$ 100 from NOE originally.

1

P.2 S V3 #1 COLOSSUE.

MEM LOCATION 0001-0002 for #31BA (14) identifies V.8.79

CONTAINS CRT HEADER 0123-013C

CONTAINS BASIC HEADER 014A-01A6

UA4 & UA5 - CONTAINS LISTING FOR -

LISTING FOR

SYSTEM SOFTWARE V8.79 & INTERCOLOR 3621 CC11 .V8.79-02

979008 01/15/80

SOME COMMENTS BY TREVOR TAYLOR

For

Intecolor 3621 and Compucolor II

BASIC HEADER in UA4
CRT HEADER in UA4
ESCAPES in UA4
50 Hz to 60Hz in UA4

Lending Period : 30 days
Return to :

Douglas A. Van Putte
18 Cross Bow Dr.
Rochester, NY 14624
CHIP Librarian

```
FCS>ASM SYS879.SRC;03

;;      SYSTEM SOFTWARE FOR INTECOLOR 3621 AND COMPUCOLOR II
;
;       Source File:     SYS879.SRC;03  V8.79-02
;       Last Modified:   January 15, 1980  JKP III
;       Copyright (C) 1978,1979,1980
;               by Intelligent Systems Corporation
;-----------------------------------------------------------------------
;;      **********[redacted]**********
                    Modifications and Revisions
;       SEPTEMBER 15, 1978
;               1.  SOURCE REARRANGED
;                       TABLES
;                       CRT
;                       FCS
;                       MICRO DISK HANDLER
;                       BASIC
;                       CRT AND BASIC HEADER MESSAGES MOVED
;                         INTO 0-3FF
;                       FCS  DUP  COMMAND HAS BEEN REMOVED
;                         HAS A JUMP TO INVALID COMMAND IN 0-3FF
;               2.  BUGS FIXED
;                       CRT - BLIND CURSOR - BLIND A7 PROBLEM
;                       FCS - LOAD DEFAULT ADDRESS  A000 => 8200
;               3.  IMPROVEMENTS
;                       MICRO DISK HANDLER - NO 1 SEC. WAIT ON READ
;               4   KNOWN BUGS
;                       CRT - BLIND CURSOR - WRAP AROUND PAST COL 80
;                             DOES NOT CHECK STATUS OF BLIND A7
;                       FCS - STACK SOMETIMES OVERFLOWS DURING THE
;                             LOADING OF .LDA FILES
;               5.  COMPUCOLOR II AND INTECOLOR 8021 COMBINED
;                       MESSAGES - UNIT NAMES ARE IN 0-3FF
;       MAY 22, 1979
;               1.  MICRO DISK HANDLER
```

```
;                       STEP CODES CORRECTED FOR PROPER PHASE
;                       COMPUTATIONS : BYTE FOR BYTE SUBSTITUTION
;
;               2.  USE NAMELESS INITIALIZATION MESSAGES
;
;               3.  MASK INITIALIZED FOR 50 HERTZ PARTS
;
;               4.  POWER-UP SEQUENCE IS REARRANGED FOR AVOID
;                       EXTRA CRT RESETS
;
;               5.  CONTROL-X BUG IS FIXED AND ESC C IS DISABLED
;
;               6.  4 ESC CODE JUMPS INTO EPROM AREA ARE ADDED
;
;               7.  CLOCK CHANGED TO RUN FROM 00:00:00 => 23:59:59
;
;
; 8.79      November 28, 1979
;
;               Released on ECN 2194:
;                 SCN-972040-01  972040  SYS879.SRC;01  V8.79
;                       JKP.III
;
;
; 8.79-01   November 29, 1979
;
;               In the micro disk handler, add fix to always
;                'step in' to a track.  Also in micro disk
;                handler, fix bugs in the handling of retry
;                counters.    MAM
;
;               Released on ECN 2195:
;                 SCN-972040-02  972040  SYS879.SRC;02  V8.79-01
;                       JKP III
;
;
; 8.79-02   January 15, 1980
;
