```
                        ;
                        ;
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*            N E W   R O M   Vers. 4.8                  *
                        ;*               For N.E. Computer                       *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ;
                               title   Rom 4.8 for NE CP/M 2.2 with Hard-Disk
                               subttl  Copyright Studio Lg, Genova - Last rev 18/08/1984 17:30
                        ;      Programmer: Martino Stefano
                        ;      All Modified by Gallerani Paolo
                        ;
                        ;
                        ;
004B                    vers   equ    'H'     ; Version for Hard-Disk
0030                    rev    equ    48      ; Revision level
                        ;
                        ;
0000                    false  equ    0
FFFF                    true   equ    not false
                        ;
                        ;
F000                    rom    equ    0F000h          ; <--- rom start
FFFF                    necnt  equ    true            ; fdd controller by n.e.
0000                    mdcnt  equ    false           ; fdd controller by micro design
0000                    md80vid equ   false           ; video int. 80*24 by micro design
FFFF                    ne80vid equ   true            ; video int. 80*24 by n.e.
FFFF                    hard   equ    true            ; hard disk with xebec cnt
                        ;
                               page   62
```

```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*            Ram data areas equates                   *
                        ;*                                                     *
                        ;*******************************************************
                        ;
0040                    ram     equ     00040h          ; top of ram data areas
0040                    fd0otr  equ     ram             ; fdd0 old track num.
0041                    fd1otr  equ     ram+1           ; fdd1 old track num.
0042                    fd2otr  equ     ram+2           ; fdd2 old track num.
0043                    fd3otr  equ     ram+3           ; fdd3 old track num.
0044                    unit    equ     ram+4           ; fdd old unit select
0045                    hladrs  equ     ram+5           ; two byte for HL save
0047                    ix.pnt  equ     ram+7           ; routines table address
0049                    spare   equ     ram+9           ; not used
004A                    task    equ     ram+10          ; six byte for wdd cmd out
1000                    ipldma  equ     01000h          ; IPL dma address
1000                    stack   equ     ipldma          ; stack ram area for IPL
                                ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*            ASCII EQUIVALENTS                        *
                        ;*                                                     *
                        ;*******************************************************
                        ;
0007                    bell    equ     'G'-'@'         ; ring beeper
0008                    backsp  equ     'H'-'@'         ; back space char.
000C                    ffeed   equ     'L'-'@'         ; form feed char.
000D                    cr      equ     'M'-'@'         ; carriage-return char.
000A                    lf      equ     'J'-'@'         ; line-feed char.
0024                    endmsg  equ     '$'             ; end of print message
                        ;
                                page
```

```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*                      ROM                            *
                        ;*                                                     *
                        ;*******************************************************
                        ;
0000'                           aseg
                                org     100h            ; Put in standard TPA
                                ;
                                .phase  rom             ; EPROM PC
                                ;
F000    C3 F061                 jp      reset           ; jmp here to hardware reset
F003    C3 F209                 jp      cin             ; console input
F006    C3 F233                 jp      cout            ; console output
F009    C3 F202                 jp      csts            ; console status
F00C    C3 F1F8                 jp      lout            ; printer output
F00F    C3 F1F2                 jp      lsts            ; printer status
F012    C3 F56F                 jp      fdios           ; fdd I/O 128 byte
F015    C3 F572                 jp      fdiod           ; fdd I/O 256 byte
F018    C3 F752                 jp      wdini           ; wdd initialization
F01B    C3 F6B4                 jp      wdio            ; wdd I/O 256 byte
F01E    C3 F562                 jp      strout          ; print string -> DE until endmsg
F021    C3 F0CC                 jp      boot            ; load IPL and exec. it
F024    C3 F55D                 jp      printat         ; print string -> DE until endmsg at cursor -> HL
F027    C3 F344                 jp      movcurs         ; move video cursor at -> HL
F02A    C3 F4E5                 jp      initialize      ; initialize video
                                ;
F02D                    CompFlg:
F02D    30                       db     rev             ; current revision level
                                ;
                                page
```

```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*           Jump here after hardware reset            *
                        ;*                                                     *
                        ;*******************************************************
                        ;
F061                    reset:
F061    AF                      xor     a               ; clear accumulator
F062    D3 D6                   out     (fddlch),a      ; deselect any drive.
F064    31 1000                 ld      sp,stack        ; set stack pointer
F067    21 0040                 ld      hl,ram          ; top of ram data areas
F06A    06 10                   ld      b,16            ; number of byte to clear
F06C                    rst00:
F06C    77                      ld      (hl),a          ; clear ram
F06D    23                      inc     hl              ; inc. ram pointer
F06E    10 FC                   djnz    rst00           ; repeat until end of ram data areas
F070    DD 21 1500              ld      ix,ipldma+500h  ; IX = reset routines table
F074    CD F4E5                 call    initialize      ; initialize video and video table
                        ;
F077    11 F036                 ld      de,inimsg       ; D.E = initial message
F07A    CD F562                 call    strout          ; print it
                        ;
                        ;
                                if      hard            ;
                        ;
                        ;
                        ; ***********************************************
                        ; * W D B O O T                                 *
                        ; * Do Boot according if there is               *
                        ; * Hard Disk interface and reading errors      *
                        ; * else boot from Floppy                       *
                        ; ***********************************************
                        ;
                        ; Last Revision 08/05/85 17:18 by Gallerani Paolo
                        ;
                                ; Now test if there are hard disk interface
                        ;
F07D    DB B9                   in      a,(rport1)      ; load cntlr status
F07F    E6 E0                   and     11100000b       ; mask bit 5,6,7. Hard disk exist ?
F081    20 31                   jr      nz,fdbtwt       ; no, then go to fdd boot
                        ;
                        ;
F083    D3 B9                   out     (wport1),a      ; reset hard cntrl
F085    11 F123                 ld      de,wdfdmsg      ; D.E = question message
F088    CD F55A                 call    pr@12.12        ; print it @ 12,12
                        ;
                        ;
F08B                    WaitKey:
                                ; wait for push 'F' or wdd ready
F08B    CD F202                 call    csts            ; get console status
                                                        ; key pushed ?
F08E    28 19                   jr      z,CkRdy         ; no, then check for wdd ready
F090    CD F209                 call    cin             ; wait one char.
F093    CB AF                   res     5,a             ; convert up-case
F095    FE 46                   cp      'F'             ; is 'F' ?
```

```
                        ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*                 Rom data areas                      *
                        ;*                                                     *
                        ;*******************************************************
                        ;
F02E                    DPBIPL: ;
                                ; Disk Parameter Block
                                ; for fd & wd IPL boot
                                ;
F02E    00                      defb    0               ; first unit & side 0
F02F    0000                    defw    0               ; first track  = 0000
F031    01                      defb    1               ; first sector = 1
F032    1000                    defw    ipldma          ; ipl dma address
F034    00                      defb    0               ; read code
F035    01                      defb    1               ; 1 sector to read
                                                        ; for wdd read
                        ;
                        ;
                        ;
F036                    inimsg:
F036    0C 0D 0A 13             defb    ffeed,cr,lf,19,'L'
F03A    4C
F03B    20 20 4E 45             defb    '  NEW FIRMWARE Vers. ',vers
F03F    57 20 46 49
F043    52 4D 57 41
F047    52 45 20 56
F04B    65 72 73 2E
F04F    20 48
F051    20 72 65 76             defb    ' rev ',rev/10+'0','.',rev mod 10+'0'
F055    20 34 2E 38
F059    20 20 13 40             defb    '  ',19,'@',bell,cr,lf,endmsg
F05D    07 0D 0A 24
                                ;
                                page
```

```
F097  20 F2                   jr      nz,WaitKey      ; no, then ignore
                              ;
                              ; Selected boot from floppy
                              ; Wait for Hard-Disk ready
                              ;
F099  11 F15E                 ld      de,fdbtmsg      ; D.E = 'ok for .. ' message
F09C  CD F55A                 call    pr@12.12        ; print it @ 12,12
                        ;
F09F                    WtHdRdy:
F09F  CD F792                 call    wtwdrdy         ; wdd ready ?
F0A2  20 FB                   jr      nz,WtHdRdy      ; no, loop
F0A4  CD F752                 call    wdini           ; yes, then initialize wdd
F0A7  18 17                   jr      FddBot          ; and go to fdd boot
                        ;
F0A9                    CkRdy:
F0A9  CD F792                 call    wtwdrdy         ; wdd ready ?
F0AC  20 DD                   jr      nz,WaitKey      ; no, return to test console
F0AE  CD F752                 call    wdini           ; else can initialize wdd
F0B1  AF                      xor     a               ; set A=0 for wdd boot
F0B2  18 1B                   jr      WdBoot          ; do boot
                        ;
                        ; Wait aproax 3 seconds
                        ; then boot from floppy
                        ;
F0B4                    fdbtwt:
F0B4  06 04                   ld      b,4
F0B6                    timer:
F0B6  11 8000                 ld      de,8000h        ; set soft timer
F0B9                    timer1:
F0B9  1B                      dec     de              ; timer 1 down
F0BA  7B                      ld      a,e             ;
F0BB  B2                      or      d               ;
F0BC  20 FB                   jr      nz,timer1       ;
F0BE  10 F6                   djnz    timer           ; timer 2 down
                        ;
                        ; Restore Unit 1 (option)
                        ;
F0C0                    FddBot:
F0C0  3E 02                   ld      a,2             ;
F0C2  D3 D6                   out     (fddlch),a      ; set unit 1
F0C4  AF                      xor     a               ; restore cmd without verify
F0C5  D3 D0                   out     (fddcmd),a      ; send out to 1771
F0C7  CD F641                 call    waitfd          ; wait until end command
F0CA  18 47                   jr      FdBoot          ; set boot from FDD
                              ;
                              page
```

```
                        ;
                        ;
                        ; ***********************************************
                        ; *            Boot Entry point                 *
                        ; *     If A=0 then boot from Hard else Floppy  *
                        ; ***********************************************
                        ;
F0CC                    boot:
F0CC    B7                      or      a               ; Is A = 0 ?
F0CD    20 44                   jr      nz,FdBoot       ; no, then go to fdd boot
                        ;
                        ;
                        ; ***********************************************
                        ; *            Boot from Hard                   *
                        ; ***********************************************
                        ;
F0CF                    WdBoot:
F0CF    F5                      push    af              ; save flag
F0D0    21 F02E                 ld      hl,DPBIPL       ; H.L -> IPL boot para adrs
F0D3    CD F6B4                 call    wdio            ; read one sector
                                ;
                                ; Check for Errors
                                ; then test for IPL ok
                                ;
F0D6                    ipltest:
F0D6    B7                      or      a               ; read error ?
F0D7    20 21                   jr      nz,ioerr        ; yes, then goto I/O error
                                ; test for IPL
F0D9    21 1006                 ld      hl,ipldma+6     ; H.L = IPL message
F0DC    7E                      ld      a,(hl)          ; get 1' char.
F0DD    FE 49                   cp      'I'             ; is 'I' ?
F0DF    20 14                   jr      nz,noipl        ; no, then go to no IPL
F0E1    23                      inc     hl              ; H.L point next char.
F0E2    7E                      ld      a,(hl)          ; get 2' char.
F0E3    FE 50                   cp      'P'             ; is 'P' ?
F0E5    20 0E                   jr      nz,noipl        ; no, then go to no IPL
F0E7    23                      inc     hl              ; H.L point next char.
F0E8    7E                      ld      a,(hl)          ; get 3' char.
F0E9    FE 4C                   cp      'L'             ; is 'L' ?
F0EB    20 08                   jr      nz,noipl        ; yes, then goto IPL ok
                                ;
                                ; IPL has been loaded
                                ; compute entry point
                                ;
F0ED                    iplok:
F0ED    F1                      pop     af              ; IPL Flag
F0EE    B7                      or      a               ;
F0EF    CA 1000                 jp      z,ipldma        ; if A = 0 then wdd bios boot
F0F2    C3 1003                 jp      ipldma+3        ;          else fdd bios boot
                                ;
                                page
```

```
                                ;
                                ;
                                ; ***** No IPL Error *****
F0F5                    noipl:
F0F5    11 F1A9                 ld      de,noiplmsg     ; D.E = no IPL message
F0F8    18 03                   jr      io.01           ; print it and retry
                                ;
                                ; Print I/O error msg then boot from floppy
                                ; ***** I/O Error *****
F0FA                    ioerr:
F0FA    11 F19A                 ld      de,ioermsg      ; D.E = I/O error message
F0FD                    io.01:
F0FD    CD F562                 call    strout          ; print it
F100    F1                      pop     af              ; take off boot flag
F101    11 F1BD                 ld      de,setnew       ; D.E = set new disk message
F104    CD F562                 call    strout          ; print message
F107                    waitcr:
F107    CD F209                 call    cin             ; wait one char.
