH,0DDH,0AFH
107B    DD AF
```

```
         107D                   ;
                                KYGRAP:
                                ;
                                ; Graphic characters
                                ;
         107D    06 00                  LD      B,0
         107F    21 1092                LD      HL,GRPTAB
         1082    09                     ADD     HL,BC
         1083    7E                     LD      A,(HL)          ;Get from graphic key table
         1084    A7                     AND     A               ;Should generate some code
         1085    C8                     RET     Z               ;No
         1086    FE 80                  CP      80H             ;1 byte code?
         1088    F5                     PUSH    AF
         1089    3E 01                  LD      A,1             ;Assume not
         108B    DC 0F55                CALL    C,PUTCHR        ;Was 2 byte code, put header byte
         108E    F1                     POP     AF
         108F    C3 0F55                JP      PUTCHR
                                ;
         1092                   GRPTAB:
         1092    4F 47 41 42            DB      4FH,47H,41H,42H,43H,44H,45H
         1096    43 44 45
         1099    46 4D 4E 57            DB      46H,4DH,4EH,57H,00H,49H,00H
         109D    00 49 00
         10A0    84 82 81 85            DB      84H,82H,81H,85H,5FH,5DH,80H
         10A4    5F 5D 80
         10A7    83 00 5B 5A            DB      83H,00H,5BH,5AH,54H,58H,55H
         10AB    54 58 55
         10AE    53 4A 56 00            DB      53H,4AH,56H,00H,00H,5EH,4BH
         10B2    00 5E 4B
         10B5    00 00 50 00            DB      00H,00H,50H,00H,52H,4CH,59H
```

```
10B9    52 4C 59
10BC    00 51 00 5C             DB      00H,51H,00H,5CH,48H,00H
10C0    48 00
                        ;
10C2                    UPDATE:
                        ;
                        ; Update pointer
                        ;
10C2    23                      INC     HL
10C3    7D                      LD      A,L
10C4    FE 18                   CP      18H             ;Check buffer boundary
10C6    C0                      RET     NZ
10C7    21 FBF0                 LD      HL,KEYBUF
10CA    C9                      RET
10CB                    CHGET:
                        ;
                        ; Get one character from keyboard
                        ;
10CB    E5                      PUSH    HL
10CC    D5                      PUSH    DE
10CD    C5                      PUSH    BC
10CE    CD FDC2                 CALL    H.CHGE
10D1    CD 0D6A                 CALL    CHSNS           ;Character already there?
10D4    20 0B                   JR      NZ,CHGET2       ;Yes, do not touch cursor
10D6    CD 09DA                 CALL    CKDPC0          ;Display cursor if disabled
10D9                    CHGET1:
10D9    CD 0D6A                 CALL    CHSNS           ;Any character in buffer?
10DC    28 FB                   JR      Z,CHGET1        ;No, wait
10DE    CD 0A27                 CALL    CKERC0          ;Erase cursor if disabled
10E1                    CHGET2:
10E1    21 FC9B                 LD      HL,INTFLG
```

```
10E4    7E                      LD      A,(HL)
10E5    FE 04                   CP      4               ;Code for pause?
10E7    20 02                   JR      NZ,CHGET3       ;No
10E9    36 00                   LD      (HL),0          ;Clear this
10EB                    CHGET3:
10EB    2A F3FA                 LD      HL,(GETPNT)
10EE    4E                      LD      C,(HL)          ;Save pressed key
10EF    CD 10C2                 CALL    UPDATE          ;Update [GETPNT]
10F2    22 F3FA                 LD      (GETPNT),HL     ;Set new [GETPNT]
10F5    79                      LD      A,C             ;Pass result to Acc
10F6    C3 08DB                 JP      PBDHRT
10F9                    CKCNTC:
                        ;
                        ; Check ctl-C
                        ;
10F9    E5                      PUSH    HL
10FA    21 0000                 LD      HL,0            ;To disable CONTinuing
10FD    CD 03FB                 CALL    ISCNTC
1100    E1                      POP     HL
1101    C9                      RET
                        ;
                        SUBTTL - MSXIO -  Music routines
```

```
1102                    WRTPSG:
                        ;
                        ; Write data to specified register of GI sound chip
                        ; Entry - (E)=data,(A)=register number
                        ; Exit - All regs preserved
                        ;
                        ; GI Reg# - usage
                        ;
                        ; 0     voice A fine tune
                        ; 1     voice A coarse tune
                        ; 2     voice B fine tune
                        ; 3     voice B coarse tune
                        ; 4     voice C fine tune
                        ; 5     voice C coarse tune
                        ; 7 B7,B6    = Reg 14,15 Input Output flags
                        ;   B5,B4,B3 = voice C,B,A noise enable (0=enabled)
                        ;   B2,B1,B0 = voice C,B,A tone enable (0=enabled)
                        ; 8     voice A volume (0..15 = volume, 16=use envelope)
                        ; 9     voice B volume (0..15 = volume, 16=use envelope)
                        ; 10    voice C volume (0..15 = volume, 16=use envelope)
                        ; 11-12 envelope period
                        ; 13    envelope shape (0..15)
                        ; 14    joystick 1 port
                        ; 15    joystick 2 port
                        ;
1102    F3                      DI
1103    D3 A0                   OUT     (PSG.LW),A      ;LATCH ADDRESS
1105    F5                      PUSH    AF
1106    7B                      LD      A,E
1107    D3 A1                   OUT     (PSG.DW),A      ;OUTPUT DATA
```

```
1109    FB                      EI
110A    F1                      POP     AF
110B    C9                      RET
110C                    INGI:
                        ;
                        ; Input data from PAD
                        ;
110C    3E 0E                   LD      A,PSG.PA
110E                    RDPSG:
110E    D3 A0                   OUT     (PSG.LW),A
1110    DB A2                   IN      A,(PSG.DR)
1112    C9                      RET
1113                    BEEP:
                        ;
                        ; BEEP causes a 'bell' sound
                        ;
                        ; Exit - all registers are destroyed
                        ;
1113    AF                      XOR     A               ;[A]=fine tune register for voice A
1114    1E 55                   LD      E,01010101B     ;data to be written on R0
1116    CD 1102                 CALL    WRTPSG
1119    5F                      LD      E,A             ;0 to coarse tune register
111A    3C                      INC     A
111B    CD 1102                 CALL    WRTPSG          ;R1 coarse
111E    1E BE                   LD      E,10111110B     ;enable voice [A] tone
1120    3E 07                   LD      A,7             ;[A]=voice enable register
1122    CD 1102                 CALL    WRTPSG          ;R7
1125    5F                      LD      E,A             ;set volume to 7
1126    3C                      INC     A               ;[A]=voice A volume register
1127    CD 1102                 CALL    WRTPSG          ;R8
112A    01 07D0                 LD      BC,07D0H
```

```
112D    CD 1133                 CALL    CSDLY1
1130    C3 04BD                 JP      GICINI          ;reset GI sound chip
1133                    CSDLY1:
                        ;
                        ; Delay by [BC]
                        ;
1133    0B                      DEC     BC
1134    E3                      EX      (SP),HL
1135    E3                      EX      (SP),HL
1136    78                      LD      A,B
1137    B1                      OR      C
1138    20 F9                   JR      NZ,CSDLY1
113A    C9                      RET
                        ;
113B                    ACTION:
                        ;
                        ; Get action information from specified music queue. Perform
                        ; action with synchronization.  Called by interrupt routine
                        ; in time.
                        ;
                        ; - Action information -
                        ;
                        ;    ITEM 1 - 2  BYTES
                        ;
                        ;           + Number of bytes that follow this item
                        ;           |
                        ;          NNNTTTTTTTTTTTTT
                        ;                  |
                        ;                  +Period of time
                        ;
                        ; ITEM 2, 3, 4 - FROM 1 TO 5 BYTES
```

```
                               ;
                               ;   IF HO 2 BITS = 0 then this is the HO byte of the tone period.
                               ;   IF HO 2 BITS = 2 then this is just a volume control byte.
                               ;       IF BIT 4 IS ON, envelope control is in effect, and bits
                               ;       0-3 give shape number of envelope.
                               ;       IF BIT 4 IS OFF, BITS 0-3 give amplitude number.
                               ;   IF HO 2 BITS = 3 THEN this byte will be followed by a 2 byte
                               ;       envelope period, HO first.
                               ;
                               ; ENTRY - (A)=Channel count number (0..2)
                               ;
113B    47                             LD      B,A             ;Save channel number
113C    CD 1470                        CALL    GETVCP          ;Get pointer into vcb of channel
113F    2B                             DEC     HL
1140    56                             LD      D,(HL)
1141    2B                             DEC     HL
1142    5E                             LD      E,(HL)          ;[DE]=countdown timer for voice
1143    1B                             DEC     DE              ;Decrement timer
1144    73                             LD      (HL),E          ;Put it back lo first
1145    23                             INC     HL
1146    72                             LD      (HL),D
1147    7A                             LD      A,D
1148    B3                             OR      E
1149    C0                             RET     NZ              ;No action if not zero
114A    78                             LD      A,B             ;Voice 0 uses queue 0
114B    32 FB3E                        LD      (QUEUEN),A      ;Set queue ID for further 'CALL XGETQ'
114E    CD 11E2                        CALL    XGETQ
1151    FE FF                          CP      0FFH
1153    28 5B                          JR      Z,VOICOF        ;branch if EOF marker
1155    57                             LD      D,A             ;SAVE IN [D]
1156    E6 E0                          AND     0E0H            ;Get number of following items
```

```
1158    07                      RLCA
1159    07                      RLCA
115A    07                      RLCA
115B    4F                      LD      C,A             ;Save in [C]
115C    7A                      LD      A,D
115D    E6 1F                   AND     1FH             ;GET LO 5 BITS OF [D]
115F    77                      LD      (HL),A          ;Set MSB of new countdown
1160    CD 11E2                 CALL    XGETQ           ;Get LSB of new countdown
1163    2B                      DEC     HL
1164    77                      LD      (HL),A          ;Set it
1165    0C                      INC     C
1166                    MORACT:
1166    0D                      DEC     C               ;Done all items?
1167    C8                      RET     Z               ;Yes
1168    CD 11E2                 CALL    XGETQ           ;Get next item from queue
116B    57                      LD      D,A             ;Save this to [D]
116C    E6 C0                   AND     0C0H            ;Get HO 2 bits
116E    20 11                   JR      NZ,XVOL         ;Execute volume action
                        ;
                        ; Set tone
                        ;
1170    CD 11E2                 CALL    XGETQ           ;Get low byte for tone
1173    5F                      LD      E,A
1174    78                      LD      A,B             ;Get back voice number
1175    07                      RLCA                    ;X 2
1176    CD 1102                 CALL    WRTPSG          ;Output fine tune register
1179    3C                      INC     A               ;Point to coarse tune register
117A    5A                      LD      E,D             ;Restore saved value
117B    CD 1102                 CALL    WRTPSG          ;Output coarse tune reg
117E    0D                      DEC     C               ;Decrement since we took 2 bytes from queue
117F    18 E5                   JR      MORACT
```

```
1181                XVOL:
                    ;
1181    67                  LD      H,A             ;save it in [H]
1182    E6 80               AND     80H             ;BIT 7 SET?
1184    28 0F               JR      Z,XEPER
                    ;
                    ; Set volume
                    ;
1186    5A                  LD      E,D             ;[A] has junk in ho which shouldn't matter
1187    78                  LD      A,B             ;Get back voice number
1188    C6 08               ADD     A,8             ;Regs 8,9,10
118A    CD 1102             CALL    WRTPSG          ;Output amplitude reg
118D    7B                  LD      A,E
118E    E6 10               AND     10H             ;Check envelope generate bit
1190    3E 0D               LD      A,0DH           ;Reg 13 for shape
1192    C4 1102             CALL    NZ,WRTPSG       ;Set envelope shape if enabled
1195                XEPER:
                    ;
                    ; Set envelope period
                    ;
1195    7C                  LD      A,H
1196    E6 40               AND     01000000B       ;See if set envelope period
1198    28 CC               JR      Z,MORACT        ;No
119A    CD 11E2             CALL    XGETQ           ;Get ho byte of envelope period
119D    57                  LD      D,A
119E    CD 11E2             CALL    XGETQ           ;Get low byte of envelope period
11A1    5F                  LD      E,A
11A2    3E 0B               LD      A,0BH           ;Register 11 for fine tune
11A4    CD 1102             CALL    WRTPSG
11A7    3C                  INC     A               ;Point to coarse tune
11A8    5A                  LD      E,D
```

```
11A9    CD 1102             CALL    WRTPSG
11AC    0D                  DEC     C
11AD    0D                  DEC     C
11AE    18 B6               JR      MORACT
11B0                VOICOF:
                    ;
                    ; Comes here when an EOF mark has been found for a specified
                    ; channel
                    ;
11B0    78                  LD      A,B
11B1    C6 08               ADD     A,8             ;Set appropriate reg #
11B3    1E 00               LD      E,0
11B5    CD 1102             CALL    WRTPSG          ;Turn off volume
11B8    04                  INC     B
11B9    21 FB3F             LD      HL,MUSICF
11BC    AF                  XOR     A
11BD    37                  SCF
11BE                RSTFL1:
11BE    17                  RLA
11BF    10 FD               DJNZ    RSTFL1
11C1    A6                  AND     (HL)            ;Get that bit
11C2    AE                  XOR     (HL)            ;Turn it off
11C3    77                  LD      (HL),A
11C4                STRTMS:
                    ;
                    ; STRTMS starts the background music task if:
                    ; 1) - it is currently idle (MUSICF=0) and
                    ; 2) - there is work queued for it (PLYCNT .GTR. 0)
                    ;
11C4    3A FB3F             LD      A,(MUSICF)
11C7    B7                  OR      A
```

```
11C8    C0                    RET     NZ              ;return if background task is active
11C9    21 FB40               LD      HL,PLYCNT
11CC    7E                    LD      A,(HL)
11CD    B7                    OR      A
11CE    C8                    RET     Z               ;return if nothing for it to do
11CF    35                    DEC     (HL)            ;1 less thing for it to do
11D0    21 0001               LD      HL,1
11D3    22 FB41               LD      (VCBA),HL       ;start it playing now
11D6    22 FB66               LD      (VCBB),HL
11D9    22 FB8B               LD      (VCBC),HL
11DC    3E 07                 LD      A,0111B         ;Trigger!
11DE    32 FB3F               LD      (MUSICF),A
11E1    C9                    RET
11E2                  XGETQ:
                      ;
11E2    3A FB3E               LD      A,(QUEUEN)      ;Get queue ID
11E5    E5                    PUSH    HL
11E6    D5                    PUSH    DE
11E7    C5                    PUSH    BC
11E8    CD 14AD               CALL    GETQ            ;Get a byte from a specified queue
11EB    C3 08DB               JP      PBDHRT          ;pop H, D, B and return
                      ;
                      SUBTTL - MSXIO -  Joystick and Paddle interface
```

```
11EE                GTSTCK:
                    ;
11EE    3D                  DEC     A
11EF    FA 1200             JP      M,KYSTCK        ;STICK(0) - read cursor keys
11F2    CD 120C             CALL    SLSTCK          ;Read joystick
11F5    21 1233             LD      HL,STKTBL
11F8                STICK1:
11F8    E6 0F               AND     0FH
11FA    5F                  LD      E,A
11FB    16 00               LD      D,0
11FD    19                  ADD     HL,DE
11FE    7E                  LD      A,(HL)
11FF    C9                  RET
1200                KYSTCK:
                    ;
1200    CD 1226             CALL    GTROW8          ;Read keyboard
1203    0F                  RRCA                    ;Move cursor status to lower four bits
1204    0F                  RRCA
1205    0F                  RRCA
1206    0F                  RRCA
1207    21 1243             LD      HL,KSTKTB
120A    18 EC               JR      STICK1
120C                SLSTCK:
                    ;
                    ; Select proper joystick and read from it
                    ;
120C    47                  LD      B,A
120D    3E 0F               LD      A,PSG.PB
120F    F3                  DI
1210    CD 110E             CALL    RDPSG           ;Read what is currently output to port B
```

```
1213    10 06                   DJNZ    SLSTC1          ;STICK(1)
1215    E6 DF                   AND     0DFH            ;Make sure P8 is low state
1217    F6 4C                   OR      4CH             ;Select joystick 2, enable P6,P7
1219    18 04                   JR      SLSTC2
121B                    SLSTC1:
                        ;
121B    E6 AF                   AND     0AFH            ;Select joystick 1, make sure P8 is low state
121D    F6 03                   OR      3               ;Enable P6,P7
121F                    SLSTC2:
121F    D3 A1                   OUT     (PSG.DW),A
1221    CD 110C                 CALL    INGI            ;Read status of joystick port
1224    FB                      EI
1225    C9                      RET
1226                    GTROW8:
                        ;
                        ; Get keyboard's 8th row, bit assignments are as follows.
                        ;
                        ; RDULxxxS
                        ; ||||   |
                        ; ||||   +- space
                        ; |||+----- left
                        ; ||+------ up
                        ; |+------- down
                        ; +-------- right
                        ;
1226    F3                      DI
1227    DB AA                   IN      A,(PPI.CR)
1229    E6 F0                   AND     0F0H
122B    C6 08                   ADD     A,8
122D    D3 AA                   OUT     (PPI.CW),A
122F    DB A9                   IN      A,(PPI.BR)
```

```
1231    FB                      EI
1232    C9                      RET
                        ;
1233                    STKTBL:
1233    00                      DB      0               ;RLBF
1234    05                      DB      5               ;RLB
1235    01                      DB      1               ;RL F
1236    00                      DB      0               ;RL
1237    03                      DB      3               ;R BF
1238    04                      DB      4               ;R B
1239    02                      DB      2               ;R F
123A    03                      DB      3               ;R
123B    07                      DB      7               ; LBF
123C    06                      DB      6               ; LB
123D    08                      DB      8               ; L F
123E    07                      DB      7               ; L
123F    00                      DB      0               ; BF
1240    05                      DB      5               ; B
1241    01                      DB      1               ;   F
1242    00                      DB      0               ;
                        ;
1243                    KSTKTB:
1243    00                      DB      0               ;RBFL
1244    03                      DB      3               ;RBF
1245    05                      DB      5               ;RB L
1246    04                      DB      4               ;RB
1247    01                      DB      1               ;R FL
1248    02                      DB      2               ;R F
1249    00                      DB      0               ;R L
124A    03                      DB      3               ;R
124B    07                      DB      7               ; BFL
```

```
124C    00                      DB      0               ; BF
124D    06                      DB      6               ; B L
124E    05                      DB      5               ; B
124F    08                      DB      8               ; FL
1250    01                      DB      1               ; F
1251    07                      DB      7               ;   L
1252    00                      DB      0               ;
                        ;
1253                    GTTRIG:
                        ;
1253    3D                      DEC     A
1254    FA 126C                 JP      M,KEYTRG        ;STRIG(0), use keyboard
1257    F5                      PUSH    AF
1258    E6 01                   AND     1
125A    CD 120C                 CALL    SLSTCK          ;Read joystick
125D    C1                      POP     BC
125E    05                      DEC     B
125F    05                      DEC     B
1260    06 10                   LD      B,10H           ;Prepare mask pattern for trigger A
1262    FA 1267                 JP      M,TRIG1
1265    06 20                   LD      B,' '           ;Prepare mask pattern for trigger B
1267                    TRIG1:
1267    A0                      AND     B               ;Extract trigger status
1268                    TRIG2:
1268    D6 01                   SUB     1               ;Return 255 if [Acc]=0, 0 if non-0
126A    9F                      SBC     A,A
126B    C9                      RET
126C                    KEYTRG:
                        ;
126C    CD 1226                 CALL    GTROW8          ;Read keyboard
126F    E6 01                   AND     1               ;Extract space status
```

```
1271    18 F5               JR      TRIG2
1273                GTPDL:
                    ;
                    ; Get value of paddle
                    ;
                    ; Input parameters (passed via [Acc])
                    ;
                    ; 1  - Paddle A connected to joystick port 1
                    ; 2  - Paddle A connected to joystick port 2
                    ; 3  - Paddle B connected to joystick port 1
                    ; 4  - Paddle B connected to joystick port 2
                    ; 5  - Paddle C connected to joystick port 1
                    ; 6  - Paddle C connected to joystick port 2
                    ; 7  - Paddle D connected to joystick port 1
                    ; 8  - Paddle D connected to joystick port 2
                    ; 9  - Paddle E connected to joystick port 1
                    ; 10 - Paddle E connected to joystick port 2
                    ; 11 - Paddle F connected to joystick port 1
                    ; 12 - Paddle F connected to joystick port 2
                    ;
1273    3C                  INC     A               ;Force parameter 2 based
1274    A7                  AND     A
1275    1F                  RRA
1276    F5                  PUSH    AF              ;Save port # (carry reset if port 1)
1277    47                  LD      B,A
1278    AF                  XOR     A
1279    37                  SCF
127A                PDL1:
127A    17                  RLA                     ;Form mask pattern
127B    10 FD               DJNZ    PDL1
127D    47                  LD      B,A             ;Set mask pattern
```

```
127E    F1                POP     AF
127F    0E 10             LD      C,10H           ;Assume port 1
1281    11 03AF           LD      DE,03AFH
1284    30 05             JR      NC,PDLP1        ;Good assumption
1286    0E 20             LD      C,' '
1288    11 4C9F           LD      DE,4C9FH
128B             PDLP1:
128B    3E 0F             LD      A,PSG.PB
128D    F3                DI
128E    CD 110E           CALL    RDPSG           ;Get current port B content
1291    A3                AND     E
1292    B2                OR      D
1293    B1                OR      C
1294    D3 A1             OUT     (PSG.DW),A      ;Set trigger high
1296    A9                XOR     C
1297    D3 A1             OUT     (PSG.DW),A      ;Set trigger low again
1299    3E 0E             LD      A,0EH
129B    D3 A0             OUT     (PSG.LW),A
129D    0E 00             LD      C,0             ;Initialize counter
129F             PDL2:
129F    DB A2             IN      A,(PSG.DR)
12A1    A0                AND     B               ;End of pulse?
12A2    28 05             JR      Z,PDL3          ;Yes
12A4    0C                INC     C               ;Bump counter
12A5    C2 129F           JP      NZ,PDL2         ;No overflow yet
12A8    0D                DEC     C               ;Make it 255
12A9             PDL3:
12A9    FB                EI
12AA    79                LD      A,C             ;Return counted value
12AB    C9                RET
12AC             GTPAD:
```

```
                        ;
                        ; Read touch pad (NEC PC-6051 compatible)
                        ;
                        ; Input parameter (passed via [Acc])
                        ;
                        ; 0 - sense touch pad status ---
                        ; 1 - return X coordinate       |for touch pad connected
                        ; 2 - return Y coordinate       |to joystick port 1
                        ; 3 - return switch status -----
                        ;
                        ; 4 - sense touch pad status ---
                        ; 5 - return X coordinate       |for touch pad connected
                        ; 6 - return Y coordinate       |to joystick port 2
                        ; 7 - return switch status -----
                        ;
                        ; Result is returned via [Acc]. As for status, 255 is returned
                        ; if true, 0 if false.
                        ;
12AC    FE 04                   CP      4               ;Read pad connected to port 1
12AE    11 0CEC                 LD      DE,0CECH        ;Assume so
12B1    38 05                   JR      C,GTPDP1        ;Good assumption
12B3    11 03D3                 LD      DE,03D3H        ;Connected to port 2
12B6    D6 04                   SUB     4
12B8                    GTPDP1:
12B8    3D                      DEC     A               ;Argument=0?
12B9    FA 12C5                 JP      M,GTPAD0        ;If so, read pad and return status
12BC    3D                      DEC     A
12BD    3A FC9D                 LD      A,(PADX)        ;Assume PAD(1) - X coordinate
12C0    F8                      RET     M               ;Good assumption
12C1    3A FC9C                 LD      A,(PADY)        ;Return Y coordinate
12C4    C8                      RET     Z
```

```
12C5                    GTPAD0:
12C5    F5                      PUSH    AF              ;Save status (minus if PAD(0) specified)
12C6    EB                      EX      DE,HL           ;[L]=bits that are not to be modified
12C7    22 F866                 LD      (RUNFLG),HL     ;[H]=bits that are to be added
12CA    9F                      SBC     A,A
12CB    2F                      CPL
12CC    E6 40                   AND     01000000B
12CE    4F                      LD      C,A             ;0 if port 1 specified, 100 octal if port 2
12CF    3E 0F                   LD      A,PSG.PB
12D1    F3                      DI                      ;disable interrupt till done
12D2    CD 110E                 CALL    RDPSG
12D5    E6 BF                   AND     0BFH
12D7    B1                      OR      C
12D8    D3 A1                   OUT     (PSG.DW),A      ;Select proper port
12DA    F1                      POP     AF
12DB    FA 12E8                 JP      M,TRYAGN        ;PAD(0) specified
12DE    CD 110C                 CALL    INGI
12E1    FB                      EI
12E2    E6 08                   AND     8
12E4    D6 01                   SUB     1
12E6    9F                      SBC     A,A
12E7    C9                      RET
12E8                    TRYAGN:
                        ;
12E8    0E 00                   LD      C,0             ;
12EA    CD 1332                 CALL    REDPAD          ;inz
12ED    CD 1332                 CALL    REDPAD          ;sense Panel input and select X
12F0    38 28                   JR      C,PADX1         ;branch if no input
12F2    CD 1320                 CALL    REDCOD          ;read first coordinate
12F5    38 23                   JR      C,PADX1         ;branch if input released
12F7    D5                      PUSH    DE              ;save for comparison
```

```
12F8    CD 1320                 CALL    REDCOD          ;read another input
12FB    C1                      POP     BC              ;restore previos coord
12FC    38 1C                   JR      C,PADX1         ;branch if input released
12FE    78                      LD      A,B
12FF    92                      SUB     D               ;[A]=ABS(X0-X1)
1300    30 02                   JR      NC,NONEG1
1302    2F                      CPL
1303    3C                      INC     A
1304                    NONEG1:
1304    FE 05                   CP      5               ;less than 5?
1306    30 E0                   JR      NC,TRYAGN       ;no, try again
1308    79                      LD      A,C
1309    93                      SUB     E               ;[A]=ABS(Y0-Y1)
130A    30 02                   JR      NC,NONEG2
130C    2F                      CPL
130D    3C                      INC     A
130E                    NONEG2:
130E    FE 05                   CP      5               ;less than 5
1310    30 D6                   JR      NC,TRYAGN       ;no, try again
1312    7A                      LD      A,D
1313    32 FC9D                 LD      (PADX),A        ;update coordinate [X]
1316    7B                      LD      A,E
1317    32 FC9C                 LD      (PADY),A        ;update coordinate [Y]
131A                    PADX1:
131A    FB                      EI                      ;finally enable interrupt
131B    7C                      LD      A,H             ;get SENSE input value
131C    D6 01                   SUB     1
131E    9F                      SBC     A,A
131F    C9                      RET                     ;return value
1320                    REDCOD:
                        ;
```

```
                          ; Read X,Y coordinate into [D,E]
                          ;
1320    0E 0A                     LD      C,0AH           ;change to channel to [Y] when done
1322    CD 1332                   CALL    REDPAD          ;read [X]
1325    D8                        RET     C               ;return if input released
1326    55                        LD      D,L
1327    D5                        PUSH    DE
1328    0E 00                     LD      C,0             ;change to [X] after read
132A    CD 1332                   CALL    REDPAD          ;read [Y]
132D    D1                        POP     DE
132E    5D                        LD      E,L             ;store Y read out
132F    AF                        XOR     A               ;clear carry
1330    67                        LD      H,A             ;force input is OK
1331    C9                        RET
1332                      REDPAD:
                          ;
                          ; Read touch panel input into [L]
                          ; Carry set if input released during read
                          ;
1332    CD 135B                   CALL    CHKEOC          ;make sure AD completed
1335    06 08                     LD      B,8             ;input 8 bits
1337    51                        LD      D,C             ;input channel# after done
1338                      REDLOP:
1338    CB 82                     RES     0,D             ;serial clock(SCK)=1
133A    CB 92                     RES     2,D
133C    CD 136D                   CALL    OUTGI
133F    CD 110C                   CALL    INGI            ;read PAD
1342    67                        LD      H,A             ;save SENSE status
1343    1F                        RRA
1344    1F                        RRA
1345    1F                        RRA
```

```
1346   CB 15                 RL      L               ;bit 2 to LSB of [L]
1348   CB C2                 SET     0,D             ;SCK=0
134A   CB D2                 SET     2,D
134C   CD 136D               CALL    OUTGI
134F   10 E7                 DJNZ    REDLOP
1351   CB E2                 SET     4,D
1353   CB EA                 SET     5,D
1355   CD 136D               CALL    OUTGI           ;initiate another AD
1358   7C                    LD      A,H             ;LSB=SENSE status
1359   1F                    RRA                     ;SENSE status to carry
135A   C9                    RET                     ;OK if no carry
135B               CHKEOC:
                   ;
                   ; Check and wait for EOC
                   ;
135B   3E 35                 LD      A,00110101B
135D   B1                    OR      C
135E   57                    LD      D,A
135F   CD 136D               CALL    OUTGI           ;reset CS
1362               EOCCHK:
1362   CD 110C               CALL    INGI
1365   E6 02                 AND     2               ;test EOC
1367   28 F9                 JR      Z,EOCCHK
1369   CB A2                 RES     4,D             ;set CS and return
136B   CB AA                 RES     5,D
136D               OUTGI:
                   ;
                   ; Output [D] to PAD
                   ;
136D   E5                    PUSH    HL
136E   D5                    PUSH    DE
```

```
4023     136F    2A F866             LD      HL,(RUNFLG)      ;Also known as [PADWRK]
4024     1372    7D                  LD      A,L
4025     1373    2F                  CPL
4026     1374    A2                  AND     D
4027     1375    57                  LD      D,A
4028     1376    3E 0F               LD      A,PSG.PB
4029     1378    D3 A0               OUT     (PSG.LW),A
4030     137A    DB A2               IN      A,(PSG.DR)
4031     137C    A5                  AND     L
4032     137D    B2                  OR      D
4033     137E    B4                  OR      H
4034     137F    D3 A1               OUT     (PSG.DW),A
4035     1381    D1                  POP     DE
4036     1382    E1                  POP     HL
4037     1383    C9                  RET
4038                         ;
4039                         SUBTTL - MSXIO -  Misc. routines for MSXIO
```

```

1384                  STMOTR:
1384     A7                     AND     A
1385     FA 1392                JP      M,FLPMOT        ;Flip motor switch
1388                  STMOT1:
1388     20 03                  JR      NZ,MOTRON
138A     3E 09                  LD      A,00001001B     ;Stop motor
138C     C2                     DB      0C2H            ;Skip next 2 bytes ('JNZ' instruction)
138D                  MOTRON:
138D     3E 08                  LD      A,8
138F     D3 AB                  OUT     (PPI.CM),A
1391     C9                     RET
1392                  FLPMOT:
                      ;
1392     DB AA                  IN      A,(PPI.CR)
1394     E6 10                  AND     10H
1396     18 F0                  JR      STMOT1
1398                  NMI:
                      ;
                      ; NMI handler
                      ;
1398     CD FDD6                CALL    H.NMI
139B     ED 45                  RETN                    ;RETN
```

```

                             ;
139D                         INIFNK:
                             ;
                             ; Initialize function key strings
                             ;
139D    01 00A0                      LD      BC,0A0H
13A0    11 F87F                      LD      DE,FNKSTR
13A3    21 13A9                      LD      HL,FKTABL
13A6    ED B0                        LDIR
13A8    C9                           RET
                             ;
13A9                         FNKDEF:
13A9    63 6F 6C 6F                  DB      "color "
13AD    72 20
13AF                                 DS      10
13B9    61 75 74 6F                  DB      "auto "
13BD    20
13BE                                 DS      11
13C9    67 6F 74 6F                  DB      "goto "
13CD    20
13CE                                 DS      11
13D9    6C 69 73 74                  DB      "list "
13DD    20
13DE                                 DS      11
13E9    72 75 6E                     DB      "run"
13EC    0D                           DB      13
13ED                                 DS      12
13F9    63 6F 6C 6F                  DB      "color 15,4,7"
13FD    72 20 31 35
1401    2C 34 2C 37
```

```
1405    0D                          DB      13
1406                                DS      3
1409    63 6C 6F 61                 DB      "cload"
140D    64
140E    22                          DB      34
140F                                DS      10
1419    63 6F 6E 74                 DB      "cont"
141D    0D                          DB      13
141E                                DS      11
1429    6C 69 73 74                 DB      "list."
142D    2E
142E    0D 1E 1E                    DB      13,30,30
1431                                DS      8
1439    0C                          DB      12
143A    72 75 6E                    DB      "run"
143D    0D                          DB      13
143E                                DS      11
                            ;
1449                        RDVDP:
                            ;
1449    DB 99                       IN      A,(VDP.SR)
144B    C9                          RET
144C                        RSLREG:
                            ;
144C    DB A8                       IN      A,(PPI.AR)
144E    C9                          RET
144F                        WSLREG:
                            ;
144F    D3 A8                       OUT     (PPI.AW),A
1451    C9                          RET
1452                        SNSMAT:
```

```
                        ;
1452    4F                      LD      C,A
1453    F3                      DI
1454    DB AA                   IN      A,(PPI.CR)      ;Get what is currently output to Port C
1456    E6 F0                   AND     0F0H            ;Leave higher 4 bits unaffected
1458    81                      ADD     A,C
1459    D3 AA                   OUT     (PPI.CW),A      ;Select row
145B    DB A9                   IN      A,(PPI.BR)      ;Get column information of selected row
145D    FB                      EI
145E    C9                      RET
145F                    ISFLIO:
                        ;
                        ; Check if we're doing device I O
                        ;
145F    CD FEDF                 CALL    H.ISFL
1462    E5                      PUSH    HL              ;Save [H,L]
1463    2A F864                 LD      HL,(PTRFIL)     ;Get file pointer
1466    7D                      LD      A,L
1467    B4                      OR      H               ;No zero?
1468    E1                      POP     HL              ;Restore [H,L]
1469    C9                      RET
146A                    DCOMPR:
                        ;
                        ; COMPAR compares [H,L] with [D,E] unsigned
                        ;
                        ; [H,L] less than [D,E] set carry
                        ; [H,L] = [D,E] set zero
                        ;
                        ; [A] is the only register used
                        ;
146A    7C                      LD      A,H
```

```
146B    92                     SUB     D
146C    C0                     RET     NZ
146D    7D                     LD      A,L
146E    93                     SUB     E
146F    C9                     RET
1470                   GETVCP:
                       ;
                       ; Entry - [A] = voice id (0..2)
                       ; Exit  - [HL] = pointer to QLENGX for voice (within static var buf)
                       ; [A] = 0. All other registers preserved.
                       ;
1470    2E 02                  LD      L,2
1472    18 03                  JR      GETVC1
1474                   GETVC2:
                       ;
                       ; Entry - [L] = desired displacement into voice buffer
                       ; Exit - [HL] = pointer to desired variable for voice VOICEN
                       ; [A] = 0. All other registers preserved.
                       ;
1474    3A FB38                LD      A,(VOICEN)
1477                   GETVC1:
                       ;
                       ; Entry - [A] = voice id (0..2)
                       ; [L] = desired displacement into voice buffer
                       ; Exit - [HL] = pointer to desired variable for voice VOICEN
                       ; [A] = 0. All other registers preserved.
                       ;
1477    D5                     PUSH    DE
1478    11 FB41                LD      DE,VCBA
147B    26 00                  LD      H,0
147D    19                     ADD     HL,DE
```

```
147E    B7                      OR      A
147F    28 07                   JR      Z,GETVCX
1481    11 0025                 LD      DE,25H          ;VCB size
1484                    GETVCL:
1484    19                      ADD     HL,DE
1485    3D                      DEC     A
1486    20 FC                   JR      NZ,GETVCL
1488                    GETVCX:
1488    D1                      POP     DE
1489    C9                      RET
148A                    PHYDIO:
                        ;
148A    CD FFA7                 CALL    H.PHYD
148D    C9                      RET
148E                    FORMAT:
                        ;
148E    CD FFAC                 CALL    H.FORM
1491    C9                      RET
                        SUBTTL - QUEUTL -  Queue utility routines

                        ;       Copyright (C) 1980 by Microsoft Corporation
                        ;       Written by Marc Wilson
                        ;
                        ; This utility provides for multiple queues with the following
                        ; capabilities:
                        ;
                        ; Queues of varying length - 1,3,7,15,31,63,127,255
                        ;
                        ; Each queue can be any of the possible lengths
                        ; The queues can be initialized at any time and be
                        ; located anywhere a single pointer (QUEUES) provides
                        ; the address of the queue table.
                        ;
                        ; The queue table has all information for each queue,
                        ; 6 bytes per queue.  A single non-zero character can
                        ; be pushed back on top of the queue.
                        ;
                        ; The entry for each queue is as follows:
                        ;         +0      PUT OFFSET
                        ;         +1      GET OFFSET
                        ;         +2      BACK CHARACTER
                        ;         +3      QUEUE LENGTH
                        ;         +4,+5   QUEUE ADDRESS
                        ;
                        ; The utility assumes that the queue table is
                        ; valid for all queue numbers passed to the routines
                        ;
                        ;ROUTINES:
                        ; All routines assume that [A] equals the queue number,
                        ; [QUEUES] contains the address of the queue table.
```

```
; Other requirements follow.
;   GETQ    - Returns current top of queue in [A],
;             zero flag set if queue empty
;   PUTQ    - Puts byte in [E] reg on end of queue,
;             zero set if queue is full
;
;NOTE:
;  The routines are designed to be reentrant, however
;  there are some restrictions for cases involving a
;  single queue (in any case operating on different
;  queues is alright). The first restriction is that
;  the same routine cannot be reentered. The second
;  is that INITQ and POPQ do not allow PUTQ,
;  GETQ or BCKQ to be entered.
;
;  LFTQ   - Returns unused number of bytes in queue in [A] reg
;  INITQ  - Initialize queue to empty state,
;           B reg=length, (DE)=ADDR
; *** All routines destroy the registers ***
;
SUBTTL  - QUEUTL -  Queue routines
```

```

1492                PUTQ:
                    ;
                    ; Put data on queue
                    ;
1492    CD 14FA             CALL    GETPTR          ;Get queue pointers
1495    78                  LD      A,B
1496    3C                  INC     A               ;Bump PUT
1497    23                  INC     HL
1498    A6                  AND     (HL)            ;Wrap around
1499    B9                  CP      C
149A    C8                  RET     Z               ;QUEUE full
149B    E5                  PUSH    HL
149C    2B                  DEC     HL
149D    2B                  DEC     HL
149E    2B                  DEC     HL
149F    E3                  EX      (SP),HL         ;Save place to put new pointer
14A0    23                  INC     HL
14A1    4F                  LD      C,A             ;Pointer in C
14A2    7E                  LD      A,(HL)
14A3    23                  INC     HL
14A4    66                  LD      H,(HL)
14A5    6F                  LD      L,A             ;(HL) = QUEUE address
14A6    06 00               LD      B,0
14A8    09                  ADD     HL,BC           ;(HL) = Address to put char
14A9    73                  LD      (HL),E
14AA    E1                  POP     HL
14AB    71                  LD      (HL),C          ;set new pointer
14AC    C9                  RET
14AD                GETQ:
                    ;
```

```
                          ; Get data from QUEUE
                          ;
14AD    CD 14FA                     CALL    GETPTR          ;Get queue pointers
14B0    36 00                       LD      (HL),0          ;zero back character
14B2    20 1D                       JR      NZ,GETBAK
14B4    79                          LD      A,C
14B5    B8                          CP      B
14B6    C8                          RET     Z               ;QUEUE empty!
14B7    23                          INC     HL
14B8    3C                          INC     A               ;Bump GET offset
14B9    A6                          AND     (HL)            ;wrap around
14BA    2B                          DEC     HL
14BB    2B                          DEC     HL
14BC    E5                          PUSH    HL              ;Save place to store pointer
14BD    23                          INC     HL
14BE    23                          INC     HL
14BF    23                          INC     HL
14C0    4F                          LD      C,A             ;offset in C
14C1    7E                          LD      A,(HL)
14C2    23                          INC     HL
14C3    66                          LD      H,(HL)
14C4    6F                          LD      L,A             ;[HL] = QUEUE address
14C5    06 00                       LD      B,0
14C7    09                          ADD     HL,BC
14C8    7E                          LD      A,(HL)          ;get char from QUEUE
14C9    E1                          POP     HL
14CA    71                          LD      (HL),C
14CB    B7                          OR      A
14CC    C0                          RET     NZ
14CD    3C                          INC     A
14CE    3E 00                       LD      A,0
```

