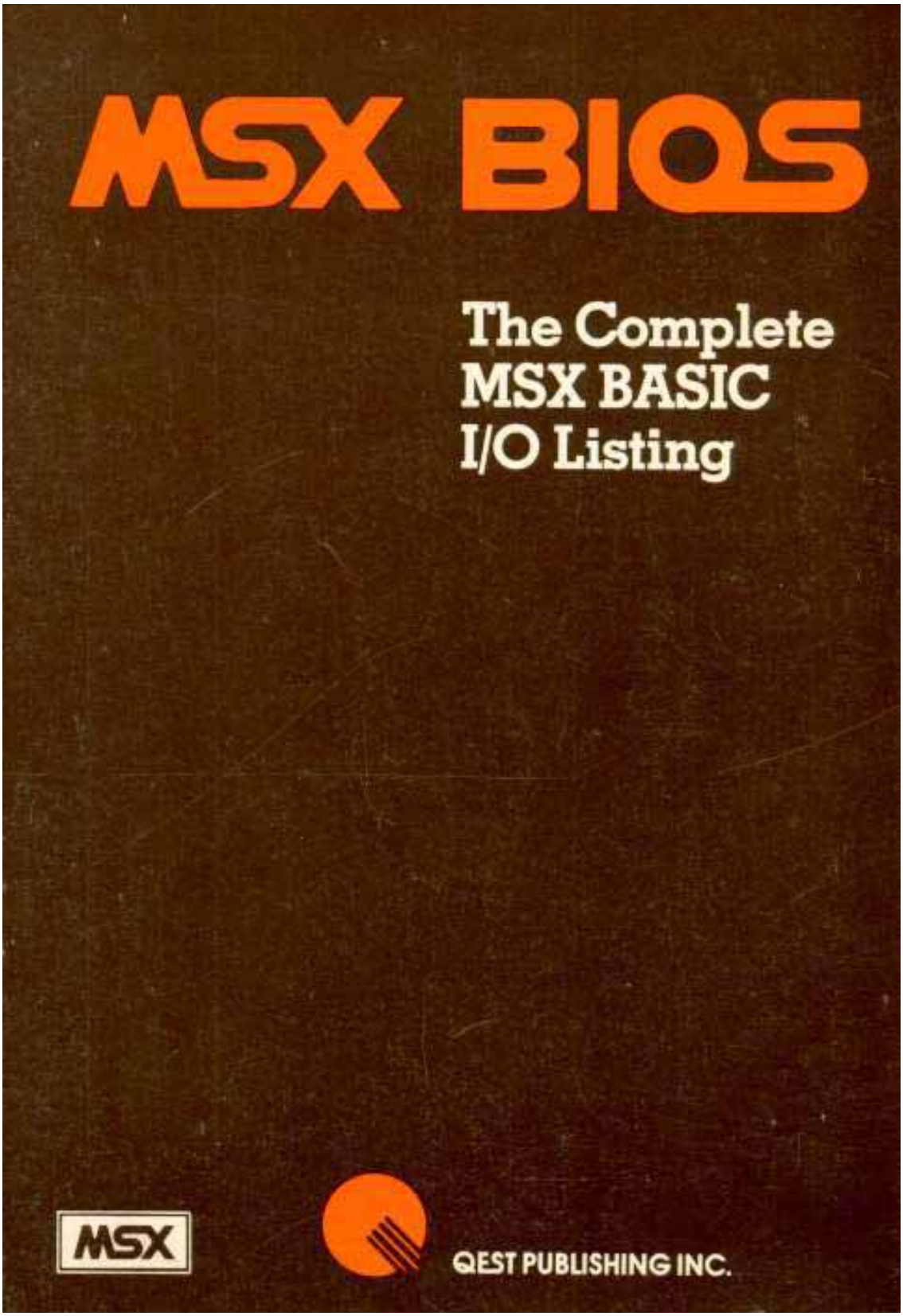


Edited: January 1985
by Steven M. Ting
Graphic design: Mervin Fong.