F10A    FE 0D                   cp      cr              ; is return ?
F10C    20 F9                   jr      nz,waitcr       ; no, wait cr
F10E    0E 07                   ld      c,bell          ;
F110    CD F233                 call    cout            ; ring beeper
                        ;
                        ;
                        ; ***********************************************
                        ; *            Boot from floppy 0               *
                        ; ***********************************************
                        ; Recalibrate Unit 0, then load IPL from it
                        ;
F113                    FdBoot:
F113    3E 01                   ld      a,1             ;
F115    D3 D6                   out     (fddlch),a      ; set unit 0
F117    F5                      push    af              ; use for Boot from floppy flag
F118    CD F619                 call    fdsek0          ; move fdd head to track 0
F11B    21 F02E                 ld      hl,DPBIPL       ; H.L -> IPL boot para adrs.
F11E    CD F572                 call    fdiod           ; read one sector
F121    18 B3                   jr      ipltest         ; check for errors
                                ;
                                page
```

```
                                ;
                                ;*******************************************************
                                ;*                                                     *
                                ;*            D a t a   a r e a                        *
                                ;*                                                     *
                                ;*******************************************************
                                ;
F123                            wdfdmsg:
F123    50 75 73 68                     defb    'Push [F] for boot from floppy disk or wait hard disk ready',endmsg
F127    20 5B 46 5D
F12B    20 66 6F 72
F12F    20 62 6F 6F
F133    74 20 66 72
F137    6F 6D 20 66
F13B    6C 6F 70 70
F13F    79 20 64 69
F143    73 6B 20 6F
F147    72 20 77 61
F14B    69 74 20 68
F14F    61 72 64 20
F153    64 69 73 6B
F157    20 72 65 61
F15B    64 79 24
                                ;
F15E                            fdbtmsg:
F15E    07 4F 6B 20                     defb    bell,'Ok for boot from floppy disk. '
F162    66 6F 72 20
F166    62 6F 6F 74
F16A    20 66 72 6F
F16E    6D 20 66 6C
F172    6F 70 70 79
F176    20 64 69 73
F17A    6B 2E 20
F17D    57 61 69 74                     defb    'Wait until hard disk ready. ',endmsg
F181    20 75 6E 74
F185    69 6C 20 68
F189    61 72 64 20
F18D    64 69 73 6B
F191    20 72 65 61
F195    64 79 2E 20
F199    24
                                ;
                                ;
F19A                            ioermsg:
F19A    0D 0A 13 48                     defb    cr,lf,19,'H','DISK ERROR',endmsg
F19E    44 49 53 4B
F1A2    20 45 52 52
F1A6    4F 52 24
                                ;
F1A9                            noiplmsg:
F1A9    0D 0A 13 48                     defb    cr,lf,19,'H','No IPL on disk.',endmsg
F1AD    4E 6F 20 49
F1B1    50 4C 20 6F
F1B5    6E 20 64 69
F1B9    73 6B 2E 24
                                ;
```

```
F1BD                            setnew:
F1BD    0D 0A 53 65                     defb    cr,lf,'Set system diskette in disk A,'
F1C1    74 20 73 79
F1C5    73 74 65 6D
F1C9    20 64 69 73
F1CD    6B 65 74 74
F1D1    65 20 69 6E
F1D5    20 64 69 73
F1D9    6B 20 41 2C
F1DD    0D 0A 61 6E                     defb    cr,lf,'and push return',19,'@',bell,endmsg
F1E1    64 20 70 75
F1E5    73 68 20 72
F1E9    65 74 75 72
F1ED    6E 13 40 07
F1F1    24
                                ;
                                        endif
                                        ;
                                        page
```

```
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*                 Console routines                      *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                               if      ne80vid
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*          Video, Keyboard, Printer routines            *
                        ;*             for video interface 80*24                 *
                        ;*               by Nuova Elettronica                    *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ;
                        ; Copyright (C) 1983, 1984 by Studio Lg, Genova - Italy
                        ; Author: Martino Stefano
                        ; Third Modify by Gallerani Paolo
                        ; Last Revision 06/08/84 21:00
                        ;
                        ; Compatility Version 4.1-1
                        ;
                               if      rev lt 41
                               .printx .*** Warning:Incompatible Video Board Driver ***.
                               endif
                        ;
                        ;
0000                    strvid  equ     0               ; start video cursor
077F                    endvid  equ     77fh            ; end video cursor
                        ;
                        ;       ***** PIO command code  ****
                        ;
000F                    mode0   equ     00fh            ; mode 0 command code (output)
004F                    mode1   equ     04fh            ; mode 1 command code (input)
008F                    mode2   equ     08fh            ; mode 2 command code (bidiretional)
00CF                    mode3   equ     0cfh            ; mode 3 command code (bit control)
                        ;
                        ;
0080                    ioadd   equ     080h            ; i/o port top address
                        ;
                               ; PIO 0 data and control port address
0080                    data0a  equ     ioadd+0         ; read/write ram 0
0081                    data0b  equ     ioadd+1         ; write printer
0082                    cont0a  equ     ioadd+2
0083                    cont0b  equ     ioadd+3
                        ;
                               ; PIO 1 data and control port address
0084                    data1a  equ     ioadd+4         ; read/write ram 1
0085                    data1b  equ     ioadd+5         ; read keyboard
0086                    cont1a  equ     ioadd+6
0087                    cont1b  equ     ioadd+7
                        ;
                               ; PIO 2 data and control port address
```

```
008B                    data2a  equ     ioadd+8         ; read/write ram 2
008B                    data2b  equ     ioadd+9         ; board flag
008A                    cont2a  equ     ioadd+10
008B                    cont2b  equ     ioadd+11
                        ;
                        ;
008C                    addreg  equ     ioadd+12                ; 6545 address register port
008D                    datreg  equ     ioadd+13                ; 6545 register port
                        ;
008E                    vidatr  equ     ioadd+14                ; video ram attibute
                        ;
008F                    beep    equ     ioadd+15                ; beeper port address
                        ;
                        ;
                        ;       ***** board flag  (data2b)  *****
                        ;
                        ;       bit     direction           function
                        ;
                        ;        0       <---             Printer Busy
                        ;        1       --->           1 = 40  0 = 80 char/line
                        ;        2       --->                Spare
                        ;        3       --->                Spare
                        ;        4       <---                Spare
                        ;        5       <---                Spare
                        ;        6       <---                Spare
                        ;        7       <---                Spare
                        ;
                        ;
                        ;       ***** Video Attributes  (vidatr)  *****
                        ;
                        ;       bit     direction           function
                        ;
                        ;        0       <--->              Blinking
                        ;        1       <--->              Reverse
                        ;        2       <--->             Under-score
                        ;        3       <--->              High-light
                        ;        4       <--->              Graphics
                        ;        5        ---               Not used
                        ;        6        ---               Not used
                        ;        7        ---               Not used
                        ;
                        ;
                                page
```

```
                        ;
                        ;
                        ;*********************************************************
                        ;* L s t S                                               *
                        ;*            Return printer status                      *
                        ;*********************************************************
                        ;
F1F2                    lsts:
F1F2    DB 89                   in      a,(data2b)      ; read board flag
F1F4    E6 01                   and     00000001b       ; mask printer busy bit
F1F6    3D                      dec     a               ; 0 -> ff, 1 -> 00
F1F7    C9                      ret                     ;   0 printer not ready
                                                        ; 255 printer ready
                        ;
                        ;
                        ;*********************************************************
                        ;* L O u t                                               *
                        ;*            Print char on printer                      *
                        ;*********************************************************
                        ;
F1F8                    lout:
F1F8    DB 89                   in      a,(data2b)      ; wait printer ready
F1FA    CB 47                   bit     0,a             ;
F1FC    20 FA                   jr      nz,lout         ;
F1FE    79                      ld      a,c             ; load char. to print
F1FF    D3 81                   out     (data0b),a      ; send out to printer
F201    C9                      ret                     ; and ret
                        ;
                        ;
                        ;*********************************************************
                        ;* C S t s                                               *
                        ;*            Return console status                      *
                        ;*********************************************************
                        ;
F202                    csts:
F202    DB 85                   in      a,(data1b)      ; read console
F204    E6 80                   and     10000000b       ; only strobe
F206    07                      rlca                    ; to bit 1
F207    3D                      dec     a               ; 1 -> 0, 0 -> ff
F208    C9                      ret                     ; 0 if no key pushed
                                                        ; 255 if key pushed
                        ;
                        ;
                        ;*********************************************************
                        ;* C I n                                                 *
                        ;*            Wait a Key from console                    *
                        ;*********************************************************
                        ;
F209                    cin:
F209    DD 2A 0047              ld      ix,(ix.pnt)     ; IX = routines table
F20D    DB 85                   in      a,(data1b)      ; wait pushed key
F20F    CB 7F                   bit     7,a             ;
F211    20 F6                   jr      nz,cin          ;
F213    F5                      push    af              ; save key
F214                    cinp00:
F214    DB 85                   in      a,(data1b)      ; wait depress key
```

```
F216  CB 7F                   bit   7,a           ;
F218  28 FA                   jr    z,cinp00      ;
F21A  F1                      pop   af            ; restore pushed key
F21B  EE 7F                   xor   01111111b     ; complements
F21D  DD CB 04 6E             bit   5,(ix+004)    ; key beep bit on ?
F221  20 02                   jr    nz,cinp11     ; no, then count
F223  D3 8F                   out   (beep),a      ; yes, then ring bell
F225                    cinp11:
F225  DD CB 04 76             bit   6,(ix+004)    ; alpha lock bit on ?
F229  C0                      ret   nz            ; no, then return
F22A  FE 61                   cp    'a'           ; yes, then convert upper case
F22C  D8                      ret   c             ; ret if A < 'a'
F22D  FE 7B                   cp    'z'+1         ;
F22F  D0                      ret   nc            ; ret if A > 'z'
F230  CB AF                   res   5,a           ; convert up-case
F232  C9                      ret                 ; and ret
                        ;
                        ;
                        ;*******************************************************
                        ;* C O u t                                             *
                        ;*            Print char on console                    *
                        ;*******************************************************
                        ;
F233                    cout:
F233  DD 2A 0047              ld    ix,(ix.pnt)   ; IX = routines table
F237  DD 6E 00                ld    l,(ix+000)    ; load video pointer (L)
F23A  DD 66 01                ld    h,(ix+001)    ; load video pointer (H)
F23D  DD 7E 05                ld    a,(ix+005)    ; load prefix location
F240  B7                      or    a             ; any prefix are set ?
F241  20 20                   jr    nz,pfxset     ; yes, then goto prefix set
F243  79                      ld    a,c           ; load char. to print
F244  FE 20                   cp    ' '           ; if char. < space
F246  DA F364                 jp    c,cntchar     ; then go to control caracter
F249  CD F3F2                 call  charout       ; print char. at HL
F24C  23                      inc   hl            ; increase video pointer
F24D                    cout11:
F24D  CD F418                 call  vidcompare    ; end of screen ?