;               In the micro disk handler, increase 'header' retry
;                count (HRTR) for improved performance with the
;                phase-locked loop speed control on the spindle
;                motor.  Improve the disk reset routine to avoid
;                unnecessary disk accesses; this is for the high
;                voltage shut-down.  Modify FCS input routine to
;                free 2 bytes for future use.    JKP III
;
;               Released on ECN 2206:
;                 SCN-972040-03  972040  SYS879.SRC;03  V8.79-02
;                       JKP III
;
;-----------------------------------------------------------------
;
;;      SYSTEM ORIGIN ADDRESSES
;

(0120)  SYSORG  EQU     0120H   ; ROM ORG FOR SYSTEM
```

```
(1F26)    BASORG  EQU     1F26H   ; ROM ORG FOR BASIC


(0020)    NOLIN   EQU     32      ; 32 LINES
(0040)    NOCHA   EQU     64      ; 64 CHARACTERS
(1000)    DSBUFS  EQU     NOLIN*NOCHA*2
(6000)    DSBUF   EQU     6000H   ; START ADDRESS OF DISPLAY FAST RAM
(7000)    X80     EQU     DSBUF+DSBUFS ; START OF DISPLAY SLOW RAM

          ; **************************************************************
          ;
          ;;      SYSTEM AND FCS ADDRESSES USED BY BASIC
          ;
A120 (6000)       ORG     6000H   ; BASIC ADDRESSES

6000 8918         DW      CMPDH
6002 8F18         DW      SUBHD
6004 7118         DW      MOVDH
6006 7A18         DW      MOVHD
6008 8318         DW      CMPHD
600A 4E19         DW      ADHLA
600C 5319         DW      ANHD
600E 5A19         DW      NEGH
6010 5B19         DW      NOTH
6012 6219         DW      ORHD
6014 6919         DW      XORHD
6016 7019         DW      SHLHD
6018 7A19         DW      SHRHD
601A 9819         DW      MULHD
601C B719         DW      DIVHD
601E 9618         DW      SPNOR
6020 D60A         DW      EMESS
6022 F20C         DW      WF2
6024 260D         DW      RF2
6026 3213         DW      CHDLR
6028 AD14         DW      PFSPC
602A BC10         DW      OPENX
602C 480B         DW      RESET
602E 2E13         DW      WR
6030 3113         DW      RD
6032 FC0C         DW      CLX

6034 4A01         DW      MBOOZ   ; DISK BASIC START-UP MESSAGE
6036 8E01         DW      MMEMO   ; MAXIMUM RAM AVAILABLE MESSAGE


          ; **************************************************************
          ;
          ;;      SYSTEM RAM AREAS
          ;
          ;       1.  RAM ORIGINS
          ;
          ;       2.  CRT RAM ALLOCATION
```

```
                ;
                ;       3.  FCS RAM ALLOCATION
                ;
                ;       4.  MICRO DISK HANDLER RAM ALLOCATION
                ;
                ;       5.  BASIC RAM ALLOCATION

(8042)  STACK   EQU     8042H   ; STACK FROM SCREEN TO HERE
(8046)  LINBF   EQU     8046H   ; BASIC LINE BUFFER
(80DE)  TRAM    EQU     80DEH   ; TEMP RAM START IN BASIC RAM
(80F0)  ORAM    EQU     80F0H   ; FCS RAM
(81AF)  CRTRAM  EQU     81AFH   ; CRT RAM

                ; ***************************************************************

                ;;      2.  CRT RAM ALLOCATION
                ;

6038 (81AF)             ORG     CRTRAM

81AF (0001)     XOUT0:  DS      1       ; CURRENT PHASE MICRO DRIVE 0
81B0 (0001)     XOUT1:  DS      1       ; CURRENT PHASE MICRO DRIVE 1
81B1 (0001)     CTRK0:  DS      1       ; CURRENT TRACK MICRO DRIVE 0