Printed in United States

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

MSX BIOS SYMBOL TABLE

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

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

F806 HOLD8
098F INESC
139D INIFNK
05D2 INIGRP
061F INIMLT
2680 INIT
0538 INIT32
049D INITIO
050E INITXT
23D5 INLIN
FCA8 INSFLG
FCA2 INTCNT
FC9B INTFLG
FCA0 INTVAL
03FB ISCNTC
145F ISFLIO
FC9E JIFFY
FCAD KANAMD
FCAC KANAST
F41F KBUF
0D89 KEYANY
FBF0 KEYBUF
0E3B KEYCOD
0C3C KEYINT
0468 KILBUF
0F10 KYEASY
107D KYGRAP
0F36 KYLOCK
0F46 KYSTOP
070F LDIRMV
0744 LDIRVM
16EE LEFTC
F954 LFPROG
14EB LFTQ
F3AF LINL32
F3AE LINL40
F3B0 LINLEN
FBB2 LINTTB
F94B LOHADR
F94D LOHCNT
F94A LOHDIR
F949 LOHMSK
F406 LOW
FCA4 LOWLIM
085D LPTOUT
F415 LPTPOS
0884 LPTSTT
15DF MAPXYC
F92F MAXDEL
F85F MAXFIL
F3EC MAXUPD
F958 MCLFLG
FB3B MCLLEN
FB3C MCLPTR
F956 MCLTAB
F672 MEMSIZ
F92D MINDEL
F3EF MINUPD
F3D7 MLTATR
F3D5 MLTCGP
F3D3 MLTCOL
F3D1 MLTNAM
F3D9 MLTPAT
F951 MOVCNT
FB3F MUSICF
F922 NAMBAS
FBE5 NEWKEY
4601 NEWSTT
F87C NLONLY
1398 NMI
F7B7 NOFUNS
1809 NSETCX
F417 NTMSXP
F862 NULBUF
FBDA OLDKEY
F6BE OLDLIN
FCB0 OLDSCR
F6C0 OLDTXT
F6BB ONEFLG
F6B9 ONELIN
FBD8 ONGSBF
F664 OPRTYP
1B45 OUTDO
FC9D PADX
FC9C PADY
F6E8 PARM1
F750 PARM2
F926 PATBAS
FC40 PATWRK
08DB PBDHRT
F953 PDIREC
148A PHYDIO
23BF PINLIN
FB40 PLYCNT
18CF PNTINI
088E POSIT
F7B4 PRMFLG
F6E6 PRMLEN
F74E PRMLN2