F250  38 07                   jr    c,cout00      ; no, then count
F252  11 FFB0                 ld    de,0ffb0h     ;
F255  19                      add   hl,de         ;
F256  CD F460                 call  updtdstart    ; update display start (scrolling)
F259                    cout00:
F259  DD 75 00                ld    (ix+000),l    ; save video pointer
F25C  DD 74 01                ld    (ix+001),h    ;
F25F  CD F43B                 call  updtvid       ; update video pointer
F262  C9                      ret                 ; and ret
                        ;
                        ;
F263                    pfxset:
                              ; any prefix are set
F263  FD 21 F278              ld    iy,pfxtab-2   ; IY -> prefix table
F267  87                      add   a,a           ; A prefix code * 2
F268  5F                      ld    e,a           ;
F269  16 00                   ld    d,0           ; DE = pfx * 2
F26B  FD 19                   add   iy,de         ; IY -> pfx routine address
F26D  11 F35F                 ld    de,clrpfx     ; DE = clear prefix address
F270  D5                      push  de            ; store in stack
F271  FD 5E 00                ld    e,(iy+000)    ;
```

```
F274  FD 56 01                ld      d,(iy+001)      ; DE = pfx routine address
F277  D5                      push    de              ; store in stack
F278  79                      ld      a,c             ; A = char. to print
F279  C9                      ret                     ; and go to pfx routine
                        ;
                        ;
F27A                    pfxtab:
F27A  F28E                    defw    pfx01           ; video attribute prefix
F27C  F421                    defw    pfx02           ; cursor off/on prefix
F27E  F29F                    defw    pfx03           ; scroll/no scroll
F280  F2AD                    defw    pfx04           ; keyboard normal/alpha look prefix
F282  F2C9                    defw    pfx05           ; escape prefix
F284  F2DD                    defw    pfx06           ; ESC '+' prefix
F2B6  F317                    defw    pfx07           ; ESC '=' prefix
F288  F2F5                    defw    pfx08           ; ESC '+' Yco prefix
F28A  F324                    defw    pfx09           ; ESC '=' Yco prefix
F28C  F2BB                    defw    pfx10           ; keyboard mute/beep
                        ;
                        ;
                        ;
F28E                    pfx01:
                              ; set video attribute
F28E  D6 40                   sub     '@'             ; subcrtact offset
F290  FE 10                   cp      16              ; max number attribute
F292  D0                      ret     nc              ; ret if A > 15
F293  2F                      cpl                     ; complements
F294  47                      ld      b,a             ; save video attribute on B
F295  DD 7E 04                ld      a,(ix+004)      ; load old video attribute
F298  F6 1F                   or      00011111b       ; clear old attribute
F29A  A0                      and     b               ; mask new attribute with scroll and keyb. para
F29B  DD 77 04                ld      (ix+004),a      ; set new video attribute
F29E  C9                      ret                     ; and return
                        ;
                        ;
                        ;
F29F                    pfx03:
                              ; set scroll/no scroll
F29F  CB 47                   bit     0,a             ; is pair
F2A1  20 05                   jr      nz,pfx033       ; no, then no scrolling
F2A3  DD CB 04 FE             set     7,(ix+004)      ; set scrolling flag
F2A7  C9                      ret                     ; and ret
F2A8                    pfx033:
F2A8  DD CB 04 BE             res     7,(ix+004)      ; reset scrolling flag
F2AC  C9                      ret                     ; and ret
                        ;
                        ;
F2AD                    pfx04:
                              ; set keyboard normal/alpha lock
F2AD  CB 47                   bit     0,a             ; is pair
F2AF  20 05                   jr      nz,pfx044       ; no, then alpha lock
F2B1  DD CB 04 F6             set     6,(ix+004)      ; set normal keyb.
F2B5  C9                      ret                     ; and ret
F2B6                    pfx044:
F2B6  DD CB 04 B6             res     6,(ix+004)      ; set alpha lock keyb.
F2BA  C9                      ret                     ; and ret
                        ;
                        ;
F2BB                    pfx10:
```

```
                                ; set keyboard mute/beep
F2BB    CB 47                   bit     0,a             ; is pair
F2BD    20 05                   jr      nz,pfx101       ; no, then beep key
F2BF    DD CB 04 EE             set     5,(ix+004)      ; set mute keyb.
F2C3    C9                      ret                     ; and ret
F2C4                    pfx101:
F2C4    DD CB 04 AE             res     5,(ix+004)      ; set beep keyb.
F2C8    C9                      ret                     ; and ret
                        ;
                        ;
F2C9                    pfx05:
                                ; escape prefix (move cursor and graphics)
F2C9    D1                      pop     de              ; take off clear prefix address
F2CA    FE 2B                   cp      '+'             ; is '+' ?
F2CC    20 05                   jr      nz,pfx055       ; no, then retry
                        ;
                                ; ESC '+'
F2CE    DD 36 05 06             ld      (ix+005),6      ; set ESC '+' prefix
F2D2    C9                      ret                     ; and return
                        ;
F2D3                    pfx055:
F2D3    FE 3D                   cp      '='             ; is '=' ?
F2D5    C2 F35F                 jp      nz,clrpfx       ; * no, then retry
                        ;
                                ; ESC '='
F2D8    DD 36 05 07             ld      (ix+005),7      ; set ESC '=' prefix
F2DC    C9                      ret                     ; and return
                        ;
                        ;pfx056:
                        ; ESC 'graphics comands'
                        ; da implementare
                        ;
                        ;pfx05end:
                        ;       push    de              ; set clear prefix address
                        ;       ret                     ; go to it
                        ;
                        ;
F2DD                    pfx06:
                                ; compute Yco in ESC '+'
F2DD    CD F307                 call    relcompute      ; subct. offset and return X,Y curs. pos.
F2E0    20 03                   jr      nz,pfx066       ; BIT 7 of A = 1 then cursor <--
F2E2    84                      add     a,h             ; BIT ' of A = 0 then cursor -->
F2E3    18 03                   jr      pfx067          ;
F2E5                    pfx066:
F2E5    94                      sub     h               ; compute Yco
F2E6    ED 44                   neg                     ;
F2E8                    pfx067:
F2E8    06 18                   ld      b,24            ; max Yco
F2EA    CD F359                 call    xy1compare      ; compare A with max Yco
F2ED    DD 77 06                ld      (ix+006),a      ; set new Yco
F2F0    3E 08                   ld      a,8             ; set ESC '+' Yco prefix
F2F2    C3 F3C9                 jp      setpfx          ; restore clear prefix address
                        ;
F2F5                    pfx08:
                                ; compute Xco in ESC '+' and move cursor
F2F5    CD F307                 call    relcompute      ; subct. offset and return X,Y curs. pos.
F2F8    20 03                   jr      nz,pfx088       ; BIT 7 of A = 1 then cursor up
F2FA    85                      add     a,l             ; BIT ' of A = 0 then cursor down
```

```
F2FB  18 03                     jr      pfx089          ;
F2FD                    pfx088:
F2FD  95                        sub     1               ; compute Xco
F2FE  ED 44                     neg                     ;
F300                    pfx089:
F300  06 50                     ld      b,80            ; max Xco
F302  CD F359                   call    xylcompare      ; compare A with max Xco
F305  18 22                     jr      pfx099          ; compute cursor address and set it
                        ;
F307                    relcompute:
F307  D6 20                     sub     ' '             ; subctract offset
F309  F5                        push    af              ; save relative Xco or Yco
F30A  01 0050                   ld      bc,80           ; BC = char/row
F30D  CD F4DC                   call    divide          ; compute actual Xco and Yco
F310  67                        ld      h,a             ; H = Yco, L = Xco
F311  F1                        pop     af              ; restore relative X/Y
F312  CB 7F                     bit     7,a             ; set flag for plus or minus
F314  CB BF                     res     7,a             ;
F316  C9                        ret                     ; and ret
                        ;
                        ;
F317                    pfx07:
                                ; compute Yco in ESC '='
F317  06 18                     ld      b,24            ; max Yco
F319  CD F357                   call    xycompare       ; compare with max Yco
F31C  DD 77 06                  ld      (ix+006),a      ; set Yco
F31F  3E 09                     ld      a,9             ; set ESC '=' Yco prefix
F321  C3 F3C9                   jp      setpfx          ; restore clear prefix address
                        ;
F324                    pfx09:
                                ; compute Xco in ESC '=' and move cursor
F324  06 50                     ld      b,80            ; max Xco
F326  CD F357                   call    xycompare       ; compare with max Xco
F329                    pfx099:
F329  DD 6E 06                  ld      l,(ix+006)      ;
F32C                    pfx09a:
F32C  26 00                     ld      h,0             ; HL = Yco
                        ;
                                rept    4
                                add     hl,hl           ; multiplay Yco per 16
                                endm
F32E  29                +       add     hl,hl           ; multiplay Yco per 16
F32F  29                +       add     hl,hl           ; multiplay Yco per 16
F330  29                +       add     hl,hl           ; multiplay Yco per 16
F331  29                +       add     hl,hl           ; multiplay Yco per 16
                        ;
F332  E5                        push    hl              ; save Yco * 16
                        ;
                                rept    2
                                add     hl,hl           ; multiplay Yco * 64
                                endm
F333  29                +       add     hl,hl           ; multiplay Yco * 64
F334  29                +       add     hl,hl           ; multiplay Yco * 64
                        ;
F335  D1                        pop     de              ; DE = Yco * 16
F336  19                        add     hl,de           ; HL = Yco * 80
F337  5F                        ld      e,a             ;
F338  16 00                     ld      d,0             ; DE = Xco
```

```
F33A   19                      add     hl,de           ; HL = Yco*80 + Xco
                               ;
F33B                           setnwcpos:
                                       ; set new cursor position
F33B   DD 75 07                ld      (ix+007),l      ; set last cursor position
F33E   DD 74 08                ld      (ix+008),h      ;
F341   C3 F259                 jp      cout00          ; set new video pointer and ret
                               ;
                               ;
F344                           movcurs:
                                       ; move cursor at H = Yco, L = Xco
F344   DD 2A 0047              ld      ix,(ix.pnt)     ; IX = routines table
F348   7C                      ld      a,h             ; load Yco
F349   06 18                   ld      b,24            ; max Yco
F34B   CD F359                 call    xylcompare      ; compare with max Yco
F34E   7D                      ld      a,l             ; load Xco
F34F   06 50                   ld      b,80            ; max Xco
F351   CD F359                 call    xylcompare      ; compare with max Xco
F354   6C                      ld      l,h             ; L = Yco, A = Xco
F355   18 D5                   jr      pfx09a          ; compute cursor address, move it and ret
                               ;
                               ;
F357                           xycompare:
                                       ; A = Xco or Yco + offset, B = max Xco or Yco
F357   D6 20                   sub     ' '             ; subcract offset
F359                           xylcompare:
F359   B8                      cp      b               ; compare with max Xco or Yco
F35A   D8                      ret     c               ; ret if A < Xco or Yco
F35B   D3 8F                   out     (beep),a        ; ring beeper
F35D   D1                      pop     de              ; restore normal return address
F35E   C9                      ret                     ; and return to clear prefix
                               ;
                               ;
F35F                           clrpfx:
                                       ; clear prefix
F35F   DD 36 05 00             ld      (ix+005),0      ; clear prefix
F363   C9                      ret                     ; and ret
                               ;
                               ;
F364                           cntchar:
                                       ; control character
F364   FE 04                   cp      4               ;
F366   D8                      ret     c               ; ret if char. < 4
F367   FE 1C                   cp      28              ;
F369   D0                      ret     nc              ; ret if char. > 27
F36A   11 F259                 ld      de,cout00       ; DE = return address
F36D   D5                      push    de              ; store in stack
F36E   FD 21 F37B              ld      iy,chr$tab - 8  ; IY -> chr$ table (char 4 - 27)
F372   87                      add     a,a             ; A = A*2
F373   5F                      ld      e,a             ;
F374   16 00                   ld      d,0             ;
F376   FD 19                   add     iy,de           ; IY -> chr$ routine address
F378   FD 5E 00                ld      e,(iy+000)      ; load routine address (L)
F37B   FD 56 01                ld      d,(iy+001)      ; load routine address (H)
F37E   D5                      push    de              ; put it on stack
F37F   11 0050                 ld      de,80           ; DE = char for line for utility
F382   C9                      ret                     ; go to chr$(xx) routine
                               ;
```

```
F383                    chr$tab:
                                ; control character table
F383    F3D2                    defw    chr$04          ; move cursor at last addressement
F385    F493                    defw    chr$05          ; clear to end of line
F387    F4A0                    defw    chr$06          ; clear to end of screen
F389    F3CE                    defw    chr$07          ; beeper
F38B    F3E9                    defw    chr$08          ; cursor left (back space)
F38D    F3CC                    defw    nochr$          ; not implemented
F38F    F3D9                    defw    chr$10          ; cursor down (line feed)
F391    F3DE                    defw    chr$11          ; cursor beginning of screen (home)
F393    F49D                    defw    chr$12          ; clear screen