```
14D0    C9                      RET
14D1                    GETBAK:
14D1    4F                      LD      C,A
14D2    06 00                   LD      B,0
14D4    21 F970                 LD      HL,QUEBAK-1
14D7    09                      ADD     HL,BC
14D8    7E                      LD      A,(HL)
14D9    C9                      RET
14DA                    INITQ:
                        ;
                        ; INITQ - Initialize QUEUE
                        ;
14DA    C5                      PUSH    BC              ;Save queue length
14DB    CD 1504                 CALL    QSTART          ;Get addr of start of QUEUE table entry
14DE    70                      LD      (HL),B          ;Clear PUT offset
14DF    23                      INC     HL
14E0    70                      LD      (HL),B          ;Clear GET offset
14E1    23                      INC     HL
14E2    70                      LD      (HL),B          ;Clear back character
14E3    23                      INC     HL
14E4    F1                      POP     AF
14E5    77                      LD      (HL),A          ;Set QUEUE length
14E6    23                      INC     HL
14E7    73                      LD      (HL),E
14E8    23                      INC     HL
14E9    72                      LD      (HL),D          ;Set QUEUE address
14EA    C9                      RET
14EB                    LFTQ:
                        ;
                        ; LFTQ - Returns number of bytes remaining in QUEUE
                        ;
```

```
14EB    CD 14FA                 CALL    CETPTR          ;Get QUEUE ptrs
14EE    78                      LD      A,B
14EF    3C                      INC     A
14F0    23                      INC     HL
14F1    A6                      AND     (HL)
14F2    47                      LD      B,A             ;B=PUT PTR+1
14F3    79                      LD      A,C
14F4    90                      SUB     B               ;subtract PUT from GET
14F5    A6                      AND     (HL)            ;make it positive UNSIGNED INTEGER
14F6    6F                      LD      L,A
14F7    26 00                   LD      H,0
14F9    C9                      RET

14FA                    GETPTR:
                        ;
                        ; QUEUE general routines
                        ;
14FA    CD 1504                 CALL    QSTART          ;Get start of QUEUE TABLE entry
14FD    46                      LD      B,(HL)          ;B = PUT OFFSET
14FE    23                      INC     HL
14FF    4E                      LD      C,(HL)          ;C = GET OFFSET
1500    23                      INC     HL
1501    7E                      LD      A,(HL)          ;A = BACK CHARACTER
1502    B7                      OR      A
1503    C9                      RET
                        ;
1504                    QSTART:
1504    07                      RLCA                    ;*2
1505    47                      LD      B,A
1506    07                      RLCA                    ;*4
1507    80                      ADD     A,B             ;*6
```

```
1508    4F                      LD      C,A
1509    06 00                   LD      B,0
150B    2A F3F3                 LD      HL,(QUEUES)
150E    09                      ADD     HL,BC
150F    C9                      RET
                        SUBTTL - MSXGRP -  Graphic driver (Print a character on GRP screen)
```

```

1510                GRPPRT:
                    ;
                    ; Print a character on the graphic screen
                    ;
1510    E5                  PUSH    HL
1511    D5                  PUSH    DE
1512    C5                  PUSH    BC
1513    F5                  PUSH    AF
1514    CD 089D             CALL    CNVCHR          ;Convert code
1517    30 62               JR      NC,JPPPAL       ;Graphic header byte, return soon
1519    20 08               JR      NZ,GPRT05       ;Converted graphic code
151B    FE 0D               CP      0DH             ;CR?
151D    28 5F               JR      Z,GRPCR         ;Do not ignore CR even on graphic screen
151F    FE 20               CP      ' '             ;Control character?
1521    38 58               JR      C,JPPPAL        ;Yes, ignore this
1523                GPRT05:
1523    CD 0752             CALL    GETPAT          ;Get character pattern in PATWRK
1526    3A F3E9             LD      A,(FORCLR)      ;Set color of character
1529    32 F3F2             LD      (ATRBYT),A
152C    2A FCB9             LD      HL,(GRPACY)
152F    EB                  EX      DE,HL           ;Current Y coordinate in [DE]
1530    ED 4B FCB7          LD      BC,(GRPACX)     ;Current X coordinate in [BC]
1534    CD 1599             CALL    SCALXY          ;Do the scaling
1537    30 42               JR      NC,JPPPAL       ;Do not print if already out of screen
1539    CD 15DF             CALL    MAPXYC          ;Map to CLOC and CMASK
153C    11 FC40             LD      DE,PATWRK
153F    0E 08               LD      C,8             ;Row counter
1541                GPRT10:
1541    06 08               LD      B,8             ;Column counter
1543    CD 1639             CALL    FETCHC          ;Get current CLOC and CMASK
```

```
1546   E5                   PUSH    HL              ;Save these
1547   F5                   PUSH    AF
1548   1A                   LD      A,(DE)          ;Get pattern for a row
1549               GPRT20:
1549   87                   ADD     A,A             ;Check each bit
154A   F5                   PUSH    AF
154B   DC 167E              CALL    C,SETC          ;Set it if 1
154E   CD 16AC              CALL    TRIGHT          ;Move 1 pixel right
1551   E1                   POP     HL              ;Assume out of screen
1552   38 04                JR      C,GPRT30        ;Good assumption, skip the rest
1554   E5                   PUSH    HL
1555   F1                   POP     AF
1556   10 F1                DJNZ    GPRT20          ;Loop till done all columns
1558               GPRT30:
1558   F1                   POP     AF              ;Restore CLOC and CMASK
1559   E1                   POP     HL
155A   CD 1640              CALL    STOREC          ;Set these
155D   CD 170A              CALL    TDOWNC          ;Move 1 pixel down
1560   38 04                JR      C,GPRT40        ;Out of screen, skip rest and return
1562   13                   INC     DE              ;Point to next row
1563   0D                   DEC     C
1564   20 DB                JR      NZ,GPRT10       ;Loop till done all rows
1566               GPRT40:
1566   CD 15D9              CALL    CHKMOD          ;Check current screen mode
1569   3A FCB7              LD      A,(GRPACX)
156C   28 06                JR      Z,GPRT50        ;We're in high-resolution mode
156E   C6 20                ADD     A,' '
1570   38 0C                JR      C,GRPCR         ;We're going out of screen
1572   18 04                JR      GPRT60
1574               GPRT50:
                   ;
```

```
1574    C6 08                ADD     A,8
1576    38 06                JR      C,GRPCR
1578                GPRT60:
1578    32 FCB7              LD      (GRPACX),A      ;Update cursor position
157B                JPPPAL:
157B    C3 08DA              JP      POPALL
157E                GRPCR:
                    ;
157E    AF                   XOR     A               ;Reset X position
157F    32 FCB7              LD      (GRPACX),A
1582    CD 15D9              CALL    CHKMOD
1585    3A FCB9              LD      A,(GRPACY)
1588    28 03                JR      Z,GPRT70
158A    C6 20                ADD     A,4*8
158C    01                   DB      1
158D                GPRT70:
158D    C6 08                ADD     A,8
158F    FE C0                CP      0C0H
1591    38 01                JR      C,GPRT80
1593    AF                   XOR     A               ;Reset Y position also
1594                GPRT80:
1594    32 FCB9              LD      (GRPACY),A
1597    18 E2                JR      JPPPAL
                    SUBTTL - MSXGRP -  (Routines for general graphics)
```

```
        1599                    SCALXY:
                                ;
                                ; SCALXY - Clips X,Y to max values in physical size and flags out
                                ; of range values.
                                ;
                                ; ENTRY [BC] = X (0 ... max X),  [DE] = Y (0 ... max Y)
                                ; EXIT  [BC] = X clipped,        [DE] = Y clipped
                                ;  CARRY is reset if one of the value was out of bound
                                ;
        1599    E5                      PUSH    HL              ;save [HL]
        159A    C5                      PUSH    BC              ;save [BC] - X coordinate
        159B    06 01                   LD      B,1             ;no-error flag
        159D    EB                      EX      DE,HL           ;Y coordinate to [HL]
        159E    7C                      LD      A,H             ;Is Y coordinate negative?
        159F    87                      ADD     A,A
        15A0    30 05                   JR      NC,YPOSTV       ;No, positive
        15A2    21 0000                 LD      HL,0            ;Substitute by 0 is negative
        15A5    18 08                   JR      YNEGTV          ;And set out of bound flag
        15A7                    YPOSTV:
                                ;
        15A7    11 00C0                 LD      DE,0C0H         ;Maximum Y+1
        15AA    E7                      RST     20H             ;Test [HL] with [DE]
        15AB    38 04                   JR      C,SCLYOK        ;if carry, not out of bound
        15AD    EB                      EX      DE,HL           ;[HL] = 192
        15AE    2B                      DEC     HL              ;Y = 191 ,maximum Y coordinate
        15AF                    YNEGTV:
        15AF    06 00                   LD      B,0             ;set out of bound flag
        15B1                    SCLYOK:
        15B1    E3                      EX      (SP),HL         ;save Y and get X to [HL]
        15B2    7C                      LD      A,H             ;Is X coordinate negative?
```

```
15B3    87                    ADD     A,A
15B4    30 05                 JR      NC,XPOSTV        ;No, positive
15B6    21 0000               LD      HL,0             ;Substitute by 0 if negative
15B9    18 08                 JR      XNEGTV           ;And set out of bound flag
15BB                  XPOSTV:
                      ;
15BB    11 0100               LD      DE,0100H         ;max X +1
15BE    E7                    RST     20H              ;Test [HL] with [DE]
15BF    38 04                 JR      C,SCLXOK
15C1    EB                    EX      DE,HL            ;[HL] = 256
15C2    2B                    DEC     HL               ;[HL] = 255 - max X coordinate
15C3                  XNEGTV:
15C3    06 00                 LD      B,0              ;error flag
15C5                  SCLXOK:
15C5    D1                    POP     DE               ;restore [DE] = Y
15C6    CD 15D9               CALL    CHKMOD
15C9    28 08                 JR      Z,HRSSCL         ;We're in high-resolution mode
15CB    CB 3D                 SRL     L                ;Divide both X and Y by 4 because we're
15CD    CB 3D                 SRL     L                ;in multi-color mode
15CF    CB 3B                 SRL     E
15D1    CB 3B                 SRL     E
15D3                  HRSSCL:
15D3    78                    LD      A,B
15D4    0F                    RRCA                     ;set carry if no error
15D5    44                    LD      B,H              ;[BC] = X
15D6    4D                    LD      C,L
15D7    E1                    POP     HL               ;restore [HL]
15D8    C9                    RET
15D9                  CHKMOD:
                      ;
                      ; Check current screen mode
```

```
         ;
15D9  3A FCAF          LD      A,(SCRMOD)
15DC  D6 02            SUB     2                 ;In what mode are we now?
15DE  C9               RET                       ;Return with the condition flag
15DF             MAPXYC:
                 ;
                 ; MAPXYC - Maps X,Y coordinates to "C" (address, mask)
                 ;
                 ; Entry: [BC] = X,   [DE] = Y
                 ;
                 ; Exit:  CLOC = [HL] -- Video Ram address
                 ; CMASK = [A] -- Bit Mask
                 ;
                 ; [ High-resolution mode ]
                 ;
                 ;      X coord - XXXXXXXX ( 8 bits, max=255)
                 ;                76543210
                 ;
                 ;      Y coord - YYYYYYYY ( 8 bits, max=191)
                 ;                76543210
                 ;
                 ;      CLOC =  YYYYYXXXXXYYY
                 ;              7654376543210
                 ;                              XXX
                 ;                              210
                 ;--------------------------------------
                 ; CMASK =      10000000 000
                 ;              01000000 001
                 ;              00100000 010
                 ;              00010000 011
                 ;              00001000 100
```

```
                          ;               00000100 101
                          ;               00000010 110
                          ;               00000001 111
                          ;
                          ; [ Multi-color mode ]
                          ;
                          ;       X coord - XXXXXX ( 6 bits, max=63 )
                          ;                 543210
                          ;
                          ;       Y coord - YYYYYY ( 6 bits, max=47 )
                          ;                 543210
                          ;
                          ;       CLOC =  YYYXXXXXYYY
                          ;               54354321210
                          ;
                          ; CMASK = 11110000 if X0=0 (even)
                          ; CMASK = 00001111 if X0=1 (odd)
                          ;
                          ;  Note:  The boundary check has already been done by a call
                          ;  to SCALXY, so no range checking is needed.
                          ;
15DF    C5                        PUSH    BC              ;Save X
15E0    CD 15D9                   CALL    CHKMOD          ;Check current screen mode
15E3    20 2E                     JR      NZ,MMPXYC       ;Multi-color mode
15E5    51                        LD      D,C             ;Save X to D also
15E6    79                        LD      A,C
15E7    E6 07                     AND     7
15E9    4F                        LD      C,A
15EA    21 160B                   LD      HL,TWOPWR       ;Table of power of two
15ED    09                        ADD     HL,BC
15EE    7E                        LD      A,(HL)          ;read bit mask CMASK
```

```
15EF    32 F92C                 LD      (CMASK),A
15F2    7B                      LD      A,E             ;Get Y coordinate
15F3    0F                      RRCA
15F4    0F                      RRCA
15F5    0F                      RRCA
15F6    E6 1F                   AND     00011111B
15F8    47                      LD      B,A
15F9    7A                      LD      A,D             ;Get X coordinate
15FA    E6 F8                   AND     11111000B
15FC    4F                      LD      C,A
15FD    7B                      LD      A,E             ;Get Y coordinate
15FE    E6 07                   AND     00000111B
1600    B1                      OR      C
1601    4F                      LD      C,A
1602    2A F3CB                 LD      HL,(GRPCGP)
1605    09                      ADD     HL,BC
1606    22 F92A                 LD      (CLOC),HL       ;Set pattern generator address
1609    C1                      POP     BC
160A    C9                      RET
160B                    TWOPWR:
                        ;
                        ; Table of power of two
                        ;
160B    80 40 20 10             DB      80H,40H,20H,10H
160F    08 04 02 01             DB      08H,04H,02H,01H
                        ;
1613                    MMPXYC:
                        ;
                        ; Map XY for multi-color mode
                        ;
1613    79                      LD      A,C             ;Get X position
```

```
1614    0F                      RRCA                    ;Even or odd?
1615    3E F0                   LD      A,11110000B     ;Assume even
1617    30 02                   JR      NC,MMPXY1       ;Good assumption
1619    3E 0F                   LD      A,00001111B     ;Odd
161B                    MMPXY1:
161B    32 F92C                 LD      (CMASK),A       ;Set up mask pattern
161E    79                      LD      A,C
161F    87                      ADD     A,A
1620    87                      ADD     A,A
1621    E6 F8                   AND     11111000B
1623    4F                      LD      C,A             ;Get lower byte
1624    7B                      LD      A,E
1625    E6 07                   AND     0111B
1627    B1                      OR      C
1628    4F                      LD      C,A
1629    7B                      LD      A,E
162A    0F                      RRCA
162B    0F                      RRCA
162C    0F                      RRCA
162D    E6 07                   AND     0111B
162F    47                      LD      B,A             ;Get higher byte
1630    2A F3D5                 LD      HL,(MLTCGP)     ;Load start address of pattern table
1633    09                      ADD     HL,BC
1634    22 F92A                 LD      (CLOC),HL
1637    C1                      POP     BC
1638    C9                      RET
```

```

1639                 FETCHC:
                     ;
                     ; FETCHC - Reads the value of the graphics accumulater
                     ;
                     ; Exit: [HL] = CLOC,  [A] = CMASK
                     ;
1639   3A F92C               LD      A,(CMASK)
163C   2A F92A               LD      HL,(CLOC)
163F   C9                    RET
1640                 STOREC:
                     ;
                     ; STOREC - Sets the graphics accumulater
                     ;
                     ; Entry: [HL] = CLOC,  [A] = CMASK
                     ;
1640   32 F92C               LD      (CMASK),A
1643   22 F92A               LD      (CLOC),HL
1646   C9                    RET
1647                 READC:
                     ;
                     ; READC - Get the attribute of the current graphics accumulater
                     ; position
                     ;
1647   C5                    PUSH    BC
1648   E5                    PUSH    HL
1649   CD 1639               CALL    FETCHC          ;Get CLOC and CMASK
164C   47                    LD      B,A             ;Save CMASK
164D   CD 15D9               CALL    CHKMOD          ;Check current screen mode
1650   20 1A                 JR      NZ,MREADC       ;Multi-color mode
1652   CD 07D7               CALL    RDVRM           ;Read VDP's VRAM (pattern)
```

```
1655    A0                      AND     B               ;Extract specified pixel
1656    F5                      PUSH    AF              ;Save whether the pixel is on or off
1657    01 2000                 LD      BC,GRPDIF
165A    09                      ADD     HL,BC
165B    CD 07D7                 CALL    RDVRM           ;Read VDP's VRAM (color)
165E    47                      LD      B,A             ;Save this to B
165F    F1                      POP     AF              ;Restore condition
1660    78                      LD      A,B             ;Restore color
1661    28 04                   JR      Z,READC1        ;Specified dot is off, return
                                                        ;background color
1663                    READC0:
1663    0F                      RRCA                    ;Specified dot is on, return foreground color
1664    0F                      RRCA
1665    0F                      RRCA
1666    0F                      RRCA
1667                    READC1:
1667    E6 0F                   AND     0FH             ;Make it a legal value
1669    E1                      POP     HL
166A    C1                      POP     BC
166B    C9                      RET
166C                    MREADC:
                        ;
166C    CD 07D7                 CALL    RDVRM           ;Read VRAM
166F    04                      INC     B               ;Check if specified pixel is even or odd
1670    05                      DEC     B
1671    F2 1667                 JP      P,READC1        ;Odd, return lower nibble
1674    18 ED                   JR      READC0          ;Even, return upper nibble
```

```

        1676                SETATR:
                            ;
                            ; SETATR - Sets the attribute (color, reverse, etc..) to be
                            ; used in future actions.
                            ;
                            ; Entry - [A] = Attribute
                            ; Exit - carry set if illegal value
                            ;
        1676    FE 10               CP      16              ;Must be less than 16
        1678    3F                  CCF
        1679    D8                  RET     C
        167A    32 F3F2             LD      (ATRBYT),A
        167D    C9                  RET
        167E                SETC:
                            ;
                            ; SETC - Sets the point indicated by the graphics accumulater
                            ; to ATTRBYT
                            ;
                            ; All registers except AF must be preserved.
                            ;
        167E    E5                  PUSH    HL
        167F    C5                  PUSH    BC
        1680    CD 15D9             CALL    CHKMOD          ;Check current screen mode
        1683    CD 1639             CALL    FETCHC
        1686    20 08               JR      NZ,MSETC        ;Multi-color mode
        1688    D5                  PUSH    DE
        1689    CD 186C             CALL    PATWRT
        168C    D1                  POP     DE
        168D    C1                  POP     BC
        168E    E1                  POP     HL
```

```
168F    C9                      RET
1690                    MSETC:
                        ;
                        ; Set a pixel in multi-color mode
                        ;
1690    47                      LD      B,A             ;Save CMASK in [B]
1691    CD 07D7                 CALL    RDVRM           ;Read VRAM
1694    4F                      LD      C,A
1695    78                      LD      A,B
1696    2F                      CPL                     ;Leave another unaffected
1697    A1                      AND     C
1698    4F                      LD      C,A
1699    3A F3F2                 LD      A,(ATRBYT)      ;Get specified color
169C    04                      INC     B               ;Check if even or odd
169D    05                      DEC     B
169E    F2 16A5                 JP      P,MSETC1        ;Odd
16A1    87                      ADD     A,A
16A2    87                      ADD     A,A
16A3    87                      ADD     A,A
16A4    87                      ADD     A,A
16A5                    MSETC1:
16A5    B1                      OR      C               ;Form new color
16A6    CD 07CD                 CALL    WRTVRM          ;Write new pattern
16A9    C1                      POP     BC
16AA    E1                      POP     HL
16AB    C9                      RET
                        SUBTTL - MSXGRP -  (Graphic cursor movements)

                        ;
                        ; UPC, DOWNC, RIGHTC, LEFTC
                        ;
                        ; These are the  C relative movement routines. They
                        ; adjust the current graphics accumulater in the indicated
                        ; direction without checking boundary conditions.
                        ;
                        ;----------------------------------------------------
                        ;
16AC                    TRIGHT:
                        ;
                        ; TRIGHT - move 1 pixel right
                        ; Return carry set if already on border
                        ;
16AC    E5                      PUSH    HL
16AD    CD 15D9                 CALL    CHKMOD
16B0    C2 1779                 JP      NZ,MTRGT
16B3    CD 1639                 CALL    FETCHC          ;Get CLOC,CMASK
16B6    0F                      RRCA                    ;Move 1 pixel right
16B7    30 4B                   JR      NC,HRZMV1       ;Within byte, just change CMASK
16B9    7D                      LD      A,L             ;Get low byte of CLOC
16BA    E6 F8                   AND     0F8H
16BC    FE F8                   CP      0F8H            ;On right edge?
16BE    3E 80                   LD      A,80H           ;Assume not
16C0    20 10                   JR      NZ,RGHTC1       ;Goot assumption
16C2    C3 175A                 JP      ONBRD1          ;On border, set carry and return
16C5                    RIGHTC:
                        ;
                        ; RIGHTC - move 1 pixel right
                        ;
```

```
16C5     E5                       PUSH     HL
16C6     CD 15D9                  CALL     CHKMOD
16C9     C2 178B                  JP       NZ,MRGTC
16CC     CD 1639                  CALL     FETCHC
16CF     0F                       RRCA                      ;move right 1 pixel
16D0     30 32                    JR       NC,HRZMV1        ;within byte, just change CMASK
16D2                      RGHTC1:
16D2     D5                       PUSH     DE
16D3     11 0008                  LD       DE,8             ;Load offset to new position
16D6     18 27                    JR       HRZMOV           ;Change CLOC also
16D8                      TLEFT:
                          ;
                          ; TLEFT - move 1 pixel left
                          ; Return carry set if already on border
                          ;
16D8     E5                       PUSH     HL
16D9     CD 15D9                  CALL     CHKMOD
16DC     C2 179C                  JP       NZ,MTLFT
16DF     CD 1639                  CALL     FETCHC           ;Get CLOC and CMASK
16E2     07                       RLCA                      ;Move 1 pixel left
16E3     30 1F                    JR       NC,HRZMV1        ;Within byte boundary, just change CMASK
16E5     7D                       LD       A,L              ;Check if we're on left edge
16E6     E6 F8                    AND      0F8H
16E8     3E 01                    LD       A,1              ;Assume not
16EA     20 0F                    JR       NZ,LEFTC1        ;Good assumption
16EC     18 6C                    JR       ONBRD1           ;We're on border, set carry and return
16EE                      LEFTC:
                          ;
                          ; LEFTC - move 1 pixel left
                          ;
16EE     E5                       PUSH     HL
```

```
16EF    CD 15D9                 CALL    CHKMOD
16F2    C2 17AC                 JP      NZ,MLFTC
16F5    CD 1639                 CALL    FETCHC
16F8    07                      RLCA                    ;move left 1 pixel
16F9    30 09                   JR      NC,HRZMV1       ;within byte boundary, just change CMASK
16FB                    LEFTC1:
16FB    D5                      PUSH    DE
16FC    11 FFF8                 LD      DE,0FFF8H       ;Load offset to new position
16FF                    HRZMOV:
16FF    19                      ADD     HL,DE           ;Add offset to new position
1700    22 F92A                 LD      (CLOC),HL       ;Update pattern address
1703    D1                      POP     DE
1704                    HRZMV1:
1704    32 F92C                 LD      (CMASK),A       ;Update CMASK
1707    A7                      AND     A               ;Clear carry
1708    E1                      POP     HL
1709    C9                      RET
170A                    TDOWNC:
                        ;
                        ; TDOWNC - move 1 pixel down.
                        ;
                        ; Return carry set if already on screen border.
                        ;
170A    E5                      PUSH    HL
170B    D5                      PUSH    DE
170C    2A F92A                 LD      HL,(CLOC)
170F    CD 15D9                 CALL    CHKMOD
1712    C2 17C6                 JP      NZ,MTDNC
1715    E5                      PUSH    HL
1716    2A F3CB                 LD      HL,(GRPCGP)
1719    11 1700                 LD      DE,1700H
```

```
171C    19                    ADD     HL,DE
171D    EB                    EX      DE,HL
171E    E1                    POP     HL
171F    E7                    RST     20H             ;Test [HL] with [DE]
                                                      ;Looks like on border?
1720    38 13                 JR      C,DWNC10        ;No
1722    7D                    LD      A,L             ;Possibly on border
1723    3C                    INC     A
1724    E6 07                 AND     7               ;Really?
1726    20 0D                 JR      NZ,DWNC10       ;No
1728    18 2F                 JR      ONBRDR          ;Yes, set carry and return
                                                      ;
172A                  DOWNC:
                      ;
                      ; DOWNC - move 1 pixel down
                      ;
172A    E5                    PUSH    HL
172B    D5                    PUSH    DE
172C    2A F92A               LD      HL,(CLOC)
172F    CD 15D9               CALL    CHKMOD
1732    C2 17DC               JP      NZ,MDNC
1735                  DWNC10:
1735    23                    INC     HL              ;move down 1 pixel
1736    7D                    LD      A,L             ;Prepare for boundary check
1737    11 00F8               LD      DE,0F8H         ;Load possible offset to new location
173A    18 31                 JR      VRTMOV          ;Check
173C                  TUPC:
                      ;
                      ; TUPC - move 1 pixel up.
                      ; Return carry set if already on screen border.
                      ;
```

```
173C    E5                       PUSH    HL
173D    D5                       PUSH    DE
173E    2A F92A                  LD      HL,(CLOC)
1741    CD 15D9                  CALL    CHKMOD
1744    C2 17E3                  JP      NZ,MTUPC
1747    E5                       PUSH    HL
1748    2A F3CB                  LD      HL,(GRPCGP)
174B    11 0100                  LD      DE,0100H
174E    19                       ADD     HL,DE
174F    EB                       EX      DE,HL
1750    E1                       POP     HL
1751    E7                       RST     20H             ;Test [HL] with [DE]
                                                         ;Looks like on border?
1752    30 14                    JR      NC,UPC10        ;No
1754    7D                       LD      A,L             ;Possibly on border
1755    E6 07                    AND     7               ;Really?
1757    20 0F                    JR      NZ,UPC10        ;No
1759                     ONBRDR:
1759    D1                       POP     DE
175A                     ONBRD1:
175A    37                       SCF                     ;Set carry indicating we're on border
175B    E1                       POP     HL
175C    C9                       RET
175D                     UPC:
                         ;
                         ; UPC - move 1 pixel up
                         ;
175D    E5                       PUSH    HL
175E    D5                       PUSH    DE
175F    2A F92A                  LD      HL,(CLOC)       ;get current position
1762    CD 15D9                  CALL    CHKMOD
```

```
1765    C2 17F8                 JP      NZ,MUPC
1768                    UPC10:
1768    7D                      LD      A,L             ;Prepare for boundary check
1769    2B                      DEC     HL              ;move up 1 pixel
176A    11 FF08                 LD      DE,0FF08H       ;Load possible offset to new location
176D                    VRTMOV:
176D    E6 07                   AND     7               ;Crossed boundary?
176F    20 01                   JR      NZ,VRTMV1       ;No, it's okay
1771    19                      ADD     HL,DE           ;Get new location
1772                    VRTMV1:
1772    22 F92A                 LD      (CLOC),HL       ;Update pattern address
1775    A7                      AND     A               ;Clear carry
1776    D1                      POP     DE
1777    E1                      POP     HL
1778    C9                      RET
1779                    MTRGT:
                        ;
                        ; Graphics cursor movement in multi-color mode
                        ; [ Horizontal movements ]
                        ;
1779    CD 1639                 CALL    FETCHC
177C    A7                      AND     A
177D    3E 0F                   LD      A,0FH           ;Assume CMASK is even
177F    FA 17C0                 JP      M,MHZMV1        ;Within byte, just change CMASK
1782    7D                      LD      A,L
1783    E6 F8                   AND     0F8H
1785    FE F8                   CP      0F8H            ;On right edge?
1787    20 0B                   JR      NZ,MRGTC1       ;No, move to next pixel
1789    18 CF                   JR      ONBRD1          ;We're on right edge, set carry and return
178B                    MRGTC:
                        ;
```

```
178B    CD 1639                       CALL    FETCHC
178E    A7                            AND     A
178F    3E 0F                         LD      A,0FH           ;Assume CMASK is even
1791    FA 17C0                       JP      M,MHZMV1        ;Good assumption
1794                          MRGTC1:
1794    D5                            PUSH    DE
1795    11 0008                       LD      DE,8            ;Next pixel is 8 byte far
                                                              ;from the current position
1798    3E F0                         LD      A,0F0H
179A    18 1F                         JR      MHCMOV
179C                          MTLFT:
                              ;
179C    CD 1639                       CALL    FETCHC
179F    A7                            AND     A
17A0    3E F0                         LD      A,0F0H          ;Assume CMASK is odd
17A2    F2 17C0                       JP      P,MHZMV1        ;Good assumption, just change CMASK
17A5    7D                            LD      A,L
17A6    E6 F8                         AND     0F8H            ;On left edge?
17A8    20 0B                         JR      NZ,MLFTC1       ;No
17AA    18 AE                         JR      ONBRD1          ;We're on left edge, set carry and return
17AC                          MLFTC:
                              ;
17AC    CD 1639                       CALL    FETCHC
17AF    A7                            AND     A
17B0    3E F0                         LD      A,0F0H          ;Assume CMASK is odd
17B2    F2 17C0                       JP      P,MHZMV1        ;Good assumption, just change CMASK
17B5                          MLFTC1:
17B5    D5                            PUSH    DE
17B6    11 FFF8                       LD      DE,0FFF8H
17B9    3E 0F                         LD      A,0FH
17BB                          MHCMOV:
```

```
17BB    19                        ADD     HL,DE
17BC    22 F92A                   LD      (CLOC),HL
17BF    D1                        POP     DE
17C0                      MHZMV1:
17C0    32 F92C                   LD      (CMASK),A
17C3    A7                        AND     A              ;Clear carry
17C4    E1                        POP     HL
17C5    C9                        RET
17C6                      MTDNC:
                          ;
                          ; [ Vertical movements ]
                          ;
17C6    E5                        PUSH    HL
17C7    2A F3D5                   LD      HL,(MLTCGP)
17CA    11 0500                   LD      DE,0500H
17CD    19                        ADD     HL,DE
17CE    E1                        POP     HL
17CF    E7                        RST     20H            ;Possibly on border?
17D0    38 0A                     JR      C,MDNC         ;No
17D2    7D                        LD      A,L            ;Check if least 3 bits are all 1's
17D3    3C                        INC     A
17D4    E6 07                     AND     7
17D6    20 04                     JR      NZ,MDNC        ;No
17D8    37                        SCF                    ;We are at the bottom border,
                                                         ;set carry and return
17D9    D1                        POP     DE
17DA    E1                        POP     HL
17DB    C9                        RET
17DC                      MDNC:
                          ;
17DC    23                        INC     HL             ;Move down 1 byte
```

```
17DD    7D                      LD      A,L
17DE    11 00F8                 LD      DE,0F8H         ;Load possible offset to next block
17E1    18 1A                   JR      MVTMOV          ;Check
17E3                    MTUPC:
                        ;
17E3    E5                      PUSH    HL
17E4    2A F3D5                 LD      HL,(MLTCGP)
17E7    11 0100                 LD      DE,0100H        ;Possibly on border?
17EA    19                      ADD     HL,DE
17EB    E1                      POP     HL
17EC    E7                      RST     20H             ;Test [HL] with [DE]
17ED    30 09                   JR      NC,MUPC         ;No
17EF    7D                      LD      A,L             ;Check if we're top of a block
17F0    E6 07                   AND     7
17F2    20 04                   JR      NZ,MUPC         ;No
17F4    37                      SCF                     ;We're on top border, set carry and return
17F5    D1                      POP     DE
17F6    E1                      POP     HL
17F7    C9                      RET
17F8                    MUPC:
                        ;
17F8    7D                      LD      A,L
17F9    2B                      DEC     HL              ;Move up 1 byte
17FA    11 FF08                 LD      DE,0FF08H       ;Load possible offset to next block
17FD                    MVTMOV:
17FD    E6 07                   AND     7               ;Wrapped to next block?
17FF    20 01                   JR      NZ,MVTMV1       ;No
1801    19                      ADD     HL,DE           ;Yes, add up offset to next block
1802                    MVTMV1:
1802    22 F92A                 LD      (CLOC),HL
1805    A7                      AND     A               ;Clear carry
```

```
 5050       1806    D1                  POP     DE
 5051       1807    E1                  POP     HL
 5052       1808    C9                  RET
 5053                               SUBTTL -MSXGRP-  (Box fill and Misc.)
```

```

1809                  NSETCX:
                      ;
                      ; NSETCX - Performs SETC, RIGHTC [HL] times
                      ;
                      ; In fact, SETC and RIGHTC are never called to increase speed,
                      ; and for the reason described below.
                      ;
                      ; Since only 2 colors can be displayed in a byte, some special
                      ; handling is required when a full-byte is set when writing left
                      ; or right extras. In this case, we can completely ignore the
                      ; background color for that byte, allowing 2 colors displayed
                      ; in a byte.
                      ;
                      ; All registers may be destroyed.
                      ;
1809    CD 15D9               CALL    CHKMOD
180C    C2 18BB               JP      NZ,MNSTCX       ;Multi-color mode
180F    E5                    PUSH    HL              ;Save count
1810    CD 1639               CALL    FETCHC          ;Get CLOC and CMASK
1813    E3                    EX      (SP),HL         ;Reget count, save CLOC
1814    87                    ADD     A,A             ;Beginig at leftmost position?
1815    38 18                 JR      C,NSTC20        ;Yes, no extra dots at the left
1817    F5                    PUSH    AF              ;Save mask pattern*2
1818    01 FFFF               LD      BC,0FFFFH
181B    0F                    RRCA
181C                  NSTC10:
181C    09                    ADD     HL,BC           ;Decrement pixel count
181D    30 45                 JR      NC,NSTCSP       ;The whole dots are within a byte
181F    0F                    RRCA
1820    30 FA                 JR      NC,NSTC10
```

```
1822    F1                    POP     AF            ;Restore mask pattern*2
1823    3D                    DEC     A             ;Form left-extra pattern
1824    E3                    EX      (SP),HL       ;Reget CLOC, save count
1825    E5                    PUSH    HL            ;Save CLOC
1826    CD 186C               CALL    PATWRT        ;Write to VRAM (pattern and color)
1829    E1                    POP     HL            ;Restore CLOC
182A    11 0008               LD      DE,8          ;Load an offset to next byte
182D    19                    ADD     HL,DE         ;Update pattern address
182E    E3                    EX      (SP),HL       ;Reget count, save CLOC
182F                NSTC20:
182F    7D                    LD      A,L           ;Get low byte of count
1830    E6 07                 AND     7             ;[A]=count mod 8
1832    4F                    LD      C,A           ;save count after byte boundary
1833    7C                    LD      A,H
1834    0F                    RRCA
1835    7D                    LD      A,L
1836    1F                    RRA
1837    0F                    RRCA
1838    0F                    RRCA                  ;[HL]=[HL]/8
1839    E6 3F                 AND     00111111B
183B    E1                    POP     HL            ;Reget CLOC
183C    47                    LD      B,A           ;[B]=counter
183D    28 14                 JR      Z,NSTC40      ;No dots in this part
183F                NSTC30:
183F    AF                    XOR     A             ;Make specified color a background color
1840    CD 07CD               CALL    WRTVRM        ;Write to VRAM (pattern)
1843    11 2000               LD      DE,GRPDIF
1846    19                    ADD     HL,DE         ;Calculate address of color table
1847    3A F3F2               LD      A,(ATRBYT)    ;Get specified color
184A    CD 07CD               CALL    WRTVRM        ;Write to VRAM (color)
184D    11 2008               LD      DE,GRPDIF+8   ;Load an offset to next byte
```

```
1850    19                      ADD     HL,DE           ;Bump CLOC
1851    10 EC                   DJNZ    NSTC30          ;Loop until done
1853                    NSTC40:
1853    0D                      DEC     C               ;dot count in char boundary
1854    F8                      RET     M               ;No dots in right extra
1855    E5                      PUSH    HL              ;Save CLOC
1856    21 185D                 LD      HL,RGTEXT       ;Load address for 'right-extra' pattern table
1859    09                      ADD     HL,BC
185A    7E                      LD      A,(HL)          ;Get pattern
185B    18 0E                   JR      NSTC50
185D                    RGTEXT:
                        ;
185D    80 C0 E0 F0             DB      80H,0C0H,0E0H,0F0H
1861    F8 FC FE                DB      0F8H,0FCH,0FEH
1864                    NSTCSP:
                        ;
1864    87                      ADD     A,A             ;Get mask pattern for the right (11111100)
1865    3D                      DEC     A
1866    2F                      CPL
1867    47                      LD      B,A             ;Save it
1868    F1                      POP     AF              ;Get mask pattern for the left (00011111)
1869    3D                      DEC     A
186A    A0                      AND     B               ;Make a pattern to write (00011100)
186B                    NSTC50:
186B    E1                      POP     HL              ;Restore CLOC ex.
```

```

186C                    PATWRT:
                        ;
                        ; PATWRT - Write a pattern to high-resolution screen
                        ;
                        ; Entry: A - Pattern to be written
                        ;       HL - Address of pattern table
                        ;   ATRBYT - Color of this pattern
                        ;
186C    47                      LD      B,A             ;Save pattern to be added
186D    CD 07D7                 CALL    RDVRM           ;Read VRAM (pattern)
1870    4F                      LD      C,A             ;Save current pattern
1871    11 2000                 LD      DE,GRPDIF
1874    19                      ADD     HL,DE           ;Form address of color table
1875    CD 07D7                 CALL    RDVRM           ;Read from VRAM (color)
1878    F5                      PUSH    AF
1879    E6 0F                   AND     0FH             ;Extract background color
187B    5F                      LD      E,A             ;Save background color
187C    F1                      POP     AF              ;Restore foreground and background color
187D    93                      SUB     E
187E    57                      LD      D,A             ;Set foreground color in the upper 4 bit
                                                        ;[B] has the specified pattern,
                                                        ;[C] has the current pattern,
                                                        ;[D] has the current foreground color
                                                        ;    shifted left 4 times,
                                                        ;[E] has the current background color,
                                                        ;[HL] has the address of color table.
187F    3A F3F2                 LD      A,(ATRBYT)      ;Get specified color
1882    BB                      CP      E               ;Same with current background?
1883    28 19                   JR      Z,SAMEBG        ;Yes
1885    87                      ADD     A,A
```

```
1886    87                      ADD     A,A
1887    87                      ADD     A,A
1888    87                      ADD     A,A
1889    BA                      CP      D               ;Same with current foreground?
188A    28 16                   JR      Z,SAMEFG        ;Yes
188C    F5                      PUSH    AF              ;Save new foreground color
188D    78                      LD      A,B
188E    B1                      OR      C
188F    FE FF                   CP      0FFH            ;All pixels are going to be set?
1891    28 17                   JR      Z,PATWR1        ;Yes, Spock will use a new repair technique
                                                        ;logically...
1893    E5                      PUSH    HL              ;Save address of color table
1894    D5                      PUSH    DE              ;Save current background color
1895    CD 18A2                 CALL    SAMEFG          ;Write to VRAM (pattern)
1898    D1                      POP     DE              ;Restore current background in [E]
1899    E1                      POP     HL              ;Restore color table address
189A    F1                      POP     AF              ;Restore new foreground color in upper
                                                        ;4 bits of [Acc]
189B    B3                      OR      E               ;Form new foreground and background color
189C    18 1A                   JR      JMPWRT          ;Write to color table
189E            SAMEBG:
                ;
189E    78                      LD      A,B
189F    2F                      CPL
18A0    A1                      AND     C
18A1    11                      DB      11H             ;Skip next 2 bytes (LXI D)
18A2            SAMEFG:
18A2    78                      LD      A,B
18A3    B1                      OR      C
18A4            WTPTAB:
18A4    11 2000                 LD      DE,GRPDIF
```