81B2 (0001)     CTRK1:  DS      1       ; CURRENT TRACK MICRO DRIVE 1
81B3 (0001)     DDRFLG: DS      1       ;* DISK DRIVE RESET FLAG (A5=RESET)  1/15/80
81B4 (0001)     AUCNT:  DS      1       ; SPARE
81B5 (0001)     CUCNT0: DS      1       ; USER COUNT MICRO DRIVE 0
81B6 (0001)     CUCNT1: DS      1       ; USER COUNT MICRO DRIVE 1
81B7 (0001)     PUP:    DS      1       ; POWER UP FLAG
81B8 (0001)     HEX:    DS      1       ; BINARY FRACTIONS OF A SECOND
81B9 (0001)             DS      1       ; SECONDS
81BA (0001)             DS      1       ; MINUTES
81BB (0001)             DS      1       ; HOURS (24)
81BC (0003)     DISPCK: DS      3       ; JMP TO DISPLAY CLOCK
81BF (0003)     ESCCRT: DS      3       ; CRT ESCAPE 30 JUMP
81C2 (0003)     OUTCRT: DS      3       ; OUTPUT TABLE 6 JUMP
81C5 (0003)     INPCRT: DS      3       ; FREE INPUT TABLE JUMP
81C8 (0003)     RST1J:  DS      3       ; TIMER 2 JUMP VECTOR

                ;;                      RAM IS CLEARED FROM HERE TO END (8200H)

81CB (0002)     VCRAD:  DS      2       ; VISIBLE CURSOR ADDRESS (+0=X, +1=Y)
81CD (0001)     ROLLN:  DS      1       ; ROLL COUNT (0=NO ROLL)
81CE (0002)     VFILL:  DS      2       ; VISIBLE FILL (+0=A7 BIT, +1=CCI)
81D0 (0002)     BFILL:  DS      2       ; BLIND FILL (+0=A7 BIT, +1=CCI)
81D2 (0002)     VHLAD:  DS      2       ; VISIBLE CURSOR HL ADDRESS
81D4 (0002)     BHLAD:  DS      2       ; BLIND CURSOR HL ADDRESS
81D6 (0001)     EXTBF:  DS      1
81D7 (0001)     SEC15:  DS      1       ; REPEAT KEY SCAN COUNTER
81D8 (0002)     PCRAD:  DS      2       ; PLOT CURSOR ADDRESS
81DA (0001)     PLOFL:  DS      1       ; CURRENT PLOT SUBMODE
81DB (0001)     PSTAT:  DS      1
```

```
81DC (0001)   ROLFL:   DS      1       ; ROLL FLAG => 0 = NO ROLL, 1 = ROLL
81DD (0001)   DUPLX:   DS      1       ; DUPLEX FLAG => 0 = LOCAL, - = FULL, + = HALF
81DE (0001)   THRUFL:  DS      1
81DF (0001)   KBDFL:   DS      1       ; KEYBOARD FLAG
81E0 (0001)   CMASK:   DS      1       ; CURRENT MASK REGISTER SETTING
81E1 (0001)   FCSFL:   DS      1       ; FILE CONTROL SYSTEM OUTPUT FLAG
81E2 (0001)   CRATE:   DS      1       ; CURRENT BAUD RATE SETTING
81E3 (0001)   INPFL:   DS      1       ; SERIAL INPUT FLAG
81E4 (0001)   LKC:     DS      1       ; LAST KEY CODE
81E5 (0001)   NKC:     DS      1       ; NEW KEY CODE
81E6 (0001)   COLFL:   DS      1       ; FLAG (FG/BG) 0=OFF, 1=ON
81E7 (0003)   JUMP:    DS      3       ; JUMP USED FOR CURSOR POSITIONING - LEFT, RIGHT, ETC
81EA (0001)   XTWO:    DS      1       ; PLOT MODE TEMPORARIES (7)
81EB (0001)   YTWO:    DS      1
81EC (0001)   XDATA:   DS      1
81ED (0001)   YDATA:   DS      1
81EE (0001)   ODDFL:   DS      1
81EF (0001)   XZERO:   DS      1
81F0 (0001)   YZERO:   DS      1
81F1 (0001)   BASFL:   DS      1       ; BASIC OUTPUT FLAG
81F2 (0001)   TEMP0:   DS      1
81F3 (0001)   TEMP1    DS      1
81F4 (0001)   TEMP2    DS      1
81F5 (0001)   TEMP3    DS      1
81F6 (0001)   TEMP4    DS      1
81F7 (0001)   TEMP5    DS      1
81F8 (0001)   OUTFL:   DS      1
81F9 (0002)   LOFL:    DS      2       ; SYSTEM OUTPUT FLAG
81FB (0002)   OUTHL:   DS      2
81FD (0001)   MS150:   DS      1       ; COUNTER FOR 150 MILLISECOND DELAY
81FE (0001)   KBCHA:   DS      1       ; KEYBOARD CHARACTER
81FF (0001)   KBRDY:   DS      1       ; KEYBOARD READY FLAG

              ; **********************************************************