F395    F4CC                    defw    chr$13          ; cursor beginning of line (carriage return)
F397    F3E2                    defw    chr$14          ; cursor right
F399    F3EC                    defw    chr$15          ; cursor up
F39B    F3CC                    defw    nochr$          ; not implemented
F39D    F3CC                    defw    nochr$          ; not implemented
F39F    F3CC                    defw    nochr$          ; not implemented
F3A1    F3B3                    defw    chr$19          ; video attribute prefix
F3A3    F3B7                    defw    chr$20          ; cursor off/on prefix
F3A5    F3BB                    defw    chr$21          ; scroll yes/no prefix
F3A7    F3BF                    defw    chr$22          ; keyboard normal/alpha look prefix
F3A9    F3C3                    defw    chr$23          ; keyboard mute/beep
F3AB    F3CC                    defw    nochr$          ; not implemented
F3AD    F3CC                    defw    nochr$          ; not implemented
F3AF    F3CC                    defw    nochr$          ; not implemented
F3B1    F3C7                    defw    chr$27          ; cursor and graphics prefix
                        ;
                        ;
F3B3                    chr$19:
                                ; video attribute prefix
F3B3    3E 01                   ld      a,001           ; set pfx code
F3B5    18 12                   jr      setpfx          ; and ret
                        ;
                        ;
F3B7                    chr$20:
                                ; cursor off/on prefix
F3B7    3E 02                   ld      a,002           ; set pfx code
F3B9    18 0E                   jr      setpfx          ; and ret
                        ;
                        ;
F3BB                    chr$21:
                                ; scroll/no scroll prefix
F3BB    3E 03                   ld      a,003           ; set pfx code
F3BD    18 0A                   jr      setpfx          ; and ret
                        ;
                        ;
F3BF                    chr$22:
                                ; keyboard normal/alpha look prefix
F3BF    3E 04                   ld      a,004           ; set pfx code
F3C1    18 06                   jr      setpfx          ; and ret
                        ;
                        ;
F3C3                    chr$23:
                                ; keyboard mute/beep prefix
F3C3    3E 0A                   ld      a,010           ; set pfx code
F3C5    18 02                   jr      setpfx          ; and ret
                        ;
                        ;
```

```
F3C7                    chr$27:
                                ; escape prefix
F3C7    3E 05                   ld      a,005           ; set pfx code
                                                        ; and ret
                        ;
F3C9    DD 77 05        setpfx: ld      (ix+005),a      ; set prefix code
                        ;
F3CC                    nochr$:
F3CC    D1                      pop     de              ; restore cout00 return address
F3CD    C9                      ret                     ; and return to caller
                        ;
                        ;
F3CE                    chr$07:
                                ; beeper
F3CE    D3 8F                   out     (beep),a        ; ring beeper
F3D0    18 FA                   jr      nochr$          ; and ret
                        ;
                        ;
F3D2                    chr$04:
                                ; move cursor at last addressement
F3D2    DD 6E 07                ld      l,(ix+007)      ; load last cursor position
F3D5    DD 66 08                ld      h,(ix+008)      ;
F3D8    C9                      ret                     ; set it and return
                        ;
                        ;
F3D9                    chr$10:
                                                        ; DE = char. for line
F3D9    19                      add     hl,de           ; HL point to next line
F3DA    D1                      pop     de              ; restore return address
F3DB    C3 F24D                 jp      cout11          ; and go to eventually scrolling
                        ;
F3DE                    chr$11:
                                ; move cursor at start of screen
F3DE    21 0000                 ld      hl,strvid       ; HL = start video
F3E1    C9                      ret                     ; return
                        ;
                        ;
F3E2                    chr$14:
                                ; cursor right
F3E2    23                      inc     hl              ; increase video pointer
F3E3    CD F418                 call    vidcompare      ; end of screen + 1 ?
F3E6    D8                      ret     c               ; no, then ret
F3E7    2B                      dec     hl              ; dec. video pointer
F3E8    C9                      ret
                        ;
F3E9                    chr$08:
                                ; back space
F3E9    11 0001                 ld      de,1            ; DE = one char. to subctract
                                                        ; go to subctact HL with DE
                        ;
F3EC                    chr$15:
                                ; cursor up
                                                        ; DE = character for line
                        ;
F3EC                    subpointer:
F3EC    AF                      xor     a               ; clear carry
F3ED    ED 52                   sbc     hl,de           ; HL = video pointer - one line
F3EF    D0                      ret     nc              ; ret if HL >= start video
```

```
F3F0  19                  add   hl,de          ; restore originally video pointer
F3F1  C9                  ret                  ; and ret
                    ;
                    ;
                    ;     ***** various routines entry point *****
                    ;
                    ;
F3F2                charout:
                          ; sen out character contained in A register at HL cursor position
F3F2  D5                  push  de             ; save register
F3F3  E5                  push  hl             ;
F3F4  F5                  push  af             ;
F3F5  CD F43B             call  updtvid        ; update video pointer
F3F8  F1                  pop   af             ; restore char. to print
F3F9  CD F3FF             call  sendout        ; send out to RAM 0
F3FC  E1                  pop   hl             ; restore register
F3FD  D1                  pop   de             ;
F3FE  C9                  ret                  ; and ret
                    ;
                    ;
F3FF                sendout:
                          ; send out character contained in reg. A to video RAM 0
F3FF  F5                  push  af             ; save character
F400                sendo0:
F400  DB 8C               in    a,(addreg)     ; load 6545 status
F402  CB 7F               bit   7,a            ; test sync bit
F404  28 FA               jr    z,sendo0       ; wait if busy
F406  E3                  ex    (sp),hl        ; delay
F407  E3                  ex    (sp),hl        ; delay
F408  DD 7E 04            ld    a,(ix+004)     ; load video attribute
F40B  F6 F0               or    11110000b      ; mask video attribute
F40D  D3 8E               out   (vidatr),a     ; send out on video attribute ram (RAM 3)
F40F  F1                  pop   af             ; restore char.
F410  F5                  push  af             ; resave char.
F411  D3 80               out   (data0a),a     ; write char. on video ram (RAM 0)
F413  AF                  xor   a              ; clear accumulator
F414  D3 8D               out   (datreg),a     ; write 0 to register 6545
F416  F1                  pop   af             ; restore character
F417  C9                  ret                  ; and ret
                    ;
                    ;
F418                vidcompare:
                          ; compare for HL = endvid+1
F418  E5                  push  hl             ; save video pointer
F419  11 0780             ld    de,endvid+1    ; DE = end video + 1
F41C  AF                  xor   a              ; clear carry
F41D  ED 52               sbc   hl,de          ; subctract HL with DE
F41F  E1                  pop   hl             ; restore video pointer
F420  C9                  ret                  ; and ret
                    ;
F421                pfx02:
                          ; set cursor off/on
F421  CB 47               bit   0,a            ; is pair
F423  20 0E               jr    nz,curson      ; no, then cursor on
                                               ; yes, then cursor off
                    ;
                    ;
F425                cursoff:
```

```
                                ; cursor off
F425    06 20                   ld      b,00100000b     ; no cursor type
F427                            outdat:
F427    3E 0A                   ld      a,10            ; cursor start register
F429    D3 8C                   out     (addreg),a      ; set 6545 cursor start register
F42B    78                      ld      a,b
F42C    D3 8D                   out     (datreg),a      ; send out to 6545 cursor start register
                                ;
F42E                            setdummy:
F42E    3E 1F                   ld      a,31            ; A = dummy location
F430    D3 8C                   out     (addreg),a      ; send out to 6545 register address
F432    C9                      ret                     ; and return
                                ;
                                ;
F433                            curson:
                                ; cursor on
F433    3E 0A                   ld      a,10            ; cursor start register
F435    D3 8C                   out     (addreg),a      ; set 6545 cursor start register
F437    06 00                   ld      b,00000000b     ; cursor fixed (blinking hardware)
F439    18 EC                   jr      outdat          ; set it
                                ;
                                ;
F43B                            updtvid:
                                ; update video pointer
F43B    DD 5E 02                ld      e,(ix+002)      ; load curently display start (L)
F43E    DD 56 03                ld      d,(ix+003)      ; load curently display start (H)
F441    19                      add     hl,de           ; compute relative position
                                                        ; and count with updtcpur
F442                            updtcpur:
                                ; update 6545 cursor position and 6545 update register at HL
F442    3E 0E                   ld      a,14            ; cursor position (H) register
F444    D3 8C                   out     (addreg),a      ; set it
F446    7C                      ld      a,h             ; load cursor address (H)
F447    D3 8D                   out     (datreg),a      ; set it
F449    3E 0F                   ld      a,15            ; cursor position (L) register
F44B    D3 8C                   out     (addreg),a      ; set it
F44D    7D                      ld      a,l             ; load cursor address (L)
F44E    D3 8D                   out     (datreg),a      ; set it
                                ;
F450                            updtureg:
                                ; update 'update register' at HL
F450    3E 12                   ld      a,18            ; update address (H) register
F452    D3 8C                   out     (addreg),a      ; set it
F454    7C                      ld      a,h             ; load cursor address (H)
F455    D3 8D                   out     (datreg),a      ; set it
F457    3E 13                   ld      a,19            ; update address (L) register
F459    D3 8C                   out     (addreg),a      ; set it
F45B    7D                      ld      a,l             ; load cursor address (L)
F45C    D3 8D                   out     (datreg),a      ; set it
F45E    18 CE                   jr      setdummy        ; set dummy location and return
                                ;
                                ;
F460                            updtdstart:
                                ; update display start
F460    DD CB 04 7E             bit     7,(ix+004)      ; test no scroll flag; is zero ?
F464    28 26                   jr      z,noupdstart    ; yes, then no scrolling
F466    E5                      push    hl              ; save video pointer
F467    DD 6E 02                ld      l,(ix+002)      ; load curently display start (L)
```

```
F46A  DD 66 03                ld      h,(ix+003)      ; load curently display start (H)
F46D  11 0050                 ld      de,80           ; DE = char. for line
F470  19                      add     hl,de           ; point to next line
F471  CB 9C                   res     3,h             ;
F473  DD 75 02                ld      (ix+002),l      ; save curently display start (L)
F476  DD 74 03                ld      (ix+003),h      ; save curently display start (H)
F479  3E 0C                   ld      a,12            ; disp. start (H) register
F47B  D3 BC                   out     (addreg),a      ; set it
F47D  7C                      ld      a,h             ; load disp. start address (H)
F47E  D3 BD                   out     (datreg),a      ; set it
F480  3E 0D                   ld      a,13            ; disp. start (L) register
F482  D3 BC                   out     (addreg),a      ; set it
F484  7D                      ld      a,l             ; load disp. start address (L)
F485  D3 BD                   out     (datreg),a      ; set it
F487  21 0780                 ld      hl,endvid+1     ; set new video pointer
F48A  18 1E                   jr      cescr1          ; go to clear to end screen
                      ;
F48C                  noupdstart:
F48C  01 0050                 ld      bc,80           ; BC = char/row
F48F  CD F4DC                 call    divide          ; HL = Xco
F492  C9                      ret                     ; and ret
                      ;
                      ;
F493                  chr$05:
F493                  cendlin:
                              ; clear to end line starting at HL
F493  E5                      push    hl              ; save video pointer
F494  CD F4CC                 call    cbeglin         ; call cursor begin of line
F497  19                      add     hl,de           ;
F498  EB                      ex      de,hl           ;
F499  E1                      pop     hl              ; restore video pointer
F49A  E5                      push    hl              ; resave it
F49B  18 07                   jr      cescr0          ; go to clear 'DE' char.
                      ;
F49D                  chr$12:
                              ; clear screen
F49D  21 0000                 ld      hl,strvid       ; HL = start video
                                                      ; and clear to end of screen
                      ;
F4A0                  chr$06:
F4A0                  cendscr:
                              ; clear to end of screen starting at HL
F4A0  E5                      push    hl              ; save video pointer
F4A1  11 07D0                 ld      de,endvid+81    ; DE = end video + one line + 1
F4A4                  cescr0:
F4A4  F3                      di                      ;
F4A5  AF                      xor     a               ; clear carry
F4A6  EB                      ex      de,hl           ; computer number of
F4A7  ED 52                   sbc     hl,de           ; character then remaining
F4A9  EB                      ex      de,hl           ; until end (line or screen)
F4AA                  cescr1:
F4AA  D5                      push    de              ; save number of char.