```
5203    18A7    19                      ADD     HL,DE
5204    18A8    18 0E                   JR      JMPWRT          ;Write to pattern table
5205    18AA            PATWR1:
5206                    ;
5207    18AA    F1                      POP     AF              ;Discard new foreground color
5208    18AB    78                      LD      A,B             ;Reget specified pattern
5209    18AC    2F                      CPL                     ;Forget current background color, 'cause
5210    18AD    E5                      PUSH    HL              ;there's no background, we display
5211    18AE    D5                      PUSH    DE              ;new pattern as background color.
5212    18AF    CD 18A4                 CALL    WTPTAB          ;Write to pattern table
5213    18B2    D1                      POP     DE
5214    18B3    E1                      POP     HL
5215    18B4    3A F3F2                 LD      A,(ATRBYT)      ;Get new color (this will be the
5216                                                            ;background color)
5217    18B7    B2                      OR      D               ;Add current foreground color
5218    18B8            JMPWRT:
5219    18B8    C3 07CD                 JP      WRTVRM          ;Write to VRAM (color)
```

```
        18BB                    MNSTCX:
                                ;
                                ; NSETCX for multicolor screen
                                ;
        18BB    E5                      PUSH    HL              ;Save counter
        18BC    CD 167E                 CALL    SETC            ;Set pixel
        18BF    CD 16C5                 CALL    RIGHTC          ;Move to right
        18C2    E1                      POP     HL              ;Restore counter
        18C3    2D                      DEC     L
        18C4    20 F5                   JR      NZ,MNSTCX
        18C6    C9                      RET
        18C7                    GTASPC:
                                ;
                                ; GTASPC - load aspect ratio for CIRCLE
                                ;
        18C7    2A F40B                 LD      HL,(ASCPCT1)
        18CA    EB                      EX      DE,HL
        18CB    2A F40D                 LD      HL,(ASCPCT2)
        18CE    C9                      RET
                                SUBTTL -MSXGRP -  (Routines for paint)
```

```

18CF                 PNTINI:
                     ;
                     ; PNTINI - Initialize border color
                     ;
18CF    F5                   PUSH    AF              ;Save specified color
18D0    CD 15D9              CALL    CHKMOD          ;In what mode are we now?
18D3    28 06                JR      Z,PNTHRS        ;High-resolution mode
18D5    F1                   POP     AF
18D6    FE 10                CP      10H             ;Legal value?
18D8    3F                   CCF                     ;Carry means illegal
18D9    18 05                JR      PNTIRT
18DB                 PNTHRS:
                     ;
18DB    F1                   POP     AF              ;Discard specified color
18DC    3A F3F2              LD      A,(ATRBYT)      ;Always ignore specified border
18DF    A7                   AND     A               ;Always legal
18E0                 PNTIRT:
18E0    32 FCB2              LD      (BRDATR),A      ;Set border color
18E3    C9                   RET                     ;Return with the condition
18E4                 SCANR:
                     ;
                     ; SCANR - scan pixels to right
                     ; Maximum number of pixels to test is passed in [DE].
                     ;
18E4    21 0000              LD      HL,0            ;Initialize PNTCNT
18E7    4D                   LD      C,L             ;Initialize PNTDFL
18E8    CD 15D9              CALL    CHKMOD          ;Check current screen mode
18EB    20 64                JR      NZ,MSCANR       ;Multi-color mode
                     ;
                     ; Scan to right in high-resolution mode
```

```
                        ; [B] set to 0 is need to suspend painting, 1 otherwise.
                        ;
                        ;       Work1 = Temporary storage for 'suspend painting'
                        ;       Work2 = Save area for pixel count to draw right
                        ;       Work3 = Save area for 'pixel changed' flag
                        ;
18ED    78                      LD      A,B
18EE    32 F866                 LD      (RUNFLG),A      ;Remember to suspend or not
18F1    AF                      XOR     A               ;Clear 'pixel changed' flag
18F2    32 F869                 LD      (WORK3),A
18F5    3A FCB2                 LD      A,(BRDATR)
18F8    47                      LD      B,A             ;Set border color to [B] for comparison
18F9                    SCANR1:
18F9    CD 1647                 CALL    READC           ;Read current color
18FC    B8                      CP      B               ;Still on border?
18FD    20 0D                   JR      NZ,SCANR2       ;No, start painting
18FF    1B                      DEC     DE              ;All pixels tested?
1900    7A                      LD      A,D
1901    B3                      OR      E
1902    C8                      RET     Z               ;Yes
1903    CD 16AC                 CALL    TRIGHT          ;Advance to right, and check if out of screen
1906    30 F1                   JR      NC,SCANR1       ;Not yet out of screen, continue
1908    11 0000                 LD      DE,0            ;All pixels has border attribute on
190B    C9                      RET                     ;this row, let BRDCNT be 0, and return
190C                    SCANR2:
                        ;
                        ; A pixel with non-border attribute is found. Start painting
                        ;
190C    CD 19AE                 CALL    CHKCHG          ;Check if pixel changed
190F    D5                      PUSH    DE              ;Save BRDCNT
1910    CD 1639                 CALL    FETCHC          ;Get current CLOC, CMASK
```

```
1913    22 F942               LD      (CSAVEA),HL     ;Set first non-border pixel encountered
1916    32 F944               LD      (CSAVEM),A
1919    11 0000               LD      DE,0            ;Initialize # of painted pixels (PNTCNT)
191C                SCANR3:
191C    13                    INC     DE              ;Update PNTCNT
191D    CD 16AC               CALL    TRIGHT          ;Move 1 pixel right
1920    38 0B                 JR      C,SCANR4        ;Out of screen
1922    CD 1647               CALL    READC           ;Read color of current pixel
1925    B8                    CP      B               ;Reached border?
1926    28 05                 JR      Z,SCANR4        ;Yes
1928    CD 19AE               CALL    CHKCHG          ;Check if pixel changed
192B    18 EF                 JR      SCANR3          ;Keep on scaning
192D                SCANR4:
                    ;
192D    D5                    PUSH    DE              ;Save PNTCNT
192E    CD 1639               CALL    FETCHC          ;Since NSETCX does not update 'C', these value
1931    E5                    PUSH    HL              ; must be saved
1932    F5                    PUSH    AF
1933    2A F942               LD      HL,(CSAVEA)     ;Set where to start painting
1936    3A F944               LD      A,(CSAVEM)
1939    CD 1640               CALL    STOREC          ;Set CLOC and CMASK
193C    EB                    EX      DE,HL           ;Set length of line to [HL] (PNTCNT)
193D    22 F867               LD      (WORK2),HL
1940    3A F866               LD      A,(WORK1)       ;Same as [RUNFLG]
1943    A7                    AND     A
1944    C4 1809               CALL    NZ,NSETCX       ;Draw [HL] pixels to the right if not suspend
1947    F1                    POP     AF              ;Restore 'last-examined-pixel' information
1948    E1                    POP     HL
1949    CD 1640               CALL    STOREC
194C    E1                    POP     HL              ;Restore PNTCNT
194D    D1                    POP     DE              ;Restore BRDCNT
```

```
 5334      194E     C3 19A9               JP        SCANL4
```

```

1951                      MSCANR:
                          ;
                          ; Scan to right in multi-color mode
                          ;
1951    CD 19C7                   CALL    MTSBRD          ;Is it border color?
1954    30 0D                     JR      NC,MSCNR1       ;No, start painting
1956    1B                        DEC     DE              ;All pixels tested?
1957    7A                        LD      A,D
1958    B3                        OR      E
1959    C8                        RET     Z               ;Yes
195A    CD 16AC                   CALL    TRIGHT          ;Advance to right, and check if out of screen
195D    30 F2                     JR      NC,MSCANR       ;Not yet out of screen, continue
195F    11 0000                   LD      DE,0            ;Out of screen, let BRDCNT be 0, and return
1962    C9                        RET
1963                      MSCNR1:
                          ;
1963    CD 1639                   CALL    FETCHC          ;Get CLOC,CMASK
1966    22 F942                   LD      (CSAVEA),HL     ;Save VRAM address
1969    32 F944                   LD      (CSAVEM),A      ;Save mask pattern
196C    21 0000                   LD      HL,0            ;Initialize PNTCNT
196F                      MSCNR2:
196F    23                        INC     HL              ;Increment PNTCNT
1970    CD 16AC                   CALL    TRIGHT          ;Advance to right, and check if out of screen
1973    D8                        RET     C               ;Going out of screen
1974    CD 19C7                   CALL    MTSBRD          ;Reached border color?
1977    30 F6                     JR      NC,MSCNR2       ;Not yet, continue
1979    C9                        RET
```

```
                               
        197A                   SCANL:
                               ;
                               ; SCANL - Scan pixels to left
                               ;
        197A    21 0000                LD      HL,0            ;Initialize PNTCNT
        197D    4D                     LD      C,L             ;Initialize PNTDFL
        197E    CD 15D9                CALL    CHKMOD          ;Check current screen mode
        1981    20 37                  JR      NZ,MSCANL       ;Multi-color mode
                               ;
                               ; Scan to left in high-resolution mode
                               ;
        1983    AF                     XOR     A               ;Clear 'pixel changed' flag
        1984    32 F869                LD      (WORK3),A
        1987    3A FCB2                LD      A,(BRDATR)
        198A    47                     LD      B,A             ;Set border color to [B] for comparison
        198B                   SCANL1:
        198B    CD 16D8                CALL    TLEFT           ;Advance to left, and check if out of screen
        198E    38 0F                  JR      C,SCANL3        ;On left edge
        1990    CD 1647                CALL    READC           ;Read color of target pixel
        1993    B8                     CP      B               ;Reached border?
        1994    28 06                  JR      Z,SCANL2        ;Yes
        1996    CD 19AE                CALL    CHKCHG          ;Check if pixel changed
        1999    23                     INC     HL              ;Update PNTCNT
        199A    18 EF                  JR      SCANL1
        199C                   SCANL2:
                               ;
        199C    CD 16C5                CALL    RIGHTC          ;'C' must specify 'last pixel painted'
        199F                   SCANL3:
        199F    E5                     PUSH    HL              ;Save PNTCNT
        19A0    ED 5B F867             LD      DE,(WORK2)      ;Load suspended pixels which remain
```

```
19A4   19                  ADD    HL,DE         ;to the right
19A5   CD 1809             CALL   NSETCX        ;Draw [HL] pixel from current 'C'
19A8   E1                  POP    HL            ;Restore PNTCNT
19A9             SCANL4:
19A9   3A F869             LD     A,(WORK3)     ;Non 0 if pixels changed attribute
19AC   4F                  LD     C,A
19AD   C9                  RET
19AE             CHKCHG:
                 ;
19AE   E5                  PUSH   HL
19AF   21 F3F2             LD     HL,ATRBYT     ;Get specified paint attribute
19B2   BE                  CP     (HL)          ;Same?
19B3   E1                  POP    HL
19B4   C8                  RET    Z             ;Yes, no change of attribute
19B5   3C                  INC    A             ;Load non 0 to [Acc]
19B6   32 F869             LD     (WORK3),A     ;Remember this temporarily
19B9   C9                  RET
19BA             MSCANL:
                 ;
                 ; Scan to left in multi-color mode
                 ;
19BA   CD 16D8             CALL   TLEFT         ;Advance to left, and check if out of screen
19BD   D8                  RET    C             ;going out of screen
19BE   CD 19C7             CALL   MTSBRD        ;Reached border color?
19C1   DA 16C5             JP     C,RIGHTC      ;Yes, adjust CLOC, CMASK and return
19C4   23                  INC    HL            ;Increment PNTCNT
19C5   18 F3               JR     MSCANL        ;Continue
19C7             MTSBRD:
                 ;
                 ; Test border subroutine for multi-color mode
                 ;
```

```
19C7    CD 1647              CALL    READC           ;Get the color of target pixel
19CA    47                   LD      B,A
19CB    3A FCB2              LD      A,(BRDATR)      ;Load specified border color
19CE    90                   SUB     B               ;Reached border?
19CF    37                   SCF                     ;Assume so
19D0    C8                   RET     Z               ;Yes, return with carry flag set
19D1    3A F3F2              LD      A,(ATRBYT)      ;Is current pixel same as ATRBYT?
19D4    B8                   CP      B
19D5    C8                   RET     Z               ;Yes, no changes made.
                                                     ;Return with carry reset
19D6    CD 167E              CALL    SETC            ;Set this pixel to ATRBYT
19D9    0E 01                LD      C,1             ;Set 'pixel-changed' flag
19DB    A7                   AND     A               ;Tell caller that we plot a dot
19DC    C9                   RET
                     SUBTTL -CASET- Cassette drivers stuff
```

```
                              ; Cassette read/write stuff
                              ;
                              ; Following driver assumes that T cycle is 279.365 nS
                              ;
                              ;  Variables referenced
                              ;       PPI.CM          To write to cassette
                              ;       PSG.DR          To read from casette
                              ;       BREAKX          Routine to check for [STOP] key pressed
                              ;
         19DD                 TAPOFF:
                              ;
         19DD    C5                   PUSH    BC
         19DE    F5                   PUSH    AF
         19DF    01 0000              LD      BC,0
         19E2                 CTWOF1:
         19E2    0B                   DEC     BC
         19E3    78                   LD      A,B             ;Test BC
         19E4    B1                   OR      C
         19E5    20 FB                JR      NZ,CTWOF1
         19E7    F1                   POP     AF
         19E8    C1                   POP     BC
         19E9                 TAPIOF:
         19E9    F5                   PUSH    AF
         19EA    3E 09                LD      A,00001001B     ;Stop motor
         19EC    D3 AB                OUT     (PPI.CM),A
         19EE    F1                   POP     AF
         19EF    FB                   EI
         19F0    C9                   RET
         19F1                 TAPOON:
                              ;
```

```
                                ; Write out header, if [A]=0 then write short header
                                ; otherwise write long header ( 5sec)
                                ;
        19F1    B7                      OR      A               ;set flag for length of header
        19F2    F5                      PUSH    AF              ;save flag
        19F3    3E 08                   LD      A,8             ;Motor on
        19F5    D3 AB                   OUT     (PPI.CM),A
        19F7    21 0000                 LD      HL,0
        19FA                    MOTRWT:
        19FA    2B                      DEC     HL
        19FB    7C                      LD      A,H
        19FC    B5                      OR      L
        19FD    20 FB                   JR      NZ,MOTRWT       ;wait till motor starts
        19FF    F1                      POP     AF              ;get back header length flag
        1A00    3A F40A                 LD      A,(HEADER)      ;get length of header
        1A03    28 02                   JR      Z,SYNCW1        ;short header
        1A05    87                      ADD     A,A
        1A06    87                      ADD     A,A
        1A07                    SYNCW1:
        1A07    47                      LD      B,A
        1A08    0E 00                   LD      C,0             ;set up counter
        1A0A    F3                      DI                      ;Don't disturb during writing to cassette
        1A0B                    SYNLP1:
        1A0B    CD 1A4D                 CALL    BIT1OT          ;Write enough marks
        1A0E    CD 1A3F                 CALL    RETRET          ;compensate overhead
        1A11    0B                      DEC     BC
        1A12    78                      LD      A,B
        1A13    B1                      OR      C
        1A14    20 F5                   JR      NZ,SYNLP1       ;loop till counter exhausts
        1A16    C3 046F                 JP      BREAKX          ;check control-stop and return
        1A19                    TAPOUT:
```

```
1A19                DATAW:
                    ;
                    ; Output a byte
                    ;
1A19    2A F406             LD      HL,(LOW)        ;get time constants for space
1A1C    F5                  PUSH    AF
1A1D    7D                  LD      A,L
1A1E    D6 0E               SUB     0EH             ;compensate loss time since last stop bit
1A20    6F                  LD      L,A
1A21    CD 1A50             CALL    BITOUT          ;output start bit
1A24    F1                  POP     AF
1A25    06 08               LD      B,8             ;Initialize counter
1A27                DATAWL:
1A27    0F                  RRCA                    ;next bit to carry
1A28    DC 1A40             CALL    C,BIT1          ;output mark if the bit is 1
1A2B    D4 1A39             CALL    NC,BIT0         ;Output space
1A2E    10 F7               DJNZ    DATAWL          ;Loop until 8 bits sent
1A30    CD 1A40             CALL    BIT1            ;Output stop bit
1A33    CD 1A40             CALL    BIT1
1A36    C3 046F             JP      BREAKX          ;Check if break pressed and return
```

```
         
1A39                    BIT0:
                        ;
                        ; Output a bit to cassette
                        ;
                        ; Absolute jumps are used to improve accuracy
                        ;
1A39    2A F406                 LD      HL,(LOW)        ;Output 0 (space)       (17 T)
1A3C    CD 1A50                 CALL    BITOUT          ;                       (18 T)
1A3F                    RETRET:
1A3F    C9                      RET                     ;                       (11 T)
1A40                    BIT1:
                        ;
1A40    CD 1A4D                 CALL    BIT1OT          ;                       (18 T)
1A43    E3                      EX      (SP),HL         ;                       (20 T)
1A44    E3                      EX      (SP),HL         ;compensate overhead    (20 T)
1A45    00                      NOP                     ;(Total 60 state)       ( 5 T)
1A46    00                      NOP                     ;                       ( 5 T)
1A47    00                      NOP                     ;                       ( 5 T)
1A48    00                      NOP                     ;                       ( 5 T)
1A49    CD 1A4D                 CALL    BIT1OT          ;To compensate time     (18 T)
1A4C    C9                      RET                     ;Don't change this      (11 T)
1A4D                    BIT1OT:
                        ;
                        ; output a single cycle
                        ;
                        ; Total number of states =16 x [L] + 16 x [H] + 71
                        ;                        =4.47uS x [L] + 4.47uS x [H] + 19.8usec
                        ;
1A4D    2A F408                 LD      HL,(HIGH)       ;                       (17 T)
1A50                    BITOUT:
```

```
1A50    F5                 PUSH    AF              ;                       (12 T)
                   ;
1A51               KEEPL:
1A51    2D                 DEC     L               ;Keep low level         ( 5 T)
1A52    C2 1A51            JP      NZ,KEEPL        ;                       (11 T)
1A55    3E 0B              LD      A,0BH           ;                       ( 8 T)
1A57    D3 AB              OUT     (PPI.CM),A      ;Output high level      (11 T)
1A59               KEEPH:
1A59    25                 DEC     H               ;keep high level        ( 5 T)
1A5A    C2 1A59            JP      NZ,KEEPH        ;                       (11 T)
1A5D    3E 0A              LD      A,0AH           ;                       ( 8 T)
1A5F    D3 AB              OUT     (PPI.CM),A      ;Output low level       (11 T)
1A61    F1                 POP     AF              ;Restore data           (12 T)
                   ;
1A62    C9                 RET                     ;                       (11 T)
1A63               TAPION:
                   ;
                   ;  Detect header block
                   ;
1A63    3E 08              LD      A,8             ;Motor on
1A65    D3 AB              OUT     (PPI.CM),A
1A67    F3                 DI
1A68    3E 0E              LD      A,0EH           ;Select PSG port A
1A6A    D3 A0              OUT     (PSG.LW),A
1A6C               SYN05:
                   ;
                   ; First, wait until a series of good pulses are found.
                   ;
1A6C    21 0457            LD      HL,0457H        ;Initialize counter
                                                   ;Number of pulse to detect header
1A6F               SYN10:
```

```
1A6F    51                      LD      D,C             ;Remember last value
1A70    CD 1B34                 CALL    CNTFUL          ;Count full cycle
1A73    D8                      RET     C               ;Aborted
1A74    79                      LD      A,C             ;Get count
1A75    FE DE                   CP      0DEH            ;0DE = Max count
1A77    30 F3                   JR      NC,SYN05        ;Too long, reset number of pulses
1A79    FE 05                   CP      5               ;5 = Min count
1A7B    38 EF                   JR      C,SYN05         ;Too short, reset number of pulses
                        ;
                        ; Now compare with last pulse width and approve this as a good pulse
                        ; if this is similar to last one.
                        ;
1A7D    92                      SUB     D               ;current - last
1A7E    30 02                   JR      NC,SYN11
1A80    2F                      CPL                     ;result was negative, negate it
1A81    3C                      INC     A
1A82                    SYN11:
1A82    FE 04                   CP      4               ;within a wow allowance?
1A84    30 E6                   JR      NC,SYN05        ;no, reset number of pulse ever seen
1A86    2B                      DEC     HL
1A87    7C                      LD      A,H
1A88    B5                      OR      L
1A89    20 E4                   JR      NZ,SYN10        ;Loop till seen enough good pulses
                        ;
1A8B                    SYN20:
                        ;
                        ; Next, calculate the mean width of pulse.
                        ;
1A8B    21 0000                 LD      HL,0            ;Initialize sum
1A8E    45                      LD      B,L             ;Initialize high byte of [BC] pair
1A8F    55                      LD      D,L             ;Loop 256 times
```

```
1A90                    SYN30:
1A90    CD 1B34                 CALL    CNTFUL
1A93    D8                      RET     C
1A94    09                      ADD     HL,BC
1A95    15                      DEC     D
1A96    C2 1A90                 JP      NZ,SYN30
1A99    01 06AE                 LD      BC,06AEH        ;compensate over head
1A9C    09                      ADD     HL,BC
                        ;
                        ; Set various values for read routine. Those are,
                        ;
                        ; LOWLIM - lower limit of the width of start bit. [H]*1.5
                        ; WINWID - width of window to count the transition.
                        ;
1A9D    7C                      LD      A,H             ;[H] has mean pulse width
1A9E    1F                      RRA
1A9F    E6 7F                   AND     7FH
1AA1    57                      LD      D,A             ;[D]=[mean]/2
1AA2    29                      ADD     HL,HL
1AA3    7C                      LD      A,H             ;[A]=[mean]x2
1AA4    92                      SUB     D               ;[A]=[mean]x1.5
1AA5    57                      LD      D,A             ;save
1AA6    D6 06                   SUB     6               ;compensate overhead at DATAR
1AA8    32 FCA4                 LD      (LOWLIM),A
                        ;
                        ; Set width of window 'WINWID'
                        ; CNTFUL takes 40T for a loop, RDBIT takes 60T for loop
                        ; set WINWID as 3 times wider than single short pulse ([mean]/2)
                        ; [WINWID]=[mean] x 1.5 x 40/60
                        ;         =[D] x 2/3
                        ;
```

```
1AAB    7A                  LD      A,D             ;get [mean width]x1.75
1AAC    87                  ADD     A,A             ;x2
1AAD    06 00               LD      B,0             ;clear quotient
1AAF                SULOP:
1AAF    D6 03               SUB     3
1AB1    04                  INC     B
1AB2    30 FB               JR      NC,SULOP        ;loop till get carry
1AB4    78                  LD      A,B             ;[A]=[mean]x1.75x2/3
1AB5    D6 03               SUB     3               ;compensate overhead in RDBIT routine
1AB7    32 FCA5             LD      (WINWID),A
1ABA    B7                  OR      A
1ABB    C9                  RET
```

```

1ABC                TAPIN:
                    ;
                    ;  Read a byte from cassette
                    ;
1ABC    3A FCA4             LD      A,(LOWLIM)
1ABF    57                  LD      D,A             ;[D] has lower limit for start bit
1AC0                DATAR:
1AC0    CD 046F             CALL    BREAKX
1AC3    D8                  RET     C               ;Aborted
1AC4    DB A2               IN      A,(PSG.DR)      ;Get cassette
1AC6    07                  RLCA                    ;High state?
1AC7    30 F7               JR      NC,DATAR        ;No
1AC9                DATAR0:
1AC9    CD 046F             CALL    BREAKX
1ACC    D8                  RET     C               ;Aborted
1ACD    DB A2               IN      A,(PSG.DR)      ;Get cassette
1ACF    07                  RLCA                    ;falling egde?
1AD0    38 F7               JR      C,DATAR0        ;No
1AD2    1E 00               LD      E,0             ;Initialize edge mask
1AD4    CD 1B1F             CALL    CNTHLF          ;Get width in [C]
1AD7                DATAR1:
1AD7    41                  LD      B,C             ;Save old width
1AD8    CD 1B1F             CALL    CNTHLF          ;Get new width in [C]
1ADB    D8                  RET     C               ;aborted
1ADC    78                  LD      A,B             ;Add width of 2 pulses
1ADD    81                  ADD     A,C
1ADE    DA 1AD7             JP      C,DATAR1        ;Pulse too long
1AE1    BA                  CP      D               ;Longer than lower limit?
1AE2    38 F3               JR      C,DATAR1        ;No
                    ;
```

```
                                ; Now, a valid start bit has been found.
                                ; [E] =  0      if NORMAL polarity,
                                ;      =255     if REVERSE polarity.
                                ;
1AE4    2E 08                           LD      L,8             ;Initialize counter
1AE6                            DATARL:
1AE6    CD 1B03                         CALL    RDBIT
1AE9    FE 04                           CP      3+1             ;Legal transitions?
1AEB    3F                              CCF
1AEC    D8                              RET     C               ;Too many transitions
1AED    FE 02                           CP      2
1AEF    3F                              CCF                     ;Set carry if 2 or 3 transitions
1AF0    CB 1A                           RR      D
                                ;
                                ; We've just assembled a bit. A check must be done to make sure
                                ; that we're at the start of next bit field.
                                ;
1AF2    79                              LD      A,C             ;Reget number of transitions
1AF3    0F                              RRCA
1AF4    D4 1B23                         CALL    NC,CNTHL0       ;Wait for next transition if 0 or 2
1AF7    CD 1B1F                         CALL    CNTHLF
1AFA    2D                              DEC     L
1AFB    C2 1AE6                         JP      NZ,DATARL       ;Loop till done
1AFE    CD 046F                         CALL    BREAKX          ;return with carry set if breaked
1B01    7A                              LD      A,D
1B02    C9                              RET
1B03                            RDBIT:
                                ;
                                ; Count number of transitions within a period specified by 'WINWID'
                                ;
                                ; length of window = 17uSec x [WINWID] + 12.3 uSec
```

```
                     ;
                     ; [D],[H] and [L] are preserved.
                     ; [E] is updated to prepare for next edge
                     ;
1B03    3A FCA5              LD      A,(WINWID)      ;Get width of window
1B06    47                   LD      B,A
1B07    0E 00                LD      C,0             ;Clear # of transitions seen
1B09                 RDBITL:
1B09    DB A2                IN      A,(PSG.DR)      ;Get a bit
1B0B    AB                   XOR     E               ;Any changes?
1B0C    F2 1B17              JP      P,NOTRAN        ;No
1B0F    7B                   LD      A,E             ;Transition seen
1B10    2F                   CPL                     ;Prepare for next transition
1B11    5F                   LD      E,A
1B12    0C                   INC     C               ;Increment # of transitions
1B13    10 F4                DJNZ    RDBITL
1B15    79                   LD      A,C             ;Get transition count
1B16    C9                   RET
1B17                 NOTRAN:
                     ;
1B17    00                   NOP                     ;Compensate time
1B18    00                   NOP
1B19    00                   NOP
1B1A    00                   NOP
1B1B    10 EC                DJNZ    RDBITL
                     ;
1B1D    79                   LD      A,C             ;Get transition count
1B1E    C9                   RET
```

```

1B1F                    CNTHLF:
                        ;
                        ; Count half cycle
                        ;       1T    =279.4nS
                        ;       period=[C] x 11.18 + 35.48uS
                        ;
1B1F    CD 046F                 CALL    BREAKX          ;Break?                 (87 T)
1B22    D8                      RET     C               ;Yes, aborted           ( 6 T)
1B23                    CNTHL0:
1B23    0E 00                   LD      C,0             ;Initialize counter     ( 8 T)
1B25                    CNTHL1:
1B25    0C                      INC     C               ;# of state for this loop
                                                        ;40T=11.18usec          ( 5 T)
1B26    28 0A                   JR      Z,TIMOUT        ;Pulse too long         ( 8 T)
1B28    DB A2                   IN      A,(PSG.DR)      ;Read cassette          (11 T)
1B2A    AB                      XOR     E               ;Desired transition?    ( 5 T)
1B2B    F2 1B25                 JP      P,CNTHL1        ;No                     (11 T)
1B2E    7B                      LD      A,E             ;Complement edge mask   ( 5 T)
1B2F    2F                      CPL                     ;                       ( 5 T)
1B30    5F                      LD      E,A             ;                       ( 5 T)
1B31    C9                      RET                     ;                       (11 T)
1B32                    TIMOUT:
                        ;
1B32    0D                      DEC     C               ;Load 255
1B33    C9                      RET
1B34                    CNTFUL:
                        ;
                        ; Count full cycle
                        ;
1B34    CD 046F                 CALL    BREAKX
```

```
 5779     1B37     D8                    RET     C               ;Aborted
 5780     1B38     DB A2                 IN      A,(PSG.DR)      ;Get cassette
 5781     1B3A     07                    RLCA                    ;Low state?
 5782     1B3B     38 F7                 JR      C,CNTFUL        ;No
 5783     1B3D     1E 00                 LD      E,0             ;Initialize edge mask
 5784     1B3F     CD 1B23               CALL    CNTHL0
 5785     1B42     C3 1B25               JP      CNTHL1
 5786                             SUBTTL - BIO -   OUTDO routine
```

```

1B45                   OUTDO:
                       ;
                       ; OUTDO ( RST 18H )
                       ;       Prints char in [A], to either terminal or disk
                       ;       or printer depending on the flags:
                       ;               PRTFLG  if non-zero print to printer
                       ;               PTRFIL  if non-zero print to disk file pointed
                       ;                       to by PTRFIL
                       ;
1B45    F5                     PUSH    AF              ;Save character
1B46    CD FEE4                CALL    H.OUTD
1B49    CD 145F                CALL    ISFLIO          ;Doing I/O to file?
1B4C    28 08                  JR      Z,LPTCOD        ;Nope, check for output to printer
1B4E    F1                     POP     AF              ;Restore char.
1B4F    DD 21 6C48             LD      IX,FILOU1       ;Jump with pointer to FILE OUT routine
1B53    C3 01FF                JP      CALBAS
                       ;
1B56                   LPTCOD:
1B56    3A F416                LD      A,(PRTFLG)      ;Output to printer?
1B59    B7                     OR      A
1B5A    28 5F                  JR      Z,TTYCHR        ;Nope, output to console
1B5C    3A F418                LD      A,(RAWPRT)      ;Print in "RAW" mode?
1B5F    A7                     AND     A
1B60    20 49                  JR      NZ,LPTCH1       ;Yes, send char to printer
1B62    F1                     POP     AF              ;restore char
                       ;
1B63                   OUTDLP:
1B63    F5                     PUSH    AF
                       ;
1B64                   NTBKS2:
```

```
1B64    FE 09                   CP      9               ;TAB?
1B66    20 0E                   JR      NZ,NOTABL       ;No
                        ;
1B68                    MORSPL:
1B68    3E 20                   LD      A,' '           ;Print a space
1B6A    CD 1B63                 CALL    OUTDLP
1B6D    3A F415                 LD      A,(LPTPOS)      ;Get current LPOS
1B70    E6 07                   AND     7               ;At TAB stop?
1B72    20 F4                   JR      NZ,MORSPL       ;No, back for more space
1B74    F1                      POP     AF              ;Discard character
1B75    C9                      RET
                        ;
1B76                    NOTABL:
1B76    D6 0D                   SUB     0DH             ;Check if CR. If so load a zero
1B78    28 0A                   JR      Z,ZERLP1        ;It is, clear LPTPOS and send CR
1B7A    38 0B                   JR      C,LPTCH0        ;Code is 0..0CH, just send.
                                                        ;without modify LPTPOS
1B7C    FE 13                   CP      " "-13          ;See if control character
1B7E    38 07                   JR      C,LPTCH0        ;Code is 0EH..1FH, ditto
1B80    3A F415                 LD      A,(LPTPOS)      ;Get LPOS
1B83    3C                      INC     A
                        ;
1B84                    ZERLP1:
1B84    32 F415                 LD      (LPTPOS),A      ;Update LPOS
                        ;
1B87                    LPTCH0:
1B87    3A F417                 LD      A,(NTMSXP)      ;Output to MSX standard printer
1B8A    A7                      AND     A
1B8B    28 1E                   JR      Z,LPTCH1        ;No mapping for KATAKANA to HIRAGANA
1B8D    F1                      POP     AF              ;restore char to print
1B8E    CD 089D                 CALL    CNVCHR          ;See if graphic header
```

```
1B91    D0                      RET     NC              ;Yep
1B92    20 23                   JR      NZ,MAPSPC       ;Graphic symbol, map to space
1B94    A7                      AND     A
1B95    F2 1BAC                 JP      P,LPTCHR
1B98    FE 86                   CP      86H             ;Graphic symbol?
1B9A    38 1B                   JR      C,MAPSPC        ;Yes, map this to space too!
1B9C    FE A0                   CP      0A0H            ;A HIRAGANA(part 1)?
1B9E    30 04                   JR      NC,NTHIRA
1BA0    C6 20                   ADD     A,' '           ;Map to KATAKANA
1BA2    18 08                   JR      LPTCHR
1BA4                    NTHIRA:
1BA4    FE E0                   CP      0E0H            ;HIRAGANA(part 2)?
1BA6    38 04                   JR      C,LPTCHR        ;No
1BA8    D6 20                   SUB     ' '             ;Map to KATAKANA
1BAA    38                      DB      38H             ;'JRC' instruction (Skip next byte)
1BAB                    LPTCH1:
1BAB    F1                      POP     AF              ;Restore char
                        ;
1BAC                    LPTCHR:
1BAC    CD 085D                 CALL    LPTOUT          ;Send character out
1BAF    D0                      RET     NC              ;Sent successful
1BB0    DD 21 73B2              LD      IX,DIOERR       ;Direct I/O error
1BB4    C3 01FF                 JP      CALBAS
1BB7                    MAPSPC:
1BB7    3E 20                   LD      A,' '
1BB9    18 F1                   JR      LPTCHR
1BBB                    TTYCHR:
                        ;
                        ; Output to console
                        ;
1BBB    F1                      POP     AF              ;Get the character
```

```
 5880     1BBC    C3 08BC                 JP      CHPUT
 5881                             SUBTTL -MSXCHR-  MSX character set
```

```

1BBF                        CGTABL:
1BBF    00 00 00 00                 DB      00H,00H,00H,00H,00H,00H,00H
1BC3    00 00 00
1BC6    00 7E 42 7E                 DB      00H,7EH,42H,7EH,42H,7EH,42H
1BCA    42 7E 42
1BCD    82 00 10 92                 DB      82H,00H,10H,92H,54H,10H,28H
1BD1    54 10 28
1BD4    44 82 00 12                 DB      44H,82H,00H,12H,14H,0F8H,14H
1BD8    14 F8 14
1BDB    34 52 92 00                 DB      34H,52H,92H,00H,10H,10H,0FEH
1BDF    10 10 FE
1BE2    10 38 54 92                 DB      10H,38H,54H,92H,00H,10H,28H
1BE6    00 10 28
1BE9    7C 92 38 54                 DB      7CH,92H,38H,54H,0FEH,00H,10H
1BED    FE 00 10
1BF0    10 10 7C 10                 DB      10H,10H,7CH,10H,10H,0FEH,00H
1BF4    10 FE 00
1BF7    7E 42 42 7E                 DB      7EH,42H,42H,7EH,42H,42H,7EH
1BFB    42 42 7E
1BFE    00 40 7E 48                 DB      00H,40H,7EH,48H,3CH,28H,7EH
1C02    3C 28 7E
1C05    08 00 FE 92                 DB      08H.00H,0FEH,92H,92H,0FEH,82H
1C09    92 FE 82
1C0C    82 86 00 04                 DB      82H,86H,00H,04H,0EEH,0A4H,0EFH
1C10    EE A4 EF
1C13    A2 EA 06 00                 DB      0A2H,0EAH,06H,00H,28H,44H,82H
1C17    28 44 82
1C1A    3C 14 24 4C                 DB      3CH,14H,24H,4CH,00H,28H,0C8H
1C1E    00 28 C8
1C21    5C EA 6C C8                 DB      5CH,0EAH,6CH,0C8H,50H,00H,7CH
```