              ;
              ;;       3.  FCS RAM ALLOCATION
              ;

8200 (8042)             ORG     STACK

8042 (0001)   SBC:     DS      1       ; SECTOR BYTE COUNT FOR DISK
8043 (0001)   CRC1:    DS      1       ; 1ST CRC BYTE FOR DISK
8044 (0001)   CRC2:    DS      1       ; 2ND CRC BYTE FOR DISK

8045 (8046)             ORG     LINBF

8046 (0001)             DS      1       ; FOR BASIC'S ","
8047 (003B)   BUFP:    DS      59      ; FCS LINE BUFFER
8082 (0027)   ZRAM:    DS      39      ; FCS STUFF / BASIC INPUT BUFFER

80A9 (80A9)             ORG     $       ; *** SHOULD EQUAL  x20A" IN BASIC ***

80A9 (8082)             ORG     ZRAM

8082 (0001)   ZFPB:    DS      1
```

```
8083 (0001)    ZFATR:   DS      1
8084 (0006)    ZFNAM:   DS      6
808A (0003)    ZFTYP:   DS      3
808D (0001)    ZFVER:   DS      1
808E (0002)    ZFSBK:   DS      2
8090 (0002)    ZFSIZ:   DS      2
8092 (0001)    ZFLBC:   DS      1
8093 (0002)    ZFLAD:   DS      2
8095 (0002)    ZFSAD:   DS      2
8097 (0001)             DS      1
8098 (0001)    ZFDBK:   DS      1
8099 (0001)    ZFDEN:   DS      1
809A (0002)    ZFAUX:   DS      2
809C (0002)    ZFHAN:   DS      2
809E (0001)    ZFFCN:   DS      1
809F (0001)    ZFDRV:   DS      1
80A0 (0002)    ZFBLK:   DS      2
80A2 (0002)    ZFBUF:   DS      2
80A4 (0002)    ZFXBC:   DS      2
80A6 (0002)    ZFPTR:   DS      2
80A8           ZFPBE:   ; END OF AUX. FPB, END OF BASIC INPUT BUFFER

80A8 (80F0)             ORG     ORAM

80F0 (0002)    DFDV:    DS      2       ;DEFAULT DEVICE (ASCII)
80F2 (0001)    DFUN:    DS      1       ;DEFAULT UNIT (ASCII)

80F3 (0002)    FPBP:    DS      2       ;FILE PARAMETER BLOCK POINTER
80F5 (0001)    OCODE:   DS      1       ;OPEN TYPE CODE
80F6 (0001)    OVERS:   DS      1       ;ORIGINAL VERSION

               ;;       SYSTEM FILE PARAMETER BLOCK ALLOCATION :

80F7 (0001)    FPB:     DS      1       ;OPEN TYPE CODE
80F8 (0001)    FATR:    DS      1       ;ATTRIBUTE BYTE
80F9 (0006)    FNAM:    DS      6       ;FILE NAME
80FF (0003)    FTYP:    DS      3       ;FILE TYPE
8102 (0001)    FVER:    DS      1       ;FILE VERSION NUMBER
8103 (0002)    FSBK:    DS      2       ;STARTING BLOCK NUMBER
8105 (0002)    FSIZ:    DS      2       ;NUMBER OF BLOCKS
8107 (0001)    FLBC:    DS      1       ;BYTE COUNT OF LAST BLOCK
8108 (0002)    FLAD:    DS      2       ;LOAD ADR. FOR "IMAGE" FILE
810A (0002)    FSAD:    DS      2       ;START ADR. FOR "IMAGE" FILE
810C (0001)             DS      1       ;SPARE
810D (0001)    FDBK:    DS      1       ;DIRECTORY BLOCK NUMBER
810E (0001)    FDEN:    DS      1       ;DIRECTORY ENTRY NUMBER
810F (0002)    FAUX:    DS      2       ;NEW FILE CLOSING SIZE, OR ...
                                        ;... AUX. BYTE COUNT FOR SEQUENTIAL ROUTINES
8111 (0002)    FHAN:    DS      2       ;HANDLER ADDRESS
8113 (0001)    FFCN:    DS      1       ;HANDLER FUNCTION CODE
8114 (0001)    FDRV:    DS      1       ;DRIVE NUMBER
8115 (0002)    FBLK:    DS      2       ;BLOCK NO. FOR TRANSFER
8117 (0002)    FBUF:    DS      2       ;BUFFER POINTER FOR TRANSFER
8119 (0002)    FXBC:    DS      2       ;BYTE COUNT FOR TRANSFER
811B (0002)    FPTR:    DS      2       ;BBUF PNTR FOR SEQUENTIAL ROUTINES
811D           FPBE:                    ;END OF SYSTEM FPB
```