F4AB  CD F43B                 call    updtvid         ; update video pointer
F4AE  D1                      pop     de              ; restore char. number
F4AF                  cescr2:
F4AF  7A                      ld      a,d             ; DE is zero ?
F4B0  B3                      or      e               ;
F4B1  28 16                   jr      z,cescr3        ; yes, then exit
```

```
F4B3                    cescr4:
F4B3    DB 8C                   in      a,(addreg)      ; load 6545 status
F4B5    CB 7F                   bit     7,a             ; test sync bit
F4B7    28 FA                   jr      z,cescr4        ; wait if busy
F4B9    E3                      ex      (sp),hl         ; delay
F4BA    E3                      ex      (sp),hl         ; delay
F4BB    3E 20                   ld      a,' '           ; load space
F4BD    D3 80                   out     (data0a),a      ; write char. on video ram (RAM 0)
F4BF    3E FF                   ld      a,11111111b     ; video normal mode attribute
F4C1    D3 8E                   out     (vidatr),a      ; send out on video attribute ram (RAM 3)
F4C3    AF                      xor     a               ; clear accumulator
F4C4    D3 8D                   out     (datreg),a      ; write 0 to register 6545
F4C6    1B                      dec     de              ; dec. char counter
F4C7    18 E6                   jr      cescr2          ; and count
F4C9                    cescr3:
F4C9    E1                      pop     hl              ; restore video pointer
F4CA    FB                      ei                      ;
F4CB    C9                      ret                     ; and return
                        ;
                        ;
F4CC                    chr$13:
F4CC                    cbeglin:
                                ; cursor beginning of line
F4CC    C5                      push    bc              ; save register
F4CD    42                      ld      b,d
F4CE    4B                      ld      c,e             ; bc = char. for line
F4CF    CD F4DC                 call    divide          ;
F4D2    2E 00                   ld      l,0             ;
F4D4    50                      ld      d,b             ;
F4D5    59                      ld      e,c             ;
F4D6    C1                      pop     bc              ; restore register
F4D7                    cbegl0:
F4D7    C8                      ret     z               ;
F4D8    19                      add     hl,de           ;
F4D9    3D                      dec     a               ;
F4DA    18 FB                   jr      cbegl0          ;
                        ;
                        ;
F4DC                    divide:
                                ; division routine: divide Hl with BC; quoto in A, resto in HL
F4DC    AF                      xor     a               ; clear accumulator and carry flag
F4DD                    hl.gt.0:
F4DD    ED 42                   sbc     hl,bc           ;
F4DF    3C                      inc     a               ;
F4E0    30 FB                   jr      nc,hl.gt.0      ;
F4E2    3D                      dec     a               ;
F4E3    09                      add     hl,bc           ;
F4E4    C9                      ret
                        ;
                        ;
                        ;
F4E5                    initialize:
                                ; PIO, 6545 and IX table initialization
                        ;
F4E5    DD 22 0047              ld      (ix.pnt),ix     ; set IX table address
F4E9    3E 8F                   ld      a,mode2         ; bidiretional mode
F4EB    D3 82                   out     (cont0a),a      ; set port a PIO 0
F4ED    D3 86                   out     (cont1a),a      ; set port a PIO 1
```

```
F4EF   D3 8A                    out     (cont2a),a      ; set port a PIO 2
F4F1   3E CF                    ld      a,mode3         ; control mode
F4F3   D3 83                    out     (cont0b),a      ;
F4F5   08                       ex      AF,AF'          ; save control mode
F4F6   3E 00                    ld      a,00000000b     ; set i/o bit
F4F8   D3 83                    out     (cont0b),a      ; set port b PIO 0
F4FA   08                       ex      AF,AF'          ; restore control mode
F4FB   D3 87                    out     (cont1b),a      ;
F4FD   08                       ex      AF,AF'          ; save control mode
F4FE   3E FF                    ld      a,11111111b     ; set i/o bit
F500   D3 87                    out     (cont1b),a      ; set port b PIO 1
F502   08                       ex      AF,AF'          ; restore control mode
F503   D3 8B                    out     (cont2b),a      ;
F505   3E F1                    ld      a,11110001b     ; set i/o bit
F507   D3 8B                    out     (cont2b),a      ; set port b PIO 2
                        ;
F509   DB 89                    in      a,(data2b)      ; read board flag
F50B   CB 8F                    res     1,a             ; set 80 char for line
F50D   D3 89                    out     (data2b),a      ;
                        ;
F50F   21 F53D                  ld      hl,tab6545      ; HL --> 6545 initialization table
F512   06 0C                    ld      b,12            ; data output counter
F514                    init00:
F514   78                       ld      a,b             ; load on a
F515   3D                       dec     a               ;
F516   D3 8C                    out     (addreg),a      ; set 6545 register
F518   7E                       ld      a,(hl)          ; get data for init
F519   D3 8D                    out     (datreg),a      ; send out on 6545 register
F51B   23                       inc     hl              ; point to next data
F51C   10 F6                    djnz    init00          ; and count until are output
                        ;
F51E   06 08                    ld      b,8             ; filling rimanents register with 00
F520                    init01:
F520   78                       ld      a,b             ;
F521   C6 0B                    add     a,11            ;
F523   D3 8C                    out     (addreg),a      ; set 6845 register
F525   AF                       xor     a               ; clear accumulator
F526   D3 8D                    out     (datreg),a      ; sen out to 6545 register
F528   10 F6                    djnz    init01          ; and count until all are output
                        ;
F52A   3E 1F                    ld      a,31            ; A = dummy location
F52C   D3 8C                    out     (addreg),a      ; send out to 6545 register
                        ;
F52E   AF                       xor     a               ; clear accumulator
F52F   2A 0047                  ld      hl,(ix.pnt)     ; HL -> routines table
F532   06 10                    ld      b,16            ; 16 byte to clear
F534                    init02:
F534   77                       ld      (hl),a          ; clear byte of table
F535   23                       inc     hl              ; increase table pointer
F536   10 FC                    djnz    init02          ; count untill all are cleared
                        ;
F538   3D                       dec     a               ; initial video attribute, scroll flag,
                                                        ; no alpha look and mute keyboard
F539   DD 77 04                 ld      (ix+004),a      ; set initial video attribute
F53C   C9                       ret                     ; and return to caller
                        ;
F53D                    tab6545:
                                ; 6545 initialization table
```

```
                        ;
0010                            .radix  16
                        ;
F53D    0B 00                   defb    0b,00           ; cursor end, cursor start
F53F    0B                      defb    0b              ; scan line-1
F540    48                      defb    48              ; mode control
F541    18                      defb    18              ; vert. sync position
F542    18                      defb    18              ; vert. displayed
F543    00                      defb    00              ; vert. total adjust
F544    1A                      defb    1a              ; vert. total-1
F545    28                      defb    28              ; vsync, hsync widths
F546    57                      defb    57              ; horiz. sync position
F547    50                      defb    50              ; horiz. displayed
F548    6F                      defb    6f              ; horiz. total-1
                        ;
000A                            .radix  10
                        ;
                        ;
                        ;
                        ;
F549                    rdcpos:
                                ; read cursor pos and set update reg = cpos reg
F549    3E 0E                   ld      a,14            ; cursor position (H) register
F54B    D3 8C                   out     (addreg),a      ; set it
F54D    DB 8D                   in      a,(datreg)      ; read cursor address (H)
F54F    67                      ld      h,a             ; store it in H reg.
F550    3E 0F                   ld      a,15            ; cursor position (L) register
F552    D3 8C                   out     (addreg),a      ; set it
F554    DB 8D                   in      a,(datreg)      ; read cursor address (L)
F556    6F                      ld      l,a             ; store it in L reg.
F557    C3 F450                 jp      updtureg        ; and update 'update register'
                        ;
                                endif
                        ;
                                page
```

```
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*             Print string pointed by DE                *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ;
                        ; *** Print @ position 12,12 ***
                        ;
F55A    21 0C0C         pr@12.12: ld    hl,12 * 256 + 12 ; @ 12,12
                        ;
                        ;
                        ; *** Print string @ positon HL ***
                        ;
F55D                    printat:
F55D    D5                      push    de              ; save string pointer
F55E    CD F344                 call    movcurs         ; move cursor at HL
F561    D1                      pop     de              ; restore str pointer and count
                                                        ; with strout
                        ;
                        ;
                        ; *** Print string pointed by DE ***
                        ;
F562                    strout:
F562    1A                      ld      a,(de)          ; load char.
F563    FE 24                   cp      endmsg          ; end message
F565    C8                      ret     z               ; yes, then return
F566    4F                      ld      c,a             ; else move to register C
F567    D5                      push    de              ; save text pointer
F568    CD F233                 call    cout            ; and print it
F56B    D1                      pop     de              ; restore text pointer
F56C    13                      inc     de              ; point to next char
F56D    18 F3                   jr      strout          ; and repeat
                                ;
                                page
```

```
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*                  Floppy disk routines                 *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ;
                               if      necnt
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*           Floppy disk driver and subroutine           *
                        ;*         for N.E. 5 Inc. floppy disk controller        *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ; Copyright (C) 1983, 1984 by Studio Lg, Genova - Italy
                        ; Author: Martino Stefano
                        ; Third Modify by Gallerani Paolo
                        ; Last Revision 06/08/84 21:00
                        ;
                        ; Compatility Version 4.2-1
                        ;
                               if      rev lt 42
                               .printx .*** Warning:Incompatible Floppy Disk Driver ***.
                               endif
                        ;
0003                    rtycnt  equ     3               ; max retries before disk I/O error
0002                    rstcnt  equ     2               ; max restore before disk I/O error
                               ;
                               page
```

```
                        ;
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*              FD 1771 I/O port                         *
                        ;*                                                       *
                        ;*********************************************************
                        ;
00D0                    fddsts  equ     0d0h            ; fdd status port
00D1                    fdtrk   equ     0d1h            ; fdd track port
00D2                    fdsec   equ     0d2h            ; fdd sector port
00D6                    fdlch   equ     0d6h            ; fdd lach port
00D7                    fddat   equ     0d7h            ; fdd data port
00D0                    fdcmd   equ     fddsts          ; fdd command port
                        ;
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*              FD 1771 Command Summary                  *
                        ;*                                                       *
                        ;*********************************************************
                        ;
0006                    fdrest  equ     00000110b       ; fdd restore command code
0016                    fdsek   equ     00010110b       ; fdd seek command code
0088                    fdrd    equ     10001000b       ; fdd read command code
00A8                    fdwt    equ     10101000b       ; fdd write command code
00D0                    fdrst   equ     11010000b       ; fdd reset int. command code
                                ;
                                page
```

```
                        ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              Routines Entry point                   *
                        ;*                                                     *
                        ;*******************************************************
                        ;
                        ;
                        ;*******************************************************
                        ;* F D I O S    Fdd I/O 128 byte                       *
                        ;*******************************************************
                        ;
F56F                    fdios:
F56F    AF                      xor     a               ; CY = 0 then 128 byte r/w
F570    18 01                   jr      fdio            ; go to commom entry point
                        ;
                        ;
                        ;*******************************************************
                        ;* F D D I O D  Fdd I/O 256 byte                       *
                        ;*******************************************************
                        ;
F572                    fdiod:
F572    37                      scf                     ; CY = 1 then 256 byte r/w
                        ;
                        ;
F573                    fdio:
F573    DD 2A 0047              ld      ix,(ix.pnt)     ; IX = routines table
F577    DD CB 09 C6             set     0,(ix+009)      ; set 256 byte r/w
F57B    38 04                   jr      c,fdio00        ; jmp if cy = 1
F57D    DD CB 09 86             res     0,(ix+009)      ; set 128 byte r/w
F581                    fdio00:
F581    7E                      ld      a,(hl)          ; get unit num. and side
F582    57                      ld      d,a             ; save for side select
F583    E6 03                   and     00000011b       ; mask bit 0 and bit 1 (unit num.)
F585    CD F661                 call    newdrv          ; select drive and restore old trk num.
F588    23                      inc     hl              ; H.L track para adrs
F589    22 0045                 ld      (hladrs),hl
F58C    0E 02                   ld      c,rstcnt        ; C = restore count
F58E                    rty00:
F58E    06 03                   ld      b,rtycnt        ; B = retry count
F590                    rty01:
F590    C5                      push    bc
F591    2A 0045                 ld      hl,(hladrs)     ; H.L track para adrs
F594    46                      ld      b,(hl)          ; B = track num.