```
 5913     1C25     50 00 7C
 5914     1C28     20 7C 44 7C          DB      20H,7CH,44H,7CH,44H,7CH,00H
 5915     1C2C     44 7C 00
 5916     1C2F     0C 70 10 FE          DB      0CH,70H,10H,0FEH,10H,10H,10H
 5917     1C33     10 10 10
 5918     1C36     00 7E 10 1E          DB      00H,7EH,10H,1EH,12H,22H,44H
 5919     1C3A     12 22 44
 5920     1C3D     08 00 00 7C          DB      08H,00H,00H,7CH,28H,28H,28H
 5921     1C41     28 28 28
 5922     1C44     4E 00 00 10          DB      4EH,00H,00H,10H,10H,10H,0FFH
 5923     1C48     10 10 FF
 5924     1C4B     00 00 00 00          DB      00H,00H,00H,00H,00H,00H,00H
 5925     1C4F     00 00 00
 5926     1C52     FF 10 10 10          DB      0FFH,10H,10H,10H,10H,10H,10H
 5927     1C56     10 10 10
 5928     1C59     10 F0 10 10          DB      10H,0F0H,10H,10H,10H,10H,10H
 5929     1C5D     10 10 10
 5930     1C60     10 10 1F 10          DB      10H,10H,1FH,10H,10H,10H,10H
 5931     1C64     10 10 10
 5932     1C67     10 10 10 FF          DB      10H,10H,10H,0FFH,10H,10H,10H
 5933     1C6B     10 10 10
 5934     1C6E     10 10 10 10          DB      10H,10H,10H,10H,10H,10H,10H
 5935     1C72     10 10 10
 5936     1C75     10 10 00 00          DB      10H,10H,00H,00H,00H,0FFH,00H
 5937     1C79     00 FF 00
 5938     1C7C     00 00 00 00          DB      00H,00H,00H,00H,00H,00H,1FH
 5939     1C80     00 00 1F
 5940     1C83     10 10 10 10          DB      10H,10H,10H,10H,00H,00H,00H
 5941     1C87     00 00 00
 5942     1C8A     F0 10 10 10          DB      0F0H,10H,10H,10H,10H,10H,10H
 5943     1C8E     10 10 10
 5944      1C91     10 1F 00 00              DB      10H,1FH,00H,00H,00H,00H,10H
 5945      1C95     00 00 10
 5946      1C98     10 10 F0 00              DB      10H,10H,0F0H,00H,00H,00H,00H
 5947      1C9C     00 00 00
 5948      1C9F     81 42 24 18              DB      81H,42H,24H,18H,18H,24H,42H
 5949      1CA3     18 24 42
 5950      1CA6     81 10 7C 10              DB      81H,10H,7CH,10H,10H,28H,44H
 5951      1CAA     10 28 44
 5952      1CAD     82 00 10 10              DB      82H,00H,10H,10H,0FEH,92H,0FEH
 5953      1CB1     FE 92 FE
 5954      1CB4     10 10 00 10              DB      10H,10H,00H,10H,10H,54H,54H
 5955      1CB8     10 54 54
 5956      1CBB     92 10 30 00              DB      92H,10H,30H,00H,00H,00H,00H
 5957      1CBF     00 00 00
 5958      1CC2     00 00 00 00              DB      00H,00H,00H,00H,00H,20H,20H
 5959      1CC6     00 20 20
 5960      1CC9     20 20 00 00              DB      20H,20H,00H,00H,20H,00H,50H
 5961      1CCD     20 00 50
 5962      1CD0     50 50 00 00              DB      50H,50H,00H,00H,00H,00H,00H
 5963      1CD4     00 00 00
 5964      1CD7     50 50 F8 50              DB      50H,50H,0F8H,50H,0F8H,50H,50H
 5965      1CDB     F8 50 50
 5966      1CDE     00 20 78 A0              DB      00H,20H,78H,0A0H,70H,28H,0F0H
 5967      1CE2     70 28 F0
 5968      1CE5     20 00 C0 C8              DB      20H,00H,0C0H,0C8H,10H,20H,40H
 5969      1CE9     10 20 40
 5970      1CEC     98 18 00 40              DB      98H,18H,00H,40H,0A0H,40H,0A8H
 5971      1CF0     A0 40 A8
 5972      1CF3     90 98 60 00              DB      90H,98H,60H,00H,10H,20H,40H
 5973      1CF7     10 20 40
 5974      1CFA     00 00 00 00              DB      00H,00H,00H,00H,00H,10H,20H
 5975     1CFE      00 10 20
 5976     1D01      40 40 40 20          DB      40H,40H,40H,20H,10H,00H,40H
 5977     1D05      10 00 40
 5978     1D08      20 10 10 10          DB      20H,10H,10H,10H,20H,40H,00H
 5979     1D0C      20 40 00
 5980     1D0F      20 A8 70 20          DB      20H,0A8H,70H,20H,70H,0A8H,20H
 5981     1D13      70 A8 20
 5982     1D16      00 00 20 20          DB      00H,00H,20H,20H,0F8H,20H,20H
 5983     1D1A      F8 20 20
 5984     1D1D      00 00 00 00          DB      00H,00H,00H,00H,00H,00H,00H
 5985     1D21      00 00 00
 5986     1D24      20 20 40 00          DB      20H,20H,40H,00H,00H,00H,78H
 5987     1D28      00 00 78
 5988     1D2B      00 00 00 00          DB      00H,00H,00H,00H,00H,00H,00H
 5989     1D2F      00 00 00
 5990     1D32      00 00 60 60          DB      00H,00H,60H,60H,00H,00H,00H
 5991     1D36      00 00 00
 5992     1D39      08 10 20 40          DB      08H,10H,20H,40H,80H,00H,70H
 5993     1D3D      80 00 70
 5994     1D40      88 98 A8 C8          DB      88H,98H,0A8H,0C8H,88H,70H,00H
 5995     1D44      88 70 00
 5996     1D47      20 60 A0 20          DB      20H,60H,0A0H,20H,20H,20H,0F8H
 5997     1D4B      20 20 F8
 5998     1D4E      00 70 88 08          DB      00H,70H,88H,08H,10H,60H,80H
 5999     1D52      10 60 80
 6000     1D55      F8 00 70 88          DB      0F8H,00H,70H,88H,08H,30H,08H
 6001     1D59      08 30 08
 6002     1D5C      88 70 00 10          DB      88H,70H,00H,10H,30H,50H,90H
 6003     1D60      30 50 90
 6004     1D63      F8 10 10 00          DB      0F8H,10H,10H,00H,0F8H,80H,0E0H
 6005     1D67      F8 80 E0
 6006      1D6A     10 08 10 E0           DB       10H,08H,10H,0E0H,00H,30H,40H
 6007      1D6E     00 30 40
 6008      1D71     80 F0 88 88           DB       80H,0F0H,88H,88H,70H,00H,0F8H
 6009      1D75     70 00 F8
 6010      1D78     88 10 20 20           DB       88H,10H,20H,20H,20H,20H,00H
 6011      1D7C     20 20 00
 6012      1D7F     70 88 88 70           DB       70H,88H,88H,70H,88H,88H,70H
 6013      1D83     88 88 70
 6014      1D86     00 70 88 88           DB       00H,70H,88H,88H,78H,08H,10H
 6015      1D8A     78 08 10
 6016      1D8D     60 00 00 00           DB       60H,00H,00H,00H,20H,00H,00H
 6017      1D91     20 00 00
 6018      1D94     20 00 00 00           DB       20H,00H,00H,00H,00H,20H,00H
 6019      1D98     00 20 00
 6020      1D9B     00 20 20 40           DB       00H,20H,20H,40H,18H,30H,60H
 6021      1D9F     18 30 60
 6022      1DA2     C0 60 30 18           DB       0C0H,60H,30H,18H,00H,00H,00H
 6023      1DA6     00 00 00
 6024      1DA9     F8 00 F8 00           DB       0F8H,00H,0F8H,00H,00H,00H,0C0H
 6025      1DAD     00 00 C0
 6026      1DB0     60 30 18 30           DB       60H,30H,18H,30H,60H,0C0H,00H
 6027      1DB4     60 C0 00
 6028      1DB7     70 88 08 10           DB       70H,88H,08H,10H,20H,00H,20H
 6029      1DBB     20 00 20
 6030      1DBE     00 70 88 08           DB       00H,70H,88H,08H,68H,0A8H,0A8H
 6031      1DC2     68 A8 A8
 6032      1DC5     70 00 20 50           DB       70H,00H,20H,50H,88H,88H,0F8H
 6033      1DC9     88 88 F8
 6034      1DCC     88 88 00 F0           DB       88H,88H,00H,0F0H,48H,48H,70H
 6035      1DD0     48 48 70
 6036      1DD3     48 48 F0 00           DB       48H,48H,0F0H,00H,30H,48H,80H
 6037      1DD7     30 48 80
 6038      1DDA     80 80 48 30          DB        80H,80H,48H,30H,00H,0E0H,50H
 6039      1DDE     00 E0 50
 6040      1DE1     48 48 48 50          DB        48H,48H,48H,50H,0E0H,00H,0F8H
 6041      1DE5     E0 00 F8
 6042      1DE8     80 80 F0 80          DB        80H,80H,0F0H,80H,80H,0F8H,00H
 6043      1DEC     80 F8 00
 6044      1DEF     F8 80 80 F0          DB        0F8H,80H,80H,0F0H,80H,80H,80H
 6045      1DF3     80 80 80
 6046      1DF6     00 70 88 80          DB        00H,70H,88H,80H,0B8H,88H,88H
 6047      1DFA     B8 88 88
 6048      1DFD     70 00 88 88          DB        70H,00H,88H,88H,88H,0F8H,88H
 6049      1E01     88 F8 88
 6050      1E04     88 88 00 70          DB        88H,88H,00H,70H,20H,20H,20H
 6051      1E08     20 20 20
 6052      1E0B     20 20 70 00          DB        20H,20H,70H,00H,38H,10H,10H
 6053      1E0F     38 10 10
 6054      1E12     10 90 90 60          DB        10H,90H,90H,60H,00H,88H,90H
 6055      1E16     00 88 90
 6056      1E19     A0 C0 A0 90          DB        0A0H,0C0H,0A0H,90H,88H,00H,80H
 6057      1E1D     88 00 80
 6058      1E20     80 80 80 80          DB        80H,80H,80H,80H,80H,0F8H,00H
 6059      1E24     80 F8 00
 6060      1E27     88 D8 A8 A8          DB        88H,0D8H,0A8H,0A8H,88H,88H,88H
 6061      1E2B     88 88 88
 6062      1E2E     00 88 C8 C8          DB        00H,88H,0C8H,0C8H,0A8H,98H,98H
 6063      1E32     A8 98 98
 6064      1E35     88 00 70 88          DB        88H,00H,70H,88H,88H,88H,88H
 6065      1E39     88 88 88
 6066      1E3C     88 70 00 F0          DB        88H,70H,00H,0F0H,88H,88H,0F0H
 6067      1E40     88 88 F0
 6068      1E43     80 80 80 00            DB       80H,80H,80H,00H,70H,88H,88H
 6069      1E47     70 88 88
 6070      1E4A     88 A8 90 68            DB       88H,0A8H,90H,68H,00H,0F0H,88H
 6071      1E4E     00 F0 88
 6072      1E51     88 F0 A0 90            DB       88H,0F0H,0A0H,90H,88H,00H,70H
 6073      1E55     88 00 70
 6074      1E58     88 80 70 08            DB       88H,80H,70H,08H,88H,70H,00H
 6075      1E5C     88 70 00
 6076      1E5F     F8 20 20 20            DB       0F8H,20H,20H,20H,20H,20H,20H
 6077      1E63     20 20 20
 6078      1E66     00 88 88 88            DB       00H,88H,88H,88H,88H,88H,88H
 6079      1E6A     88 88 88
 6080      1E6D     70 00 88 88            DB       70H,00H,88H,88H,88H,88H,50H
 6081      1E71     88 88 50
 6082      1E74     50 20 00 88            DB       50H,20H,00H,88H,88H,88H,0A8H
 6083      1E78     88 88 A8
 6084      1E7B     A8 D8 88 00            DB       0A8H,0D8H,88H,00H,88H,88H,50H
 6085      1E7F     88 88 50
 6086      1E82     20 50 88 88            DB       20H,50H,88H,88H,00H,88H,88H
 6087      1E86     00 88 88
 6088      1E89     88 70 20 20            DB       88H,70H,20H,20H,20H,00H,0F8H
 6089      1E8D     20 00 F8
 6090      1E90     08 10 20 40            DB       08H,10H,20H,40H,80H,0F8H,00H
 6091      1E94     80 F8 00
 6092      1E97     70 40 40 40            DB       70H,40H,40H,40H,40H,40H,70H
 6093      1E9B     40 40 70
 6094      1E9E     00 88 50 20            DB       00H,88H,50H,20H,70H,20H,70H
 6095      1EA2     70 20 70
 6096      1EA5     20 00 70 10            DB       20H,00H,70H,10H,10H,10H,10H
 6097      1EA9     10 10 10
 6098      1EAC     10 70 00 20            DB       10H,70H,00H,20H,50H,88H,00H
```

```
6099      1EB0      50 88 00
6100      1EB3      00 00 00 00          DB       00H,00H,00H,00H,00H,00H,00H
6101      1EB7      00 00 00
6102      1EBA      00 00 00 F8          DB       00H,00H,00H,0F8H,00H,40H,20H
6103      1EBE      00 40 20
6104      1EC1      10 00 00 00          DB       10H,00H,00H,00H,00H,00H,00H
6105      1EC5      00 00 00
6106      1EC8      00 70 08 78          DB       00H,70H,08H,78H,88H,78H,00H
6107      1ECC      88 78 00
6108      1ECF      80 80 B0 C8          DB       80H,80H,0B0H,0C8H,88H,0C8H,0B0H
6109      1ED3      88 C8 B0
6110      1ED6      00 00 00 70          DB       00H,00H,00H,70H,88H,80H,88H
6111      1EDA      88 80 88
6112      1EDD      70 00 08 08          DB       70H,00H,08H,08H,68H,98H,88H
6113      1EE1      68 98 88
6114      1EE4      98 68 00 00          DB       98H,68H,00H,00H,00H,70H,88H
6115      1EE8      00 70 88
6116      1EEB      F8 80 70 00          DB       0F8H,80H,70H,00H,10H,28H,20H
6117      1EEF      10 28 20
6118      1EF2      F8 20 20 20          DB       0F8H,20H,20H,20H,00H,00H,00H
6119      1EF6      00 00 00
6120      1EF9      68 98 98 68          DB       68H,98H,98H,68H,08H,70H,80H
6121      1EFD      08 70 80
6122      1F00      80 F0 88 88          DB       80H,0F0H,88H,88H,88H,88H,00H
6123      1F04      88 88 00
6124      1F07      20 00 60 20          DB       20H,00H,60H,20H,20H,20H,70H
6125      1F0B      20 20 70
6126      1F0E      00 10 00 30          DB       00H,10H,00H,30H,10H,10H,10H
6127      1F12      10 10 10
6128      1F15      90 60 40 40          DB       90H,60H,40H,40H,48H,50H,60H
6129      1F19      48 50 60
```

```
 6130     1F1C     50 48 00 60         DB      50H,48H,00H,60H,20H,20H,20H
 6131     1F20     20 20 20
 6132     1F23     20 20 70 00         DB      20H,20H,70H,00H,00H,00H,0D0H
 6133     1F27     00 00 D0
 6134     1F2A     A8 A8 A8 A8         DB      0A8H,0A8H,0A8H,0A8H,00H,00H,00H
 6135     1F2E     00 00 00
 6136     1F31     B0 C8 88 88         DB      0B0H,0C8H,88H,88H,88H,00H,00H
 6137     1F35     88 00 00
 6138     1F38     00 70 88 88         DB      00H,70H,88H,88H,88H,70H,00H
 6139     1F3C     88 70 00
 6140     1F3F     00 00 B0 C8         DB      00H,00H,0B0H,0C8H,0C8H,0B0H,80H
 6141     1F43     C8 B0 80
 6142     1F46     80 00 00 68         DB      80H,00H,00H,68H,98H,98H,68H
 6143     1F4A     98 98 68
 6144     1F4D     08 08 00 00         DB      08H,08H,00H,00H,0B0H,0C8H,80H
 6145     1F51     B0 C8 80
 6146     1F54     80 80 00 00         DB      80H,80H,00H,00H,00H,78H,80H
 6147     1F58     00 78 80
 6148     1F5B     F0 08 F0 00         DB      0F0H,08H,0F0H,00H,40H,40H,0F0H
 6149     1F5F     40 40 F0
 6150     1F62     40 40 48 30         DB      40H,40H,48H,30H,00H,00H,00H
 6151     1F66     00 00 00
 6152     1F69     90 90 90 90         DB      90H,90H,90H,90H,68H,00H,00H
 6153     1F6D     68 00 00
 6154     1F70     00 88 88 88         DB      00H,88H,88H,88H,50H,20H,00H
 6155     1F74     50 20 00
 6156     1F77     00 00 88 A8         DB      00H,00H,88H,0A8H,0A8H,0A8H,50H
 6157     1F7B     A8 A8 50
 6158     1F7E     00 00 00 88         DB      00H,00H,00H,88H,50H,20H,50H
 6159     1F82     50 20 50
 6160     1F85     88 00 00 00         DB      88H,00H,00H,00H,88H,88H,98H
 6161     1F89    88 88 98
 6162     1F8C    68 08 70 00         DB      68H,08H,70H,00H,00H,0F8H,10H
 6163     1F90    00 F8 10
 6164     1F93    20 40 F8 00         DB      20H,40H,0F8H,00H,18H,20H,20H
 6165     1F97    18 20 20
 6166     1F9A    40 20 20 18         DB      40H,20H,20H,18H,00H,20H,20H
 6167     1F9E    00 20 20
 6168     1FA1    20 00 20 20         DB      20H,00H,20H,20H,20H,00H,0C0H
 6169     1FA5    20 00 C0
 6170     1FA8    20 20 10 20         DB      20H,20H,10H,20H,20H,0C0H,00H
 6171     1FAC    20 C0 00
 6172     1FAF    40 A8 10 00         DB      40H,0A8H,10H,00H,00H,00H,00H
 6173     1FB3    00 00 00
 6174     1FB6    00 00 00 00         DB      00H,00H,00H,00H,00H,00H,00H
 6175     1FBA    00 00 00
 6176     1FBD    00 00 10 38         DB      00H,00H,10H,38H,7CH,0FEH,0FEH
 6177     1FC1    7C FE FE
 6178     1FC4    38 7C 00 6C         DB      38H,7CH,00H,6CH,0FEH,0FEH,0FEH
 6179     1FC8    FE FE FE
 6180     1FCB    7C 38 10 00         DB      7CH,38H,10H,00H,38H,38H,0FEH
 6181     1FCF    38 38 FE
 6182     1FD2    FE D6 10 7C         DB      0FEH,0D6H,10H,7CH,00H,10H,38H
 6183     1FD6    00 10 38
 6184     1FD9    7C FE 7C 38         DB      7CH,0FEH,7CH,38H,10H,00H,00H
 6185     1FDD    10 00 00
 6186     1FE0    78 84 84 84         DB      78H,84H,84H,84H,84H,78H,00H
 6187     1FE4    84 78 00
 6188     1FE7    00 78 FC FC         DB      00H,78H,0FCH,0FCH,0FCH,0FCH,78H
 6189     1FEB    FC FC 78
 6190     1FEE    00 40 FE 48         DB      00H,40H,0FEH,48H,70H,48H,82H
 6191     1FF2    70 48 82
 6192      1FF5     7C 00 00 00          DB      7CH,00H,00H,00H,10H,7EH,3CH
 6193      1FF9     10 7E 3C
 6194      1FFC     5A 34 00 00          DB      5AH,34H,00H,00H,00H,40H,42H
 6195      2000     00 40 42
 6196      2003     42 52 20 00          DB      42H,52H,20H,00H,00H,00H,1CH
 6197      2007     00 00 1C
 6198      200A     1C 22 02 0C          DB      1CH,22H,02H,0CH,00H,00H,00H
 6199      200E     00 00 00
 6200      2011     18 7E 18 30          DB      18H,7EH,18H,30H,6EH,00H,00H
 6201      2015     6E 00 00
 6202      2018     00 12 7E 3C          DB      00H,12H,7EH,3CH,52H,34H,00H
 6203      201C     52 34 00
 6204      201F     00 00 28 7C          DB      00H,00H,28H,7CH,2AH,22H,24H
 6205      2023     2A 22 24
 6206      2026     00 00 00 08          DB      00H,00H,00H,08H,5CH,6AH,0CH
 6207      202A     5C 6A 0C
 6208      202D     30 00 00 00          DB      30H,00H,00H,00H,08H,0EH,38H
 6209      2031     08 0E 38
 6210      2034     4C 3A 00 00          DB      4CH,3AH,00H,00H,00H,00H,3CH
 6211      2038     00 00 3C
 6212      203B     02 02 1C 00          DB      02H,02H,1CH,00H,00H,00H,00H
 6213      203F     00 00 00
 6214      2042     00 00 00 00          DB      00H,00H,00H,00H,00H,20H,0FEH
 6215      2046     00 20 FE
 6216      2049     20 7C AA B2          DB      20H,7CH,0AAH,0B2H,64H,00H,00H
 6217      204D     64 00 00
 6218      2050     80 82 82 82          DB      80H,82H,82H,82H,90H,60H,00H
 6219      2054     90 60 00
 6220      2057     1C 00 7C 02          DB      1CH,00H,7CH,02H,02H,04H,18H
 6221      205B     02 04 18
 6222      205E     00 38 00 FE          DB      00H,38H,00H,0FEH,08H,30H,50H
 6223      2062     08 30 50
 6224      2065     9E 00 20 FA          DB        9EH,00H,20H,0FAH,22H,7CH,0A2H
 6225      2069     22 7C A2
 6226      206C     A2 4C 00 40          DB        0A2H,4CH,00H,40H,44H,0F2H,4AH
 6227      2070     44 F2 4A
 6228      2073     48 88 30 00          DB        48H,88H,30H,00H,10H,0FCH,08H
 6229      2077     10 FC 08
 6230      207A     3E 04 80 7C          DB        3EH,04H,80H,7CH,00H,18H,18H
 6231      207E     00 18 18
 6232      2081     30 60 60 30          DB        30H,60H,60H,30H,18H,00H,04H
 6233      2085     18 00 04
 6234      2088     84 BE 84 84          DB        84H,0BEH,84H,84H,84H,48H,00H
 6235      208C     84 48 00
 6236      208F     00 FC 02 00          DB        00H,0FCH,02H,00H,40H,80H,7EH
 6237      2093     40 80 7E
 6238      2096     00 10 16 F8          DB        00H,10H,16H,0F8H,08H,7CH,80H
 6239      209A     08 7C 80
 6240      209D     78 00 80 80          DB        78H,00H,80H,80H,80H,80H,84H
 6241      20A1     80 80 84
 6242      20A4     88 70 00 08          DB        88H,70H,00H,08H,0FEH,08H,38H
 6243      20A8     FE 08 38
 6244      20AB     48 38 08 00          DB        48H,38H,08H,00H,04H,44H,0FEH
 6245      20AF     04 44 FE
 6246      20B2     44 44 40 3E          DB        44H,44H,40H,3EH,00H,64H,28H
 6247      20B6     00 64 28
 6248      20B9     30 FE 20 40          DB        30H,0FEH,20H,40H,3CH,00H,00H
 6249      20BD     3C 00 00
 6250      20C0     00 00 00 00          DB        00H,00H,00H,00H,00H,00H,00H
 6251      20C4     00 00 00
 6252      20C7     00 00 00 00          DB        00H,00H,00H,00H,60H,90H,60H
 6253      20CB     60 90 60
 6254     20CE    00 38 20 20        DB       00H,38H,20H,20H,20H,00H,00H
 6255     20D2    20 00 00
 6256     20D5    00 00 00 00        DB       00H,00H,00H,00H,00H,20H,20H
 6257     20D9    00 20 20
 6258     20DC    20 E0 00 00        DB       20H,0E0H,00H,00H,00H,00H,00H
 6259     20E0    00 00 00
 6260     20E3    80 40 20 00        DB       80H,40H,20H,00H,00H,00H,00H
 6261     20E7    00 00 00
 6262     20EA    30 30 00 00        DB       30H,30H,00H,00H,00H,0F8H,08H
 6263     20EE    00 F8 08
 6264     20F1    F8 08 10 20        DB       0F8H,08H,10H,20H,40H,00H,00H
 6265     20F5    40 00 00
 6266     20F8    00 F0 10 60        DB       00H,0F0H,10H,60H,40H,80H,00H
 6267     20FC    40 80 00
 6268     20FF    00 10 20 60        DB       00H,10H,20H,60H,0A0H,20H,20H
 6269     2103    A0 20 20
 6270     2106    00 00 20 F0        DB       00H,00H,20H,0F0H,90H,10H,20H
 6271     210A    90 10 20
 6272     210D    40 00 00 00        DB       40H,00H,00H,00H,0F0H,20H,20H
 6273     2111    F0 20 20
 6274     2114    20 F0 00 00        DB       20H,0F0H,00H,00H,20H,0F0H,60H
 6275     2118    20 F0 60
 6276     211B    A0 A0 20 00        DB       0A0H,0A0H,20H,00H,00H,40H,0F8H
 6277     211F    00 40 F8
 6278     2122    48 50 40 40        DB       48H,50H,40H,40H,00H,00H,00H
 6279     2126    00 00 00
 6280     2129    70 10 10 10        DB       70H,10H,10H,10H,0F8H,00H,00H
 6281     212D    F8 00 00
 6282     2130    00 F0 10 F0        DB       00H,0F0H,10H,0F0H,10H,0F0H,00H
 6283     2134    10 F0 00
 6284     2137    00 00 A8 A8        DB       00H,00H,0A8H,0A8H,08H,10H,20H
 6285     213B    08 10 20
 6286     213E    00 00 00 00          DB      00H,00H,00H,00H,0F8H,00H,00H
 6287     2142    F8 00 00
 6288     2145    00 00 F8 08          DB      00H,00H,0F8H,08H,28H,30H,20H
 6289     2149    28 30 20
 6290     214C    20 40 00 08          DB      20H,40H,00H,08H,10H,20H,60H
 6291     2150    10 20 60
 6292     2153    A0 20 20 00          DB      0A0H,20H,20H,00H,20H,0F8H,88H
 6293     2157    20 F8 88
 6294     215A    88 08 10 20          DB      88H,08H,10H,20H,00H,00H,0F8H
 6295     215E    00 00 F8
 6296     2161    20 20 20 20          DB      20H,20H,20H,20H,0F8H,00H,10H
 6297     2165    F8 00 10
 6298     2168    F8 10 30 50          DB      0F8H,10H,30H,50H,90H,10H,00H
 6299     216C    90 10 00
 6300     216F    20 F8 28 28          DB      20H,0F8H,28H,28H,28H,48H,88H
 6301     2173    28 48 88
 6302     2176    00 20 F8 20          DB      00H,20H,0F8H,20H,0F8H,20H,20H
 6303     217A    F8 20 20
 6304     217D    20 00 78 48          DB      20H,00H,78H,48H,88H,08H,08H
 6305     2181    88 08 08
 6306     2184    10 20 00 40          DB      10H,20H,00H,40H,78H,50H,90H
 6307     2188    78 50 90
 6308     218B    10 10 20 00          DB      10H,10H,20H,00H,00H,0F8H,08H
 6309     218F    00 F8 08
 6310     2192    08 08 08 F8          DB      08H,08H,08H,0F8H,00H,50H,0F8H
 6311     2196    00 50 F8
 6312     2199    50 50 10 10          DB      50H,50H,10H,10H,20H,00H,00H
 6313     219D    20 00 00
 6314     21A0    C0 08 C8 08          DB      0C0H,08H,0C8H,08H,10H,0E0H,00H
 6315     21A4    10 E0 00
 6316      21A7     00 F8 08 10            DB       00H,0F8H,08H,10H,20H,50H,88H
 6317      21AB     20 50 88
 6318      21AE     00 40 F8 48            DB       00H,40H,0F8H,48H,50H,40H,40H
 6319      21B2     50 40 40
 6320      21B5     38 00 88 88            DB       38H,00H,88H,88H,48H,08H,10H
 6321      21B9     48 08 10
 6322      21BC     20 40 00 78            DB       20H,40H,00H,78H,48H,78H,88H
 6323      21C0     48 78 88
 6324      21C3     08 10 20 00            DB       08H,10H,20H,00H,10H,0E0H,20H
 6325      21C7     10 E0 20
 6326      21CA     F8 20 20 40            DB       0F8H,20H,20H,40H,00H,0A8H,0A8H
 6327      21CE     00 A8 A8
 6328      21D1     A8 08 08 10            DB       0A8H,08H,08H,10H,20H,00H,70H
 6329      21D5     20 00 70
 6330      21D8     00 F8 20 20            DB       00H,0F8H,20H,20H,20H,40H,00H
 6331      21DC     20 40 00
 6332      21DF     40 40 60 50            DB       40H,40H,60H,50H,48H,40H,40H
 6333      21E3     48 40 40
 6334      21E6     00 20 F8 20            DB       00H,20H,0F8H,20H,20H,20H,20H
 6335      21EA     20 20 20
 6336      21ED     40 00 00 70            DB       40H,00H,00H,70H,00H,00H,00H
 6337      21F1     00 00 00
 6338      21F4     00 F8 00 00            DB       00H,0F8H,00H,00H,0F8H,08H,0D0H
 6339      21F8     F8 08 D0
 6340      21FB     20 50 88 00            DB       20H,50H,88H,00H,20H,0F8H,08H
 6341      21FF     20 F8 08
 6342      2202     30 E8 20 20            DB       30H,0E8H,20H,20H,00H,08H,08H
 6343      2206     00 08 08
 6344      2209     08 10 20 40            DB       08H,10H,20H,40H,80H,00H,20H
 6345      220D     80 00 20
 6346      2210     10 48 48 48            DB       10H,48H,48H,48H,48H,88H,00H
 6347     2214     48 88 00
 6348     2217     80 80 F8 80          DB       80H,80H,0F8H,80H,80H,80H,78H
 6349     221B     80 80 78
 6350     221E     00 F8 08 08          DB       00H,0F8H,08H,08H,08H,10H,20H
 6351     2222     08 10 20
 6352     2225     40 00 00 40          DB       40H,00H,00H,40H,0A0H,10H,08H
 6353     2229     A0 10 08
 6354     222C     08 00 00 20          DB       08H,00H,00H,20H,0F8H,20H,20H
 6355     2230     F8 20 20
 6356     2233     A8 A8 20 00          DB       0A8H,0A8H,20H,00H,00H,0F8H,08H
 6357     2237     00 F8 08
 6358     223A     08 50 20 10          DB       08H,50H,20H,10H,00H,0F0H,00H
 6359     223E     00 F0 00
 6360     2241     60 00 00 F0          DB       60H,00H,00H,0F0H,08H,00H,10H
 6361     2245     08 00 10
 6362     2248     20 40 80 90          DB       20H,40H,80H,90H,88H,0F8H,00H
 6363     224C     88 F8 00
 6364     224F     08 08 08 50          DB       08H,08H,08H,50H,20H,50H,80H
 6365     2253     20 50 80
 6366     2256     00 78 20 F8          DB       00H,78H,20H,0F8H,20H,20H,20H
 6367     225A     20 20 20
 6368     225D     18 00 40 F8          DB       18H,00H,40H,0F8H,48H,48H,50H
 6369     2261     48 48 50
 6370     2264     40 40 00 00          DB       40H,40H,00H,00H,70H,10H,10H
 6371     2268     70 10 10
 6372     226B     10 10 F8 00          DB       10H,10H,0F8H,00H,00H,0F8H,08H
 6373     226F     00 F8 08
 6374     2272     F8 08 08 F8          DB       0F8H,08H,08H,0F8H,00H,70H,00H
 6375     2276     00 70 00
 6376     2279     F8 08 08 10          DB       0F8H,08H,08H,10H,20H,00H,48H
 6377     227D     20 00 48
 6378      2280     48 48 48 48          DB       48H,48H,48H,48H,10H,20H,00H
 6379      2284     10 20 00
 6380      2287     10 50 50 50          DB       10H,50H,50H,50H,50H,58H,90H
 6381      228B     50 58 90
 6382      228E     00 40 40 40          DB       00H,40H,40H,40H,48H,48H,50H
 6383      2292     48 48 50
 6384      2295     60 00 00 F8          DB       60H,00H,00H,0F8H,88H,88H,88H
 6385      2299     88 88 88
 6386      229C     88 F8 00 F8          DB       88H,0F8H,00H,0F8H,88H,88H,08H
 6387      22A0     88 88 08
 6388      22A3     08 10 20 00          DB       08H,10H,20H,00H,00H,0C0H,00H
 6389      22A7     00 C0 00
 6390      22AA     08 08 10 E0          DB       08H,08H,10H,0E0H,00H,90H,48H
 6391      22AE     00 90 48
 6392      22B1     00 00 00 00          DB       00H,00H,00H,00H,00H,00H,60H
 6393      22B5     00 00 60
 6394      22B8     90 60 00 00          DB       90H,60H,00H,00H,00H,00H,00H
 6395      22BC     00 00 00
 6396      22BF     40 FE 40 5E          DB       40H,0FEH,40H,5EH,80H,0A0H,9EH
 6397      22C3     80 A0 9E
 6398      22C6     00 20 FE 40          DB       00H,20H,0FEH,40H,0F8H,04H,04H
 6399      22CA     F8 04 04
 6400      22CD     78 00 00 00          DB       78H,00H,00H,00H,0FCH,02H,02H
 6401      22D1     FC 02 02
 6402      22D4     04 38 00 00          DB       04H,38H,00H,00H,0FEH,0CH,30H
 6403      22D8     FE 0C 30
 6404      22DB     40 40 38 00          DB       40H,40H,38H,00H,10H,12H,1CH
 6405      22DF     10 12 1C
 6406      22E2     30 40 40 3E          DB       30H,40H,40H,3EH,00H,24H,0F2H
 6407      22E6     00 24 F2
 6408      22E9     48 48 9C AA          DB       48H,48H,9CH,0AAH,10H,00H,80H
 6409     22ED    10 00 80
 6410     22F0    9E 80 80 A0         DB      9EH,80H,80H,0A0H,0BEH,0C0H,00H
 6411     22F4    BE C0 00
 6412     22F7    44 4C 7A AA         DB      44H,4CH,7AH,0AAH,0A6H,0AAH,6CH
 6413     22FB    A6 AA 6C
 6414     22FE    00 40 EC 52         DB      00H,40H,0ECH,52H,62H,0CEH,4AH
 6415     2302    62 CE 4A
 6416     2305    4C 00 00 38         DB      4CH,00H,00H,38H,54H,92H,0A2H
 6417     2309    54 92 A2
 6418     230C    A2 4C 00 04         DB      0A2H,4CH,00H,04H,0BEH,84H,84H
 6419     2310    BE 84 84
 6420     2313    9E A4 5C 00         DB      9EH,0A4H,5CH,00H,08H,4CH,0C6H
 6421     2317    08 4C C6
 6422     231A    46 44 44 38         DB      46H,44H,44H,38H,00H,20H,18H
 6423     231E    00 20 18
 6424     2321    20 16 8A CA         DB      20H,16H,8AH,0CAH,18H,00H,00H
 6425     2325    18 00 00
 6426     2328    20 70 D8 8C         DB      20H,70H,0D8H,8CH,06H,02H,00H
 6427     232C    06 02 00
 6428     232F    3E 84 BE 84         DB      3EH,84H,0BEH,84H,9CH,0A6H,18H
 6429     2333    9C A6 18
 6430     2336    00 08 7E 08         DB      00H,08H,7EH,08H,7EH,38H,4CH
 6431     233A    7E 38 4C
 6432     233D    3A 00 E0 24         DB      3AH,00H,0E0H,24H,24H,7EH,0A4H
 6433     2341    24 7E A4
 6434     2344    A4 68 00 20         DB      0A4H,68H,00H,20H,0FCH,24H,62H
 6435     2348    FC 24 62
 6436     234B    A0 62 3C 00         DB      0A0H,62H,3CH,00H,04H,44H,7CH
 6437     234F    04 44 7C
 6438     2352    C6 AA 92 64         DB      0C6H,0AAH,92H,64H,00H,20H,20H
 6439     2356    00 20 20
 6440     2359    78 20 78 22          DB      78H,20H,78H,22H,1CH,00H,00H
 6441     235D    1C 00 00
 6442     2360    48 FC 4A 42          DB      48H,0FCH,4AH,42H,4CH,40H,00H
 6443     2364    4C 40 00
 6444     2367    08 BC CA 8A          DB      08H,0BCH,0CAH,8AH,0BCH,08H,30H
 6445     236B    BC 08 30
 6446     236E    00 08 08 0E          DB      00H,08H,08H,0EH,08H,78H,8CH
 6447     2372    08 78 8C
 6448     2375    72 00 38 84          DB      72H,00H,38H,84H,80H,0FCH,0C2H
 6449     2379    80 FC C2
 6450     237C    02 38 00 00          DB      02H,38H,00H,00H,42H,42H,42H
 6451     2380    42 42 42
 6452     2383    62 04 18 00          DB      62H,04H,18H,00H,7CH,08H,30H
 6453     2387    7C 08 30
 6454     238A    DC 62 92 7C          DB      0DCH,62H,92H,7CH,00H,20H,2CH
 6455     238E    00 20 2C
 6456     2391    F4 24 64 E4          DB      0F4H,24H,64H,0E4H,26H,00H,7CH
 6457     2395    26 00 7C
 6458     2398    18 20 5C 82          DB      18H,20H,5CH,82H,02H,7CH,00H
 6459     239C    02 7C 00
 6460     239F    40 60 DC 62          DB      40H,60H,0DCH,62H,42H,0C2H,5CH
 6461     23A3    42 C2 5C
 6462     23A6    00 10 30 20          DB      00H,10H,30H,20H,70H,48H,0CEH
 6463     23AA    70 48 CE
 6464     23AD    84 00 00 00          DB      84H,00H,00H,00H,00H,00H,00H
 6465     23B1    00 00 00
 6466     23B4    00 00 00 00          DB      00H,00H,00H,00H,00H,00H,00H
 6467     23B8    00 00 00
 6468     23BB    00 00 00 00          DB      00H,00H,00H,00H
 6469                             SUBTTL - MSXINL, Screen editor - Line input and function character
```

```

23BF                PINLIN:
                    ;
                    ; Main entry point
                    ;
23BF    CD FDDB             CALL    H.PINL
23C2    3A F6AA             LD      A,(AUTFLG)      ;During AUTO mode?
23C5    A7                  AND     A
23C6    20 0D               JR      NZ,INLIN        ;Yes, then fake INLIN to prevent Û from
                                                    ;deleting line number
23C8    2E 00               LD      L,0
23CA    18 14               JR      INLIN1
23CC                QINLIN:
                    ;
                    ; Output question mark then get input
                    ;
23CC    CD FDE0             CALL    H.QINL
23CF    3E 3F               LD      A,'?'
23D1    DF                  RST     18H
23D2    3E 20               LD      A,' '
23D4    DF                  RST     18H
23D5                INLIN:
23D5    CD FDE5             CALL    H.INLI
23D8    2A F3DC             LD      HL,(CSRY)
23DB    2D                  DEC     L
23DC    C4 0C29             CALL    NZ,TERMIN       ;Terminate previous line
23DF    2C                  INC     L
23E0                INLIN1:
23E0    22 FBCA             LD      (FSTPOS),HL     ;Mark first position
23E3    AF                  XOR     A
23E4    32 FC9B             LD      (INTFLG),A
```

```
23E7                    INLIN2:
23E7    CD 10CB                 CALL    CHGET
23EA    21 2437                 LD      HL,SCITBL-2
23ED    0E 0B                   LD      C,0BH           ;SCI Max
23EF    CD 0919                 CALL    INDJMP          ;Do functions
23F2    F5                      PUSH    AF
23F3    C4 23FF                 CALL    NZ,INLOUT       ;Output a character
23F6    F1                      POP     AF
23F7    30 EE                   JR      NC,INLIN2       ;Not a terminator
                        ;
                        ; return to BASIC (break or CR)
                        ;
23F9    21 F55D                 LD      HL,BUFMIN
23FC    C8                      RET     Z               ;Cnt-C, return with carry set
23FD    3F                      CCF                     ;No, return carry clear
23FE                    RETURN:
23FE    C9                      RET
```

```

2419/23FF                 INLOUT:
                          ;
23FF     F5                       PUSH    AF              ;Save character to output
2400     FE 09                    CP      9               ;TAB?
2402     20 0F                    JR      NZ,OUTNTB       ;Nope
2404     F1                       POP     AF              ;Discard stack
2405                      OUTTAB:
2405     3E 20                    LD      A,' '           ;Map to space
2407     CD 23FF                  CALL    INLOUT
240A     3A F3DD                  LD      A,(CSRX)
240D     3D                       DEC     A               ;Make it zero based.
240E     E6 07                    AND     7               ;Reached TAB stop?
2410     20 F3                    JR      NZ,OUTTAB       ;Not yet,  continue...
2412     C9                       RET
2413                      OUTNTB:
                          ;
2413     F1                       POP     AF              ;Restore character
2414     21 FCA8                  LD      HL,INSFLG       ;points insert mode flag
2417     FE 01                    CP      1               ;Graphic header byte?
2419     28 0B                    JR      Z,INLOT0        ;Yes, send as is
241B     FE 20                    CP      ' '             ;control char?
241D     38 09                    JR      C,INLOT1        ;branch if so. - Reset insert mode
241F     F5                       PUSH    AF              ;save char to output
2420     7E                       LD      A,(HL)          ;get insert mode flag
2421     A7                       AND     A               ;test
2422     C4 24F2                  CALL    NZ,INSERT       ;if insert mode, make room to insert
2425     F1                       POP     AF              ;restore char to output
2426                      INLOT0:
2426     DF                       RST     18H             ;output char
2427     C9                       RET
```

```
2428                    INLOT1:
                        ;
2428    36 00                   LD      (HL),0          ;reset insert mode
242A    DF                      RST     18H             ;send this control char
242B    3E                      DB      3EH
242C                    SETINS:
242C    3E                      DB      3EH             ;Set insert mode and exit
242D                    SETOVW:
242D    AF                      XOR     A               ;Set overwrite mode
242E    F5                      PUSH    AF
242F    CD 0A2E                 CALL    CKERCS
2432    F1                      POP     AF
2433    32 FCAA                 LD      (CSTYLE),A
2436    C3 09E1                 JP      CKDPCS
```

```
2439                    SCITBL:
                        ;
                        ; Table of function characters
                        ;
2439    08                      DB      08H             ;Delete previous char
243A    2561                    DW      DELETE
243C    12                      DB      12H             ;Toggle insert flag
243D    24E5                    DW      TGLINS
243F    1B                      DB      1BH             ;Escape
2440    23FE                    DW      RETURN
2442    02                      DB      02H             ;Back word
2443    260E                    DW      LBCKWD
2445    06                      DB      06H             ;Next word
2446    25F8                    DW      LNXTWD
2448    0E                      DB      0EH
2449    25D7                    DW      LAPPND
244B    05                      DB      05H             ;Erase to end of line
244C    25B9                    DW      TRUNC
244E    03                      DB      03H             ;Abort
244F    24C5                    DW      LBREAK
2451    0D                      DB      0DH             ;Carriage return
2452    245A                    DW      LCRRET
2454    15                      DB      15H             ;Delete whole line
2455    25AE                    DW      LERASE
2457    7F                      DB      7FH             ;Delete character at cursor
2458    2550                    DW      LDELNX
                        SUBTTL - MSXINL, Screen editor -  Process special characters
```

```

 245A                  LCRRET:
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                       ; ;
                       ; Carriage return ;
                       ; ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 245A    CD 266C               CALL    GTFRST          ;L=line number of first visual
 245D    3A F6AA               LD      A,(AUTFLG)      ;During AUTO mode?
 2460    A7                    AND     A
 2461    28 02                 JR      Z,NOTAUT        ;No
 2463    26 01                 LD      H,1             ;Always get from top of line during AUTO mode
 2465                  NOTAUT:
 2465    E5                    PUSH    HL
                       ;
                       ; Put logical starting at L into BUF
                       ;
 2466    CD 0A2E               CALL    CKERCS
 2469    E1                    POP     HL
 246A    11 F55E               LD      DE,BUF          ;Line buffer pointer
 246D    06 FE                 LD      B,0FEH          ;Max count
 246F    2D                    DEC     L
 2470                  LCR1:
 2470    2C                    INC     L
 2471                  LCR2:
 2471    D5                    PUSH    DE              ;Save buffer pointer
 2472    C5                    PUSH    BC              ;Save buffer count
 2473    CD 0BD8               CALL    GETVRM          ;Get current character in Acc
 2476    C1                    POP     BC              ;Restore buffer count
 2477    D1                    POP     DE              ;Restore buffer pointer
 2478    A7                    AND     A               ;Null?
```