DIRECTORY BUFFER

```
811D                    DBF:                    ;DIR BLOCK BUFFER
811D (0001)             DBLK:   DS      1       ;"THIS" DIR BLOCK NUMBER
811E (0001)             MDBLK:  DS      1       ;MAX. DIR BLOCK NUMBER
811F (007E)                     DS      126     ;REMAINDER OF 128. BYTE DIR BLOCK BUFFER
819D                    DBFE:                   ;END OF DIR BLOCK BUFFER

                        ; THE FOLLOWING 18. BYTES ARE THE DIR BLOCK BUFFER EXTENSION
                        ; USED BY CLOSE WHEN THE "FREE" ENTRY MOVES TO THE NEXT BLOCK:

819D (0002)             XFHAN:  DS      2       ;AUX. HANDLER ADDRESS
819F (0001)             XFFCN:  DS      1       ;AUX. HANDLER FUNCTION CODE
81A0 (0001)             XFDRV:  DS      1       ;AUX. DRIVE NUMBER
81A1 (0002)             XFBLK:  DS      2       ;AUX. BLOCK NUMBER
81A3 (0002)             XFBUF:  DS      2       ;AUX. BUFFER POINTER
81A5 (0002)             XFXBC:  DS      2       ;AUX. BYTE COUNT
81A7 (0004)                     DS      4       ; *** USED BY COPY ***
81AB (0002)             TMP1:   DS      2       ; USED BY COPY & MAYBE OTHERS ?
81AD (0002)                     DS      2

81AF (81AF)                     ORG     $       ; *** SHOULD NOT BE PAST "CRTRAM" ***


                        ; ***************************************************************
                        ;
                        ;;      4.  MICRO DISK HANDLER RAM ALLOCATION
                        ;

                        ;       MICRO DISK HANDLER "TEMP" RAM ALLOCATION :

81AF (80DE)                     ORG     TRAM

80DE (0002)             TEMPHL: DS      2       ;* FREE FOR FUTURE USE  1/15/80  JKP III
80E0 (0001)             BRTRY:  DS      1       ; BLOCK RETRY COUNTER
80E1 (0001)             RFLG:   DS      1       ; "RESTORE" FLAG / COUNTER
80E2 (0001)             CRTRY:  DS      1       ; "CHUNK" RETRY COUNTER
80E3 (0002)             OBC:    DS      2       ; OLD BYTE COUNT

80E5 (0001)             TFCN:   DS      1       ; FUNCTION CODE
80E6 (0001)             TDRV:   DS      1       ; DRIVE NUMBER
80E7 (0002)             TBLK:   DS      2       ; BLOCK NUMBER
80E9 (0002)             TMEM:   DS      2       ; MEMORY BUFFER POINTER
80EB (0002)             TBC:    DS      2       ; BYTE COUNT

80ED (0001)             OSEC:   DS      1       ; OLD SECTOR NUMBER
80EE (0001)             SEC:    DS      1       ; SECTOR NUMBER
80EF (0001)             TRK:    DS      1       ; TRACK NUMBER

80F0 (80F0)                     ORG     $       ; *** SHOULD NOT BE PAST "ORAM" ***


                        ; ***************************************************************
                        ;
                        ;;      5.  BASIC RAM ALLOCATION

                        ;       SEE SECTION 1 FOR BASIC LINE BUFFER AND RAM ALLOCATION
```

```
;  ************************************************************************
;
;;      INTECOLOR 3621 & COMPUCOLOR II SOFTWARE V8.79-02
;       32 LINES BY 64 CHARACTERS
;       COPYRIGHT (C) 1974,1975,1976,1977,1978,1979,1980
;        BY INTELLIGENT SYSTEMS CORPORATION
;
;       PROGRAMMED BY CHARLES A. MUENCH, ET AL.
;       8080 CPU CLOCK=17.9712/9=1.9968 MHZ
;       MIN TIME INTERVAL=16X.5008=8.0128 US
;       MIN COUNT FOR INTERVAL TIMERS=64.1 US
;       15.6 COUNTS = 1 MS
;       78 COUNTS = 5 MS
;       156 COUNTS = 10 MS
;       195 COUNTS = 12.5 MS
;       255 COUNTS = 16.346 MS
;       195 COUNTS X 8 = 100 MS
;                  X 40 = .5 SEC.
;                  X 80 = 1 SEC.
;                  X 160 = 2 SEC.
;                  X 240 = 3 SEC.

;;      PORT ASSIGNMENTS

(0000)  MFIOA   EQU     0
(0000)  RXBUF   EQU     MFIOA
(0001)  EXTIN   EQU     MFIOA+1
(0002)  RSTBF   EQU     MFIOA+2
(0003)  STAT5   EQU     MFIOA+3
(0004)  COMND   EQU     MFIOA+4
(0005)  BAUD    EQU     MFIOA+5
(0006)  TXBUF   EQU     MFIOA+6
(0007)  EXTOT   EQU     MFIOA+7
(0008)  MASK    EQU     MFIOA+8
(0009)  TIME1   EQU     MFIOA+9
(000A)  TIME2   EQU     MFIOA+10
(000B)  TIME3   EQU     MFIOA+11
(000C)  TIME4   EQU     MFIOA+12
(000D)  TIME5   EQU     MFIOA+13

(005C)  KAP0    EQU     05CH
(0079)  KAP1    EQU     079H
(003B)  KAP2    EQU     03BH
(009F)  KAP3    EQU     09FH
(0021)  KAP4    EQU     021H    ; 50 HERTZ  , 006H = 60 HERTZ
(000C)  KAP5    EQU     00CH

(0060)  CRTCHIP EQU     60H
(0060)  CRT0    EQU     CRTCHIP+0
(0061)  CRT1    EQU     CRTCHIP+1
```

```