F595    23                      inc     hl              ; skip track hi byte for Wdd comp.
F596    23                      inc     hl              ; H.L sector para adrs
F597    7E                      ld      a,(hl)          ; A = sector num.
F598    D3 D2                   out     (fddsec),a      ; set sector num.
F59A    78                      ld      a,b             ; set new track
F59B    D3 D7                   out     (fdddat),a      ; number and
F59D    3E 16                   ld      a,fddsek        ; execute seek
F59F    D3 D0                   out     (fddcmd),a      ; command
F5A1    CD F623                 call    errlget         ; wait until end command
F5A4    B7                      or      a               ; error detect ?
F5A5    20 54                   jr      nz,fdderchk     ; yes ! go to error check
```

```
F5A7                            trkok:
                                        ; track and sector are ok
F5A7    23                              inc     hl              ; H.L dma para adrs
F5A8    5E                              ld      e,(hl)          ; E = dma adrs low
F5A9    23                              inc     hl              ;
F5AA    56                              ld      d,(hl)          ; D = dma adrs high
F5AB    23                              inc     hl              ; H.L R/W para adrs (0 = read ,1 = write)
F5AC    DD CB 09 46                     bit     0,(ix+009)      ; test bit 0 of (ix+009) for fdd num byte
F5B0    3E 80                           ld      a,128           ; if bit 0 = 0 then 128 byte r/w
F5B2    28 01                           jr      z,trkok1        ;
F5B4    AF                              xor     a               ; else 256 byte r/w
F5B5                            trkok1:
F5B5    47                              ld      b,a             ; B = num. byte to R/W
F5B6    0E D7                           ld      c,fdddat        ; C = fdd data I/O port
                                ;
F5B8    7E                              ld      a,(hl)          ; A = R/W para
F5B9    EB                              ex      de,hl           ; HL = dma adrs
F5BA    B7                              or      a               ; test for read or write
F5BB    20 18                           jr      nz,fdwrite      ; if nz then write
F5BD                            fdread:
                                        ; else fdd read
F5BD    3E 88                           ld      a,fddrd         ; fdd read command
F5BF    D3 D0                           out     (fddcmd),a      ; exec. command
F5C1    CD F63C                         call    fddelay         ; wait aproax 56 microS
F5C4    18 03                           jr      fdrd01
                                ;
F5C6                            fdrd00:
F5C6    0F                              rrca                    ; busy bit --> CY
F5C7    30 22                           jr      nc,fdioend      ; if busy = 0 then end read
F5C9                            fdrd01:
F5C9    DB D0                           in      a,(fddsts)      ; test fdd status
F5CB    CB 4F                           bit     1,a             ; data request active ?
F5CD    28 F7                           jr      z,fdrd00        ; no ! then test busy bit
F5CF    ED A2                           ini                     ; read one byte
F5D1    20 F6                           jr      nz,fdrd01       ; and wait until all are read
                                ;
F5D3    18 16                           jr      fdioend         ; go to end fdio
F5D5                            fdwrite:
                                        ; fdd write
F5D5    3E A8                           ld      a,fddwt         ; fdd write command
F5D7    D3 D0                           out     (fddcmd),a      ; exec. command
F5D9    CD F63C                         call    fddelay         ; wait aproax 56 microS
F5DC    18 03                           jr      fdwt01
                                ;
F5DE                            fdwt00:
F5DE    0F                              rrca                    ; busy bit --> CY
F5DF    30 0A                           jr      nc,fdioend      ; if busy = 0 then end write
F5E1                            fdwt01:
F5E1    DB D0                           in      a,(fddsts)      ; test fdd status
F5E3    CB 4F                           bit     1,a             ; data request active ?
F5E5    28 F7                           jr      z,fdwt00        ; no ! then test busy bit
F5E7    ED A3                           outi                    ; write one byte
F5E9    20 F6                           jr      nz,fdwt01       ; and wait until all are write
                                ;
F5EB                            fdioend:
                                        ; end of read or write
F5EB    60                              ld      h,b             ; save byte counter
F5EC    CD F633                         call    err2get         ; wait until end command
```

```
F5EF    B7                      or      a               ; error occurs ?
F5F0    20 09                   jr      nz,fdderchk     ; then error check else A=0
F5F2    B4                      or      h               ; all byte are read or write ?
F5F3    3E 02                   ld      a,00000010b     ; set density error
F5F5    20 04                   jr      nz,fdderchk     ; no, then i/o error
F5F7    AF                      xor     a               ; clear accumulator for normal return
F5F8    C1                      pop     bc              ; else restore BC
F5F9    18 18                   jr      fddret          ; and ret to call
                        ;
                        ;
F5FB                    fdderchk:
                                ; fdd check for any error
F5FB    C1                      pop     bc              ; B = rtycnt. C = rstcnt
F5FC    F5                      push    af              ; save fdd status
F5FD    E6 03                   and     00000011b       ; bit 0 = time-out error
                                                        ; bit 1 = density  error
F5FF    20 13                   jr      nz,fddrt1       ; ret if one
F601    F1                      pop     af              ; restore fdd status
F602    05                      dec     b               ; retry count down
F603    20 8B                   jr      nz,rty01        ; repeat until rtycnt = 0
F605    0D                      dec     c               ; restore count down
F606    28 0B                   jr      z,fddret        ; retry error return
F608    F5                      push    af              ; save fdd status
F609    CD F619                 call    fdsek0          ; fdd seek to track 0
F60C    B7                      or      a               ; error ?
F60D    20 05                   jr      nz,fddrt1       ; then rty err
F60F    F1                      pop     af              ; restore fdd status
F610    C3 F58E                 jp      rty00           ; and retry
F613                    fddret:
F613    F5                      push    af              ; save fdd status
F614    AF              fddrt1: xor     a               ; clear register A
F615    D3 D6                   out     (fddlch),a      ; deselect any drive
F617    F1                      pop     af              ; restore fdd status
F618    C9                      ret                     ; and ret
                        ;
                        ;
F619                    fdsek0:
                                ; fdd seek to track 0
F619    C5                      push    bc              ; save register
F61A    3E 06                   ld      a,fddrest       ; fdd restore command
F61C    D3 D0                   out     (fddcmd),a      ; exec. command
F61E    CD F641                 call    waitfd          ; wait until end command
F621    C1                      pop     bc              ; restore register
F622    C9                      ret                     ; and ret
                        ;
                        ;
F623                    erriget:
                                ; return error in A register for type 1 command
F623    CD F641                 call waitfd             ; wait until end command
F626    B7                      or      a               ; time out error ?
F627    C0                      ret     nz              ; yes , return to call
F628    78                      ld      a,b             ; else load fdd status
F629    E6 10                   and     00010000b       ; mask bit 4 (seek error)
F62B    CB 27                   sla     a               ; move to bit 5
F62D    4F                      ld      c,a             ; save seek error bit
F62E    78                      ld      a,b             ; reload fdd status
F62F    E6 08                   and     00001000b       ; mask bit 3 (crc error)
F631    B1                      or      c               ; A: bit 3 = crc ,bit 5 = seek
```

```
F632  C9                        ret                     ; return to call
                        ;
F633                    err2get:
                                ; return error in A register for type 2 command
F633  CD F641                   call    waitfd          ; wait until end command
F636  B7                        or      a               ; time out error ?
F637  C0                        ret     nz              ; yes ,return to caller
F638  78                        ld      a,b             ; load fdd status
F639  E6 5C                     and     01011100b       ; mask wrt-prtc,rnf,crc,lst-dat error
F63B  C9                        ret                     ; and return
                        ;
F63C                    fddelay:
                                rept    4
                                ex      (sp),hl         ; delay beetwen write command reg.
                                endm                    ; to read status reg.
F63C  E3              +         ex      (sp),hl         ; delay beetwen write command reg.
F63D  E3              +         ex      (sp),hl         ; delay beetwen write command reg.
F63E  E3              +         ex      (sp),hl         ; delay beetwen write command reg.
F63F  E3              +         ex      (sp),hl         ; delay beetwen write command reg.
F640  C9                        ret


                        ;
F641                    waitfd:
                                ; wait until fdd busy is reset.
F641  CD F63C                   call    fddelay         ; wait aproax 56 microS
F644  06 02                     ld      b,2             ; set soft timer
F646                    wait00:
F646  11 0000                   ld      de,0            ; for aproax five seconds
F649                    wait01:
F649  DB D0                     in      a,(fddsts)      ; input to fdd status
F64B  CB 47                     bit     0,a             ; test busy bit
F64D  28 0F                     jr      z,wait02        ; jump if no command is in progress
F64F  1B                        dec     de              ;
F650  7A                        ld      a,d             ; timer down
F651  B3                        or      e               ;
F652  20 F5                     jr      nz,wait01       ;
F654  05                        dec     b               ;
F655  20 EF                     jr      nz,wait00       ; time out
F657                    timeout:
F657  3E D0                     ld      a,fddrst        ; reset fdd controller
F659  D3 D0                     out     (fddcmd),a      ; exec. command
F65B  3E 01                     ld      a,00000001b     ; set time-out bit error
F65D  C9                        ret                     ; and ret
F65E                    wait02:
F65E  47                        ld      b,a             ; save fdd status in B register
F65F  AF                        xor     a               ; clear accumulator for
F660  C9                        ret                     ; normal return
                        ;
F661                    newdrv:
                                ; select new drive and restore old trk num.
F661  E5                        push    hl              ; save para pointer
F662  4F                        ld      c,a             ;
F663  06 00                     ld      b,0             ; BC = new unit num.
F665  21 F68A                   ld      hl,fddtab       ;
F668  09                        add     hl,bc           ; HL. byte to select drive
F669  5E                        ld      e,(hl)          ; E = byte to select drive
F66A  7A                        ld      a,d             ; A = unit num. and side
F66B  E6 10                     and     00010000b       ; mask bit 4 (side)
```

```
F66D   CB 27                    sla     a               ; move side to bit 5
F66F   B3                       or      e               ; A = unit + 5/8 sel + side
F670   D3 D6                    out     (fddlch),a      ; select new drive and side
F672   21 0044                  ld      hl,unit         ; H.L old unit num.
F675   7E                       ld      a,(hl)          ; A = old unit num.
F676   B9                       cp      c               ; old unit equ new unit ?
F677   28 0F                    jr      z,drvequ        ; skip if yes
F679   71                       ld      (hl),c          ; save new unit num.
F67A   21 0040                  ld      hl,fd0otr       ;
F67D   E5                       push    hl              ;
F67E   85                       add     a,l             ; * Version for less space (4/6)
F67F   6F                       ld      l,a             ;       add only low byte *

                        ;;      ld      e,a             ;
                        ;;      ld      d,0             ; DE = old unit num.
                        ;;      add     hl,de           ; H.L old trk of old unit num.
F680   DB D1                    in      a,(fddtrk)      ; save track num.
F682   77                       ld      (hl),a          ; of old unit num.
F683   E1                       pop     hl              ;
F684   09                       add     hl,bc           ; H.L old trk of new unit num.
F685   7E                       ld      a,(hl)          ; select old track
F686   D3 D1                    out     (fddtrk),a      ; num. of new unit
F688                    drvequ:
F688   E1                       pop     hl              ; restore para pointer
F689   C9                       ret                     ; and ret
                        ;
F68A                    fddtab:
                                ; if fdd controller by n.e.
F68A   01 02 04 08              defb    01,02,04,08     ; unit 0 to 3 are 5 inch (bit 5 = 0)
                        ;
                                page
```

```
                endif
;
                page
```

```
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*                Hard disk routines                     *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ;
                                if      hard            ; hard disk exist
                        ;
                        ; ***************************************
                        ; * Size Table for                      *
                        ; *            Hard Disk BASF 6188      *
                        ; ***************************************
                        ;
0168                    Cyls    equ     360             ; number of cylinders
0004                    Heads   equ     4               ; number of heads
                        ;
F68E                    initab:
F68E    01 68                   defb    high Cyls,low Cyls; number of cylinders
F690    04                      defb    Heads           ; number of heads
F691    00 80                   defb    0,128           ; starting reduced current cylinder
F693    00 40                   defb    0,64            ; starting write precompensation cylinder
F695    0B                      defb    11              ; maximum ECC data burst length
                        ;
                        ;
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*      Winchester disk driver and subroutine            *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                        ; Copyright (C) 1983, 1984 Studio Lg, Genova - Italy
                        ; Author: Martino Stefano
                        ; Third Modify by Gallerani Paolo
                        ; Last Revision 06/08/84 21:07
                        ;
                        ; Compatility Version 4.2-1
                        ;
                                if      rev lt 42
                                .printx .*** Warning:Incompatible Hard Disk Driver ***.