F74C PRMPRV
F6E4 PRMSTK
FD89 PROCNM
FB35 PRSCNT
F416 PRTFLG
F864 PTRFIL
F6A9 PTRFLG
0F55 PUTCHR
F3F8 PUTPNT
1492 PUTQ
23CC QINLIN
F971 QUEBAK
F959 QUETAB
FB3E QUEUEN
F3F3 QUEUES
F418 RAWPRT
F380 RDPRIM
110E RDPSG
01B6 RDSLT
7E1A RDSLTW
1449 RDVDP
07D7 RDVRM
1647 READC
F3F7 REPCNT
FC6A REQSTP
F3DF RG0SAV
F3E0 RG1SAV
F3E1 RG2SAV
F3E2 RG3SAV
F3E3 RG4SAV
F3E4 RG5SAV
F3E5 RG6SAV
F3E6 RG7SAV
16C5 RIGHTC
F857 RNDX
FAF5 RS2IQ
144C RSLREG
F955 RTPROG
FC9A RTYCNT
FCBE RUNBNF
F866 RUNFLG
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      10100011B
                    ;
                    ;  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
```

Key code table (0DA5H..0EC4H)

```
                                ;
                                ;       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
```

Key code table (0DA5H..0EC4H)

```
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
```

```
                                        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
```

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, 00BlH
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

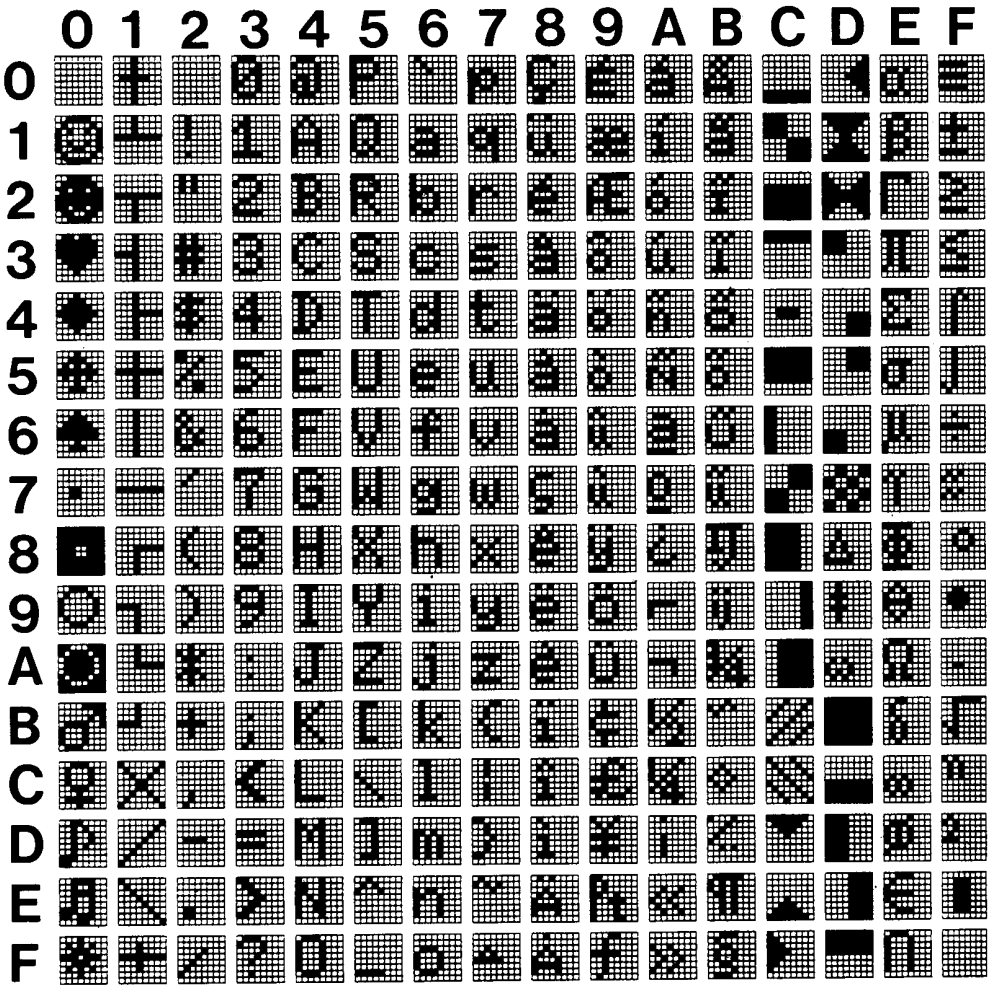
Name: CALBAS, 0159H
Function: Performs far_call (i.e., inter-slot call) into BASIC interpreter.
Entry: Address in [IX]
Returns: Who knows?
Modifies: ditto

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.