```
2479   28 14                 JR      Z,LCRNUL        ;Yes, ignore this
247B   FE 20                 CP      ' '             ;Special graphic character?
247D   30 0B                 JR      NC,LCRNRM       ;No, proceed normally
247F   05                    DEC     B               ;Decrement BUF size counter before storing
2480   28 1D                 JR      Z,LBLKSP        ;At end of BUF, so ignore this
2482   4F                    LD      C,A
2483   3E 01                 LD      A,1             ;Store header byte for graphic symbol
2485   12                    LD      (DE),A
2486   13                    INC     DE
2487   79                    LD      A,C
2488   C6 40                 ADD     A,'@'
248A                 LCRNRM:
248A   12                    LD      (DE),A          ;Store byte in buffer
248B   13                    INC     DE              ;Bump buffer pointer
248C   05                    DEC     B               ;Decrement BUF size counter
248D   28 10                 JR      Z,LBLKSP        ;At end of BUF
248F                 LCRNUL:
248F   24                    INC     H               ;Next column
2490   3A F3B0               LD      A,(LINLEN)      ;Max column reached?
2493   BC                    CP      H               ;
2494   30 DB                 JR      NC,LCR2         ;Not yet
2496   D5                    PUSH    DE              ;Save buffer pointer
2497   CD 0C1D               CALL    GETTRM          ;Is this line terminated?
249A   D1                    POP     DE              ;Restore buffer pointer
249B   26 01                 LD      H,1             ;Assume not, start from top of next line
249D   28 D1                 JR      Z,LCR1          ;No
249F                 LBLKSP:
                     ;
                     ; Suppress trailing blanks, [DE]=last+1
                     ;
249F   1B                    DEC     DE              ;Back up buffer pointer
```

```
24A0    1A                      LD      A,(DE)          ;Get stored character
24A1    FE 20                   CP      ' '             ;Is it space?
24A3    28 FA                   JR      Z,LBLKSP        ;Yes, ignore this
24A5    E5                      PUSH    HL
24A6    D5                      PUSH    DE
24A7    CD 09E1                 CALL    CKDPCS
24AA    D1                      POP     DE
24AB    E1                      POP     HL
                        ;
                        ; Terminate
                        ;
24AC    13                      INC     DE              ;Point past last valid character
24AD    AF                      XOR     A               ;Load terminator
24AE    12                      LD      (DE),A          ;Put it in BUF
24AF                    FAKECR:
24AF    3E 0D                   LD      A,0DH           ;Load character to echo to console
24B1    A7                      AND     A               ;Reset Z-flag, (say not break)
24B2                    LNXTLN:
24B2    F5                      PUSH    AF              ;Save this flag
24B3    CD 0C29                 CALL    TERMIN
24B6    CD 088E                 CALL    POSIT           ;Save current cursor position
24B9    3E 0A                   LD      A,0AH
24BB    DF                      RST     18H             ;Move cursor to start of next line
24BC    AF                      XOR     A               ;Clear possible INSFLG
24BD    32 FCA8                 LD      (INSFLG),A
24C0    F1                      POP     AF              ;Restore flags
24C1    37                      SCF                     ;Set carry indicating end of input
24C2    E1                      POP     HL              ;Discard return address (XRA A;RET)
24C3    C9                      RET                     ;If break, Z flag is set
24C4                    LBREKO:
                        ;
```

```
                          ; Control-C input
                          ;
24C4    2C                        INC     L               ;Bump line counter
24C5                      LBREAK:
24C5    CD 0C1D                   CALL    GETTRM          ;Line terminated?
24C8    28 FA                     JR      Z,LBREK0        ;No, check next line
24CA    CD 242D                   CALL    SETOVW          ;Set to overwrite mode
24CD    AF                        XOR     A               ;Load 0 in Acc, and set Z flag
24CE    32 F55E                   LD      (BUF),A         ;Say no character in BUF
24D1    26 01                     LD      H,1             ;Set to first column
24D3    E5                        PUSH    HL              ;Save cursor position
24D4    CD 04BD                   CALL    GICINI          ;Initialize sound chip and queue
24D7    CD 0454                   CALL    CKSTTP          ;Check if STOP trap is active or not
24DA    E1                        POP     HL
24DB    38 D2                     JR      C,FAKECR        ;Yes, fake CR
24DD    3A FBB1                   LD      A,(BASROM)      ;Executing BASIC program in ROM?
24E0    A7                        AND     A
24E1    20 CC                     JR      NZ,FAKECR       ;Yes, fake CR
24E3    18 CD                     JR      LNXTLN
```

```

24E5                    TGLINS:
                        ;
                        ; Toggle insert mode flag
                        ;
24E5    21 FCA8                 LD      HL,INSFLG       ;Get current insert flag
24E8    7E                      LD      A,(HL)
24E9    EE FF                   XOR     0FFH            ;Toggle insert status and affect Z flag
24EB    77                      LD      (HL),A
24EC    CA 242D                 JP      Z,SETOVW        ;Set to overwrite mode
24EF    C3 242C                 JP      SETINS          ;Set to insert mode
24F2                    INSERT:
                        ;
                        ; Insert a blank
                        ;
24F2    CD 0A2E                 CALL    CKERCS          ;Erase cursor before operation
24F5    2A F3DC                 LD      HL,(CSRY)
24F8    0E 20                   LD      C,' '           ;Load raw code for space
24FA                    INS1:
24FA    E5                      PUSH    HL              ;Save current cursor position
24FB                    INS2:
24FB    C5                      PUSH    BC              ;Save previous character
24FC    CD 0BD8                 CALL    GETVRM          ;Get current character in C
24FF    D1                      POP     DE              ;Restore previous character in [E]
2500    C5                      PUSH    BC              ;Save current character
2501    4B                      LD      C,E             ;C=previous character
2502    CD 0BE6                 CALL    PUTVRM          ;Put it on screen
2505    C1                      POP     BC              ;Restore current character in C
2506    3A F3B0                 LD      A,(LINLEN)      ;Check if end of line
2509    24                      INC     H               ;Bump column counter
250A    BC                      CP      H               ;End of line?
```

```
250B    7A                      LD      A,D             ;Get current attribute in Acc
250C    30 ED                   JR      NC,INS2         ;If not, continue till end of line
                        ;
                        ; Now we just finished a line, code of character wrapped to next
                        ; line is held in [C].
                        ;
250E    E1                      POP     HL              ;Restore current cursor position
250F    CD 0C1D                 CALL    GETTRM          ;Is this line terminated?
2512    28 37                   JR      Z,INS6          ;Line not terminated on this visual
                        ;
                        ; The current line is terminated. A check must be made to
                        ; determine if a wrapped character is a space, or we're inserting
                        ; at the end-of-line. If so, we have to open a next line to
                        ; insert.
                        ;
2514    79                      LD      A,C             ;Move last character to A for comparison
2515    FE 20                   CP      ' '
2517    F5                      PUSH    AF              ;Save the condition
2518    20 0A                   JR      NZ,INS3         ;No, open next line
251A    3A F3B0                 LD      A,(LINLEN)      ;Are we trying to insert at the EOL?
251D    BC                      CP      H               ;
251E    28 04                   JR      Z,INS3          ;Yes, open next line
2520    F1                      POP     AF              ;Discard stack
2521    C3 09E1                 JP      CKDPCS          ;Display cursor again
2524                    INS3:
                        ;
2524    CD 0C2A                 CALL    UNTERM          ;Unterminate this line
2527    2C                      INC     L               ;Go to next row
2528    C5                      PUSH    BC              ;Save character code
2529    E5                      PUSH    HL              ;Save position of character in operation
252A    CD 0C32                 CALL    GETLEN          ;Bottom of screen?
```

```
252D    BD                      CP      L               ;
252E    38 05                   JR      C,INS4          ;Yes
                        ;
                        ; Scroll down starting at line L
                        ;
2530    CD 0AB7                 CALL    INSLN0          ;Insert a blank line there
2533    18 0F                   JR      INS5
2535                    INS4:
                        ;
                        ; Scroll up
                        ;
2535    21 F3DC                 LD      HL,CSRY
2538    35                      DEC     (HL)
2539    20 01                   JR      NZ,INS45
253B    34                      INC     (HL)
253C                    INS45:
253C    2E 01                   LD      L,1
253E    CD 0A88                 CALL    DELLN0
2541    E1                      POP     HL
2542    2D                      DEC     L
2543    E5                      PUSH    HL
2544                    INS5:
2544    E1                      POP     HL
2545    C1                      POP     BC
2546    F1                      POP     AF              ;Restore flags
2547    CA 09E1                 JP      Z,CKDPCS        ;If we were trying to insert at the
                                                        ;end-of-line, nothing else to do
254A    2D                      DEC     L               ;Cancel next 'INR L'
254B                    INS6:
                        ;
                        ; Not end of logical line, pass character to next line
```

```
 6796                             ;
 6797     254B    2C                        INC     L               ;Bump row counter
 6798     254C    26 01                     LD      H,1             ;Start from first column
 6799     254E    18 AA                     JR      INS1            ;Pass character to next line
```

```
2550                LDELNX:
                    ;
                    ; Delete current character
                    ;
2550    3A F3B0             LD      A,(LINLEN)
2553    BC                  CP      H               ;At rightmost position?
2554    20 05               JR      NZ,LDELX1       ;Nope
2556    CD 0C1D             CALL    GETTRM          ;Is this a terminated line?
2559    20 3A               JR      NZ,DELET5       ;Yes, place a space there.
255B                LDELX1:
255B    3E 1C               LD      A,1CH           ;Move cursor right
255D    DF                  RST     18H
255E    2A F3DC             LD      HL,(CSRY)       ;Fall into 'delete prev. character'
2561                DELETE:
                    ;
                    ; Delete previous character
                    ;
2561    E5                  PUSH    HL
2562    CD 0A2E             CALL    CKERCS
2565    E1                  POP     HL
2566    25                  DEC     H               ;Are we at top of line?
2567    C2 257A             JP      NZ,DELET2       ;No
256A    24                  INC     H               ;Yes
256B    E5                  PUSH    HL              ;Save current cursor position
256C    2D                  DEC     L               ;Look a line above
256D    28 0A               JR      Z,DELET1        ;At top of screen
256F    3A F3B0             LD      A,(LINLEN)
2572    67                  LD      H,A
2573    CD 0C1D             CALL    GETTRM          ;Is previous line terminated?
2576    20 01               JR      NZ,DELET1       ;Yes
```

```
2578    E3                      EX      (SP),HL         ;No, substitue by current HL
2579                    DELET1:
2579    E1                      POP     HL              ;Get saved cursor position
257A                    DELET2:
257A    22 F3DC                 LD      (CSRY),HL       ;Set new cursor position
257D                    DELET3:
257D    3A F3B0                 LD      A,(LINLEN)
2580    BC                      CP      H
2581    28 12                   JR      Z,DELET5        ;Just over strike with blank
2583    24                      INC     H
2584                    DELET4:
2584    CD 0BD8                 CALL    GETVRM          ;Get current character and attribute
2587    25                      DEC     H
2588    CD 0BE6                 CALL    PUTVRM          ;Output it to left of current position
258B    24                      INC     H
258C    24                      INC     H
258D    3A F3B0                 LD      A,(LINLEN)
2590    3C                      INC     A
2591    BC                      CP      H
2592    20 F0                   JR      NZ,DELET4       ;Do next till end of visual
2594    25                      DEC     H
2595                    DELET5:
2595    0E 20                   LD      C,' '           ;Load raw code for space
2597    CD 0BE6                 CALL    PUTVRM
259A    CD 0C1D                 CALL    GETTRM
259D    C2 09E1                 JP      NZ,CKDPCS       ;End of line, all done
25A0    E5                      PUSH    HL
25A1    2C                      INC     L
25A2    26 01                   LD      H,1
25A4    CD 0BD8                 CALL    GETVRM          ;Get first character next visual
25A7    E3                      EX      (SP),HL
```

```
 6862     25A8    CD 0BE6                CALL    PUTVRM          ;Put at last position last line
 6863     25AB    E1                     POP     HL
 6864     25AC    18 CF                  JR      DELET3
```

```
          
25AE                 LERASE:
                     ;
                     ; Erase logical line
                     ;
25AE    CD 0A2E              CALL    CKERCS
25B1    CD 266C              CALL    GTFRST          ;Set L=first visual this logical line
25B4    22 F3DC              LD      (CSRY),HL
25B7    18 05                JR      TRUNC1
25B9                 TRUNC:
                     ;
                     ; Truncate logical line
                     ;
25B9    E5                   PUSH    HL
25BA    CD 0A2E              CALL    CKERCS
25BD    E1                   POP     HL
25BE                 TRUNC1:
25BE    CD 0C1D              CALL    GETTRM          ;Is this line terminated?
25C1    F5                   PUSH    AF              ;Save the condition
25C2    CD 0AEE              CALL    EOL             ;Erase to end-of-line
25C5    F1                   POP     AF              ;Restore condition
25C6    20 05                JR      NZ,DPCSOW       ;Yes
25C8    26 01                LD      H,1             ;Go to next line
25CA    2C                   INC     L               ;Bump row counter
25CB    18 F1                JR      TRUNC1          ;And continue
25CD                 DPCSOW:
                     ;
25CD    CD 09E1              CALL    CKDPCS
25D0    AF                   XOR     A
25D1    32 FCA8              LD      (INSFLG),A
25D4    C3 242D              JP      SETOVW
```

```
25D7                   LAPPND:
                       ;
                       ; Append to current line
                       ;
25D7    CD 0A2E                CALL    CKERCS          ;Erase cursor
25DA    2A F3DC                LD      HL,(CSRY)       ;Get current cursor position
25DD    2D                     DEC     L
25DE                   LAP1:
25DE    2C                     INC     L
25DF    CD 0C1D                CALL    GETTRM          ;Line terminated?
25E2    28 FA                  JR      Z,LAP1          ;No, look at next line
25E4    3A F3B0                LD      A,(LINLEN)
25E7    67                     LD      H,A
25E8    24                     INC     H
25E9                   LAP2:
25E9    25                     DEC     H               ;Reached start of line?
25EA    28 07                  JR      Z,LAP3          ;Yes
25EC    CD 0BD8                CALL    GETVRM          ;Get a character at the cursor
25EF    FE 20                  CP      ' '             ;Space?
25F1    28 F6                  JR      Z,LAP2          ;Yes, skip this
25F3                   LAP3:
25F3    CD 0A5B                CALL    ADVCUR          ;Advance cursor to point to end of line
25F6    18 D5                  JR      DPCSOW          ;Re-display cursor
25F8                   LNXTWD:
                       ;
                       ; Move to next word
                       ;
25F8    CD 0A2E                CALL    CKERCS
25FB    CD 2634                CALL    PRVCHK
25FE                   LNW1:
25FE    CD 2624                CALL    NXTCHK          ;Still in word?
```

```
2601    28 CA                   JR      Z,DPCSOW        ;Reached screen bottom, abort
2603    38 F9                   JR      C,LNW1          ;Yes
2605                    LNW2:
2605    CD 2624                 CALL    NXTCHK          ;Reached word?
2608    28 C3                   JR      Z,DPCSOW        ;Reached screen bottom, abort
260A    30 F9                   JR      NC,LNW2         ;Not yet
260C    18 BF                   JR      DPCSOW
260E                    LBCKWD:
                        ;
                        ; Move to previous word
                        ;
260E    CD 0A2E                 CALL    CKERCS
2611                    LBW1:
2611    CD 2634                 CALL    PRVCHK          ;Still in separator?
2614    28 B7                   JR      Z,DPCSOW        ;Reached screen top, abort
2616    30 F9                   JR      NC,LBW1         ;Yes
2618                    LBW2:
2618    CD 2634                 CALL    PRVCHK          ;Reached separator?
261B    28 B0                   JR      Z,DPCSOW        ;Reached screen top, abort
261D    38 F9                   JR      C,LBW2          ;Not yet
261F    CD 0A5B                 CALL    ADVCUR
2622    18 A9                   JR      DPCSOW
2624                    NXTCHK:
                        ;
                        ; Move right and check
                        ;
2624    2A F3DC                 LD      HL,(CSRY)       ;Get current cursor position
2627    CD 0A5B                 CALL    ADVCUR          ;Advance cursor
262A    CD 0C32                 CALL    GETLEN          ;Get an actual height of screen
262D    5F                      LD      E,A             ;[D],[E] hold the dead end position
262E    3A F3B0                 LD      A,(LINLEN)
```

```
2631    57                      LD      D,A
2632    18 09                   JR      PRVCK1
2634                    PRVCHK:
                        ;
                        ; Move left and check
                        ;
2634    2A F3DC                 LD      HL,(CSRY)       ;Get current cursor position
2637    CD 0A4C                 CALL    BS              ;Regress cursor
263A    11 0101                 LD      DE,0101H        ;[D],[E] hold the dead end position
263D                    PRVCK1:
                        ;
                        ; Check current character
                        ; Carry set if the character is regarded as separator
                        ;
263D    2A F3DC                 LD      HL,(CSRY)       ;Get updated cursor position
2640    E7                      RST     20H             ;Reached dead end?
2641    C8                      RET     Z               ;Yes, return with Z flag
2642    11 2668                 LD      DE,RESZRO       ;Jump to RESZRO when done
2645    D5                      PUSH    DE
2646    CD 0BD8                 CALL    GETVRM          ;Get ASCII code of character at [H],[L]
2649    FE 30                   CP      '0'             ;Set carry if  "0".."9"
264B    3F                      CCF
264C    D0                      RET     NC
264D    FE 3A                   CP      ':'
264F    D8                      RET     C
2650    FE 41                   CP      'A'             ;Set carry if "A".."Z"
2652    3F                      CCF
2653    D0                      RET     NC
2654    FE 5B                   CP      'Z'+1
2656    D8                      RET     C
2657    FE 61                   CP      'a'             ;Set carry if "a".."z"
```

```
2659    3F                      CCF
265A    D0                      RET     NC
265B    FE 7B                   CP      'z'+1
265D    D8                      RET     C
265E    FE 86                   CP      86H             ;Check for Hiragana (86H)
2660    3F                      CCF
2661    D0                      RET     NC
2662    FE A0                   CP      0A0H
2664    D8                      RET     C
2665    FE A6                   CP      0A6H
2667    3F                      CCF
2668                    RESZRO:
2668    3E 00                   LD      A,0             ;Reset Z flag without affecting C flag
266A    3C                      INC     A
266B    C9                      RET
                        ;
                        ; Set H,L to first visual line in logical line
                        ;
266C                    GTFRST:
266C    2D                      DEC     L               ;Look a line just above
266D    28 05                   JR      Z,GTFST1        ;If we're at top of screen, all done
266F    CD 0C1D                 CALL    GETTRM          ;Get terminator
2672    28 F8                   JR      Z,GTFRST        ;More to get above in this logical
2674                    GTFST1:
2674    2C                      INC     L               ;L=line number of first visual
2675    3A FBCA                 LD      A,(FSTPOS)      ;Get first line
2678    BD                      CP      L               ;Same?
2679    26 01                   LD      H,1             ;Assume not
267B    C0                      RET     NZ              ;Good assumption
267C    2A FBCA                 LD      HL,(FSTPOS)     ;Get first line and column
267F    C9                      RET
```

```
 7020                                     END
```

MSX BIOS CROSS REFERENCE

```
 - BIOS CROSS REFERENCE LISTING -

ACTION       1#   2664    3518#
ADVCUR       1#   1930    2166#   6917    6947    6954
ALPJMP       1#   2892#   3041
ASCPCT1      1#   5236
ASCPCT2      1#   5238
ATRBAS       1#   1163    1255    1296    1387    1437
ATRBYT       1#   4407    4725    4756    5113    5168    5215    5256    5404    5431
AUTFLG       1#   6476    6599
BAKCLR       1#   1574    1584    1660    1684
BASROM       1#    923    2571    6699
BDRCLR       1#   1690
BEEP         1#    170    1914    3485#
BEGIN       30#
BIT0         1#   5517    5523#
BIT1         1#   5516    5519    5520    5533#
BIT1OT       1#   5494    5535    5542    5544#
BITOUT       1#   5511    5530    5552#
BRDATR       1#   5259    5282    5377    5427
BREAKX       1#    167    1008#   1733    5500    5521    5666    5672    5712    5755    5778
BS           1#   1916    1932    2144#   2297    6965
BUF          1#   6610    6692
BUFEND       1#   2087    2391    2478    2497
BUFMIN       1#   6513
CALATR       1#    136    1430#
CALBAS       1#    252     363#   2768    5803    5871
CALESL       1#    412     419#
CALLF        1#     90     366#
CALPAT       1#    135    1413#
CALSLT       1#     57     365     404#    437
CAPST        1#   3055    3193    3275
CGCAP1       1#   3201    3203#
CGPBAS       1#   1140    1159    1471    2083
```

```
CGPNT       1#   1473   1474   1521   1525
CGSND1      1#   3247   3249#
CGTABL      1#     40   5883#
CHCLTX      1#   1650   1677#
CHGBD1      1#   1687   1691#
CHGBDR      1#   1570   1583   1652   1688#
CHGCAP      1#    237   3198#
CHGCLR      1#    119   1141   1164   1644#
CHGET       1#    157   3403#  6502
CHGET1      1#   3414#  3416
CHGET2      1#   3412   3418#
CHGET3      1#   3422   3424#
CHGMOD      1#    118   1704   1714#
CHGSND      1#    238   3244#
CHKBUF      1#   2799#  2823
CHKCHG      1#   5300   5313   5385   5401#
CHKEOC      1#   3980   4003#
CHKMOD      1#   4442   4460   4520   4533#  4589   4683   4736   4787   4803   4818   4833
          4859   4883   4898   4925   5070   5247   5268   5370
CHKRAM      1#     31    681#
CHKSCR      1#   1544   1700   1820   2071   2126   2451#  2813
CHPLP1      1#   1732#  1736
CHPLP2      1#   1738#  1753
CHPUT       1#    158   1813#  5880
CHPUT1      1#   1825   1837#  2195
CHPUT3      1#   1842   1850#
CHRGTR      1#     51
CHSNS       1#    156   2807#  3411   3415
CHSNS1      1#   2814   2822#
CKCNTC      1#    169   3431#
CKDPC0      1#    943   2051#  3413
CKDPCS      1#   1826   2059#  6562   6658   6757   6790   6856   6892
```

```
CKERC0      1#    953    2106#   3417
CKERCS      1#   1822    2114#   6559    6608    6718    6819    6870    6879    6900    6923    6938
CKRM05      1#    717#    778
CKRM10      1#    732#    774
CKRM15      1#    725     729     737#
CKRM20      1#    739#    748     750
CKRM25      1#    746     751#
CKRM30      1#    758     766#
CKRM35      1#    769     775#
CKRM50      1#    801#    854
CKRM55      1#    810#    850
CKRM60      1#    805     812#
CKRM65      1#    814#    823     827
CKRM70      1#    821     828#
CKRM75      1#    835     843#
CKRM80      1#    846     851#
CKSTTP      1#    963     983#   6696
CLICKW      1#   3241#   3243
CLIKFL      1#   2718    3234    3238
CLIKSW      1#   3231
CLOC        1#   4614    4652    4663    4672    4843    4858    4882    4897    4924    4936    4989
        5048
CLPRIM      1#    418
CLRSPR      1#    126    1372#
CLRTX1      1#   1555    1557#
CLRTX2      1#   1563#   1566
CLRTXT      1#   1142    1165    1547#   1924    1941    1943
CLS         1#    171    1705#
CLSHRS      1#   1265    1545    1568#
CLSMLT      1#   1314    1546    1581#
CLSPR2      1#   1389#   1411
CLSPR3      1#   1403    1407#
```

```
CLSSUB      1#    1542#   1711
CMASK       1#    4598    4634    4662    4671    4846    4992
CNSDFG      1#    2363    2450    2594    2818
CNTFUL      1#    5585    5616    5774#   5782
CNTHL0      1#    5708    5757#   5784
CNTHL1      1#    5759#   5765    5785
CNTHLF      1#    5678    5681    5709    5749#
CNTPUT      1#    1849    1869#
CNTTBL      1#    1912#
CNVCH1      1#    1797#   1807
CNVCH2      1#    1795    1800#
CNVCH3      1#    1791    1803#
CNVCHR      1#     161    1781#   1839    2421    4397    5848
CODSAV      1#    2076    2129
CR          1#    1926    2206#   2221    2257
CRTCNT      1#    2355    2376    2439    2596
CSAVEA      1#    5303    5321    5353
CSAVEM      1#    5304    5322    5354
CSDLY1      1#    3504    3506#   3515
CSHOME      1#    1560    1922    1965    2201#
CSRSW       1#    2049    2055    2063    2110    2118
CSRX        1#    1827    2004    2196    6528
CSRY        1#    1851    1901    2073    2128    2183    2375    6493    6719    6776    6813    6835
          6872    6901    6953    6964    6972
CSTYLE      1#    2041    2089    6561
CTWOF1      1#    5455#   5459
CURLIN      1#     995    3116
DATAR       1#    5665#   5670
DATAR0      1#    5671#   5676
DATAR1      1#    5679#   5685    5687
DATARL      1#    5694#   5711
DATAW     5502#
```

```
 - BIOS CROSS REFERENCE LISTING -

DATAWL        1#    5514#   5518
DCOMPR        1#      59    4146#
DELET1        1#    6826    6830    6832#
DELET2        1#    6822    6834#
DELET3        1#    6836#   6864
DELET4        1#    6841#   6850
DELET5        1#    6809    6839    6852#
DELETE        1#    6569    6814#
DELLN0        1#    1868    2222#   6782
DELLN1        1#    2240#   2249
DIOERR        1#    5870
DISSC1        1#    1175    1182#
DISSCR        1#     108    1131    1150    1176#   1249    1290
DLN        1953     2215#
DOWN          1#    1864    1936    1959    2173#
DOWN1         1#    2180    2185#
DOWNC         1#     216    4876#
DPCSOW        1#    6886    6890#   6918    6927    6931    6933    6941    6945    6948
DSFKCL        1#    2395#   2398
DSPCS1        1#    2091    2093#   2098
DSPCSR        1#    2058    2066#
DSPFK1        1#    2386    2389#
DSPFK2        1#    2413#   2437
DSPFK4        1#    2405#   2408
DSPFK5        1#    2417#   2422    2430
DSPFK6        1#    2425    2428#
DSPFK8        1#    2423    2426#
DSPFKE        1#    2403    2411    2438#
DSPFNK        1#     175    2366#   2821
DWNC10        1#    4869    4873    4885#
EASYTB        1#    2937#   3163
ELN           1#    1949    2226    2250    2263    2289    2301#   2356
```

```
EMSITB   1124#
ENASCR      1#    109    1145    1169#   1268    1317
ENASLT      1#     61     476#    498     976
ENESLT      1#    478     484#
ENSTOP      1#   2761
ENTESC      1#   1928    1982#
EOCCHK      1#   4011#   4014
EOL         1#   1945    2308#   2334    6884
EOP         1#   1947    2327#   2342
ERACSR      1#   2113    2121#
ERAFNK      1#    174    2346#
ERASPR      1#   1167    1266    1315    1384#
EREOL1      1#   2319#   2325
ESCCNT      1#   1843    1984
ESCTBL      1#   1939#   1992
EXABO1      1#    964     971#
EXCABO      1#    936     960#
EXPTBL      1#    364     880     974
FAKECR      1#   6667#   6698    6701
FETCHC      1#    220    4418    4656#   4681    4737    4789    4805    4820    4835    4946    4957
         4969    4979    5073    5302    5318    5352
FILOU1      1#   5802
FILVRM      1#    115    1383    1559    1575    1580    1664#
FKTABL      1#   4071
FLPMOT      1#   4043    4052#
FLVRM1      1#   1667#   1674
FNKDEF   4075#
FNKFLG      1#   3093
FNKINT      1#   3097    3114#
FNKSB       1#    173    1567    2359#
FNKSTR      1#   2384    2387    3104    4070
FNKSWI      1#   2390    2815
```

```
 - BIOS CROSS REFERENCE LISTING -

FORCLR        1#    1385    1655    1679    4406
FORMAT        1#     246    4201#
FSTPOS        1#    2236    6498    7014    7018
GENCLK        1#    3218    3230#
GET1L1        1#    2464    2476#
GET1LN        1#    2243    2282    2465#
GET8B         1#    2086    2458#
GETBAK        1#    4293    4321#
GETLEN        1#    2007    2177    2223    2259    2336    2590#   6764    6955
GETPAT        1#    1506#   4405
GETPNT        1#    1005    1027    2803    3226    3425    3428
GETPTR        1#    4263    4291    4351    4364#
GETQ          1#    3678    4287#
GETTRM        1#    2231    2269    2562#   2586    6644    6688    6741    6808    6829    6855    6882
          6905    7010
GETVC1        1#    1096    4168    4176#
GETVC2        1#     250    4169#
GETVCL        1#    4190#   4193
GETVCP        1#     249    3547    4161#
GETVCX        1#    4188    4194#
GETVRM        1#    2075    2501#   6618    6725    6842    6860    6913    6977
GETYPR        1#      63
GICIN1        1#    1083#   1109
GICINI        1#     146    1056#   3505    6695
GORSET        1#    2000    2031#
GOSET         1#    1998    2020#
GPRT05        1#    4399    4404#
GPRT10        1#    4416#   4440
GPRT20        1#    4422#   4431
GPRT30        1#    4428    4432#
GPRT40        1#    4437    4441#
GPRT50        1#    4444    4448#
```

```
GPRT60       1#    4447     4452#
GPRT70       1#    4462     4465#
GPRT80       1#    4468     4470#
GRPACX       1#    4410     4443     4453     4459
GRPACY       1#    4408     4461     4471
GRPATR       1#    1254
GRPCGP       1#    1576     4612     4862     4901
GRPCOL       1#    1573
GRPCR        1#    4401     4446     4451     4456#
GRPDIF       1#    4688     5111     5115     5153     5202
GRPHED       1#    1787
GRPNAM       1#    1256     1283
GRPPAT       1#    1252
GRPPRT       1#     138     4389#
GRPTAB       1#    3365     3377#
GSPAD1       1#    1422     1425#
GSPSIZ       1#     137     1420     1440#
GTASPC       1#     228     5232#
GTFRST       1#    6598     6871     7007#    7011
GTFST1       1#    7009     7012#
GTPAD        1#     186     3867#
GTPAD0       1#    3893     3899#
GTPAT1       1#    1526#    1540
GTPDL        1#     187     3807#
GTPDP1       1#    3888     3891#
GTROW8       1#    2689     3698     3726#    3804
GTSTCK       1#     184     3683#
GTTRIG       1#     185     3783#
H.CHGE       1#    3410
H.CHPU       1#    1819
H.DSPC       1#    2070
H.DSPF       1#    2370
```

```
(MSX BASIC ROM BIOS )   Macro-80                                PAGE XREF - 9
 - BIOS CROSS REFERENCE LISTING -

H.ERAC       1#    2125
H.ERAF       1#    2350
H.FORM       1#    4203
H.INIP       1#    1470
H.INLI       1#    6492
H.ISFL       1#    4139
H.KEYC       1#    2993
H.KEYI       1#    2621
H.KYEA       1#    3160
H.LPTO       1#    1730
H.LPTS       1#    1759
H.NMI        1#    4061
H.OUTD       1#    5798
H.PHYD       1#    4199
H.PINL       1#    6475
H.QINL       1#    6486
H.TIMI       1#    2625
H.TOTE       1#    1703
HEADER       1#    5485
HIGH         1#    5551
HRSSCL       1#    4521     4526#
HRZMOV       1#    4811     4841#
HRZMV1       1#    4791     4807     4822     4837     4845#
ILN          1#    1951     2251#
INDJMP       1#    1889#    1897     1994     6505
INESC        1#    1846     1987#
INESC1       1#    1989     1995#
INESC2       1#    2005     2009#
INGI         1#    1051     3476#    3723     3915     3987     4012
INIFNK       1#      99     4065#
INIGR1       1#    1260#    1263     1264
INIGRP       1#     129     1245#    1722
```

```
INIML1       1#    1300#   1313
INIML2       1#    1302#   1310
INIML3       1#    1305#   1308
INIMLT       1#     130    1286#   1723
INIPAT       1#    1143    1166    1466#
INIPT1       1#    1477#   1490
INIT         1#     919
INIT32       1#     128    1146#   1720
INITIO       1#      98    1038#
INITQ        1#    1088    4328#
INITXT       1#     127    1127#   1719
INLIN        1#     164    6478    6491#
INLIN1       1#    6481    6497#
INLIN2       1#    6501#   6509
INLOT0       1#    6538    6546#
INLOT1       1#    6540    6549#
INLOUT       1#    6507    6519#   6527
INS1         1#    6721#   6799
INS2         1#    6723#   6735
INS3         1#    6752    6755    6758#
INS4       6766    6772#
INS45        1#    6778    6780#
INS5         1#    6771    6786#
INS6         1#    6742    6793#
INSERT       1#    6544    6714#
INSFLG       1#    6536    6677    6708    6894
INSLN0       1#    2258#   6770
INSLN1       1#    2279#   2288
INTCNT       1#    2638    2647
INTFLG       1#     927     944    3217    3419    6500
INTRET       1#    2624    2672    2720    2723    2731#
INTVAL       1#    2645
```

```
ISCNTC      1#    168     922#   3437
ISFLIO      1#    247    4135#   5799
JFLVRM      1#   1579#   1593
JIFFY       1#   2651    2653
JMPBC       1#   1887    1902    1905#
JMPWRT      1#   5191    5204    5218#
JPPPAL      1#   4398    4403    4412    4454#   4472
JPUTCH      1#   3025    3032#   3060
KAIUEO      1#   3260    3265#
KANAMD      1#   1053    3256
KANANO      1#   3267    3290#
KANASF      1#   3269    3307#
KANAST      1#   3002    3173
KANJNO      1#   3261    3324#
KANJSF      1#   3263    3341#
KEEPH       1#   5560#   5562
KEEPL       1#   5555#   5557
KEYANY      1#   2795    2828#
KEYBUF      1#   3401
KEYCHK      1#   2719    2746#
KEYCK1      1#   2753#   2760
KEYCK2      1#   2773#   2779
KEYCK3      1#   2778    2781#
KEYCK4      1#   2730    2780    2785#
KEYCK5      1#   2789#   2798
KEYCOD      1#   2848    2983#
KEYINT      1#     97    2603#
KEYNOM      1#   2896    3053#
KEYSFT      1#   2895    3050#
KEYTRG      1#   3786    3802#
KILBUF      1#    251     962    1002#
KSTKTB      1#   3703    3765#
```

```
KY1CNT   2900    2914#
KY1NOM   2902    2903#
KY1SFC   2899    2924#
KY1SFT   2901    2908#
KYALP       1#   2865    3034#
KYANY1      1#   2844#   2852
KYC1TB      1#   2898#   3063
KYCLA0      1#   2999    3005#
KYCLAS      1#   2995    3007#   3017
KYCLS    2881    3150#
KYCOD1      1#   2863    3061#
KYEASY      1#   2867    2875    2879    2883    3156#
KYFNC1      1#   3086    3090#
KYFNC2      1#   3098#   3120
KYFNC3      1#   3107#   3113
KYFUNC      1#   2873    3080#
KYGRAP      1#   3001    3360#
KYJTAB      1#   2859#   2992
KYKAN1      1#   3262    3264    3268    3270#
KYKANA      1#   3004    3252#
KYKLOK      1#   2871    3169#
KYLOCK      1#   2869    3189#
KYNUM       1#   2861    3018#
KYSTCK      1#   3686    3696#
KYSTOP      1#   2877    3206#
KYSTP1      1#   3214    3216#
LAP1        1#   6903#   6906
LAP2        1#   6910#   6915
LAP3        1#   6912    6916#
LAPPND      1#   6579    6896#
LBCKWD      1#   6575    6934#
LBLKSP      1#   6626    6637    6648#   6655
```

```
LBREAK      1#   6583    6687#
LBREK0      1#   6682#   6689
LBW1        1#   6939#   6942
LBW2        1#   6943#   6946
LCR1        1#   6613#   6647
LCR2        1#   6615#   6642
LCRNRM      1#   6624    6633#
LCRNUL      1#   6622    6638#
LCRRET      1#   6585    6592#
LDELNX      1#   6589    6801#
LDELX1      1#   6807    6810#
LDIMV1      1#   1457#   1464
LDIRMV      1#    116    1452#   2479
LDIRVM      1#    117    1493#   2498
LDIVM1      1#   1497#   1504
LEFT        1#   1963    2148    2153#
LEFTC       1#    213    4828#
LEFTC1      1#   4826    4838#
LERASE      1#   6587    6866#
LF          1#   1860#   1920
LFTQ        1#    205    4347#
LINL32      1#   1154
LINL40      1#   1135
LINLEN      1#   1136    1155    2003    2139    2150    2323    2401    2474    2492    2542    6640
         6731    6753    6805    6827    6837    6847    6907    6957
LINTTB      1#   1561
LNW1        1#   6925#   6928
LNW2        1#   6929#   6932
LNXTLN      1#   6670#   6702
LNXTWD      1#   6577    6919#
LOC         1#   1955    1979#
LOW         1#   5506    5529
```

```
LOWLIM      1#   5638    5663
LPT.DW      1#    623#   1740
LPT.SB      1#    624#   1055    1742    1744
LPT.ST      1#    625#
LPTABO      1#   1734    1748#
LPTCH0      1#   5833    5836    5843#
LPTCH1      1#   5811    5846    5864#
LPTCHR      1#   5852    5858    5861    5867#   5874
LPTCOD      1#   5800    5805#
LPTOUT      1#    159    1726#   5868
LPTPOS      1#   1751    5824    5837    5841
LPTSTT      1#    160    1735    1757#
MAPSPC      1#   5850    5854    5872#
MAPXYC      1#    219    4413    4540#
MDNC        1#   4884    5006    5010    5016#
MHCMOV      1#   4966    4987#
MHZMV1      1#   4949    4960    4972    4982    4991#
MLFTC       1#   4834    4977#
MLFTC1      1#   4975    4983#
MLTATR      1#   1295
MLTCGP      1#   1591    4650    5001    5025
MLTNAM      1#   1297    1333
MLTPAT      1#   1293
MMPXY1      1#   4631    4633#
MMPXYC      1#   4590    4624#
MNSTCX      1#   5071    5221#   5230
MORACT      1#   3577#   3596    3619    3631
MORSPL      1#   5821#   5826
MOTRON      1#   4045    4048#
MOTRWT      1#   5479#   5483
MREADC      1#   4684    4706#
MRGTC       1#   4804    4955#
```

```
MRGTC1        1#    4953     4961#
MSCANL        1#    5371     5411#    5420
MSCANR        1#    5269     5336#    5347
MSCNR1        1#    5341     5350#
MSCNR2        1#    5356#    5361
MSETC         1#    4738     4745#
MSETC1        1#    4759     4764#
MTDNC         1#    4860     4996#
MTLFT         1#    4819     4967#
MTRGT         1#    4788     4941#
MTSBRD        1#    5340     5360     5417     5421#
MTUPC         1#    4899     5022#
MUPC          1#    4926     5030     5033     5038#
MUSCLL        1#    1073#    1076
MUSICF        1#    1070     2657     3642     3657     3670
MUSINT        1#    2660#    2669
MUSITB        1#    1098     1114#
MVTMOV        1#    5021     5043#
MVTMV1        1#    5045     5047#
NAMBAS        1#    1138     1157     1553     2557
NEWKEY        1#    2752     2771     2788
NMI           1#     124     4057#
NMSFTB        1#    2885#    3027
NOKEY         1#    3187#
NONEG1        1#    3935     3938#
NONEG2        1#    3943     3946#
NOSTOP        1#    2763     2766     2769#
NOTABL        1#    5819     5830#
NOTAUT        1#    6601     6603#
NOTRAN        1#    5730     5738#
NSETCX        1#     227     5055#    5328     5395
NSTC10        1#    5080#    5084
```

```
NSTC20       1#   5076    5094#
NSTC30       1#   5108#   5117
NSTC40       1#   5107    5118#
NSTC50       1#   5125    5139#
NSTCSP       1#   5082    5130#
NTBKS2       1#   5817#
NTBOTM       1#   2379    2381#
NTHIRA       1#   5856    5859#
NTINTT       1#   2642    2646#
NTMSXP       1#   5844
NXTCHK       1#   6926    6930    6949#
OLDKEY       1#   1029    1031    2725    2726    2787
OLDSCR       1#   1134    1153    1702
ONBRD1       1#   4797    4827    4914#   4954    4976
ONBRDR       1#   4874    4912#
ONGSBF       1#   3145    3147
OUTDLP       1#    248    5814#   5823
OUTDO        1#     55    5788#
OUTGI        1#   3986    3995    3999    4010    4017#
OUTNTB       1#   6523    6533#
OUTTAB       1#   6525#   6531
PADX         1#   3895    3950
PADX1        1#   3926    3928    3932    3953#
PADY         1#   3897    3952
PATBAS       1#   1161    1253    1294    1380    1427
PATWR1       1#   5181    5205#
PATWRK       1#   1523    4414
PATWRT       1#   4740    5089    5142#
PBDHRT       1#   1832#   2853    3430    3679
PDL1         1#   3833#   3835
PDL2         1#   3856#   3861
PDL3         1#   3859    3863#
```