                                endif
                        ;
                        ;
                                page
```

```
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*      HARD DISK OUTPUT AND INPUT PORTS                 *
                        ;*                                                       *
                        ;*********************************************************
                        ;
00B8                    wport0  equ     0b8h            ; sasi write port 0 - write data
00B9                    wport1  equ     0b9h            ; sasi write port 1 - software reset
00BA                    wport2  equ     0bah            ; sasi write port 2 - cntlr select
00BB                    wport3  equ     0bbh            ; sasi write port 3 - not used
00B8                    rport0  equ     0b8h            ; sasi read  port 0 - read data
00B9                    rport1  equ     0b9h            ; sasi read  port 1 - read status
00BA                    rport2  equ     0bah            ; sasi read  port 2 - not used
00BB                    rport3  equ     0bbh            ; sasi read  port 3 - not used
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*      HARD DISK VARIOUS EQUATES                        *
                        ;*                                                       *
                        ;*********************************************************
                        ;
0000                    reqbit  equ     000h            ; request line bit position
0001                    reqmsk  equ     001h            ; request mask for bit test
0001                    busybit equ     001h            ; busy line bit position
0002                    busymsk equ     002h            ; busy mask for bit test
0002                    msgbit  equ     002h            ; message line bit position
0004                    msgmsk  equ     004h            ; message mask for bit test
0003                    cdbit   equ     003h            ; command/data bit position
0008                    cdmsk   equ     008h            ; command/data mask for bit test
0004                    iobit   equ     004h            ; input/output bit position
0010                    iomsk   equ     010h            ; input/output mask for bit test
0002                    errmsk  equ     002h            ; test for an error
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*      HARD DISK CONTROLLER COMMAND EQUATES             *
                        ;*                                                       *
                        ;*********************************************************
                        ;
0000                    drvrdy  equ     000h            ; test drive ready command
0004                    format  equ     004h            ; format command code
0008                    read    equ     008h            ; read command code
000A                    write   equ     00ah            ; write command code
0003                    sense   equ     003h            ; status sense command code
000C                    initl   equ     00ch            ; initialize disk size command
000B                    seek    equ     00bh            ; seek command size
0001                    recal   equ     001h            ; recalibrate command code
0003                    sense   equ     003h            ; sense status command code
00E0                    ramdiag equ     0e0h            ; ram diagnostic command code
                                ;
                                ;
F696                    parowmsg:
F696    0D 0A 48 61             defb    cr,lf,'HardDisk Parameter overflow',endmsg
F69A    72 64 44 69
F69E    73 6B 20 50
```

```
F6A2    61 72 61 6D
F6A6    65 74 65 72
F6AA    20 6F 76 65
F6AE    72 66 6C 6F
F6B2    77 24

                                ;
                                page
```

```
                                ;
                                ;
                                ;*******************************************************
                                ;*                                                     *
                                ;*              Routines Entry point                   *
                                ;*                                                     *
                                ;*******************************************************
                                ;
                                ;
F6B4                            wdio:
                                        ; wdd i/o 256 byte
F6B4    DB B9                           in      a,(rport1)      ; load cntlr status
F6B6    E6 E0                           and     11100000b       ; mask bit 5,6,7. Hard disk exist ?
F6B8    C0                              ret     nz              ; ret if not
F6B9    22 0045                         ld      (hladrs),hl     ; save r/w para adrs
F6BC                            wdio1:
F6BC    7E                              ld      a,(hl)          ; A = unit & side
F6BD    47                              ld      b,a             ; save in B for unit
F6BE    E6 30                           and     00110000b       ; mask side number
                                        rept    4               ;
                                        srl     a               ; move side number to bit 0,1
                                        endm                    ;
F6C0    CB 3F           +               srl     a               ; move side number to bit 0,1
F6C2    CB 3F           +               srl     a               ; move side number to bit 0,1
F6C4    CB 3F           +               srl     a               ; move side number to bit 0,1
F6C6    CB 3F           +               srl     a               ; move side number to bit 0,1
F6C8    FE 02                           cp      Heads/2         ; test for side overflow
F6CA    D2 F732                         jp      nc,parower      ; if side > (Heads/2)-1 then para overflow
F6CD    CB 40                           bit     0,b             ; test unit number
F6CF    28 02                           jr      z,wdio2         ; jump if unit 0
F6D1    C6 02                           add     a,(Heads/2)     ;
F6D3                            wdio2:
F6D3    4F                              ld      c,a             ; C = head address (0 to Heads-1)
F6D4    23                              inc     hl              ; H.L = track para adrs.
F6D5    5E                              ld      e,(hl)          ; E = low byte track number
F6D6    23                              inc     hl              ; point to high byte
F6D7    56                              ld      d,(hl)          ; D = track high byte
F6D8    23                              inc     hl              ; point to sector
F6D9    E5                              push    hl              ; save sec para pointer
F6DA    A7                              and     a               ; cy=0
F6DB    21 0167                         ld      hl,Cyls-1       ; Max # of Cyl
F6DE    ED 52                           sbc     hl,de           ; compare with par
F6E0    38 4F                           jr      c,parov0        ; overflow: out of disk bounds
F6E2    21 0000                         ld      hl,0            ; reset product
F6E5    44                              ld      b,h             ; BC = head address
                                        rept    Heads           ; Repeat # of Heads Times
                                        add     hl,de           ; HL = track * Heads
                                        endm                    ;
F6E6    19              +               add     hl,de           ; HL = track * Heads
F6E7    19              +               add     hl,de           ; HL = track * Heads
F6E8    19              +               add     hl,de           ; HL = track * Heads
F6E9    19              +               add     hl,de           ; HL = track * Heads
F6EA    09                              add     hl,bc           ; HL = (track * Heads) + head address
                                        rept    5               ;
                                        add     hl,hl           ; HL = (track * Heads + head address)*32
                                        endm                    ;
```

```
F6EB  29          +          add   hl,hl          ; HL = (track * Heads + head address)*32
F6EC  29          +          add   hl,hl          ; HL = (track * Heads + head address)*32
F6ED  29          +          add   hl,hl          ; HL = (track * Heads + head address)*32
F6EE  29          +          add   hl,hl          ; HL = (track * Heads + head address)*32
F6EF  29          +          add   hl,hl          ; HL = (track * Heads + head address)*32
F6F0  EB                     ex    de,hl          ; DE = (track * Heads + head address)*32
F6F1  E1                     pop   hl             ; H.L = track para adrs.
F6F2  7E                     ld    a,(hl)         ; A = sector number
F6F3  3D                     dec   a              ; convert sector to base 0
F6F4  FE 20                  cp    32             ; test for sector overflow
F6F6  30 3A                  jr    nc,parower     ; if sector-1 > 31 then para overflow
F6F8  4F                     ld    c,a            ; BC= sector number
F6F9  EB                     ex    de,hl          ; HL= (trk*Hds+hda)*32 , D.E = sec para adrs
F6FA  09                     add   hl,bc          ; HL= (track * Hds + head address)*32 + sec num
F6FB  7C                     ld    a,h            ; A = middle address
F6FC  65                     ld    h,l            ;
F6FD  6F                     ld    l,a            ; swap H with L
F6FE  22 004C                ld    (task+2),hl    ; set middle and low address
F701  EB                     ex    de,hl          ; H.L = sec para adrs
F702  23                     inc   hl             ; H.L = dma para adrs
F703  5E                     ld    e,(hl)         ;
F704  23                     inc   hl             ;
F705  56                     ld    d,(hl)         ; DE = dma address
F706  D5                     push  de             ; save dma address
F707  23                     inc   hl             ; H.L = r/w para adrs
F708  7E                     ld    a,(hl)         ; A = r/w flag (0 = read, 1 = write)
F709  F5                     push  af             ;
F70A  23                     inc   hl             ; H.L = block count
F70B  7E                     ld    a,(hl)         ; A = num sec to r/w
F70C  32 004E                ld    (task+4),a     ; set block count
F70F  5F                     ld    e,a            ; E = num sec to r/w
F710  F1                     pop   af             ; A = r/w flag
F711  E1                     pop   hl             ; HL = dma addess
F712  01 00B8                ld    bc,wport0      ; wdd data port
                                                  ; B = 0 = r/w 256 byte
F715  B7                     or    a              ; read or write ?
F716  20 23                  jr    nz,wdwrite     ; A = 1 then write
                             ;
F718                  wdread:
                             ; wdd read 256 byte
F718  CD F77F                call  selcntlr       ; select the controller
F71B  3E 08                  ld    a,read         ; read sector command
F71D  CD F79E                call  taskout        ; send out to controller
F720                  wdrd1:
F720  CD F7DE                call  reqwait        ; wait for controller request
F723  C0                     ret   nz             ; no cntlr request then error (wd.rty)
F724  E6 08                  and   cdmsk          ; check for input status
F726  20 05                  jr    nz,wdsts       ; C/D- active, read status
F728  ED B2                  inir                 ; read 256 byte (one sector)
F72A  1D                     dec   e              ; sector counter down
F72B  20 F3                  jr    nz,wdrd1       ; no zero then read next sector
F72D                  wdsts:
F72D  CD F7B2                call  getstat        ; get completion status
F730  C9                     ret                  ; return status
                      ;
                      ;
F731  E1              parov0: pop   hl             ; pop parameter pointer
F732                  parower:
```

```
                                ; wdd i/o para overflow
F732    11 F696                 ld      de,parowmsg     ; D.E = par overflow message
F735    CD F562                 call    strout          ; print it
F738    3E 40                   ld      a,40h           ; 'Parameter Overflow'
F73A    C9                      ret                     ; return with error
                        ;
                        ;
F73B                    wdwrite:
                                ; wdd write 256 byte
F73B    CD F77F                 call    selcntlr        ; select the controller
F73E    3E 0A                   ld      a,write         ; write sector command
F740    CD F79E                 call    taskout         ; send out to controller
F743                    wdwt1:
F743    CD F7DE                 call    reqwait         ; wait for controller request
F746    C0                      ret     nz              ; no cntlr request then error
F747    E6 08                   and     cdmsk           ; check for input status
F749    20 E2                   jr      nz,wdsts        ; C/D- active, read status
F74B    ED B3                   otir                    ; write 256 byte (one sector)
F74D    1D                      dec     e               ; sector counter down
F74E    20 F3                   jr      nz,wdwt1        ; no zero then read next sector
F750    18 DB                   jr      wdsts           ; return with status
                        ;
                        ;
F752                    wdini:
                                ; initialize Drive Characteristics
F752    D3 B9                   out     (wport1),a      ; send out a reset pulse
                        ;
F754                    wddrdy:
F754    CD F792                 call    wtwdrdy         ; wdd ready ?
F757    C0                      ret     nz              ; error if not
                        ;
F758    CD F77F                 call    selcntlr        ; select the controller
F75B    3E 0C                   ld      a,initl         ; inizialize disk size command
F75D    CD F79E                 call    taskout         ; send out the command
F760    21 F68E                 ld      hl,initab       ; point to drive size tab.