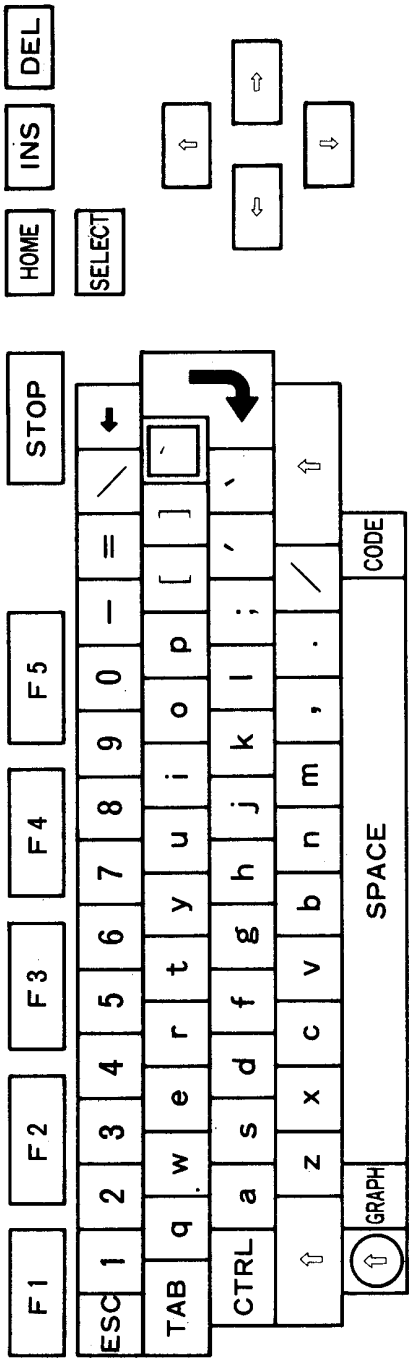
```
  ***
 *   *
 *   *
 * * *
 *   *
 *   *
  ***
```

o Decode International (USA)

INT			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	ƒ F4	√ FB
		Shift	◙ 0A		² FD	ⁿ FC			∫ F5	
	Code		δ EB	ƒ 9F	‡ D9	§ BF	¢ 9B	ÿ 98	α E0	β E1
		Shift	Δ D8	ı AD	Pt 9E	₸ BE	£ 9C	Ÿ 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		ˠ 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	~	a 61	b 62
		Shift	" 22	~ 7E	< 3C	> 3E	? 3F	´ dead key	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		ì B3	o B5	μ E6	ñ A4	ó A2	ú A3	â 83	ô 93
		Shift	I B2	Õ B4		Ñ A5		ll 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	õ 95	ê 88	è 8A	á A0	à 85
		Shift			É 90					