```
PDLP1       1#   3840    3843#
PHYDIO      1#    244    4197#
PINLIN      1#    163    6471#
PLYCNT      1#   3660
PNTHRS      1#   5248    5253#
PNTINI      1#    229    5242#
PNTIRT      1#   5252    5258#
POPALL      1#   1113    1821    1830#   4455
POSIT       1#    172    1766#   6673
PPI.AR      1#    257#    292     334     413     479     552     614#    763     776     799     840
         852      891    4118
PPI.AW      1#    258#    354     358     447     451     482     556     582     617#    694     718
         786      802     863     894     899     904     909     914    4122
PPI.BR      1#    615#   1016    1022    2756    3743    4132
PPI.CM      1#    619#    692    3204    3239    3250    4050    5465    5477    5559    5564    5573
PPI.CR      1#    616#   1012    1019    2748    3739    4054    4128
PPI.CW      1#    618#    696    1015    1021    2755    3742    4131
PRTFLG      1#   5806
PRVCHK      1#   6924    6940    6944    6960#
PRVCK1      1#   6959    6967#
PSG.DR      1#    607#   3179    3483    3857    4030    5668    5674    5728    5763    5780
PSG.DW      1#    606#   3186    3472    3722    3850    3852    3912    4034
PSG.LW      1#    605#   3178    3469    3482    3854    4029    5576
PSG.PA      1#    609#   3480
PSG.PB      1#    610#   3710    3844    3907    4028
PTRFIL      1#   4141
PUT1L1      1#   2486    2494#
PUT1LN      1#   2245    2284    2440    2487#
PUT8B       1#   2102    2482#
PUTCHR      1#   2893    2894    3033    3078    3111    3155    3168    3219#   3273    3373    3375
PUTPNT      1#   1004    1026    2804    3223    3229
PUTQ        1#    206    4259#
```

```
PUTVRM     1#   1854    2105    2131    2300    2512#   6729    6844    6854    6862
QINLIN     1#    166    6482#
QSTART     1#   4333    4368    4377#
QUEBAK     1#   4324
QUEUEN     1#   3560    3674
QUEUES     1#   4384
RAMLOW     1#    296     871     872
RAWPRT     1#   5809
RDBIT      1#   5695    5715#
RDBITL     1#   5727#   5735    5744
RDESLT     1#    291     299#
RDPSG      1#    148    3481#   3712    3846    3909
RDSLT      1#     49     289#    304    1482    1531
RDVDP      1#    241    4112#
RDVRM      1#    111    1606#   4685    4690    4708    4750    5151    5155
READC      1#    225    4674#   5285    5310    5382    5425
READC0     1#   4696#   4712
READC1     1#   4694    4701#   4711
READYR     1#   2767
REDCOD     1#   3927    3930    3959#
REDLOP     1#   3983#   3996
REDPAD     1#   3924    3925    3964    3969    3975#
REPCNT     1#   1033    2721    2784
REQSTP     1#    965     988     991
REQTRP     1#    967    2634    2644    2701    2704    2707    2710    2713    3127#
RESZRO     1#   6975    7000#
RETRET     1#   5495    5531#
RETURN     1#   6516#   6573
RG0SAV     1#   1205    1214    1232    1273    1322
RG1SAV     1#   1173    1180    1219    1237    1278    1327    1376    1400    1444
RGHTC1     1#   4796    4808#
RGTEXT     1#   5122    5126#
```

```
 - BIOS CROSS REFERENCE LISTING -

RIGHT         1#    1855    1961    2135#   2170
RIGHTC        1#     212    4798#   5227    5390    5418
RSET10        1#    2038    2043#
RSLREG        1#     239    4116#
RSTFL1        1#    3645#   3647
RSTMOD        1#    1969    1976#
RUBOUT        1#    1853    2293#
RUNFLG        1#    3902    4023    5279
SAMEBG        1#    5170    5192#
SAMEFG        1#    5176    5185    5198#
SAVSTK        1#     979
SCALXY        1#     218    4411    4475#
SCANL         1#     231    5364#
SCANL1        1#    5379#   5387
SCANL2        1#    5384    5388#
SCANL3        1#    5381    5391#
SCANL4        1#    5334    5397#
SCANR         1#     230    5261#
SCANR1        1#    5284#   5293
SCANR2        1#    5287    5296#
SCANR3        1#    5306#   5314
SCANR4        1#    5309    5312    5315#
SCITBL        1#    6503    6564#
SCLXOK        1#    4513    4518#
SCLYOK        1#    4497    4502#
SCNCNT        1#    2670
SCRMOD        1#    1133    1152    1251    1292    1551    1648    2455    2540    4537
SELEXP        1#     301     342     420     486     544#
SELPRM        1#     290     331     411     477     500#
SETATR        1#     224    4714#
SETC          1#     226    4425    4727#   5226    5435
SETCHK        1#    2352    2372    2446#
```

```
SETGRP     1#    133   1267   1269#  1336
SETINS     1#   6554#  6713
SETMLT     1#    134   1316   1318#
SETMOD     1#   1967   1970#
SETOVW     1#   6556#  6690   6712   6895
SETRD      1#    113   1454   1610   1630#  2505
SETREG     1#   1337   1345   1347#
SETRG1     1#   1340   1343   1349#
SETRG2     1#   1357#  1360
SETSCM     1#   1227   1244   1285   1335#
SETT32     1#    132   1168   1228#
SETTRM     1#   1858   2584#
SETTXT     1#    131   1144   1210#
SETWRT     1#    114   1257   1298   1472   1496   1600   1615#  1666   2317   2516
SFTKEY     1#   2382   2764   2816   2996   3022   3036   3064   3084   3151   3210   3258
SLEXP1     1#    563#   566
SLPRM1     1#    509#   513
SLPRM2     1#    531#   533
SLSTC1     1#   3713   3717#
SLSTC2     1#   3716   3721#
SLSTCK     1#   2678   2682   3687   3705#  3789
SLTTBL     1#    430    492    915    917
SNSMAT     1#    242   4124#
SSLTLP     1#    881#   887
STATFL     1#   2631
STCSSW     1#   2029   2048#
STICK1     1#   3689#  3704
STKTBL     1#   3688   3747#
STMOT1     1#   4044#  4056
STMOTR     1#    199   4041#
STOCSR     1#   1866   2143   2165   2182#  2189   2211
STOP       1#    981
```

```
STOREC      1#    222    4435    4665#   5323    5331
STRTMS      1#    149    3651#
STSTYL      1#   2027    2040#
SULOP       1#   5649#   5652
SYN05       1#   5577#   5589    5591    5602
SYN10       1#   5583#   5606
SYN11       1#   5597    5600#
SYN20       1#   5608#
SYN30       1#   5615#   5620
SYNCHR      1#     46
SYNCW1      1#   5486    5489#
SYNLP1      1#   5493#   5499
T32ATR      1#   1162
T32CGP      1#   1158
T32COL      1#   1662
T32NAM      1#   1156    1242
T32PAT      1#   1160
TAB         1#   1918    2190#   2199
TAPIN       1#    194    5659#
TAPIOF      1#    195    5462#
TAPION      1#    193    5568#
TAPOFF      1#    198    5450#
TAPOON      1#    196    5469#
TAPOUT      1#    197    5501#
TDOWNC      1#    217    4436    4850#
TERMIN      1#   2314    2579#   6495    6672
TGLINS      1#   6571    6704#
TIMOUT      1#   5762    5770#
TLEFT       1#   4812#   5380    5415
TOTEXT      1#    176     973    1696#
TRGFLG      1#   2694
TRIG1       1#   3794    3796#
```

```
 - BIOS MSX CROSS REFERENCE LISTING -

TRIG2          1#    3798#   3806
TRIGHT         1#    4426    4781#   5292    5308    5346    5358
TRPTBL         1#    2633    2643    2700    2703    2706    2709    2712    3121
TRUNC          1#    6581    6874#
TRUNC1         1#    6873    6881#   6889
TRYAGN         1#    3914    3921#   3940    3948
TTYCHR         1#    5808    5875#
TTYPOS         1#    1829
TUPC           1#     215    4890#
TWOPWR         1#    4595    4617#
TXTCGP         1#    1139
TXTNAM         1#    1137    1225
UNTERM         1#    2582#   6760
UP             1#    1934    1957    2159#
UPC            1#     214    4918#
UPC10          1#    4908    4911    4927#
UPDATE         1#    3225    3393#   3427
V.COLR         1#     600#
VADDR          1#    2316    2473    2491    2504    2515    2521#
VADDR1         1#    2543    2546#
VADDR2         1#    2545    2550#
VCBA           1#    3666    4184
VCBB           1#    3667
VCBC           1#    3668
VDP.CW         1#     597#   1197    1200    1623    1627    1638    1641
VDP.DRW        1#     596#   1261    1306    1458    1485    1499    1604    1613    1669    2321    2508
         2518
VDP.SR         1#     598#   2622    4114
VOICAQ         1#    1080
VOICEN         1#    4175
VOICOF         1#    3563    3632#
VRTMOV         1#    4889    4931#
```

```
 - BIOS MSX CROSS REFERENCE LISTING -

VRTMV1      1#    4933    4935#
WATINT      1#     945#    951
WINWID      1#    5655    5724
WORK1       1#    5326
WORK2       1#    5325    5393
WORK3       1#    5281    5376    5398    5409
WRESED      1#     305     347#
WRESLT      1#     332     339#
WRPRIM      1#     338
WRSLT       1#      53     329#    346
WRTPSG      1#     147    1044    1047    1050    1092    1112    3443#   3493    3496    3499    3502
        3591    3594    3608    3612    3625    3628    3640
WRTVDP      1#     110    1186#   1218    1224    1236    1241    1277    1282    1326    1332    1365
        1379    1694
WRTVRM      1#     112    1393    1397    1409    1595#   4766    5110    5114    5219
WSLREG      1#     240    4120#
WTPTAB      1#    5201#   5212
XEPER       1#    3601    3613#
XGETQ       1#    3561    3573    3580    3587    3620    3622    3672#
XNEGTV      1#    4508    4516#
XPOSTV      1#    4506    4509#
XVOL        1#    3583    3597#
YNEGTV      1#    4492    4500#
YPOSTV      1#    4490    4493#
ZERLP1      1#    5832    5840#
```

MSX BIOS SYMBOL TABLE

042C	ABORT	10F9	CKCNTC	0A88	DELLN0
F847	ARG	FBD9	CLIKFL	FD99	DEVICE
F7B5	ARYTA2	F3DB	CLIKSW	F662	DIMFLG
F6C4	ARYTAB	F935	CLINEF	0577	DISSCR
F40B	ASCPCT1	F3B2	CLMLST	F665	DONUM
F40D	ASCPCT2	F92A	CLOC	F6B5	DOT
F931	ASPECT	F38C	CLPRIM	0A61	DOWN
F928	ATRBAS	06A8	CLRSPR	172A	DOWNC
F3F2	ATRBYT	0848	CLS	FCBD	DRWANG
F6AA	AUTFLG	F92C	CMASK	FCBB	DRWFLG
F6AD	AUTINC	F936	CNPNTS	FCBC	DRWSCL
F6AB	AUTLIN	F3DE	CNSDFG	F699	DSCPTR
F3EA	BAKCLR	08B0	CNVCH1	F698	DSCTMP
FBB1	BASROM	08B2	CNVCH2	0B2B	DSPFNK
F3EB	BDRCLR	08B4	CNVCH3	1B63	DUTDLP
1113	BEEP	089D	CNVCHR	0570	ENASCR
FC48	BOTTOM	FBCC	CODSAV	025E	ENASLT
FCB2	BRDATR	F66A	CONLO	267F	ENDBIOS
046F	BREAKX	F668	CONSAV	F660	ENDBUF
3FDC	BRKTXT	F666	CONTXT	F6A1	ENDFOR
F55E	BUF	F669	CONTYP	F40F	ENDPRG
FC18	BUFEND	F939	CPCNT	FFCA	ENDWRK
F55D	BUFMIN	F93B	CPCNT8	026B	ENESLT
06F9	CALATR	F938	CPLOTF	FBB0	ENSTOP
01FF	CALBAS	F93D	CRCSUM	0989	ENTESC
022E	CALESL	F3B1	CRTCNT	0B15	ERAFNK
0205	CALLF	F3FC	CS120	F414	ERRFLG
06E4	CALPAT	F942	CSAVEA	F6B3	ERRLIN
0217	CALSLT	F944	CSAVEM	F6B7	ERRTXT
FCAB	CAPST	F941	CSCLXY	FCC1	EXPTBL
FCB1	CASPRV	FCA9	CSRSW	F7F8	FACLO
F933	CENCNT	F3DD	CSRX	F7C5	FBUFFR
F924	CGPBAS	F3DC	CSRY	1639	FETCHC
F91F	CGPNT	F93F	CSTCNT	F871	FILNM2
1BBF	CGTABL	FCAA	CSTYLE	F860	FILTAB
0F3D	CHGCAP	F41C	CURLIN	0815	FILVRM
07F7	CHGCLR	F945	CXOFF	13A9	FKTABL
10CB	CHGET	F947	CYOFF	FCAE	FLBMEM
084F	CHGMOD	F7F6	DAC	F6A6	FLGINP
0F7A	CHGSND	F6A3	DATLIN	FBCE	FNKFLG
0D62	CHKBUF	F6C8	DATPTR	0B26	FNKSB
02D7	CHKRAM	146A	DCOMPR	F87F	FNKSTR
0B9F	CHKSCR	F7F4	DECCNT	FBCD	FNKSWI
08BC	CHPUT	268C	DECSUB	F3E9	FORCLR
08DF	CHPUT1	F7F2	DECTM2	148E	FORMAT
2686	CHRGTR	F7F0	DECTMP	F3F5	FRCNEW
0D6A	CHSNS	F6CA	DEFTBL	F69B	FRETOP

FBCA	FSTPOS	FEEE	H.DSKC	FE67	H.MERG
F7BA	FUNACT	FE12	H.DSKF	FE3A	H.MKD
F3FA	GETPNT	FE17	H.DSKI	FE30	H.MKI
1474	GETVC2	FDEF	H.DSKO	FE35	H.MKS
1470	GETVCP	FDA9	H.DSPC	FDF9	H.NAME
2689	GETYPR	FDB3	H.DSPF	FF3E	H.NEWS
04BD	GICINI	FEA3	H.EOF	FDD6	H.NMI
FCB7	GRPACX	FDAE	H.ERAC	FEB7	H.NODE
FCB9	GRPACY	FDB8	H.ERAF	FE58	H.NOFO
F3CD	GRPATR	FF02	H.ERRF	FF34	H.NOTR
F3CB	GRPCGP	FFB1	H.ERRO	FE62	H.NTFL
F3C9	GRPCOL	FEFD	H.ERRP	FF2F	H.NTFN
FCA6	GRPHED	FF70	H.EVAL	FF6B	H.NTPL
F3C7	GRPNAM	FE2B	H.FIEL	FE5D	H.NULO
F3CF	GRPPAT	FE7B	H.FILE	FF75	H.OKNO
1510	GRPPRT	FE85	H.FILO	FDEA	H.ONGO
0704	GSPSIZ	FF1B	H.FINE	FEE4	H.OUTD
18C7	GTASPC	FF7A	H.FING	FEB2	H.PARD
12AC	GTPAD	FF16	H.FINI	FFA7	H.PHYD
1273	GTPDL	FF5C	H.FINP	FDDB	H.PINL
11EE	GTSTCK	FEA8	H.FOPS	FFC5	H.PLAY
1253	GTTRIG	FFAC	H.FORM	FEBC	H.POSD
FCB3	GXPOS	FF9D	H.FRET	FEF8	H.PRGE
FCB5	GYPOS	FF66	H.FRME	FF52	H.PRTF
F40A	HEADER	FF93	H.FRQI	FFA2	H.PTRG
FE1C	H.ATTR	FEC6	H.GEND	FDE0	H.QINL
FEAD	H.BAKU	FE4E	H.GETP	FF07	H.READ
FE76	H.BINL	FF43	H.GONE	FF4D	H.RETU
FE71	H.BINS	FE8A	H.INDS	FE26	H.RSET
FF8E	H.BUFL	FDC7	H.INIP	FE8F	H.RSLF
FDC2	H.CHGE	FDE5	H.INLI	FECB	H.RUNC
FDA4	H.CHPU	FE03	H.IPL	FE94	H.SAVD
FF48	H.CHRG	FEDF	H.ISFL	FE6C	H.SAVE
FED0	H.CLEA	FF7F	H.ISMI	FF98	H.SCNE
FE0D	H.CMD	FF2A	H.ISRE	FFC0	H.SCRE
FF57	H.COMP	FDCC	H.KEYC	FE53	H.SETF
FE08	H.COPY	FD9A	H.KEYI	FDF4	H.SETS
FEE9	H.CRDO	FDFE	H.KILL	FF39	H.SNGF
FF20	H.CRUN	FDD1	H.KYEA	FEDA	H.STKE
FF25	H.CRUS	FF89	H.LIST	FD9F	H.TIMI
FE49	H.CVD	FE99	H.LOC	FDBD	H.TOTE
FE3F	H.CVI	FE9E	H.LOF	FF61	H.TRMN
FE44	H.CVS	FED5	H.LOPD	FF84	H.WIDT
FEF3	H.DDGR	FFB6	H.LPTO	F408	HIGH
FEC1	H.DEVN	FFBB	H.LPTS	FC4A	HIMEM
FE80	H.DGET	FE21	H.LSET	F83E	HOLD
FF11	H.DIRD	FF0C	H.MAIN	F836	HOLD2

F806	HOLD8	15DF	MAPXYC	18CF	PNTINI
098F	INESC	F92F	MAXDEL	088E	POSIT
139D	INIFNK	F85F	MAXFIL	F7B4	PRMFLG
05D2	INIGRP	F3EC	MAXUPD	F6E6	PRMLEN
061F	INIMLT	F958	MCLFLG	F74E	PRMLN2
2680	INIT	FB3B	MCLLEN	F74C	PRMPRV
0538	INIT32	FB3C	MCLPTR	F6E4	PRMSTK
049D	INITIO	F956	MCLTAB	FD89	PROCNM
050E	INITXT	F672	MEMSIZ	FB35	PRSCNT
23D5	INLIN	F92D	MINDEL	F416	PRTFLG
FCA8	INSFLG	F3EF	MINUPD	F864	PTRFIL
FCA2	INTCNT	F3D7	MLTATR	F6A9	PTRFLG
FC9B	INTFLG	F3D5	MLTCGP	0F55	PUTCHR
FCA0	INTVAL	F3D3	MLTCOL	F3F8	PUTPNT
03FB	ISCNTC	F3D1	MLTNAM	1492	PUTQ
145F	ISFLIO	F3D9	MLTPAT	23CC	QINLIN
FC9E	JIFFY	F951	MOVCNT	F971	QUEBAK
FCAD	KANAMD	FB3F	MUSICF	F959	QUETAB
FCAC	KANAST	F922	NAMBAS	FB3E	QUEUEN
F41F	KBUF	FBE5	NEWKEY	F3F3	QUEUES
0D89	KEYANY	4601	NEWSTT	F418	RAWPRT
FBF0	KEYBUF	F87C	NLONLY	F380	RDPRIM
0E3B	KEYCOD	1398	NMI	110E	RDPSG
0C3C	KEYINT	F7B7	NOFUNS	01B6	RDSLT
0468	KILBUF	1809	NSETCX	7E1A	RDSLTW
0F10	KYEASY	F417	NTMSXP	1449	RDVDP
107D	KYGRAP	F862	NULBUF	07D7	RDVRM
0F36	KYLOCK	FBDA	OLDKEY	1647	READC
0F46	KYSTOP	F6BE	OLDLIN	F3F7	REPCNT
070F	LDIRMV	FCB0	OLDSCR	FC6A	REQSTP
0744	LDIRVM	F6C0	OLDTXT	F3DF	RG0SAV
16EE	LEFTC	F6BB	ONEFLG	F3E0	RG1SAV
F954	LFPROG	F6B9	ONELIN	F3E1	RG2SAV
14EB	LFTQ	FBD8	ONGSBF	F3E2	RG3SAV
F3AF	LINL32	F664	OPRTYP	F3E3	RG4SAV
F3AE	LINL40	1B45	OUTDO	F3E4	RG5SAV
F3B0	LINLEN	FC9D	PADX	F3E5	RG6SAV
FBB2	LINTTB	FC9C	PADY	F3E6	RG7SAV
F94B	LOHADR	F6E8	PARM1	16C5	RIGHTC
F94D	LOHCNT	F750	PARM2	F857	RNDX
F94A	LOHDIR	F926	PATBAS	FAF5	RS2IQ
F949	LOHMSK	FC40	PATWRK	144C	RSLREG
F406	LOW	08DB	PBDHRT	F955	RTPROG
FCA4	LOWLIM	F953	PDIREC	FC9A	RTYCNT
085D	LPTOUT	148A	PHYDIO	FCBE	RUNBNF
F415	LPTPOS	23BF	PINLIN	F866	RUNFLG
0884	LPTSTT	FB40	PLYCNT	F87D	SAVEND

FCBF	SAVENT
FB36	SAVSP
F6B1	SAVSTK
F6AF	SAVTXT
FB39	SAVVOL
1599	SCALXY
197A	SCANL
18E4	SCANR
2439	SCITBL
F3F6	SCNCNT
FCAF	SCRMOD
02A3	SELEXP
027E	SELPRM
1676	SETATR
167E	SETC
0602	SETGRP
0659	SETMLT
07EC	SETRD
05B4	SETT32
0C2B	SETTRM
0594	SETTXT
07DF	SETWRT
FBEB	SFTKEY
F94F	SKPCNT
120C	SLSTCK
FCC9	SLTATR
FCC5	SLTTBL
FD09	SLTWRK
1452	SNSMAT
F3E7	STATFL
F674	STKTOP
1384	STMOTR
0A69	STOCSR
1640	STOREC
F6C6	STREND
6678	STROUT
11C4	STRTMS
F6A5	SUBFLG
F7BC	SWPTMP
2683	SYNCHR
F3C3	T32ATR
F3C1	T32CGP
F3BF	T32COL
F3BD	T32NAM
F3C5	T32PAT
1ABC	TAPIN
19E9	TAPIOF
1A63	TAPION
19DD	TAPOFF
19F1	TAPOON
1A19	TAPOUT
170A	TDOWNC
F6A7	TEMP
F6BC	TEMP2
F69D	TEMP3
F69F	TEMP8
F7B8	TEMP9
F678	TEMPPT
F67A	TEMPST
083B	TOTEXT
F7C4	TRCFLG
F3E8	TRGFLG
FC4C	TRPTBL
F661	TTYPOS
173C	TUPC
F3B9	TXTATR
F3B7	TXTCGP
F3B5	TXTCOL
F3B3	TXTNAM
F3BB	TXTPAT
F676	TXTTAB
175D	UPC
F39A	USRTAB
F663	VALTYP
F6C2	VARTAB
FB41	VCBA
FB66	VCBB
FB8B	VCBC
F419	VLZADR
F41B	VLZDAT
F975	VOICAQ
F9F5	VOICBQ
FA75	VOICCQ
FB38	VOICEN
FCA5	WINWID
F385	WRPRIM
01D1	WRSLT
1102	WRTPSG
057F	WRTVDP
07CD	WRTVRM
144F	WSLREG

APPENDIX A

```
                                   TITLE   MSX USA version
                                   SUBTTL  Symbol definition
                                   page 36
                                   .Z80
  0000'                            ASEG

                                   .COMMENT %

                                  Differences between Japanese version and overseas versions

                                1)  The default screen mode has been changed from 1 to 0.
                                2)  The default border color has been changed  from  7  to  4.   The
                                    default function  key  string  for  F6 key has been also changed
                                    to reflect this change.
                                3)  The character generator pattern has been changed.
                                4)  The Hiragana to Katakana conversion in LPT  output  routine  has
                                    been removed.
                                5)  The ASCII load problem has been fixed.
                                6)  The null device name problem has been fixed.
                                7)  The format symbol in PRINT USING  statement  has  been  changed.
                                8)  The reserved key  matrix  area  now  has  a  table  for  ten-key
                                    support
                                            United States           United Kingdom
                            Vsync:          60Hz                    50Hz
                            Screen size:    39 (default)            37 (default)
                            Layout:         QWERTY                  QWERTY
                            Deadkey:        4 deadkeys supported.   4 deadkeys supported.
                            Currency:       Dollar sign              British Pound sign
                            Special note:   None                    None
                            Status:         Finalized               Finalized

                                            %
  009C                          POND    EQU     9CH             ;character code for pound sign
  0006                          DEADNUM EQU     6

                                PRINTV  MACRO   VALUE
                                        IF1
                                        .PRINTX * VALUE bytes left *
                                        ENDIF
                                        ENDM
                                ;
                                ;       MSX ROM references
                                ;
  006C                          INITXT          EQU     6CH     ;initialize screen to 40 character text
  0132                          CHGCAP          EQU     132H
  0F10                          KYEASY          EQU     0F10H
  0F55                          PUTCHR          EQU     0F55H   ;put a character in queue
  0F64                          GENCLK          EQU     0F64H   ;generate click sound
  10C2                          UPDATE          EQU     10C2H   ;update put/get pointer
  FBEB                          SFTKEY          EQU     0FBEBH  ;current shift key status
  FCAB                          CAP_LOCK        EQU     0FCABH  ;capital lock status     (CAPST)
  FCAC                          DEAD_STATUS     EQU     0FCACH  ;current dead-key status (KANAST)
                                                                ; if 0 no preceding dead-key
                                                                ; if 1              dead-key
                                                                ; if 2      shifted-dead-key
                                                                ; if 3         code-dead-key
                                                                ; if 4   code-shift-dead-key

                                        IF1
                                        .PRINTX / USA version /
                                        ENDIF                   ;IF1
                                        ORG     2BH
                                ;
                                ; The format of ID byte is as follows
                                ;
                                ;  2BH: b7 b6 b5 b4 b3 b2 b1 b0
                                ;        |  |  |  |  |  |  |  |
                                ;        |  |  |  |  +--+--+--+-- kind of character generator
                                ;        |  |  |  |                 0:Japanese  1:International
                                ;        |  +--+--+-------------- format of date
                                ;        |                          0:Y-M-D 1:M-D-Y 2:D-M-Y
                                ;        +----------------------- frequency of interrupt
                                ;                                   1:50Hz  0:60Hz
                                ;
 002B    11                             DEFB    00010001B          ;UK - DEFB      1010001B
                                ;
                                ;  2CH: b7 b6 b5 b4 b3 b2 b1 b0
                                ;        |  |  |  |  |  |  |  |
                                ;        |  |  |  |  +--+--+--+-- kind of keyboard
                                ;        |  |  |  |                 0:Japan     1:International
                                ;        |  |  |  |                 2:French    3:UK       4:DIN
                                ;        |  |  |  |
                                ;        +--+--+--+-------------- version of BASIC (print using etc.)
                                ;
 002C    11                             DEFB    11H                ;UK - DEFB      13H
                                ;
                                ;  34H .. 37H
                                ;
                                ;       Range of first byte for 2-byte characters such as KANJI
                                ;
```

```
                              ORG      0D9BH
                        ;
 0D9B    1021                 DEFW     KEYCOD

                              SUBTTL   Key code table (0DA5H..0EC4H)
```

```
                                      ORG     0DA5H

                              ;*****************************************************************************
                              ;                                                                           *
                              ;       Table of codes for various shift  conditions.   Note  that  0FFH     *
                              ;       (255) is reserved for dead-key.                                     *
                              ;                                                                           *
                              ;*****************************************************************************

                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                              ;
                              ;       Keyboard encode table for 'QWERTY' layout
                              ;
                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                              ;
                              ;       Normal codes
                              ;
 0DA5                         NORMAL:
 0DA5    30 31 32 33                  DEFB    '01234567'
 0DA9    34 35 36 37
 0DAD    38 39 2D 3D                  DEFB    '89-=[];'          '89-= \ [];'
 0DB1    5C 5B 5D 3B
 0DB5    27 60 2C 2E                  DEFB    ''';./',0FFH,'ab'   ;''' ` ,./',0ffH,'ab'
 0DB9    2F FF 61 62
 0DBD    63 64 65 66                  DEFB    'cdefghij'
 0DC1    67 68 69 6A
 0DC5    6B 6C 6D 6E                  DEFB    'klmnopqr'
 0DC9    6F 70 71 72
 0DCD    73 74 75 76                  DEFB    'stuvwxyz'
 0DD1    77 78 79 7A
```

```
                                ;
                                ;       Codes when shift key pressed
                                ;
 0DD5                           SHIFT:
 0DD5      29 21 40 23                  DEFB    ')!@#$%&'           ')!@#$% ^ &'
 0DD9      24 25 5E 26
 0DDD      2A 28 5F 2B                  DEFB    '*(_+|≐:'           '*(_+|{}:'
 0DE1      7C 7B 7D 3A
 0DE5      22 7E 3C 3E                  DEFB    '"°§¶?',0FFH,'AB'   '"~<>?',0ffH,'AB'
 0DE9      3F FF 41 42
 0DED      43 44 45 46                  DEFB    'CDEFGHIJ'
 0DF1      47 48 49 4A
 0DF5      4B 4C 4D 4E                  DEFB    'KLMNOPQR'
 0DF9      4F 50 51 52
 0DFD      53 54 55 56                  DEFB    'STUVWXYZ'
 0E01      57 58 59 5A

                                ;
                                ;       Codes when graph key pressed
                                ;
                                ;               0    1    2    3    4    5    6    7
 0E05                           GRAPH:
 0E05      09 AC AB BA                  DEFB    009H,0ACH,0ABH,0BAH,0EFH,0BDH,0F4H,0FBH ;0
 0E09      EF BD F4 FB
 0E0D      EC 07 17 F1                  DEFB    0ECH,007H,017H,0F1H,01EH,001H,00DH,006H ;1
 0E11      1E 01 0D 06
 0E15      05 BB F3 F2                  DEFB    005H,0BBH,0F3H,0F2H,01DH,0FFH,0C4H,011H ;2
 0E19      1D FF C4 11
 0E1D      BC C7 CD 14                  DEFB    0BCH,0C7H,0CDH,014H,015H,013H,0DCH,0C6H ;3
 0E21      15 13 DC C6
 0E25      DD C8 0B 1B                  DEFB    0DDH,0C8H,00BH,01BH,0C2H,0DBH,0CCH,018H ;4
 0E29      C2 DB CC 18
```

```
  0E2D    D2 12 C0 1A                 DEFB    0D2H,012H,0C0H,01AH,0CFH,01CH,019H,00FH ;5
  0E31    CF 1C 19 0F

                                ;
                                ;       Codes when graph and shift keys pressed
                                ;
                                ;                 0     1     2     3     4     5     6     7
  0E35                          GRAPH_SHIFT:
  0E35    0A 00 FD FC                 DEFB    00AH,000H,0FDH,0FCH,000H,000H,0F5H,000H ;0
  0E39    00 00 F5 00
  0E3D    00 08 1F F0                 DEFB    000H,008H,01FH,0F0H,016H,002H,00EH,004H ;1
  0E41    16 02 0E 04
  0E45    03 F7 AE AF                 DEFB    003H,0F7H,0AEH,0AFH,0F6H,0FFH,0FEH,000H ;2
  0E49    F6 FF FE 00
  0E4D    FA C1 CE D4                 DEFB    0FAH,0C1H,0CEH,0D4H,010H,0D6H,0DFH,0CAH ;3
  0E51    10 D6 DF CA
  0E55    DE C9 0C D3                 DEFB    0DEH,0C9H,00CH,0D3H,0C3H,0D7H,0CBH,0A9H ;4
  0E59    C3 D7 CB A9
  0E5D    D1 00 C5 D5                 DEFB    0D1H,000H,0C5H,0D5H,0D0H,0F9H,0AAH,0F8H ;5
  0E61    D0 F9 AA F8
                                ;
                                ;       Codes when code key pressed
                                ;
                                ;                 0     1     2     3     4     5     6     7
  0E65                          CODE:
  0E65    EB 9F D9 BF                 DEFB    0EBH,09FH,0D9H,0BFH,09BH,098H,0E0H,0E1H ;0
  0E69    9B 98 E0 E1
  0E6D    E7 87 EE E9                 DEFB    0E7H,087H,0EEH,0E9H,000H,0EDH,0DAH,0B7H ;1
  0E71    00 ED DA B7
  0E75    B9 E5 86 A6                 DEFB    0B9H,0E5H,086H,0A6H,0A7H,0FFH,084H,097H ;2
  0E79    A7 FF 84 97
  0E7D    8D 8B 8C 94                 DEFB    08DH,08BH,08CH,094H,081H,0B1H,0A1H,091H ;3
  0E81      81 B1 A1 91
  0E85      B3 B5 E6 A4                     DEFB    0B3H,0B5H,0E6H,0A4H,0A2H,0A3H,083H,093H ;4
  0E89      A2 A3 83 93
  0E8D      89 96 82 95                     DEFB    089H,096H,082H,095H,088H,08AH,0A0H,085H ;5
  0E91      88 8A A0 85

                                    ;
                                    ;       Codes when code and shift keys pressed
                                    ;
                                    ;                0     1     2     3     4     5     6     7
  0E95                              CODE_SHIFT:
  0E95      D8 AD 9E BE                     DEFB    0D8H,0ADH,09EH,0BEH,09CH,09DH,000H,000H ;0
  0E99      9C 9D 00 00
  0E9D      E2 80 00 00                     DEFB    0E2H,080H,000H,000H,000H,0E8H,0EAH,0B6H ;1
  0EA1      00 E8 EA B6
  0EA5      B8 E4 8F 00                     DEFB    0B8H,0E4H,08FH,000H,0A8H,0FFH,08EH,000H ;2
  0EA9      A8 FF 8E 00
  0EAD      00 00 00 99                     DEFB    000H,000H,000H,099H,09AH,0B0H,000H,092H ;3
  0EB1      9A B0 00 92
  0EB5      B2 B4 00 A5                     DEFB    0B2H,0B4H,000H,0A5H,000H,0E3H,000H,000H ;4
  0EB9      00 E3 00 00
  0EBD      00 00 90 00                     DEFB    000H,000H,090H,000H,000H,000H,000H,000H ;5
  0EC1      00 00 00 00

                                            IF1
                                            IF      ($-NORMAL) NE (48*6)
                                            .PRINTX * Table length not correct *
                                            ENDIF
                                            ENDIF
```

```
                                        ORG     0F17H
                                ;
  0F17    1003                          DEFW    EASYTB-48
                                        SUBTTL  Dead key handler (0F1FH..0F34H)
                                ORG     0F1FH
                        ;
 0F1F                   DEAD_KEY:
 0F1F    3A FBEB                LD      A,(SFTKEY)
 0F22    5F                     LD      E,A
 0F23    F6 FE                  OR      11111110B       ;extract shift key status only
 0F25    CB 63                  BIT     4,E             ;code key pressed?
 0F27    20 02                  JR      NZ,DEAD_KEY1    ;no
 0F29    E6 FD                  AND     11111101B
 0F2B                   DEAD_KEY1:
 0F2B    2F                     CPL
 0F2C    3C                     INC     A               ;make 1..4
 0F2D    32 FCAC                LD      (DEAD_STATUS),A
 0F30    18 32                  JR      GENCLK          ;generate click sound

                                PRINTV  %(0F35H-$)




                                ORG     0F5AH
                        ;
 0F5A    105B                   DEFW    NEW_UPDATE

                                SUBTTL  Keyboard encoder (0F83H..10C1H)
                                ORG     0F83H
                        ;
                        ;       Beginning of the table-driven key encoder
                        ;
                        ;       [C] = raw code for pressed key
                        ;
 0F83                   INTKEY:
 0F83    3A FBEB                LD      A,(SFTKEY)      ;get current shift key status
 0F86    5F                     LD      E,A             ;save shift key status in [E]
 0F87    1F                     RRA                     ;move control key status to carry
 0F88    1F                     RRA
 0F89    F5                     PUSH    AF              ;remember control  key  status  (carry
                                                        ;reset if pressed)
 0F8A    7B                     LD      A,E             ;restore shift key status
 0F8B    2F                     CPL
 0F8C    30 10                  JR      NC,IS_CONTROL   ;control key being pressed
                        ;
                        ;       Get an  offset  into  SFTTAB  using current shift key status and
                        ;       code lock status.
                        ;
 0F8E    1F                     RRA
 0F8F    1F                     RRA
 0F90    07                     RLCA
 0F91    E6 03                  AND     11B
 0F93    CB 4F                  BIT     1,A             ;is graph shift on?
 0F95    20 09                  JR      NZ,INTKEY_1     ;yes, ignore code key
 0F97    CB 63                  BIT     4,E             ;is code pressed?
 0F99    20 05                  JR      NZ,INTKEY_1     ;no
 0F9B    F6 04                  OR      100B            ;set code bit
 0F9D    11                     DEFB    11H             ;'LD DE,XXXX' instruction
                        ;
```

```
                                ;       Control key is being pressed.  Ignore the graph  and  code  lock
                                ;       status.
                                ;
  0F9E                          IS_CONTROL:
  0F9E    E6 01                         AND     1               ;valid is only shift key status
                                ;
                                ;       Now we have in [Acc] '00000CGS'
                                ;                                   |||
                                ;                                   ||+-- shift \
                                ;                                   |+--- graph  >-- 1 when pressed
                                ;                                   +---- code  /
                                ;
  0FA0                          INTKEY_1:
  0FA0    5F                            LD      E,A
  0FA1    87                            ADD     A,A
  0FA2    83                            ADD     A,E
  0FA3    87                            ADD     A,A
  0FA4    87                            ADD     A,A
  0FA5    87                            ADD     A,A
  0FA6    87                            ADD     A,A
  0FA7    5F                            LD      E,A
  0FA8    16 00                         LD      D,0
  0FAA    21 0DA5                       LD      HL,NORMAL
  0FAD    19                            ADD     HL,DE           ;[HL] = the address of table
  0FAE    42                            LD      B,D             ;[BC] = offset into code table
  0FAF    09                            ADD     HL,BC
  0FB0    F1                            POP     AF              ;restore control key status into carry
  0FB1    7E                            LD      A,(HL)          ;get real code
  0FB2    3C                            INC     A               ;dead-key?
  0FB3    CA 0F1F                       JP      Z,DEAD_KEY      ;yes
  0FB6    3D                            DEC     A               ;should code be generated?
  0FB7    C8                            RET     Z               ;no code should be generated
```

```
 0FB8    38 16                    JR      C,WASNT_CONTROL ;control was not pressed
 0FBA    E6 DF                    AND     11011111B       ;force to upper case
 0FBC    D6 40                    SUB     40H             ;make control character
 0FBE    FE 20                    CP      ' '
 0FC0    D0                       RET     NC              ;cannot make control code
 0FC1                     JPUTCHR:
 0FC1    18 92                    JR      PUTCHR          ;skip  2  byte  code  check  and  case
                                                          ;translation
                          ;
 0FC3                     KYFUNC:
 0FC3    3A FBEB                  LD      A,(SFTKEY)
 0FC6    0F                       RRCA
 0FC7    38 04                    JR      C,KYFNC1
 0FC9    79                       LD      A,C
 0FCA    C6 05                    ADD     A,5
 0FCC    4F                       LD      C,A
 0FCD                     KYFNC1:
 0FCD    C3 0EC5                  JP      0EC5H
                          ;
 0FD0                     WASNT_CONTROL:
 0FD0    FE 20                    CP      ' '             ;2 byte code?
 0FD2    30 0B                    JR      NC,NOT_2BYTE    ;no
 0FD4    F5                       PUSH    AF
 0FD5    3E 01                    LD      A,1             ;put graphic header byte
 0FD7    CD 0F55                  CALL    PUTCHR
 0FDA    F1                       POP     AF
 0FDB    C6 40                    ADD     A,40H           ;add offset
 0FDD    18 E2                    JR      JPUTCHR         ;skip case translation
                          ;
                          ;       Check if case translation is necessary
                          ;
 0FDF                     NOT_2BYTE:
```