                (0062)  CRT2    EQU     CRTCHIP+2
                (0063)  CRT3    EQU     CRTCHIP+3
                (0064)  CRT4    EQU     CRTCHIP+4
                (0065)  CRT5    EQU     CRTCHIP+5
                (0066)  CRT6    EQU     CRTCHIP+6
                (0076)  ROLDA   EQU     CRTCHIP+16+6    ; "SLOW" ADDRESSING
                (007C)  CRXDA   EQU     CRTCHIP+16+12   ; PREVENTS DG FROM
                (007D)  CRYDA   EQU     CRTCHIP+16+13   ; BEING STARVED
                (006A)  STOPIT  EQU     CRTCHIP+10
                (006E)  STARTIT EQU     CRTCHIP+14

MEM VERSIONS    (1F26)  BASICW  EQU     BASORG  ; BASIC ENTRY POINTS AND ROUTINES FOR TABLES
0001 - 0002     (1F2C)  BASICE  EQU     BASICW+06H
The # 01BA(H)   (1F38)  BASICI  EQU     BASICW+12H
identifies V.8.79 (1F3B)  BASEX   EQU     BASICW+15H
                (1F3E)  AUTOX   EQU     BASICW+18H

                        ;*****************************************
                        ;;      START OF SYSTEM SOFTWARE
START OF ROM
                80F0 (0000)             ORG     0
                0000            TIMX1:
                0000 C3BA01     START:  JMP     STARX
                0003 381F       INITAD: DW      BASICI  ; BASIC INITIAL ADDRESS
                0005 (0001)             DS      1
UM4             0006 00         MODE:   DB      0       ; 0=LOCAL, 1=HALF, 2=FULL
                0007 01         PROLL:  DB      1       ; 0=PAGE   1=ROLL

                0008 (0008)             ORG     08H
                0008 C3C881     TIMX2:  JMP     RST1J
                000B 8D00       AESCTB: DW      ESCTB   ; ADDRESS OF ESCAPE TABLE
                000D 0D         CARET:  DB      0DH
                000E B0         CHDEL:  DB      -80
                000F (0001)             DS      1

                0010 (0010)             ORG     10H
                0010 C35302     XINTR:  JMP     UPDATE  ; CLOCK UPDATE
                0013 (0001)             DS      1
                0014 08         DELTA:  DB      8       ; NUMERATOR OF TIME
                0015 0F         TAU:    DB      15      ; DENOMINATOR OF TIME
                0016 C0         RATEA:  DB      0C0H    ; 9600 BAUD + 1 STOP BIT
                0017 08         RCOMD:  DB      08H     ; BAUD  1X=08H    8X=18H

                0018 (0018)             ORG     18H
                0018 C30409     TIMX3:  JMP     TIM3X
                001B F4         TPROG:  DB      -12     ; 150MS
                001C 05         CHTIM:  DB      5       ; SCANS PER CHARACTER FOR REPEAT
                001D FF         KEDEL:  DB      255     ; KEYBOARD SCAN RATE * 64 MICROSECONDS
                001E FF81       KERDY:  DW      KBRDY

                0020 (0020)             ORG     20H
                0020 C3BD03     RXSER:  JMP     ISERL
                0023 40                 DB      64      ; NUMBER OF CHARACTERS
                0024 C30B18     KEYTST: JMP     RTST
                0027 14         STEPCD: DB      STPTIM  ; MICRO DISK STEP RATE

                0028 (0028)             ORG     28H
                0028 C3B804     TXSER:  JMP     OSERL
```

```
002B C39803    KEYCO:   JMP      KBREP     ; PROCESS KEY CODE
002E 04        CDMK:    DB       4         ; 4 FOR PROPER CLOCK OPERATION
002F A0        RATEB:   DB       0A0H      ; 4800 BAUD + 1 STOP BIT

0030 (0030)             ORG      30H
0030 C32805    TIMX4:   JMP      TIM4X
0033 C3B403    BASOUT:  JMP      CRTUBE    ; PROCESS CHAR WITH BASIC FLAG
0036 4280      ACRTSP:  DW       STACK     ; ADDRESS OF TOP OF STACK FOR CRT MODE


0038 (0038)             ORG      38H
0038           TIMX5:
0038 C3B601    BEGEX:   JMP      BEGIN
003B C30D05    BREAK:   JMP      BRAKE     ; PROCESS BREAK CODE
003E FEB1      KCHAR:   DW       KBCHA


                        ;*****************************************************
                        ;;       DISK HANDLER VALUES AND VECTORS

0040 4344      IDEV:    DB       'CD'      ;INITIAL DEFAULT DEVICE
0042 30        IUNT:    DB       '0'       ;INITIAL DEFAULT UNIT

0043 C3C11A    HDVCT:   JMP      CDHD      ;JMP TO HANDLER
0046 4344      CDNM:    DB       'CD'      ;DEVICE NAME
0048 02        CDNU:    DB       2         ;NUMBER OF UNITS
0049 0A        CDSEC:   DB       NSEC
004A 50        CDK:     DB       UPTIM/STPTIM
004B (0002)             DS       2

004D FF                 DB       255       ; SPACE FOR MD AND DF HANDLERS
004E (0009)             DS       9

0057 FF                 DB       255