INTERNATIONAL MSX VERSIONS

o Layout International (USA)

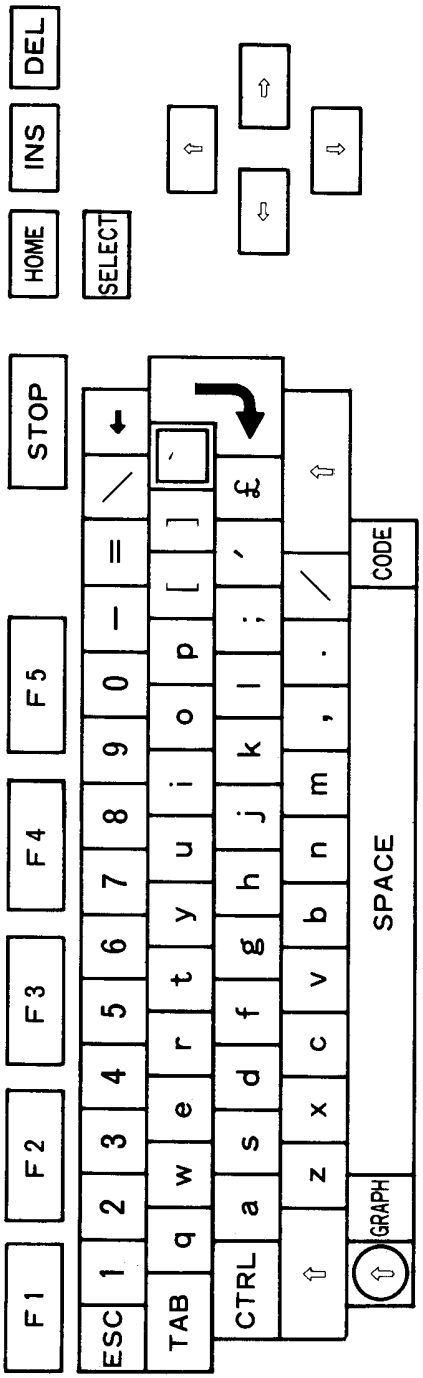


o Decode UK

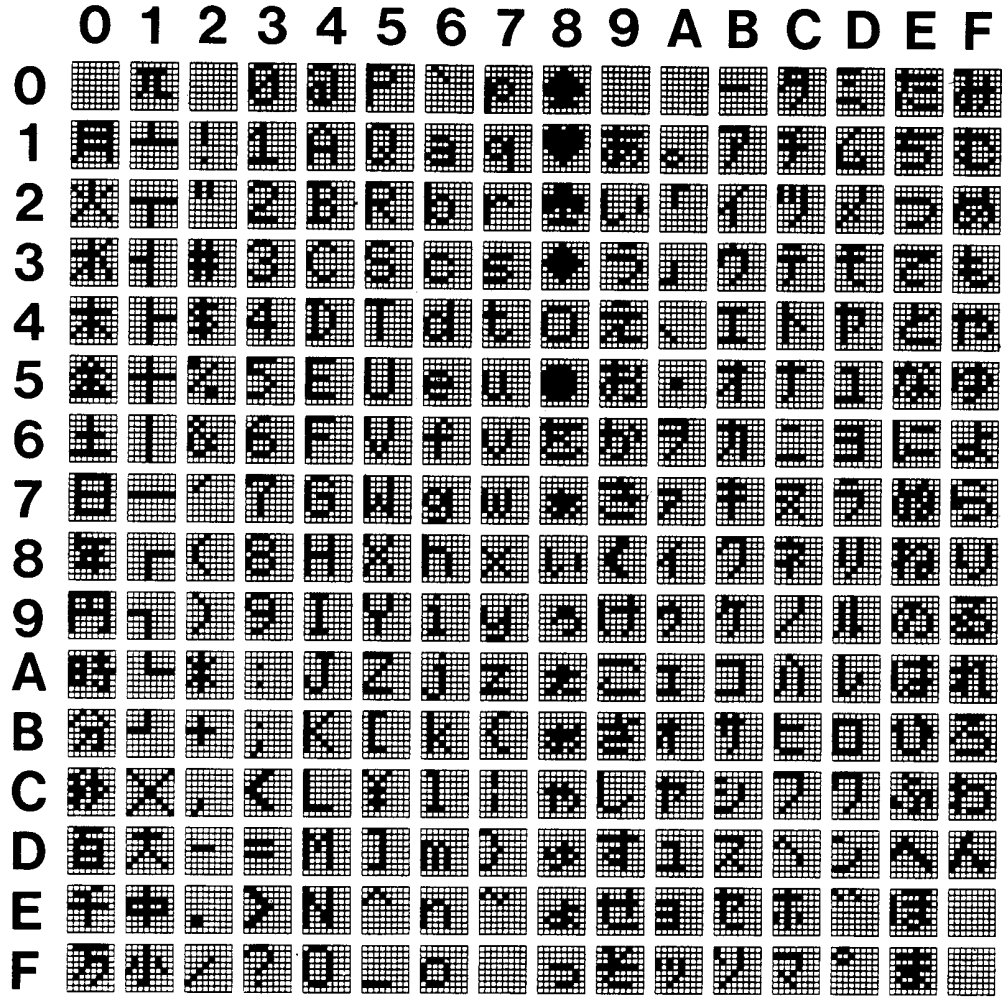
UK			0	1	2	3	4	5	6	7
0	Normal		0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37
		Shift	) 29	! 21	@ 40	# 23	$ 24	% 25	^ 5E	& 26
	Graph		○ 09	¼ AC	½ AB	¼ BA	η EF	‰ BD	ƒ F4	√ FB
		Shift	◙ 0A		² FD	" FC			∫ F5	
	Code		δ EB	ƒ 9F	‡ D9	§ BF	¢ 9B	ÿ 98	α E0	β E1
		Shift	Δ D8	¡ AD	Pt 9E	¶ BE	£ 9C	¥ 9D		
1	Normal		8 38	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		ì 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	ù 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)

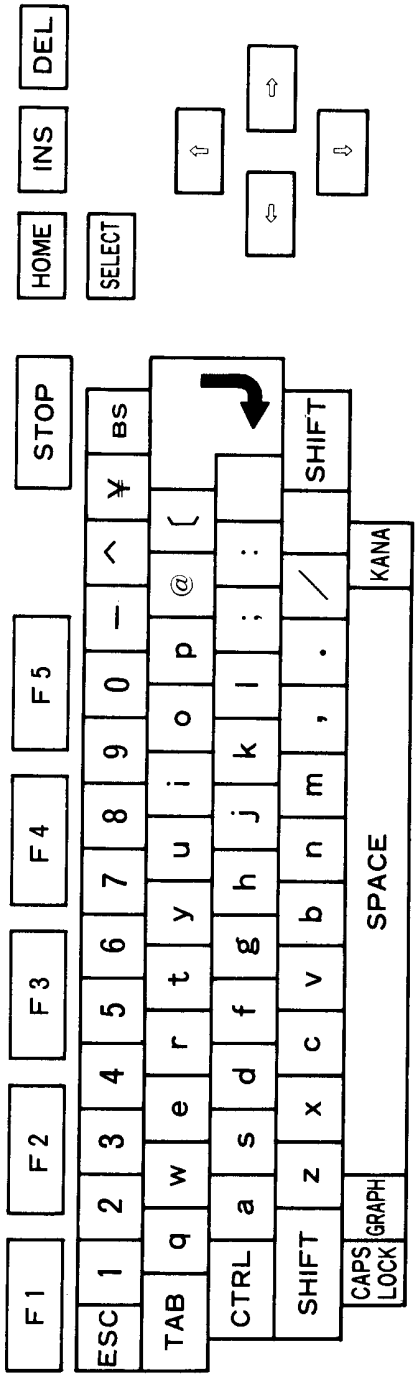