```
0FDF    21 FCAB                 LD      HL,CAP_LOCK      ;capital lock active?
0FE2    34                      INC     (HL)
0FE3    35                      DEC     (HL)
0FE4    28 0A                   JR      Z,CHECK_DEAD     ;no
0FE6    FE 61                   CP      'a'              ;normal alphabet?
0FE8    38 27                   JR      C,CHECK_SPECIAL  ;no, check if special alphabet
0FEA    FE 7B                   CP      'z'+1
0FEC    30 23                   JR      NC,CHECK_SPECIAL
0FEE    E6 DF                   AND     11011111B        ;force to upper case
0FF0                    CHECK_DEAD:
0FF0    ED 5B FCAC              LD      DE,(DEAD_STATUS)
0FF4    1C                      INC     E                ;dead-key active?
0FF5    1D                      DEC     E
0FF6    28 C9                   JR      Z,JPUTCHR        ;no
0FF8    57                      LD      D,A              ;save encoded code
0FF9    F6 20                   OR      00100000B        ;force to lower case
0FFB    21 1066                 LD      HL,VOWELS+DEADNUM-1
0FFE    0E 06                   LD      C,DEADNUM
1000    ED B9                   CPDR                     ;is input character vowel?
1002    7A                      LD      A,D              ;restore code
1003    20 BC                   JR      NZ,JPUTCHR       ;no
1005    23                      INC     HL
1006    0E 06                   LD      C,DEADNUM
1008                    DEAD1:
1008    09                      ADD     HL,BC
1009    1D                      DEC     E
100A    20 FC                   JR      NZ,DEAD1
100C    7E                      LD      A,(HL)           ;get from table
100D    CB 6A                   BIT     5,D              ;is input code lower or upper?
100F    20 B0                   JR      NZ,JPUTCHR       ;lower, no case translation necessary
1011                    CHECK_SPECIAL:
1011    0E 1F                   LD      C,TABLE_LENGTH   ;number of special alphabets
```

```
 1013    21 109D                 LD      HL,SPECIAL_UPPER-1
 1016    ED B9                   CPDR                       ;found in lower case table?
 1018    20 A7                   JR      NZ,JPUTCHR         ;no
 101A    0E 1F                   LD      C,TABLE_LENGTH     ;number of special alphabets
 101C    23                      INC     HL                 ;compensate [HL] so it points to the
                                                            ;data that matched
 101D    09                      ADD     HL,BC              ;add table length to get address of
                                                            ;the character
 101E    7E                      LD      A,(HL)             ;get code from table
 101F    18 A0                   JR      JPUTCHR
                         ;
                         ;       Here with raw code in [C]
                         ;
 1021                    KEYCOD:
 1021    79                      LD      A,C                ;get raw code
 1022    21 1B96                 LD      HL,KYJTAB
 1025    CD FDCC                 CALL    0FDCCH
 1028    16 0F                   LD      D,0FH
 102A                    KYCLAS:
 102A    BE                      CP      (HL)
 102B    23                      INC     HL
 102C    5E                      LD      E,(HL)
 102D    23                      INC     HL
 102E    D5                      PUSH    DE
 102F    D8                      RET     C
 1030    D1                      POP     DE
 1031    18 F7                   JR      KYCLAS
                         ;
 1033                    EASYTB:
 1033    00                      DEFB    0                  ;Shift          (48)
 1034    00                      DEFB    0                  ;Control        (49)
 1035    00                      DEFB    0                  ;Graph          (50)
```

```
  1036    00                        DEFB    0               ;Cap lock       (51)
  1037    00                        DEFB    0               ;Kana lock      (52)
  1038    00                        DEFB    0               ;F1             (53)
  1039    00                        DEFB    0               ;F2             (54)
  103A    00                        DEFB    0               ;F3             (55)
  103B    00                        DEFB    0               ;F4             (56)
  103C    00                        DEFB    0               ;F5             (57)
  103D    1B                        DEFB    27              ;Escape         (58)
  103E    09                        DEFB    9               ;Tab            (59)
  103F    00                        DEFB    0               ;Stop           (60)
  1040    08                        DEFB    8               ;Back space     (61)
  1041    18                        DEFB    'X'-'@'         ;Select         (62)
  1042    0D                        DEFB    13              ;Enter          (63)
  1043    20                        DEFB    32              ;Space          (64)
  1044    0C                        DEFB    12              ;Clear          (65)
  1045    12                        DEFB    'R'-'@'         ;Insert         (66)
  1046    7F                        DEFB    127             ;Rubout         (67)
  1047    1D                        DEFB    29              ;Left           (68)
  1048    1E                        DEFB    30              ;Up             (69)
  1049    1F                        DEFB    31              ;Down           (70)
  104A    1C                        DEFB    28              ;Right          (71)
                            ;
                            ;       For additional key matrix
                            ;
  104B    00                        DEFB    0               ;               (72)
  104C    00                        DEFB    0               ;               (73)
  104D    00                        DEFB    0               ;               (74)
  104E    30                        DEFB    '0'             ;               (75)
  104F    31                        DEFB    '1'             ;               (76)
  1050    32                        DEFB    '2'             ;               (77)
  1051    33                        DEFB    '3'             ;               (78)
  1052    34                        DEFB    '4'             ;               (79)
  1053    35                       DEFB    '5'           ;                  (80)
  1054    36                       DEFB    '6'           ;                  (81)
  1055    37                       DEFB    '7'           ;                  (82)
  1056    38                       DEFB    '8'           ;                  (83)
  1057    39                       DEFB    '9'           ;                  (84)
  1058    2D                       DEFB    '-'           ;                  (85)
  1059    2C                       DEFB    ','           ;                  (86)
  105A    2E                       DEFB    '.'           ;                  (87)
                           ;
  105B                     NEW_UPDATE:
  105B    AF                       XOR     A             ;clear DEAD_STATUS since code generated
  105C    32 FCAC                  LD      (DEAD_STATUS),A
  105F    18 61                    JR      UPDATE
                           ;
  1061                     VOWELS:
  1061    61 65 69 6F              DEFB    'aeiouy'
  1065    75 79
                           ;
                           ;       Table of codes when vowels are used with a dead key.
                           ;
                           ;
                           ;       For 'dead-key' (non-shifted)
                           ;
  1067    85                       DEFB    85H           ;a accent grave
  1068    8A                       DEFB    8AH           ;e accent grave
  1069    8D                       DEFB    8DH           ;i accent grave
  106A    95                       DEFB    95H           ;o accent grave
  106B    97                       DEFB    97H           ;u accent grave
  106C    79                       DEFB    'y'
                           ;
                           ;       For shifted dead-key
                           ;
```

```
 106D    A0                        DEFB    0A0H           ;a accent egu
 106E    82                        DEFB    82H            ;e accent egu
 106F    A1                        DEFB    0A1H           ;i accent egu
 1070    A2                        DEFB    0A2H           ;o accent egu
 1071    A3                        DEFB    0A3H           ;u accent egu
 1072    79                        DEFB    'y'
                           ;
                           ;       For code dead-key
                           ;
 1073    83                        DEFB    83H            ;a accent circonflex
 1074    88                        DEFB    88H            ;e accent circonflex
 1075    8C                        DEFB    8CH            ;i accent circonflex
 1076    93                        DEFB    93H            ;o accent circonflex
 1077    96                        DEFB    96H            ;u accent circonflex
 1078    79                        DEFB    'y'
                           ;
                           ;       For shifted-code dead key
                           ;
 1079    84                        DEFB    84H            ;a umlaut
 107A    89                        DEFB    89H            ;e umlaut
 107B    8B                        DEFB    8BH            ;i umlaut
 107C    94                        DEFB    94H            ;o umlaut
 107D    81                        DEFB    81H            ;u umlaut
 107E    98                        DEFB    98H            ;y umlaut
                           ;
                           ;       Table of special alphabets
                           ;
                           ;       Used to determine if a key should be affected by capital lock
                           ;
 107F                      SPECIAL_ALPHABET:

 107F    83                        DEFB    83H            ;a accent circonflex
```

```
  1080    88                      DEFB    88H             ;e accent circonflex
  1081    8C                      DEFB    8CH             ;i accent circonflex
  1082    93                      DEFB    93H             ;o accent circonflex
  1083    96                      DEFB    96H             ;u accent circonflex

  1084    84                      DEFB    84H             ;a umlaut
  1085    89                      DEFB    89H             ;e umlaut
  1086    8B                      DEFB    8BH             ;i umlaut
  1087    94                      DEFB    94H             ;o umlaut
  1088    81                      DEFB    81H             ;u umlaut
  1089    98                      DEFB    98H             ;y umlaut

  108A    A0                      DEFB    0A0H            ;a accent egu
  108B    82                      DEFB    82H             ;e accent egu
  108C    A1                      DEFB    0A1H            ;i accent egu
  108D    A2                      DEFB    0A2H            ;o accent egu
  108E    A3                      DEFB    0A3H            ;u accent egu

  108F    85                      DEFB    85H             ;a accent grave
  1090    8A                      DEFB    8AH             ;e accent grave
  1091    8D                      DEFB    8DH             ;i accent grave
  1092    95                      DEFB    95H             ;o accent grave
  1093    97                      DEFB    97H             ;u accent grave

  1094    B1                      DEFB    0B1H            ;a tilda
  1095    B3                      DEFB    0B3H            ;i tilda
  1096    B5                      DEFB    0B5H            ;o tilda
  1097    B7                      DEFB    0B7H            ;u tilda
  1098    A4                      DEFB    0A4H            ;n tilda

  1099    86                      DEFB    86H             ;a circle
  109A    87                      DEFB    87H             ;c cedille
```

```
 109B    91                    DEFB    91H            ;ae
 109C    B9                    DEFB    0B9H           ;ij
 109D    79                    DEFB    'y'
 001F                TABLE_LENGTH    EQU     $-SPECIAL_ALPHABET
                     ;
 109E                SPECIAL_UPPER:
 109E    41                    DEFB    'A'            ;A accent circonflex
 109F    45                    DEFB    'E'            ;E accent circonflex
 10A0    49                    DEFB    'I'            ;I accent circonflex
 10A1    4F                    DEFB    'O'            ;O accent circonflex
 10A2    55                    DEFB    'U'            ;U accent circonflex

 10A3    8E                    DEFB    8EH            ;A umlaut
 10A4    45                    DEFB    'E'            ;E umlaut
 10A5    49                    DEFB    'I'            ;I umlaut
 10A6    99                    DEFB    99H            ;O umlaut
 10A7    9A                    DEFB    9AH            ;U umlaut
 10A8    59                    DEFB    'Y'            ;Y umlaut

 10A9    41                    DEFB    'A'            ;A accent egu
 10AA    90                    DEFB    90H            ;E accent egu
 10AB    49                    DEFB    'I'            ;I accent egu
 10AC    4F                    DEFB    'O'            ;O accent egu
 10AD    55                    DEFB    'U'            ;U accent egu

 10AE    41                    DEFB    'A'            ;A accent grave
 10AF    45                    DEFB    'E'            ;E accent grave
 10B0    49                    DEFB    'I'            ;I accent grave
 10B1    4F                    DEFB    'O'            ;O accent grave
 10B2    55                    DEFB    'U'            ;U accent grave

 10B3    B0                    DEFB    0B0H           ;A tilda
```

```
  10B4    B2                      DEFB    0B2H            ;I tilda
  10B5    B4                      DEFB    0B4H            ;O tilda
  10B6    B6                      DEFB    0B6H            ;U tilda
  10B7    A5                      DEFB    0A5H            ;N tilda

  10B8    8F                      DEFB    8FH             ;A circle
  10B9    80                      DEFB    80H             ;C cedille
  10BA    92                      DEFB    92H             ;AE
  10BB    B8                      DEFB    0B8H            ;IJ
  10BC    59                      DEFB    'Y'

                                  IF      TABLE_LENGTH NE ($-SPECIAL_UPPER)
                                  .PRINTX * Upper case table inconsistent *
                                  ENDIF

                                  PRINTV  %(10C2H-$)

                                  SUBTTL  Function key content
```

```
                                        ORG     1404H
                                ;
                                ;       Patch to change the default border color to 4
                                ;
 1404    34                             DEFB    '4'             ;change default border color to 4

                                        SUBTTL  Dispatch table (1B94H..1BAAH)
                                        ORG     1B94H
                                ;
                                ;       Patch to ignore the katakana to hiragana mapping
                                ;
  1B94    18 16                         JR      1BACH
                                ;
  1B96                          KYJTAB:
  1B96    30                            DEFB    48
  1B97    83                            DEFB    LOW INTKEY
  1B98    33                            DEFB    51
  1B99    10                            DEFB    LOW KYEASY
  1B9A    34                            DEFB    52
  1B9B    36                            DEFB    LOW 0F36H       ;capital lock
  1B9C    35                            DEFB    53
  1B9D    10                            DEFB    LOW KYEASY      ;code
  1B9E    3A                            DEFB    58
  1B9F    C3                            DEFB    LOW KYFUNC      ;function key
  1BA0    3C                            DEFB    60
  1BA1    10                            DEFB    LOW KYEASY
  1BA2    3D                            DEFB    61
  1BA3    46                            DEFB    LOW 0F46H       ;stop key
  1BA4    41                            DEFB    65
  1BA5    10                            DEFB    LOW KYEASY
  1BA6    42                            DEFB    66
  1BA7    06                            DEFB    LOW 0F06H       ;CLS/HOME key
  1BA8    FF                            DEFB    255
  1BA9    10                            DEFB    LOW KYEASY

                                        IF2
                                        IF      (HIGH INTKEY) NE 0FH
                                        .PRINTX * INTKEY not on 0FxxH *
                                ENDIF
                                IF      (HIGH KYFUNC) NE 0FH
                                .PRINTX * KYFUNC not on 0FxxH *
                                ENDIF
                                ENDIF

                                PRINTV  %(1BABH-$)
                                SUBTTL  Character font
```

```
                                        ORG     1BBFH
                                        .list

                                        (Font Image of each version)

                                        1BBFH to 23BEH
```

```
MSX USA version Macro-80        3.44    01-Jan-85       PAGE    12
currency symbol and print formatter symbols

                                         ORG     3499H
                                 ;
  3499    24                             DEFB    '$'        ;UK - 9CH, Pound Sign

                                         ORG     3549H      ;UK - 9CH, Pound sign
                                 ;
  3549    24                             DEFB    '$'

                                 ;
                                 ;       Patch code to fix ":xxx" file names
                                 ;
                                         ORG     5600H
  5600    CD 7FB7                        CALL    PATCH1

                                         ORG     60E3H
  60E3    5C                             DEFB    '\'

                                         ORG     60F1H
  60F1    5C                             DEFB    '\'

                                         ORG     6109H
  6109    26                             DEFB    '&'

                                         ORG     611FH
  611F    5C                             DEFB    '\'

                                         ORG     6126H
  6126    24                             DEFB    '$'        ;UK - 9CH, Pound sign

                                         ORG     6135H
  6135    24                             DEFB    '$'        ;UK - 9CH, Pound Sign
                                         SUBTTL  Miscellaneous patches
```

```
                                        ORG     738AH
                                ;
                                ;       Patch to allow graphic characters in ASCII load
                                ;
  738A    FE 0A                         CP      0AH             ;line feed?
  738C    28 EE                         JR      Z,737CH         ;yes, ignore this

                                        ORG     7754H
                                ;
                                ;       TCONST
          ; 60*120*4/2 = 14400  ;        Store original value - do not change
          ; 50*120*4/2 = 12000  ;
  7754    40                            DEFB    40H             ;UK - 0 (2nd byte of mantissa)
  7755    00                            DEFB    00H             ;UK - 0 (3rd byte of mantissa)
  7756    45                            DEFB    45H             ;UK - 45H (exponent)
  7757    14                            DEFB    14H             ;UK - 12H (1st byte of mantissa)

                                        ORG     7D2EH
                                ;
                                ;       Patch to change to 40 character mode
                                ;
  7D2E    CD 006C                       CALL    INITXT

                                        ORG     7F55H
                                ;
                                ;       Patch to change to 37 character mode
                                ;
  7F55    27                            DEFB    39              ;39 character mode for NTSC

                                        ORG     7F92H           ;UK - 37 character mode for PAL
```

```
                              ;       Patch to change the default border color to 4
                              ;
  7F92    04                          DEFB    4

                              ;
                              ;       Patch code to fix ":xxx" file names
                              ;
                                      ORG     7FB7H
  7FB7                        PATCH1:
  7FB7    11 FD89                     LD      DE,0FD89H       ;load PROCNM
  7FBA    A7                          AND     A               ;is device name null?
  7FBB    C0                          RET     NZ              ;no
  7FBC    04                          INC     B               ;yes, fake 1
  7FBD    C9                          RET

  7FBE                        LASTWR  EQU     $

                                      END
```

```
Miscellaneous patches

Macros:
PRINTV

Symbols:
FCAB    CAP_LOCK        0FF0    CHECK_DEAD      1011    CHECK_SPECIAL
0132    CHGCAP          0E65    CODE            0E95    CODE_SHIFT
1008    DEAD1           0006    DEADNUM         0F1F    DEAD_KEY
0F2B    DEAD_KEY1       FCAC    DEAD_STATUS     1033    EASYTB
0F64    GENCLK          0E05    GRAPH           0E35    GRAPH_SHIFT
006C    INITXT          0F83    INTKEY          0FA0    INTKEY_1
0F9E    IS_CONTROL      0FC1    JPUTCHR         1021    KEYCOD
102A    KYCLAS          0F10    KYEASY          0FCD    KYFNC1
0FC3    KYFUNC          1B96    KYJTAB          7FBE    LASTWR
105B    NEW_UPDATE      0DA5    NORMAL          0FDF    NOT_2BYTE
7FB7    PATCH1          009C    POND            0F55    PUTCHR
FBEB    SFTKEY          0DD5    SHIFT           107F    SPECIAL_ALPHABET
109E    SPECIAL_UPPER   001F    TABLE_LENGTH    10C2    UPDATE
1061    VOWELS          0FD0    WASNT_CONTROL


No Fatal error(s)
```

List of some ROM BIOS calls used by BASIC:

Name:	SYNCHR, 0008H
Function:	Checks if the current character pointed by HL is the one we want. If not, generates 'Syntax error', otherwise falls into CHRGTR.
Entry:	HL, character to be checked be placed at the next location to this RST.
Returns:	HL points to next character, A has the character. Carry flag set if number, Z flag set if end of statement.
Modifies:	AF, HL

Name:	CHRGTR, 0010H
Function:	Gets next character (or token) from BASIC text.
Entry:	HL
Returns:	HL points to next character, A has the character. Carry flag set if number, Z flag set if end of statement encountered.
Modifies:	AF, HL

Name:	OUTDO, 0018H
Function:	Outputs to current device
Entry:	A, PTRFIL, PRTFLG
Returns:	None
Modifies:	None

Name:	DCOMPR, 0020H
Function:	Compares HL with DE
Entry:	HL, DE
Returns:	Flags
Modifies:	AF

Name:	GETYPR, 0028H
Function:	Returns the type of FAC
Entry:	FAC
Returns:	Flags
Modifies:	AF

Name:	CALLF, 0030H
Function:	Performs far_call (i.e., inter-slot call)
Entry:	None
Returns:	Who knows?
Modifies:	ditto
Note:	Calling sequence is as follows.

```
RST     6
DB      destination slot
DW      destination address
```

For precise description about parameters, see CALSLT.

Name:	CHSNS, 009CH
Function:	Checks the status of keyboard buffer.
Entry:	None
Returns:	Z flag reset if there's any character in buffer
Modifies:	AF

Name:	CHGET, 009FH
Function:	Waits until any characters are typed, and return with the character code.
Entry:	None
Returns:	Character code in [Acc]
Modifies:	AF

Name:	CHPUT, 00A2H
Function:	Outputs a character to console.
Entry:	Character code to be output in [Acc]
Returns:	None
Modifies:	None

Name:	LPTOUT, 00A5H
Function:	Outputs a character to LPT
Entry:	Character code to be output in [Acc]
Returns:	Carry flag set if aborted
Modifies:	F

Name:	LPTSTT, 00A8H
Function:	Checks line printer status
Entry:	None
Returns:	255 in [Acc] and Z flag reset if printer ready, 0 and Z flag set if not.
Modifies:	AF

Name:	CNVCHR, 00ABH
Function:	Checks graphic header byte and convert code
Entry:	Character code in [Acc]
Returns:	Carry flag reset - graphic header byte Carry flag set, Z flag set - converted graphic co Carry flag set, Z flag reset - non converted code
Modifies:	AF

Name:	PINLIN, 00AEH
Function:	Accepts a line from console until a CR or STOP is typed, and stores the line in buffer
Entry:	None
Returns:	Address of buffer top-1 in [HL], carry flag set if STOP is typed.
Modifies:	All

Name:	INLIN, 00B1H
Function:	Same as PINLIN, except in case AUTFLG is set.
Entry:	None
Returns:	Address of buffer top-1 in [HL], carry flag set if STOP is pressed.
Modifies:	All

Name:	QINLIN, 00B4H
Function:	Outputs a '?' mark and a space then fall into INLIN.
Entry:	None
Returns:	Address of buffer top-1 in [HL], carry flag set if STOP is pressed.
Modifies:	All

Name:	BREAKX, 00B7H
Function:	Checks the status of Control-STOP key
Entry:	None
Returns:	Carry flag set if being pressed
Modifies:	AF
Note:	This routine is used to check Control-STOP when interrupts are disabled.

Name:	ISCNTC, 00BAH
Function:	Checks the status of SHIFT-STOP key
Entry:	None
Returns:	None
Modifies:	None

Name:	CKCNTC, 00BDH
Function:	Same as ISCNTC, used by BASIC
Entry:	None
Returns:	None
Modifies:	None

Name:	BEEP, 00C0H
Function:	Beeps buzzer, reset sound chip.
Entry:	None
Returns:	None
Modifies:	All

Name:	CLS, 00C3H
Function:	Clears screen
Entry:	None
Returns:	None
Modifies:	AF, BC, DE

Name:	POSIT, 00C6H
Function:	Locates cursor at specified position.
Entry:	Column in [H], row in [L]
Returns:	None
Modifies:	AF

Name:	FNKSB, 00C9H
Function:	Checks if function key display is active. If so, displays it, otherwise do nothing.
Entry:	FNKFLG
Returns:	None
Modifies:	All

Name:	ERAFNK, 00CCH
Function:	Erases function key display
Entry:	None
Returns:	None
Modifies:	All

Name:	DSPFNK, 00CFH
Function:	Displays function key display
Entry:	None
Returns:	None
Modifies:	All

Name:	TOTEXT, 00D2H
Function:	Forces screen to text mode
Entry:	None
Returns:	None
Modifies:	All

Following are used to access game I/O

Name:	GTSTCK, 00D5H
Function:	Returns the current status of joy stick
Entry:	Joy stick ID in [Acc]
Returns:	Direction in [Acc]
Modifies:	All

Name:	GTTRIG, 00D8H
Function:	Returns the current status of trigger button
Entry:	Trigger button ID in [Acc]
Returns:	Returns 0 in [Acc] if not pressed, 255 otherwise.
Modifies:	AF

Name:	GTPAD, 00DBH
Function:	Checks current status of touch PAD
Entry:	ID in [Acc]
Returns:	Value in [Acc]
Modifies:	All

Name:	GTPDL, 00DEH
Function:	Returns the value of paddle
Entry:	Paddle ID in [Acc]
Returns:	Value in [Acc]
Modifies:	All

Following are used to access cassette tape

Name:	TAPION, 00E1H
Function:	Sets motor on and reads header from tape
Entry:	None
Returns:	Carry flag set if aborted
Modifies:	All

Name:	TAPIN, 00E4H
Function:	Inputs from tape
Entry:	None
Returns:	Data in [Acc], carry flag set if aborted.
Modifies:	All

Name:	TAPIOF, 00E7H
Function:	Stops reading from tape
Entry:	None
Returns:	None
Modifies:	None

Name:	TAPOON, 00EAH
Function:	Sets motor on and writes header block to cassette.
Entry:	[Acc] holds non-0 value if a long header desired, 0 if a short header desired.
Returns:	Carry flag set if aborted
Modifies:	All

Name:	TAPOUT, 00EDH
Function:	Outputs to tape
Entry:	Data to be output in [Acc]
Returns:	Carry flag set if aborted
Modifies:	All

Name:	TAPOOF, 00F0H
Function:	Stops writing to tape
Entry:	None
Returns:	None
Modifies:	None

Name:	STMOTR, 00F3H
Function:	Sets cassette motor
Entry:	0 in [Acc] to stop, 1 to start, 255 to flip.
Returns:	None
Modifies:	AF

Following are used to handle queues

Name:	LFTQ, 00F6H
Function:	Returns how many bytes are left in queue
Entry:	
Returns:	
Modifies:	

Name:	PUTQ, 00F9H
Function:	Puts a byte in queue
Entry:	
Returns:	
Modifies:	

Following are used by GENGRP and ADVGRP modules

Name:	FETCHC, 0114H
Function:	Fetches current physical address and mask pattern.
Entry:	None
Returns:	Address in [HL], mask pattern in [Acc]
Modifies:	A, HL

Name:	STOREC, 0117H
Function:	Stores to physical address and mask pattern
Entry:	Address in [HL], mask pattern in [Acc]
Returns:	None
Modifies:	None

Name:	GTASPC, 0126H
Function:	Returns aspect ratio
Entry:	None
Returns:	DE, HL
Modifies:	DE, HL

Name:	PNTINI, 0129H
Function:	Initializes for PAINT
Entry:	
Returns:	
Modifies:	

Name:	SCANR, 012CH
Function:	Scans pixels to right
Entry:	
Returns:	
Modifies:	

Name:	SCANL, 012FH
Function:	Scans pixels to left
Entry:	
Returns:	
Modifies:	

Following are the additional entries

Name:	CHGCAP, 0132H
Function:	Changes the status of CAP lamp
Entry:	0 in [Acc] to turn off the lamp, non 0 otherwise.
Returns:	None
Modifies:	AF

Name:	CHGSND, 0135H
Function:	Changes the status of 1 bit sound port.
Entry:	0 in [Acc] to turn off, non 0 otherwise.
Returns:	None
Modifies:	AF

Name:	RSLREG, 0138H
Function:	Reads what is currently output to primary slot register.
Entry:	None
Returns:	Result in [Acc]
Modifies:	A

Name:	WSLREG, 013BH
Function:	Writes to primary slot register.
Entry:	Value in [Acc]
Returns:	None
Modifies:	None

Name:	RDVDP, 013EH
Function:	Reads VDP's status register.
Entry:	None
Returns:	Data in [Acc]
Modifies:	A

Name:	SNSMAT, 0141H
Function:	Returns the status of specified row of a keyboard matrix.
Entry:	Row # in [Acc]
Returns:	Status in [Acc], corresponding bit is reset to 0 if being pressed.
Modifies:	AF

Name:	ISFLIO, 014AH
Function:	Checks if we're doing device I/O
Entry:	None
Returns:	Non zero if so, zero otherwise
Modifies:	AF

Name:	OUTDLP, 014DH
Function:	Outputs to LPT
Entry:	Code in [Acc]
Returns:	None
Modifies:	F
Note:	This entry differs from LPTOUT in that: 1) TABs are expanded to spaces, 2) HIRAGANA and graphics symbol are converted when non-MSX printer is in use, 3) a jump to 'device I/O error' is made when aborted.

Name:	KILBUF, 0156H
Function:	Clears keyboard buffer
Entry:	None
Returns:	None
Modifies:	HL

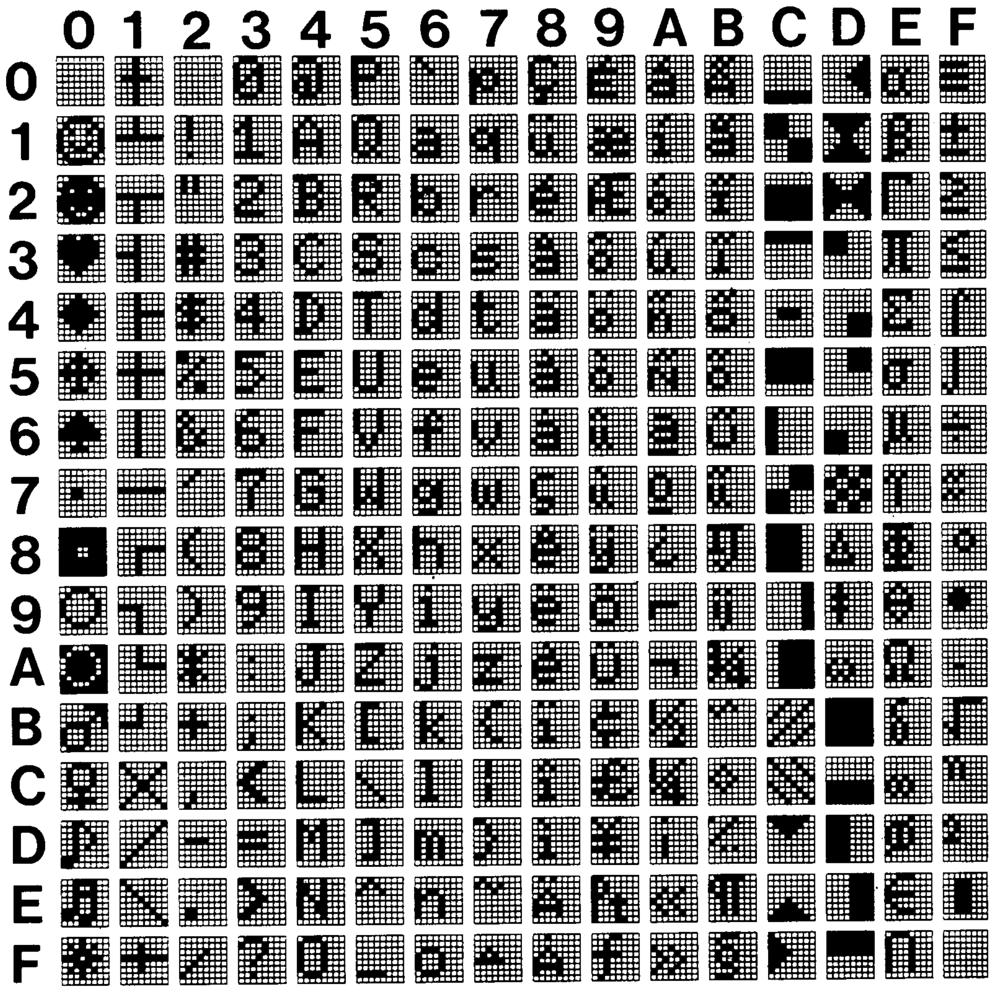
Name:	CALBAS, 0159H
Function:	Performs far_call (i.e., inter-slot call) into BASIC interpreter.
Entry:	Address in [IX]
Returns:	Who knows?
Modifies:	ditto

APPENDIX B

INTERNATIONAL MSX VERSIONS

o Character Set (Common to DIN, French, INT, UK, and USA)

Character Code Table (International)



Note: The font of the character '0' (Zero) is different for DIN version. See figure.