F763    06 08                   ld      b,8             ; set up a byte counter
F765    CD F7DE                 call    reqwait         ; wait for controller request
F768    C0                      ret     nz              ; no zero then error
F769                    iniz00:
F769    7E                      ld      a,(hl)          ; get a byte to send out
F76A    D3 B8                   out     (wport0),a      ; send it to the controller
F76C    23                      inc     hl              ; bump the drive size tab. pointer
F76D    10 FA                   djnz    iniz00          ; decrement the byte count and
                                                        ; wait until all are output
F76F    CD F7B2                 call    getstat         ; get completion status
F772    C0                      ret     nz              ; ret if error completion
                                ;
F773    CD F77F                 call    selcntlr        ; select the controller
F776    3E 01                   ld      a,recal         ; recalibrate command code
F778    CD F79E                 call    taskout         ; send command to cntlr
F77B    CD F7B2                 call    getstat         ; get completion status
F77E    C9                      ret                     ; ret with status
                        ;
F77F                    selcntlr:
                                ; selects the default controller
F77F    DB B9                   in      a,(rport1)      ; read status port
F781    E6 02                   and     busymsk         ; mask busy bit
F783    20 FA                   jr      nz,selcntlr     ; loop if busy
```

```
F785  3E 01                   ld      a,1             ; cntrl default select code
F787  D3 B8                   out     (wport0),a      ; send it to trasparent latch
F789  D3 BA                   out     (wport2),a      ; generate a select strobe
F78B                    selc00:
F78B  DB B9                   in      a,(rport1)      ; get cntrl response
F78D  E6 02                   and     busymsk         ; isolate the busy mask
F78F  28 FA                   jr      z,selc00        ; wait for cntrl busy
F791  C9                      ret                     ; busy has arrived,exit
                        ;
F792                    wtwdrdy:
                              ; test for drive ready
F792  CD F77F                 call    selcntlr        ; select the controller
F795  3E 00                   ld      a,drvrdy        ; drive ready command
F797  CD F79E                 call    taskout         ; send out the command
F79A  CD F7B2                 call    getstat         ; get completion status
F79D  C9                      ret                     ; and ret to caller
                        ;
                        ;
F79E                    taskout:
                              ; send out the command contained in A register
F79E  E5                      push    hl              ; Save Ptr
F79F  21 004A                 ld      hl,task         ; Point to table
F7A2  77                      ld      (hl),a          ; store command
F7A3  16 06                   ld      d,6             ; six bytes
F7A5  CD F7DE                 call    reqwait         ; wait for request
F7A8  C0                      ret     nz              ; no, error
F7A9  7E                task1: ld     a,(hl)          ; get one byte
F7AA  D3 B8                   out     (wport0),a      ; send it
F7AC  23                      inc     hl              ; point to next
F7AD  15                      dec     d               ; dec counter
F7AE  20 F9                   jr      nz,task1        ; loop until zero
F7B0  E1                      pop     hl              ; restore
F7B1  C9                      ret                     ; cmd send
                        ;
F7B2                    getstat:
F7B2  CD F7DE                 call    reqwait         ; wait for request
F7B5  C0                      ret     nz              ; no, error
F7B6  DB B8                   in      a,(rport0)      ; get status byte
F7B8  57                      ld      d,a             ; save
F7B9  CD F7DE                 call    reqwait         ; wait for another
F7BC  C0                      ret     nz              ; no, error
F7BD  DB B8                   in      a,(rport0)      ; get fill byte
F7BF  7A                      ld      a,d             ; restore
F7C0  E6 02                   and     errmsk          ; only status bit
F7C2  C8                      ret     z               ; no error, return A=0
                              ;
                              ; Get Sense Status
                              ;
F7C3  CD F77F                 call    selcntlr        ; select the controller
F7C6  3E 03                   ld      a,sense         ; sense status command
F7C8  CD F79E                 call    taskout         ; send out to controller
F7CB  CD F7DE                 call    reqwait         ; wait for controller request
F7CE  C0                      ret     nz              ; no cntlr request then error
F7CF  DB B8                   in      a,(rport0)      ; get error code
F7D1  5F                      ld      e,a             ; save
                              rept    3               ; three bytes to discard
                              in      a,(rport0)      ; get one bytes
                              endm                    ;
```

```
F7D2  DB B8       +         in      a,(rport0)      ; get one bytes
F7D4  DB B8       +         in      a,(rport0)      ; get one bytes
F7D6  DB B8       +         in      a,(rport0)      ; get one bytes
F7D8  CD F7B2               call    getstat         ; get completion status
F7DB  7B                    ld      a,e             ; restore error code
F7DC  B7                    or      a               ; set flag
F7DD  C9                    ret                     ; return it
                      ;
                      ;
F7DE                  reqwait:
F7DE  C5                    push    bc              ; save  register
F7DF  D5                    push    de              ;
F7E0  06 08                 ld      b,8             ; set soft timer for aproax 15 Second
F7E2                  reqwt0:
F7E2  11 0000               ld      de,0            ;
F7E5                  reqwt1:
F7E5  DB B9                 in      a,(rport1)      ; get cntlr status bits
F7E7  4F                    ld      c,a             ; save on c
F7E8  E6 01                 and     reqmsk          ; only request bit (A=0 or A=1)
F7EA  20 09                 jr      nz,reqwtex      ; exit if (REQ=1)
F7EC  1B                    dec     de              ; timer 1 down
F7ED  7A                    ld      a,d             ; check for elapsed
F7EE  B3                    or      e               ;
F7EF  20 F4                 jr      nz,reqwt1       ; no, loop
F7F1  10 EF                 djnz    reqwt0          ; timer 2 down
F7F3  0E FF                 ld      c,0ffh          ; time out error (A=FF)
F7F5                  reqwtex:
F7F5  3D                    dec     a               ; A=A-1    (A=0  ,  A=FF)
                                                    ; set flag (Z=REQ, NZ=timeout)
F7F6  79                    ld      a,c             ; return status on a
F7F7  D1                    pop     de              ; restore
F7F8  C1                    pop     bc              ;       registers
F7F9  C9                    ret                     ; and ret
                      ;
                            else                    ; hard disk don't exist
                      ;
                      wdio:
                            ld      a,1             ; wdd i/o error
                            ret                     ;
                      ;
                      wdini:
                            ret                     ; return
                      ;
                            endif
                      ;
                      ;
                      ;
                            .dephase                ; end of this program
                            end     100h
```

Macros:

Symbols:

ADDREG	008C	BACKSP	0008	BEEP	008F	BELL	0007
BOOT	F0CC	BUSYBI	0001	BUSYMS	0002	CBEGL0	F4D7
CBEGLI	F4CC	CDBIT	0003	CDMSK	0008	CENDLI	F493
CENDSC	F4A0	CESCR0	F4A4	CESCR1	F4AA	CESCR2	F4AF
CESCR3	F4C9	CESCR4	F4B3	CHAROU	F3F2	CHR$04	F3D2
CHR$05	F493	CHR$06	F4A0	CHR$07	F3CE	CHR$08	F3E9
CHR$10	F3D9	CHR$11	F3DE	CHR$12	F49D	CHR$13	F4CC
CHR$14	F3E2	CHR$15	F3EC	CHR$19	F3B3	CHR$20	F3B7
CHR$21	F3BB	CHR$22	F3BF	CHR$23	F3C3	CHR$27	F3C7
CHR$TA	F383	CIN	F209	CINP00	F214	CINP11	F225
CKRDY	F0A9	CLRPFX	F35F	CNTCHA	F364	COMPFL	F02D
CONT0A	0082	CONT0B	0083	CONT1A	0086	CONT1B	0087
CONT2A	008A	CONT2B	008B	COUT	F233	COUT00	F259
COUT11	F24D	CR	000D	CSTS	F202	CURSOF	F425
CURSON	F433	CYLS	0168	DATA0A	0080	DATA0B	0081
DATA1A	0084	DATA1B	0085	DATA2A	0088	DATA2B	0089
DATREG	008D	DIVIDE	F4DC	DPBIPL	F02E	DRVEQU	F688
DRVRDY	0000	ENDMSG	0024	ENDVID	077F	ERR1GE	F623
ERR2GE	F633	ERRMSK	0002	FALSE	0000	FD0OTR	0040
FD1OTR	0041	FD2OTR	0042	FD3OTR	0043	FDBOOT	F113
FDBTMS	F15E	FDBTWT	F0B4	FDDBOT	F0C0	FDDCMD	00D0
FDDDAT	00D7	FDDELA	F63C	FDDERC	F5FB	FDDLCH	00D6
FDDRD	0088	FDDRES	0006	FDDRET	F613	FDDRST	00D0
FDDRT1	F614	FDDSEC	00D2	FDDSEK	0016	FDDSTS	00D0
FDDTAB	F68A	FDDTRK	00D1	FDDWT	00A8	FDIO	F573
FDIO00	F581	FDIOD	F572	FDIOEN	F5EB	FDIOS	F56F
FDRD00	F5C6	FDRD01	F5C9	FDREAD	F5BD	FDSEK0	F619
FDWRIT	F5D5	FDWT00	F5DE	FDWT01	F5E1	FFEED	000C
FORMAT	0004	GETSTA	F7B2	HARD	FFFF	HEADS	0004
HL.GT.	F4DD	HLADRS	0045	INIMSG	F036	INIT00	F514
INIT01	F520	INIT02	F534	INITAB	F68E	INITIA	F4E5
INITL	000C	INIZ00	F769	IO.01	F0FD	IOADD	0080
IOBIT	0004	IOERMS	F19A	IOERR	F0FA	IOMSK	0010
IPLDMA	1000	IPLOK	F0ED	IPLTES	F0D6	IX.PNT	0047
LF	000A	LOUT	F1F8	LSTS	F1F2	MD80VI	0000
MDCNT	0000	MODE0	000F	MODE1	004F	MODE2	008F
MODE3	00CF	MOVCUR	F344	MSGBIT	0002	MSGMSK	0004
NE80VI	FFFF	NECNT	FFFF	NEWDRV	F661	NOCHR$	F3CC
NOIPL	F0F5	NOIPLM	F1A9	NOUPDS	F48C	OUTDAT	F427
PAROV0	F731	PAROWE	F732	PAROWM	F696	PFX01	F28E
PFX02	F421	PFX03	F29F	PFX033	F2A8	PFX04	F2AD
PFX044	F2B6	PFX05	F2C9	PFX055	F2D3	PFX06	F2DD
PFX066	F2E5	PFX067	F2E8	PFX07	F317	PFX08	F2F5
PFX088	F2FD	PFX089	F300	PFX09	F324	PFX099	F329
PFX09A	F32C	PFX10	F2BB	PFX101	F2C4	PFXSET	F263
PFXTAB	F27A	PR@12.	F55A	PRINTA	F55D	RAM	0040
RAMDIA	00E0	RDCPOS	F549	READ	0008	RECAL	0001
RELCOM	F307	REQBIT	0000	REQMSK	0001	REQWAI	F7DE
REQWT0	F7E2	REQWT1	F7E5	REQWTE	F7F5	RESET	F061
REV	0030	ROM	F000	RPORT0	00B8	RPORT1	00B9
RPORT2	00BA	RPORT3	00BB	RST00	F06C	RSTCNT	0002
RTY00	F58E	RTY01	F590	RTYCNT	0003	SEEK	000B
SELC00	F78B	SELCNT	F77F	SENDO0	F400	SENDOU	F3FF
SENSE	0003	SETDUM	F42E	SETNEW	F1BD	SETNWC	F33B
SETPFX	F3C9	SPARE	0049	STACK	1000	STROUT	F562
STRVID	0000	SUBPOI	F3EC	TAB654	F53D	TASK	004A
TASK1	F7A9	TASKOU	F79F	TIMEOU	F657	TIMER	F0B6

```
TIMER1  F0B9    TRKOK   F5A7    TRKOK1  F5B5    TRUE    FFFF


Rom 4.8 for NE CP/M 2.2 with Hard-Disk  MACRO-80 3.36   17-Mar-80       PAGE    S-1
Copyright Studio Lg, Genova - Last rev 18/08/1984 17:30

UNIT    0044    UPDTCP  F442    UPDTDS  F460    UPDTUR  F450
UPDTVI  F43B    VERS    0048    VIDATR  008E    VIDCOM  F418
WAIT00  F646    WAIT01  F649    WAIT02  F65E    WAITCR  F107
WAITFD  F641    WAITKE  F08B    WDBOOT  F0CF    WDDRDY  F754
WDFDMS  F123    WDINI   F752    WDIO    F6B4    WDIO1   F6BC
WDIO2   F6D3    WDRD1   F720    WDREAD  F718    WDSTS   F72D
WDWRIT  F73B    WDWT1   F743    WPORT0  00B8    WPORT1  00B9
WPORT2  00BA    WPORT3  00BB    WRITE   000A    WTHDRD  F09F
WTWDRD  F792    XY1COM  F359    XYCOMP  F357


No  Fatal error(s)
```