o Decode Japanese 1

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	一 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			あ 87	う 89	え 8A	お 8B	や 8C
	Caps	ヲ A6			ア A7	ウ A9	エ AA	オ AB	ヤ AC
1		ゆ 8D	よ 8E					「 A2	
	Caps	ュ AD	ョ AE					「 A2	
2			」 A3	A4	。 A1	・ A5			
	Caps		」 A3	、 A4	。 A1	・ A5			
3				い 88					
	Caps			ィ A8					
5									っ 8F
	Caps								ッ AF

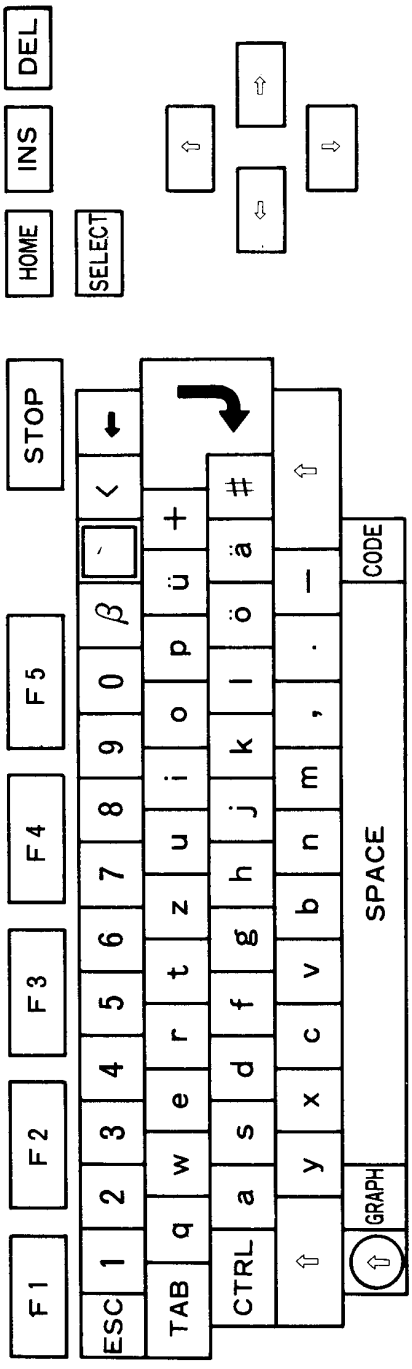
INTERNATIONAL MSX VERSIONS

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	€ 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	´ dead key	> 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	¨ dead key	≥ 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	å 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	ä 85