0058 (0005)             DS       5

                        ;*********************************************
                        ;;       CONTROL CHARACTER TABLE

005D 53043E1F  TBL00:   DW       FREE,AUTOX        ; @,A / 0,1
0061 F7075204           DW       POTON,CRSXY       ; B,C / 2,3
0065 53045304           DW       FREE,FREE         ; D,E / 4,5
0069 52041A05           DW       CCI,BEL           ; F,G / 6,7
006D B702FA02           DW       HOME,TAB          ; H,I / 8,9
0071 06034305           DW       LF,ELINE          ; J,K / 10,11
0075 7F02BB02           DW       ERASE,CR          ; L,M / 12,13
0079 31038F03           DW       A7ON,BA7OF        ; N,O / 14,15
007D ED04FE02  TBL24:   DW       PRINT,CRSRT       ; X,Y / 24,25
0081 39035204           DW       CRSLT,ESCAP       ; Z,ESC / 26,27
0085 3F038403           DW       CRSUP,COMOF       ; \,] / 28,29
0089 83038803           DW       COMON,BLINK       ; ^,_ / 30,31

                        ;;       ESCAPE SEQUENCE TABLE

014(D) 008D 53045204 ESCTB:   DW       FREE,BLIND        ; @,A / 0,1
0091 46045304           DW       CHPLO,FREE        ; B,C / 2,3   DISABLE ESC C <<8.79>>
0095 FF162C1F           DW       ESCD,BASICE       ; D,E / 4,5
0099 A504F416           DW       FULL,ESCQ         ; F,Q / 6,7
```

```
0201 (D)
        009D A3040090           DW      HALF,9000H      ; H,I / 8,9       ; ESC I => 9000H
        00A1 E704DC04           DW      DOWN,ROLL       ; J,K / 10,11
        00A5 A6044704           DW      LOCAL,TMODE     ; L,M / 12,13
        00A9 47045304           DW      TMODE,FREE      ; N,O / 14,15
        00AD 00405304           DW      04000H,FREE     ; P,Q / 16,17     ; ESC P => 4000H
        00B1 520400A0           DW      BRATE,0A000H    ; R,S / 18,19     ; ESC S => 0A000H
        00B5 00825304           DW      8200H,FREE      ; T,U / 20,21     ; ESC T => 8200H
        00B9 5304261F           DW      FREE,BASICW     ; V,W / 22,23
        00BD DD045204           DW      PAGE,TEST       ; X,Y / 24,25
        00C1 00485304           DW      04800H,VISIB    ; Z,[ / 26,27     ; ESC Z => 4800H
        00C5 00500058           DW      05000H,05800H   ; \,] / 28,29     ; ESC \ => 5000H
                                                                          ; ESC ] => 5800H
0201 (D) 00C9 BF81A701          DW      ESCCRT,CRTMO    ; ^,_ / 30,31
                  (USER)

                        ;;      OUTPUT FLAG TABLE

        00CD B004AA04 OUTBL:    DW      BEGOT,KEYOT     ; 0,1
        00D1 61056905           DW    ⊗ LTPEN,XYMIT     ; 2,3
        00D5 9C05A505           DW      LINE,CARR       ; 4,5
        00D9 C281AD05           DW      OUTCRT,CRET     ; 6,7

                        ;;      INPUT FLAG TABLE

        00DD 25040504 INPTB:    DW      SVCHA,SBCHA     ; 0,1
        00E1 09075404           DW      PLOTX,CURSO     ; 2,3
        00E5 61086804           DW      CPLOX,VCRSY     ; 4,5
        00E9 4A047304           DW      CCIX,BCRSX      ; 6,7
        00ED 80044E04           DW      BCRSY,BCCIX     ; 8,9
        00F1 C581C581           DW      INPCRT,INPCRT   ; 10,11
        00F5 53043717           DW      FREEX,FCSX      ; 12,13
        00F9 F9175304           DW      S1OUT,FREEX     ; 14,15
        00FD C581C581           DW      INPCRT,INPCRT   ; 16,17
        0101 C604C581           DW      BRATX,INPCRT    ; 18,19
        0105 C581C581           DW      INPCRT,INPCRT   ; 20,21
        0109 C5813B1F           DW      INPCRT,BASEX    ; 22,23
        010D C5817A02           DW      INPCRT,TESTX    ; 24,25
        0111 C5810105           DW      INPCRT,ESCAX    ; 26,27
        0115 C581C581           DW      INPCRT,INPCRT   ; 28,29
        0119 C581C581           DW      INPCRT,INPCRT   ; 30,31

        011D (0120)             ORG     SYSORG

        0120 C3AD0A   DUP00:    JMP     IVC     ; FCS  DUPLICATE ROUTINE
                                                ;    DISABLED - INVALID COMMAND
                                                ;    WITH SECOND CD DISK JUMP INTO EPROM

                        ;;      SPECIAL HEADER MESSAGES FOR CRT MODE AND BASIC

0291 (D) 0123 06020B1D /CRTMSG: DB      6,2,11,29
        0127 43525420           DB      'CRT MODE ',22,'V8.79-02',18,10,11,239
        012B 4D4F4445
        012F 20165638
        0133 2E37392D
        0137 3032120A
0315 (D) 013B 0BEF                       extra to ...

        013D (014A)             ORG     0014AH
```