```
  ***
 *   *
 *   *
 * * *
 *   *
 *   *
  ***
```

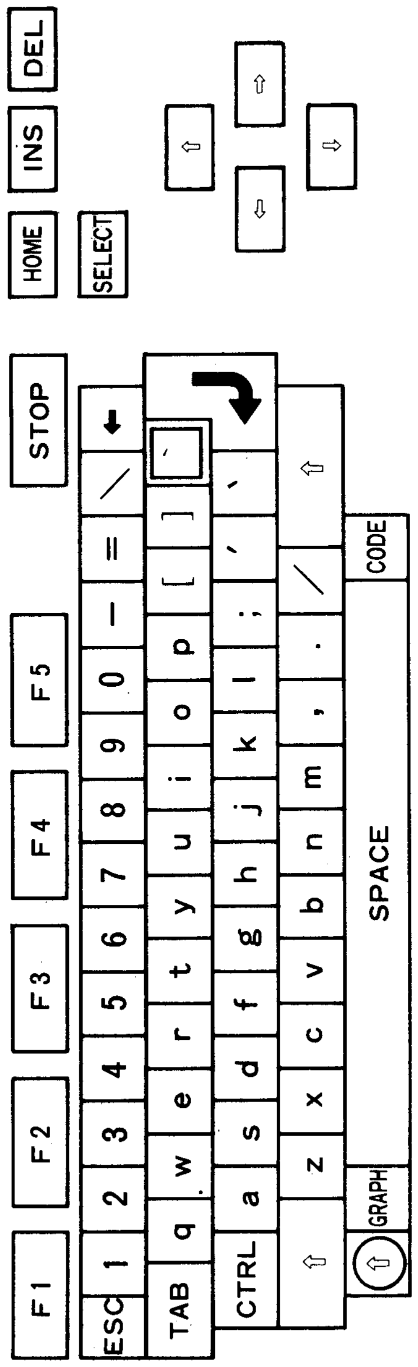
INTERNATIONAL MSX VERSIONS

- Decode International (USA)

I N T			0	1	2	3	4	5	6	7
0	Normal		0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37
		Shift	) 29	! 21	@ 40	# 23	$ 24	% 25	^ 5E	& 26
	Graph		○ 09	¼ AC	½ AB	¾ BA	η EF	‰ BD	⌠ F4	√ FB
		Shift	◙ 0A		^{2} FD	^{n} FC			⌡ F5	
	Code		δ EB	ƒ 9F	‡ D9	§ BF	¢ 9B	ÿ 98	α E0	β E1
		Shift	∆ D8	¡ AD	Pt 9E	¶ BE	£ 9C	Y 9D		
1	Normal		8 38	9 39	- 2D	= 3D	\ 5C	[5B	] 5D	; 3B
		Shift	* 2A	(28	_ 5F	+ 2B	¦ 7C	{ 7B	} 7D	: 3A
	Graph		∞ EC	• 07	─ 17	± F1	╲ 1E	☺ 01	♪ 0D	♠ 06
		Shift		◘ 08	┼ 1F	≡ F0	│ 16	☻ 02	♫ 0E	♦ 04
	Code		γ E7	ç 87	ε EE	θ E9		φ ED	ω DA	ũ B7
		Shift	Γ E2	Ç 80				Φ E8	Ω EA	Ũ B6
2	Normal		' 27	` 60	, 2C	. 2E	/ 2F	` dead key	a 61	b 62
		Shift	" 22	~ 7E	< 3C	> 3E	? 3F	´	A 41	B 42
	Graph		♣ 05	BB	≦ F3	≧ F2	╱ 1D	`	▬ C4	┴ 11
		Shift	♥ 03	≈ F7	« AE	» AF	÷ F6	´	▌ FE	
	Code		ĳ B9	σ E5	å 86	a A6	o A7	^	ä 84	ù 97
		Shift	Ĳ B8	Σ E4	Å 8F		¿ A8	¨	Ä 8E	
3	Normal		c 63	d 64	e 65	f 66	g 67	h 68	i 69	j 6A
		Shift	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A
	Graph		◇ BC	▟ C7	▼ CD	├ 14	┼ 15	┤ 13	▄ DC	▌ C6
		Shift	FA	▚ C1	▲ CE	▗ D4	┼ 10	▖ D6	▀ DF	█ CA
	Code		ì 8D	ï 8B	î 8C	ö 94	ü 81	ã B1	í A1	æ 91
		Shift				Ö 99	Ü 9A	Ã B0		Æ 92
4	Normal		k 6B	l 6C	m 6D	n 6E	o 6F	p 70	q 71	r 72
		Shift	K 4B	L 4C	M 4D	N 4E	O 4F	P 50	Q 51	R 52
	Graph		▌ DD	█ C8	♂ 0B	┘ 1B	▄ C2	█ DB	╲╲ CC	┌ 18
		Shift	▐ DE	▐ C9	♀ 0C	▘ D3	▀ C3	▚ D7	╱╱ CB	┌ A9
	Code		ĩ B3	õ B5	µ E6	ñ A4	ó A2	ú A3	â 83	ô 93
		Shift	Ĩ B2	Õ B4		Ñ A5		‼ E3		
5	Normal		s 73	t 74	u 75	v 76	w 77	x 78	y 79	z 7A
		Shift	S 53	T 54	U 55	V 56	W 57	X 58	Y 59	Z 5A
	Graph		⋈ D2	┬ 12	▂ C0	└ 1A	▶ CF	╳ 1C	┐ 19	☼ 0F
		Shift	⧗ D1		▆ C5	▝ D5	◀ D0	● F9	¬ AA	○ F8
	Code		ë 89	û 96	é 82	ò 95	ê 88	è 8A	á A0	à 85
		Shift			É 90					

INTERNATIONAL MSX VERSIONS

o Layout International (USA)

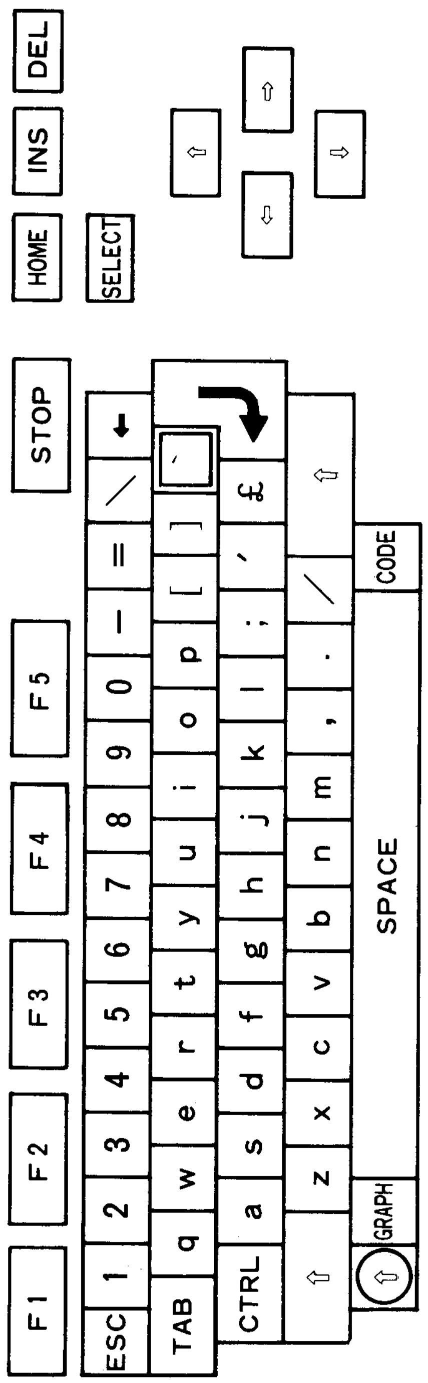

F 1 F 2 F 3 F 4 F 5 STOP
HOME INS DEL
SELECT
ESC 1 2 3 4 5 6 7 8 9 0 − = /
TAB q w e r t y u i o p [ ]
CTRL a s d f g h j k l ; ' `
z x c v b n m , . /
GRAPH SPACE CODE

o Decode UK

UK			0	1	2	3	4	5	6	7
0	Normal		0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37
		Shift	) 29	! 21	@ 40	# 23	$ 24	% 25	^ 5E	& 26
	Graph		○ 09	¼ AC	½ AB	¼ BA	ɳ EF	‰ BD	⌠ F4	√ FB
		Shift	◙ 0A		² FD	ⁿ FC			⌡ F5	
	Code		δ EB	ƒ 9F	‡ D9	§ BF	¢ 9B	ÿ 98	α E0	β E1
		Shift	∆ D8	¡ AD	Pt 9E	¶ BE	£ 9C	¥ 9D		
1	Normal		8 38	9 39	– 2D	= 3D	\ 5C	[5B	] 5D	; 3B
		Shift	* 2A	(28	_ 5F	+ 2B	| 7C	{ 7B	} 7D	: 3A
	Graph		∞ EC	• 07	— 17	± F1	╲ 1E	☺ 01	♪ 0D	♠ 06
		Shift		◘ 08	┼ 1F	≡ F0	│ 16	☻ 02	♫ 0E	♦ 04
	Code		γ E7	ç 87	ε EE	θ E9	60	ϕ ED	◍ DA	ũ B7
		Shift	Γ E2	Ç 80				Φ E8	Ω EA	Ũ B6
2	Normal		' 27	£ 9C	, 2C	. 2E	/ 2F	` dead key	a 61	b 62
		Shift	" 22	~ 7E	< 3C	> 3E	? 3F	´	A 41	B 42
	Graph		♣ 05	⌐ BB	≤ F3	≥ F2	╱ 1D	`	▬ C4	┴ 11
		Shift	♥ 03	≈ F7	« AE	» AF	÷ F6	´	▌ FE	
	Code		ij B9	σ E5	å 86	a A6	o A7	^	ä 84	ù 97
		Shift	IJ B8	Σ E4	Å 8F		¿ A8	¨	A 8E	
3	Normal		c 63	d 64	e 65	f 66	g 67	h 68	i 69	j 6A
		Shift	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A
	Graph		◇ BC	▞ C7	▼ CD	├ 14	┼ 15	┤ 13	▄ DC	▌ C6
		Shift	. FA	▚ C1	▲ CE	▖ D4	┼ 10	▖ D6	▀ DF	▐ CA
	Code		ì 8D	ï 8B	î 8C	ö 94	ü 81	ã B1	í A1	æ 91
		Shift				Ö 99	Ü 9A	Ã B0		Æ 92
4	Normal		k 6B	l 6C	m 6D	n 6E	o 6F	p 70	q 71	r 72
		Shift	K 4B	L 4C	M 4D	N 4E	O 4F	P 50	Q 51	R 52
	Graph		▌ DD	▌ C8	♂ 0B	┘ 1B	▄ C2	█ DB	╲╲ CC	┌ 18
		Shift	▐ DE	▐ C9	♀ 0C	▘ D3	▬ C3	▩ D7	╱╱ CB	┌ A9
	Code		ı B3	õ B5	µ E6	ñ A4	ó A2	ú A3	â 83	ô 93
		Shift	I B2	Õ B4		Ñ A5		Π E3		
5	Normal		s 73	t 74	u 75	v 76	w 77	x 78	y 79	z 7A
		Shift	S 53	T 54	U 55	V 56	W 57	X 58	Y 59	Z 5A
	Graph		⧓ D2	┬ 12	▬ C0	└ 1A	▶ CF	╳ 1C	┐ 19	☼ 0F
		Shift	⧗ D1		█ C5	▘ D5	◀ D0	● F9	┐ AA	○ F8
	Code		ë 89	û 96	é 82	ò 95	ê 88	è 8A	á A0	à 85
		Shift			É 90					

INTERNATIONAL MSX VERSIONS

o Layout UK



o Character Code Table (Japanese)

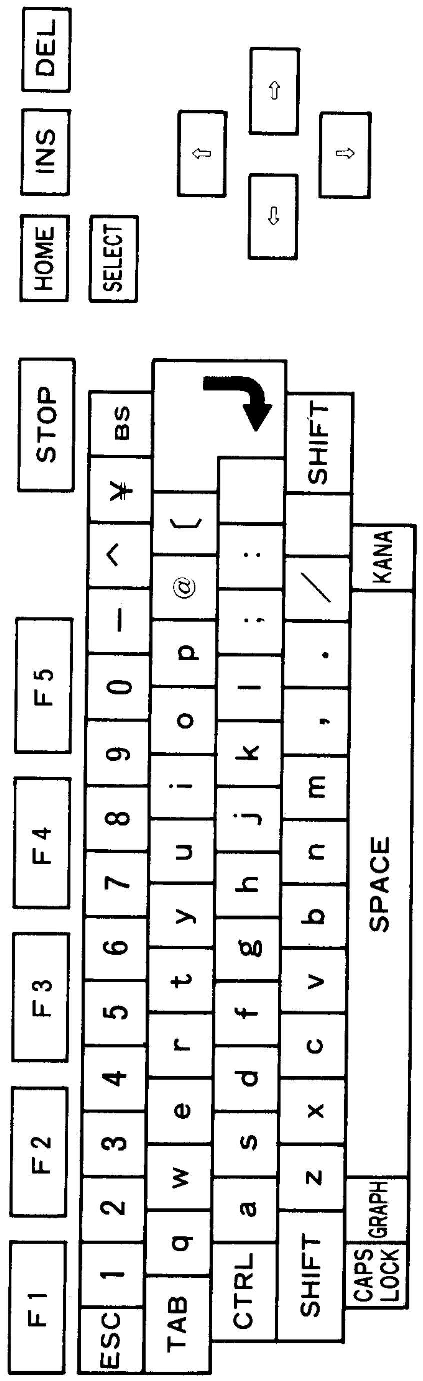
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		π		0	@	P	`	p	♠			ー	タ	ミ	た	み
1	月	┴	!	1	A	Q	a	q	♥	あ	。	ア	チ	ム	ち	む
2	火	┬	"	2	B	R	b	r	♣	い	「	イ	ツ	メ	つ	め
3	水	┤	#	3	C	S	c	s	♦	う	」	ウ	テ	モ	て	も
4	木	├	$	4	D	T	d	t	○	え	、	エ	ト	ヤ	と	や
5	金	┼	%	5	E	U	e	u	●	お	・	オ	ナ	ユ	な	ゆ
6	土	│	&	6	F	V	f	v	を	か	ヲ	カ	ニ	ヨ	に	よ
7	日	─	'	7	G	W	g	w	ぁ	き	ァ	キ	ヌ	ラ	ぬ	ら
8	年	┌	(	8	H	X	h	x	ぃ	く	ィ	ク	ネ	リ	ね	り
9	円	┐	)	9	I	Y	i	y	ぅ	け	ゥ	ケ	ノ	ル	の	る
A	時	└	*	:	J	Z	j	z	ぇ	こ	ェ	コ	ハ	レ	は	れ
B	分	┘	+	;	K	[	k	{	ぉ	さ	ォ	サ	ヒ	ロ	ひ	ろ
C	秒	╳	,	<	L	¥	l	¦	ゃ	し	ャ	シ	フ	ワ	ふ	わ
D	百	大	-	=	M	]	m	}	ゅ	す	ュ	ス	ヘ	ン	へ	ん
E	千	中	.	>	N	^	n	~	ょ	せ	ョ	セ	ホ	゛	ほ	
F	万	小	/	?	O	_	o		っ	そ	ッ	ソ	マ	゜	ま	

o Decode Japanese 1

JIS			0	1	2	3	4	5	6	7
0	Normal		0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37
		Shift		! 21	" 22	# 23	$ 24	% 25	& 26	' 27
	Graph		万 0F	日 07	月 01	火 02	水 03	木 04	金 05	土 06
	Kana		わ FC	ぬ E7	ふ EC	あ 91	う 93	え 94	お 95	や F4
		Caps	ワ DC	ヌ C7	フ CC	ア B1	ウ B3	エ B4	オ B5	ヤ D4
1	Normal		8 38	9 39	- 2D	^ 5E	¥ 5C	@ 40	[5B	; 3B
		Shift	(28	) 29	= 3D	~ 7E	¦ 7C	‘ 60	{ 7B	+ 2B
	Graph		百 0D	千 E0	─ 17		円 09		○ 84	♣ 82
	Kana		ゆ F5	よ F6	ほ EE	へ ED	ー B0	゛ DE	゜ DF	れ FA
		Caps	ユ D5	ヨ D6	ホ CE	ヘ CD	ー B0	゛ DE	゜ DF	レ DA
2	Normal		: 3A	] 5D	, 2C	. 2E	/ 2F		a 61	b 62
		Shift	* 2A	} 7D	< 3C	> 3E	? 3F	_ 5F	A 41	B 42
	Graph		♥ 81	● 85	小 1F	大 1D	♠ 80	◆ 83		┘ 1B
	Kana		け 99	む F1	ね E8	る F9	め F2	ろ FB	ち E1	こ 9A
		Caps	ケ B9	ム D1	ネ C8	ル D9	メ D2	ロ DB	チ C1	コ BA
3	Normal		c 63	d 64	e 65	f 66	g 67	h 68	i 69	j 6A
		Shift	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A
	Graph		└ 1A	├ 14	┌ 18	┼ 15	┤ 13	時 0A	│ 16	
	Kana		そ 9F	し 9C	い 92	は EA	き 97	く 98	に E6	ま EF
		Caps	ソ BF	シ BC	イ B2	ハ CA	キ B7	ク B8	ニ C6	マ CF
4	Normal		k 6B	l 6C	m 6D	n 6E	o 6F	p 70	q 71	r 72
		Shift	K 4B	L 4C	M 4D	N 4E	O 4F	P 50	Q 51	R 52
	Graph			中 1E	分 0B			π 10		┬ 12
	Kana		の E9	り F8	も F3	み F0	ら F7	せ 9E	た E0	す 9D
		Caps	ノ C9	リ D8	モ D3	ミ D0	ラ D7	セ BE	タ C0	ス BD
5	Normal		s 73	t 74	u 75	v 76	w 77	x 78	y 79	z 7A
		Shift	S 53	T 54	U 55	V 56	W 57	X 58	Y 59	Z 5A
	Graph		秒 0C	┐ 19		┴ 11		╳ 1C	年 08	
	Kana		と E4	か 96	な E5	ひ EB	て E3	さ 9B	ん FD	つ E2
		Caps	ト C4	カ B6	ナ C5	ヒ CB	テ C3	サ BB	ン DD	ツ C2

INTERNATIONAL MSX VERSIONS

o Layout Japanese



o Decode Japanese 2

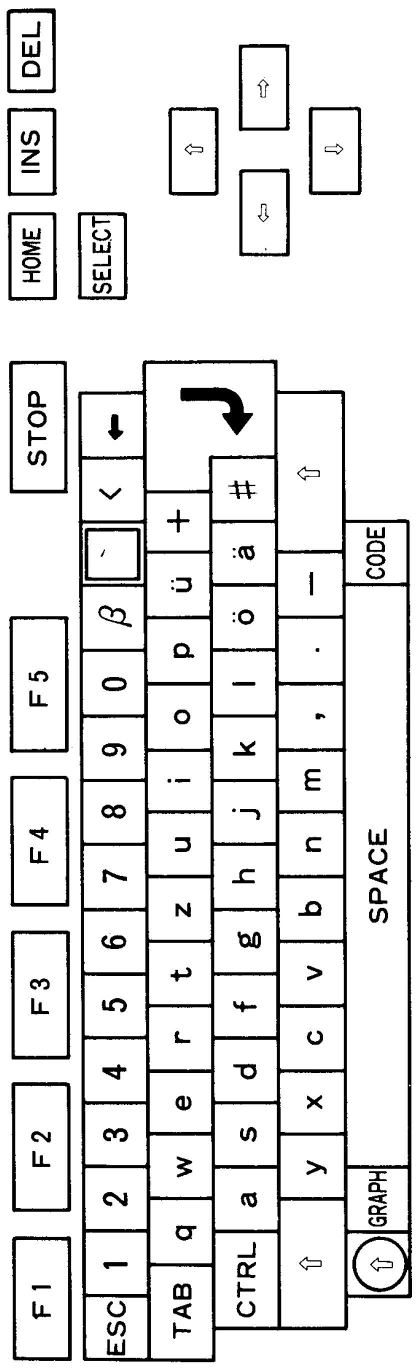
KANA+SHIFT		0	1	2	3	4	5	6	7
0		を 86			ぁ 87	ぅ 89	ぇ 8A	ぉ 8B	ゃ 8C
	Caps	ヲ A6			ァ A7	ゥ A9	ェ AA	ォ AB	ャ AC
1		ゅ 8D	ょ 8E					「 A2	
	Caps	ュ AD	ョ AE					「 A2	
2			」 A3	A4	。 A1	・ A5			
	Caps		」 A3	、 A4	。 A1	・ A5			
3				ぃ 88					
	Caps			ィ A8					
5									っ 8F
	Caps								ッ AF

o Decode DIN

DIN			0	1	2	3	4	5	6	7
0	Normal		0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37
		Shift	= 3D	! 21	" 22	§ BF	$ 24	% 25	& 26	/ 2F
	Graph		○ 09	¼ AC	½ AB	¾ BA	η EF	‰ BD	⌠ F4	╱ 1D
		Shift	◙ 0A		² FD	ⁿ FC		÷ F6	⌡ F5	╲ 1E
	Code		δ EB	¦ 7C	@ 40	ε EE	ç 87	¢ 9B	γ E7	\ 5C
		Shift	Δ D8	¡ AD	Pt 9E	¶ BE	Ç 80	£ 9C	Γ E2	
1	Normal		8 38	9 39	β E1	` dead key	< 3C	ü 81	+ 2B	ö 94
		Shift	(28	) 29	? 3F	´	> 3E	Ü 9A	* 2A	Ö 99
	Graph		∞ EC	• 07	♪ 0D	` 60	« AE	☺ 01	± F1	♠ 06
		Shift		◘ 08	♫ 0E	´ 27	» AF	☻ 02	┼ 1F	♦ 04
	Code		[5B	] 5D	θ E9	^ dead key	≤ F3	ϕ ED	ω DA	ũ B7
		Shift	{ 7B	} 7D	¿ A8	¨	≥ F2	Φ E8	Ω EA	Ũ B6
2	Normal		ä 84	# 23	, 2C	. 2E	- 2D		a 61	b 62
		Shift	Ä 8E	^ 5E	; 3B	: 3A	_ 5F		A 41	B 42
	Graph		♣ 05	~ 7E	√ FB	│ 16	─ 17		▬ C4	┴ 11
		Shift	♥ 03	⌒ BB	≈ F7		≡ F0		▮ FE	
	Code		ĳ B9	σ E5	ȧ 86	ª A6	º A7		α E0	ù 97
		Shift	Ĳ B8	Σ E4	Å 8F					
3	Normal		c 63	d 64	e 65	f 66	g 67	h 68	i 69	j 6A
		Shift	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A
	Graph		◇ BC	▞ C7	▼ CD	├ 14	┼ 15	┤ 13	▄ DC	▌ C6
		Shift	- FA	▚ C1	▲ CE	▗ D4	┼ 10	▖ D6	▀ DF	█ CA
	Code		ì 8D	ï 8B	î 8C	ƒ 9F	ÿ 98	ã B1	í A1	æ 91
		Shift						Ã B0		Æ 92
4	Normal		k 6B	l 6C	m 6D	n 6E	o 6F	p 70	q 71	r 72
		Shift	K 4B	L 4C	M 4D	N 4E	O 4F	P 50	Q 51	R 52
	Graph		▌ DD	█ C8	♂ 0B	┘ 1B	▬ C2	█ DB	╲ CC	┌ 18
		Shift	▐ DE	▐ C9	♀ 0C	▘ D3	▀ C3	▩ D7	╱ CB	┌ A9
	Code		ĩ B3	õ B5	µ E6	ñ A4	ó A2	ú A3	â 83	ô 93
		Shift	Ĩ B2	Õ B4		Ñ A5		Π E3		
5	Normal		s 73	t 74	u 75	v 76	w 77	x 78	z 7A	y 79
		Shift	S 53	T 54	U 55	V 56	W 57	X 58	Z 5A	Y 59
	Graph		⧓ D2	┬ 12	▄ C0	└ 1A	▶ CF	╳ 1C	┐ 19	☼ 0F
		Shift	⧗ D1	‡ D9	█ C5	▝ D5	◀ D0	● F9	┐ AA	○ F8
	Code		ë 89	û 96	é 82	ò 95	ê 88	è 8A	à A0	á 85
		Shift			É 90					¥ 9D

INTERNATIONAL MSX VERSIONS

o Layout DIN

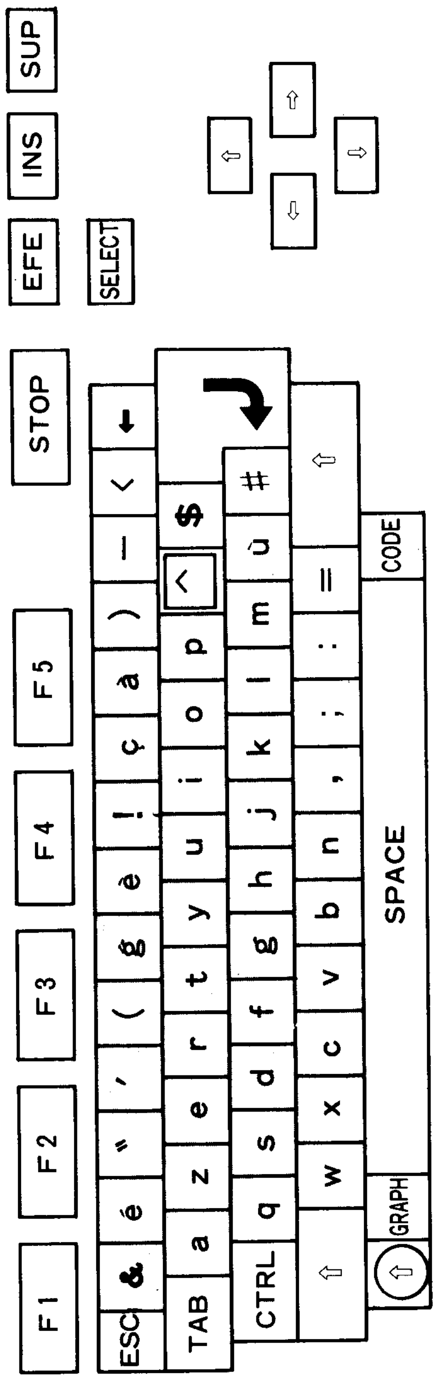


F 1
F 2
F 3
F 4
F 5
STOP
HOME
INS
DEL
SELECT
ESC 1 2 3 4 5 6 7 8 9 0 ß ´ < ←
TAB q w e r t z u i o p ü +
CTRL a s d f g h j k l ö ä #
⇧ y x c v b n m , . — ⇧
GRAPH
SPACE
CODE

o Decode French

FR			0	1	2	3	4	5	6	7
0	Normal		à 85	& 26	é 82	" 22	' 27	(28	§ BF	è 8A
		Shift	0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37
	Graph		○ 09	£ AC	½ AB	¼ BA	˜ BB	η EF	⌠ F4	√ FB
		Shift	◙ 0A	| 16	² FD	ⁿ FC	≈ F7		⌡ F5	
	Code		δ EB	¦ 7C	@ 40	α E0	` 60	{ 7B	^ 5E	ε EE
		Shift	Δ D8	¡ AD	É 90	Pt 9E		[5B	¶ BE	~ 7E
1	Normal		! 21	ç 87	) 29	– 2D	< 3C	^	$ 24	m 6D
		Shift	8 38	9 39	° F8	_ 5F	> 3E	¨ dead key	* 2A	M 4D
	Graph		∞ EC	• 07	☺ 01	─ 17	« AE	^	♪ 0D	♠ 06
		Shift		◘ 08	☻ 02	┼ 1F	» AF	¨	♫ 0E	♦ 04
	Code		γ E7	θ E9	} 7D	∅ ED	≤ F3	^	¢ 9B	ũ B7
		Shift	Γ E2	Ç 80	] 5D	Φ E8	≥ F2	¨		Ũ B6
2	Normal		ù 97	# 23	; 3B	: 3A	= 3D		q 71	b 62
		Shift	% 25	£ 9C	. 2E	/ 2F	+ 2B		Q 51	B 42
	Graph		♣ 05	‰ BD	÷ F6	╲ 1E	± F1		▬ C4	┴ 11
		Shift	♥ 03			╱ 1D	≡ F0		▌ FE	
	Code		ĳ B9	σ E5	à 86	a̲ A6	o̲ A7		ä 84	β E1
		Shift	Ĳ B8	Σ E4	Å 8F	\ 5C			Ä 8E	
3	Normal		c 63	d 64	e 65	f 66	g 67	h 68	i 69	j 6A
		Shift	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A
	Graph		◇ BC	▞ C7	▼ CD	├ 14	┼ 15	┤ 13	▄ DC	▌ C6
		Shift	- FA	▚ C1	▲ CE	▖ D4	┼ 10	▖ D6	▀ DF	█ CA
	Code		ì 8D	ï 8B	î 8C	ö 94	ü 81	ã B1	í A1	æ 91
		Shift				Ö 99	Ü 9A	Ã B0		Æ 92
4	Normal		k 6B	l 6C	, 2C	n 6E	o 6F	p 70	a 61	r 72
		Shift	K 4B	L 4C	? 3F	N 4E	O 4F	P 50	A 41	R 52
	Graph		▌ DD	▌ C8	♂ 0B	┘ 1B	▆ C2	█ DB	⧹ CC	┌ 18
		Shift	▐ DE	▐ C9	♀ 0C	▪ D3	▔ C3	▩ D7	⧸ CB	┌ A9
	Code		ĩ B3	õ B5	μ E6	ñ A4	ó A2	ú A3	â 83	ô 93
		Shift	Ĩ B2	Õ B4	¿ A8	Ñ A5		∏ E3		
5	Normal		s 73	t 74	u 75	v 76	z 7A	x 78	y 79	w 77
		Shift	S 53	T 54	U 55	V 56	Z 5A	X 58	Y 59	W 57
	Graph		⧓ D2	┬ 12	▂ C0	└ 1A	▶ CF	╳ 1C	┐ 19	☼ 0F
		Shift	⧗ D1	‡ D9	█ C5	▘ D5	◀ D0	● F9	┐ AA	
	Code		ë 89	û 96	ÿ 98	ò 95	ê 88	ƒ 9F	á A0	ω DA
		Shift							¥ 9D	Ω EA

o Layout French



```
      ;
      ;       Following short  routines  are  to perform inter-slot read/write
      ;       and call facility.
      ;


      ;
      ;       Read primitive
      ;
F380  (RDPRIM, 5)
                    OUT     PPI.AW          ;Select primary slot
                    MOV     E,M             ;Read from slot
                    JMPR    WRPRM1          ;Restore current setting
      ;
      ;       Write primitive
      ;
F385  (WRPRIM, 7)
                    OUT     PPI.AW          ;Select primary slot
                    MOV     M,E             ;Write to slot
            WRPRM1: MOV     A,D             ;Load current setting
                    OUT     PPI.AW          ;Restore current setting
                    RET
      ;
      ;       Call primitive
      ;
F38C  (CLPRIM, 14)
                    OUT     PPI.AW          ;Select primary slot
                    EXAF                    ;Restore [Acc] and flags
                    CALL    CLPRIM+12       ;Perform indirect call by IX
                    EXAF                    ;Save possible returned value
                    POP     PSW             ;Get old slot status
                    OUT     PPI.AW          ;Restore it
                    EXAF                    ;Restore possible returned
                                            ;value
                    RET
                    IX
                    PCHL
```

```
          Following are definition of hooks and their functions

          name           - name of hook
          where          - where in what module it is used
          purpose        - what purpose it is used for

FD9A  (HOKJMP,0)
      ;       name:           H.KEYI
      ;       where:          MSXIO, at the beginning of interrupt handler
      ;       purpose:        to do  additional  interrupt  handling  such  as
      ;                       RS232C
      ;
FD9A  (H.KEYI,5)
      ;       name:           H.TIMI
      ;       where:          MSXIO, in timer interrupt handler
      ;       purpose:        to allow other  interrupt  handling  invoked  by
      ;                       timer
      ;
FD9F  (H.TIMI,5)
      ;       name:           H.CHPU
      ;       where:          MSXIO, at  the  beginning  of  CHPUT  (CHaracter
      ;                       outPUT) routine
      ;       purpose:        to allow other console output devices to be used
      ;
FDA4  (H.CHPU,5)
      ;       name:           H.DSPC
      ;       where:          MSXIO, at  the  beginning  of  DSPCSR   (DiSPlay
      ;                       CurSoR) routine
      ;       purpose:        to allow other console output devices to be used
      ;
FDA9  (H.DSPC,5)
      ;       name:           H.ERAC
      ;       where:          MSXIO, at the beginning of ERACSR (ERAse CurSoR)
      ;                       routine
      ;       purpose:        to allow other console output devices to be used
```

```
FDAE  (H.ERAC,5)
      ;       name:        H.DSPF
      ;       where:       MSXIO, at  the  beginning  of  DSPFNK   (DiSPlay
      ;                    FuNction Key) routine
      ;       purpose:     to allow other console output devices to be used
      ;
FDB3  (H.DSPF,5)
      ;       name:        H.ERAF
      ;       where:       MSXIO,   at   the  beginning  of  ERAFNK  (ERAse
      ;                    FuNction Key) routine
      ;       purpose:     to allow other console output devices to be used
      ;
FDB8  (H.ERAF,5)
      ;       name:        H.TOTE
      ;       where:       MSXIO, at the beginning of TOTEXT (force  screen
      ;                    TO TEXT mode) routine
      ;       purpose:     to allow other console output devices to be used
      ;
FDBD  (H.TOTE,5)
      ;       name:        H.CHGE
      ;       where:       MSXIO, at  the  beginning  of  CHGET  (CHaracter
      ;                    GET) routine
      ;       purpose:     to allow other console input devices to be used
      ;
FDC2  (H.CHGE,5)
      ;       name:        H.INIP
      ;       where:       MSXIO, at the beginning  of  INIPAT  (INItialize
      ;                    PATtern) routine
      ;       purpose:     to allow other character sets to be used
      ;
FDC7  (H.INIP,5)
      ;       name:        H.KEYC
      ;       where:       MSXIO, at the beginning of  KEYCOD  (KEY  CODer)
      ;                    routine
      ;       purpose:     to allow other key assignments to be used
      ;
FDCC  (H.KEYC,5)
```

```
     ;       name:         H.KYEA
     ;       where:        MSXIO, at  the  beginning  of  KYEASY (KeY EASY)
     ;                     routine
     ;       purpose:      to allow other key assignments to be used
     ;
FDD1 (H.KYEA,5)
     ;       name:         H.NMI
     ;       where:        MSXIO, at  the  beginning  of  NMI (Non Maskable
     ;                     Interrupt) routine
     ;       purpose:      to allow NMI handling
     ;
FDD6 (H.NMI, 5)
     ;       name:         H.PINL
     ;       where:        MSXINL, at  the  beginning  of  PINLIN  (Program
     ;                     INput LINe) routine
     ;       purpose:      to allow other console input  devices  or  other
     ;                     input design to be used
     ;
FDDB (H.PINL,5)
     ;       name:         H.QINL
     ;       where:        MSXINL, at  the  beginning  of  QINLIN (Question
     ;                     mark and INput LINe) routine
     ;       purpose:      to allow other console input  devices  or  other
     ;                     input design to be used
     ;
FDE0 (H.QINL,5)
     ;       name:         H.INLI
     ;       where:        MSXINL, at the beginning of INLIN  (INput  LINe)
     ;                     routine
     ;       purpose:      to allow other console input  devices  or  other
     ;                     input design to be used
     ;
FDE5 (H.INLI,5)
     ;       name:         H.ONGO
     ;       where:        MSXSTS, at the  beginning  of  ONGOTP  (ON  GOTo
     ;                     Procedure) routine
     ;       purpose:      to allow other interrupting devices to be used
```

```
      ;
FDEA  (H.ONGO,5)
      ;       name:            H.DSKO
      ;       where:           MSXSTS, at the beginning of DSKO$ (DiSK Output)
      ;                        routine
      ;       purpose:         to install disk driver
      ;
FDEF  (H.DSKO,5)
      ;       name:            H.SETS
      ;       where:           MSXSTS, at  the  beginning  of  SETS  (SET
      ;                        attributeS) routine
      ;       purpose:         to install disk driver
      ;
FDF4  (H.SETS,5)
      ;       name:            H.NAME
      ;       where:           MSXSTS, at the beginning of NAME (reNAME) routine
      ;       purpose:         to install disk driver
      ;
FDF9  (H.NAME,5)
      ;       name:            H.KILL
      ;       where:           MSXSTS, at the beginning  of  KILL  (KILL  file)
      ;                        routine
      ;       purpose:         to install disk driver
      ;
FDFE  (H.KILL,5)
      ;       name:            H.IPL
      ;       where:           MSXSTS, at the beginning of IPL (Initial Program
      ;                        Load) routine
      ;       purpose:         to install disk driver
      ;
FE03  (H.IPL, 5)
      ;       name:            H.COPY
      ;       where:           MSXSTS, at the beginning of  COPY  (COPY  files)
      ;                        routine
      ;       purpose:         to install disk driver
      ;
FE08  (H.COPY,5)
```

```
      ;       name:           H.CMD
      ;       where:          MSXSTS,  at  the  beginning  of  CMD  (CoMmanD)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE0D  (H.CMD, 5)
      ;       name:           H.DSKF
      ;       where:          MSXSTS, at the beginning  of  DSKF  (DiSK  Free)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE12  (H.DSKF,5)
      ;       name:           H.DSKI
      ;       where:          MSXSTS, at  the  beginning  of DSKI (DiSK Input)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE17  (H.DSKI,5)
      ;       name:           H.ATTR
      ;       where:          MSXSTS, at the beginning  of  ATTR$  (ATTRibute)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE1C  (H.ATTR,5)
      ;       name:           H.LSET
      ;       where:          MSXSTS, at  the  beginning  of  LSET  (Left SET)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE21  (H.LSET,5)
      ;       name:           H.RSET
      ;       where:          MSXSTS, at the beginning  of  RSET  (Right  SET)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE26  (H.RSET,5)
      ;       name:           H.FIEL
      ;       where:          MSXSTS,  at  the  beginning  of  FIELD  (FIELD)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE2B  (H.FIEL,5)
      ;       name:           H.MKI$
      ;       where:          MSXSTS, at the  beginning  of  MKI$  (MaKe  Int)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE30  (H.MKI$,5)
      ;       name:           H.MKS$
      ;       where:          MSXSTS, at  the  beginning of MKS$ (Make Single)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE35  (H.MKS$,5)
      ;       name:           H.MKD$
      ;       where:          MSXSTS, at the beginning of MKD$  (Make  Double)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE3A  (H.MKD$,5)
      ;       name:           H.CVI
      ;       where:          MSXSTS, at  the  beginning  of CVI (Convert Int)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE3F  (H.CVI,5)
      ;       name:           H.CVS
      ;       where:          MSXSTS, at the beginning of  CVS  (Convert  Sng)
      ;                       routine
      ;       purpose:        to install disk driver
      ;
FE44  (H.CVS,5)
      ;       name:           H.CVD
      ;       where:          MSXSTS, at  the  beginning  of CVD (Convert Dbl)
      ;                        routine
      ;       purpose:         to install disk driver
      ;
FE49  (H.CVD,5)
      ;       name:            H.GETP
      ;       where:           SPCDSK, at the GETPTR (GET file PoinTeR) routine
      ;       purpose:         to install disk driver
      ;
FE4E  (H.GETP,5)
      ;       name:            H.SETF
      ;       where:           SPCDSK, at the SETFIL (SET FILe pointer) routine
      ;       purpose:         to install disk driver
      ;
FE53  (H.SETF,5)
      ;       name:            H.NOFO
      ;       where:           SPCDSK, at the NOFOR (NO FOR clause) routine
      ;       purpose:         to install disk driver
      ;
FE58  (H.NOFO,5)
      ;       name:            H.NULO
      ;       where:           SPCDSK, at the NULOPN (NULl file OPeN) routine
      ;       purpose:         to install disk driver
      ;
FE5D  (H.NULO,5)
      ;       name:            H.NTFL
      ;       where:           SPCDSK, at the NTFL0 (NoT FiLe number 0) routine
      ;       purpose:         to install disk driver
      ;
FE62  (H.NTFL,5)
      ;       name:            H.MERG
      ;       where:           SPCDSK, at  the  MERGE  (MERGE  program  files)
      ;                        routine
      ;       purpose:         to install disk driver
      ;
FE67  (H.MERG,5)
      ;       name:            H.SAVE
      ;       where:           SPCDSK, at the SAVE routine
```

```
     ;       purpose:         to install disk driver
     ;
FE6C (H.SAVE,5)
     ;       name:            H.BINS
     ;       where:           SPCDSK, at the BINSAV (BINary SAVe) routine
     ;       purpose:         to install disk driver
     ;
FE71 (H.BINS,5)
     ;       name:            H.BINL
     ;       where:           SPCDSK, at the BINLOD (BINary LOaD) routine
     ;       purpose:         to install disk driver
     ;
FE76 (H.BINL,5)
     ;       name:            H.FILE
     ;       where:           SPCDSK, at the FILES command
     ;       purpose:         to install disk driver
     ;
FE7B (H.FILE,5)
     ;       name:            H.DGET
     ;       where:           SPCDSK, at the DGET (Disk GET) routine
     ;       purpose:         to install disk driver
     ;
FE80 (H.DGET,5)
     ;       name:            H.FILO
     ;       where:           SPCDSK, at the FILOU1 (FILe OUt 1) routine
     ;       purpose:         to install disk driver
     ;
FE85 (H.FILO,5)
     ;       name:            H.INDS
     ;       where:           SPCDSK, at the  INDSKC  (INput  DiSK  Character)
     ;                        routine
     ;       purpose:         to install disk driver
     ;
FE8A (H.INDS,5)
     ;       name:            H.RSLF
     ;       where:           SPCDSK, to re-select old drive
     ;       purpose:         to install disk driver
     ;
FE8F (H.RSLF,5)
     ;       name:           H.SAVD
     ;       where:          SPCDSK, to save current drive
     ;       purpose:        to install disk driver
     ;
FE94 (H.SAVD,5)
     ;       name:           H.LOC
     ;       where:          SPCDSK, at the LOC (LOCation) function
     ;       purpose:        to install disk driver
     ;
FE99 (H.LOC, 5)
     ;       name:           H.LOF
     ;       where:          SPCDSK, at the LOF (Length Of File) function
     ;       purpose:        to install disk driver
     ;
FE9E (H.LOF, 5)
     ;       name:           H.EOF
     ;       where:          SPCDSK, at the EOF (End Of File) function
     ;       purpose:        to install disk driver
     ;
FEA3 (H.EOF, 5)
     ;       name:           H.FPOS
     ;       where:          SPCDSK, at the FPOS (File POSition) function
     ;       purpose:        to install disk driver
     ;
FEA8 (H.FPOS,5)
     ;       name:           H.BAKU
     ;       where:          SPCDSK, at the BAKUPT (BAcK UP) routine
     ;       purpose:        to install disk driver
     ;
FEAD (H.BAKU,5)
     ;       name:           H.PARD
     ;       where:          SPCDEV, at the PARDEV (PARse DEVice name)
     ;                       routine
     ;       purpose:        to expand logical device names
     ;
```

```
FEB2  (H.PARD,5)
      ;       name:          H.NODE
      ;       where:         SPCDEV, at the NODEVN (NO DEVice Name) routine
      ;       purpose:       to set other default device
      ;
FEB7  (H.NODE,5)
      ;       name:          H.POSD
      ;       where:         SPCDEV, at the POSDSK (POSsibly DiSK) routine
      ;       purpose:       to install disk driver
      ;
FEBC  (H.POSD,5)
      ;       name:          H.DEVN
      ;       where:         SPCDEV, at the DEVNAM (DEVice NAMe) routine
      ;       purpose:       to expand logical device names
      ;
FEC1  (H.DEVN,5)
      ;       name:          H.GEND
      ;       where:         SPCDEV, at the GENDSP (GENeral device
      ;                      DiSPatcher)
      ;       purpose:       to expand logical device names
      ;
FEC6  (H.GEND,5)
      ;       name:          H.RUNC
      ;       where:         BIMISC, at the RUNC (RUN Clear) routine
      ;       purpose:
      ;
FECB  (H.RUNC,5)
      ;       name:          H.CLEA
      ;       where:         BIMISC, at the CLEARC (CLEAR Clear) routine
      ;       purpose:
      ;
FED0  (H.CLEA,5)
      ;       name:          H.LOPD
      ;       where:         BIMISC, at the LOPDFT (LOop and set DeFaulT)
      ;                      routine
      ;       purpose:       to use other defaults for variables
      ;
```

```
FED5  (H.LOPD,5)
      ;      name:          H.STKE
      ;      where:         BIMISC, at the STKERR (STacK ERRor) routine
      ;      purpose:
      ;
FEDA  (H.STKE,5)
      ;      name:          H.ISFL
      ;      where:         BIMISC, at the ISFLIO (IS FiLe I/O) routine
      ;      purpose:
      ;
FEDF  (H.ISFL,5)
      ;      name:          H.OUTD
      ;      where:         BIO, at the OUTDO (OUT DO) routine
      ;      purpose:
      ;
FEE4  (H.OUTD,5)
      ;      name:          H.CRDO
      ;      where:         BIO, at the CRDO (CRlf DO) routine
      ;      purpose:
      ;
FEE9  (H.CRDO,5)
      ;      name:          H.DSKC
      ;      where:         BIO, at the DSKCHI (DiSK CHaracter Input)
      ;                     routine
      ;      purpose:
      ;
FEEE  (H.DSKC,5)
      ;      name:          H.DOGR
      ;      where:         GENGRP, at the DOGRPH (DO GRaPH) routine
      ;      purpose:
      ;
FEF3  (H.DOGR,5)
      ;      name:          H.PRGE
      ;      where:         BINTRP, at the PRGEND (PRoGram END) routine
      ;      purpose:
      ;
FEF8  (H.PRGE,5)
```

```
      ;       name:              H.ERRP
      ;       where:             BINTRP, at the ERRPRT (ERRor PRinT) routine
      ;       purpose:
      ;
FEFD  (H.ERRP,5)
      ;       name:
      ;       where:             BINTRP
      ;       purpose:
      ;
FF02  (H.ERRF,5)
      ;       name:              H.READ
      ;       where:             BINTRP, at the READY entry
      ;       purpose:
      ;
FF07  (H.READ,5)
      ;       name:              H.MAIN
      ;       where:             BINTRP, at the MAIN entry
      ;       purpose:
      ;
FF0C  (H.MAIN,5)
      ;       name:              H.DIRD
      ;       where:             BINTRP, at the DIRDO (DIRect statement DO).
      ;       purpose:
      ;
FF11  (H.DIRD,5)
      ;       name:
      ;       where:             BINTRP
      ;       purpose:
      ;
FF16  (H.FINI,5)
      ;       name:
      ;       where:             BINTRP
      ;       purpose:
      ;
FF1B  (H.FINE,5)
      ;       name:
      ;       where:             BINTRP
```

```
     ;       purpose:
     ;
FF20 (H.CRUN,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
FF25 (H.CRUS,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
FF2A (H.ISRE,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
FF2F (H.NTFN,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
FF34 (H.NOTR,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
FF39 (H.SNGF,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
FF3E (H.NEWS,5)
     ;       name:
     ;       where:          BINTRP
     ;       purpose:
     ;
```

```
FF43  (H.GONE,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF48  (H.CHRG,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF4D  (H.RETU,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF52  (H.PRTF,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF57  (H.COMP,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF5C  (H.FINP,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF61  (H.TRMN,5)
      ;       name:
      ;       where:           BINTRP
      ;       purpose:
      ;
FF66  (H.FRME,5)
      ;       name:
```

```
      ;       where:          BINTRP
      ;       purpose:
      ;
FF6B  (H.NTPL,5)
      ;       name:
      ;       where:          BINTRP
      ;       purpose:
      ;
FF70  (H.EVAL,5)
      ;       name:
      ;       where:          BINTRP
      ;       purpose:
      ;
FF75  (H.OKNO,5)
      ;       name:
      ;       where:          BINTRP
      ;       purpose:
      ;
FF7A  (H.FING,5)
      ;       name:           H.ISMI
      ;       where:          BINTRP, at the ISMID$ (IS MID$) routine
      ;       purpose:
      ;
FF7F  (H.ISMI,5)
      ;       name:           H.WIDT
      ;       where:          BINTRP, at the WIDTHS (WIDTH) routine
      ;       purpose:
      ;
FF84  (H.WIDT,5)
       ;       name:           H.LIST
      ;       where:          BINTRP, at the LIST routine
      ;       purpose:
      ;
FF89  (H.LIST,5)
      ;       name:           H.BUFL
      ;       where:          BINTRP, at the BUFLIN (BUFfer LINe) routine
      ;       purpose:
        ;
FF8E    (H.BUFL,5)
        ;       name:           H.FRQI
        ;       where:          BINTRP, at the FRQINT routine
        ;       purpose:
        ;
FF93    (H.FRQI,5)
        ;       name:
        ;       where:          BINTRP
        ;       purpose:
        ;
FF98    (H.SCNE,5)
        ;       name:           H.FRET
        ;       where:          BISTRS, at the FRETMP (FREe up TeMPoraries)
        ;                       routine
        ;       purpose:
        ;
FF9D    (H.FRET,5)
        ;       name:           H.PTRG
        ;       where:          BIPTRG, at the PTRGET (PoinTeR GET) routine
        ;       purpose:        to use other variable names than default
        ;
FFA2    (H.PTRG,5)
        ;       name:           H.PHYD
        ;       where:          MSXIO, at the PHYDIO (PHYsical Disk I/O) routine
        ;       purpose:        to install disk driver
        ;
FFA7    (H.PHYD,5)
        ;       name:           H.FORM
        ;       where:          MSXIO, at the FORMAT (disk FORMATter) routine
        ;       purpose:        to install disk driver
        ;
FFAC    (H.FORM,5)
        ;       name:           H.ERRO
        ;       where:          BINTRP, at the ERROR routine
        ;       purpose:        to trap errors from application programs
        ;
```

```
FFB1  (H.ERRO,5)
      ;       name:           H.LPTO
      ;       where:          MSXIO, at the LPTOUT (Line PrinTer OUTput)
      ;                       routine
      ;       purpose:        to use other printer than default
      ;
FFB6  (H.LPTO,5)
      ;       name:           H.LPTS
      ;       where:          MSXIO, at the LPTSTT (Line PrinTer STaTus)
      ;                       routine
      ;       purpose:        to use other printer than default
      ;
FFBB  (H.LPTS,5)
      ;       name:           H.SCRE
      ;       where:          MSXSTS, at the entry to SCREEN statement.
      ;       purpose:        To expand SCREEN statement.
      ;
FFC0  (H.SCRE,5)
      ;       name:           H.PLAY
      ;       where:          MSXSTS, at the entry to PLAY statement.
      ;       purpose:        To expand PLAY statement.
      ;
FFC5  (H.PLAY,5)
      ;
FFCA  (ENDWRK,0)                ;end of work area
```

ISBN 0-933063-00-8