		Shift			É 90					¥ 9D

INTERNATIONAL MSX VERSIONS

o Layout DIN

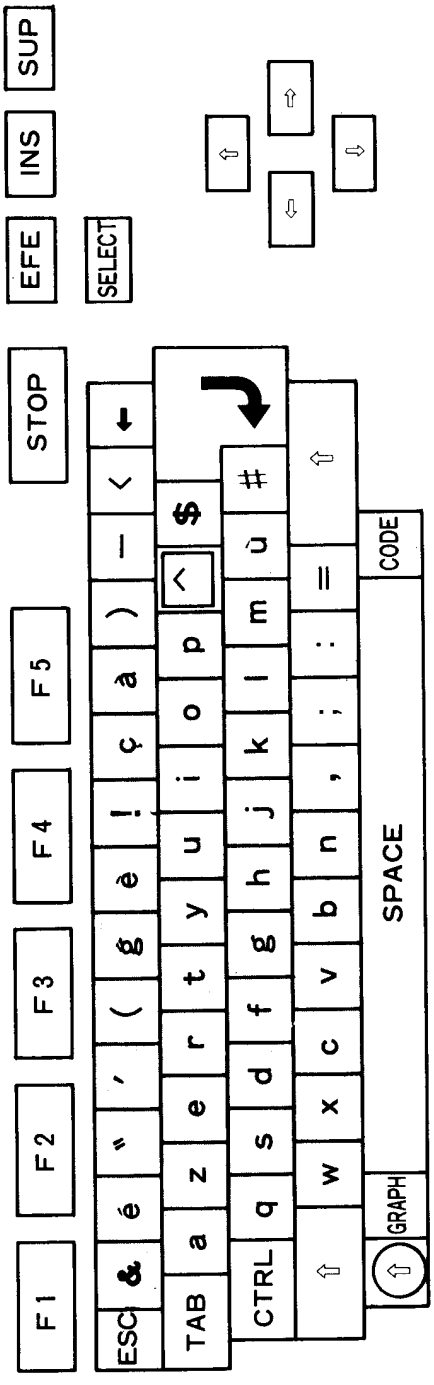


o Decode French

FR			0	1	2	3	4	5	6	7
0	Normal		â 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	ƒ F4	√ FB
		Shift	◙ 0A	| 16	² FD	ⁿ FC	≈ F7		∫ F5	
	Code		δ EB	¦ 7C	@ 40	α E0	` 60	¦ 7B	^ 5E	∈ 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	φ ED	≦ F3	^	¢ 9B	ũ B7
		Shift	Γ E2	C 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	å 86	a A6	o 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		Π 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

INTERNATIONAL MSX VERSIONS

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