FITTED TO 8.79 (I) 10/85 (UA4)

ESC U -- 5400(H) | PACK SMALLER ... LIKE ... INTO ROM.
ESC V -- 5C00(H)
ESC O -- C000(H) – HIGH MEMORY
ESC Q -- E000(H) – HIGH MEMORY
ESC @ -- 82A0(H) – CTA COMPILER

ESC c -- SPARE

8·79 II still has original jumps.

```
0330(D)  014A 06010E0B MBOO2:   DB      6,1,14,11
         014E 4449534B          DB      'DISK BASIC'
         0152 20424153
         0156 4943
         0158 1D162056          DB      29,22,' V8.79 ',19,'COPYRIGHT (C) 1979',16,' BY '
         015C 382E3739
         0160 2013434F
         0164 50595249
         0168 47485420
         016C 28432920
         0170 31393739
         0174 10204259
         0178 20
         0179 494E5445          DB      'INTELLIGENT SYSTEMS'
         017D 4C4C4947
         0181 454E5420
         0185 53595354
         0189 454D53
0396(D)  018C 0D0A              DB      13,10
         018E 124D4158 MMEMO:   DB      18,'MAXIMUM RAM AVAILABLE ?',0
         0192 494D554D
         0196 2052414D
         019A 20415641
         019E 494C4142
         01A2 4C45203F
0422(D)  01A6 00

         01A7 212301    CRTMO:  LXI     H,CRTMSG
         01AA CD2A18            CALL    OSTR
         01AD 2A3600            LHLD    ACRTSP
         01B0 F9                SPHL
         01B1 213800            LXI     H,BEGEX
         01B4 E5                PUSH    H
         01B5 E9                PCHL

                        ;;      BEGEX LOOP TIME ,NO WAIT = 33 US
                        ;        WITH PROM WAIT = 40.5 US

         01B6 FB        BEGIN:  EI
         01B7 C33800            JMP     BEGEX

         01BA 113800    STARX:  LXI     D,BEGEX ; SET RETURN ON STACK
         01BD 2A3600            LHLD    ACRTSP  ; SET UP STACK ADRESS
         01C0 F9                SPHL
         01C1 D5                PUSH    D

                                                ; <<8.79>> CRT RESET MOVED FROM HERE

         01C2 3E03              MVI     A,3
         01C4 D304              OUT     COMND   ;RESET MFIOA
         01C6 3AB781            LDA     PUP     ; IS THIS REAL POWERUP ?
         01C9 FE97              CPI     97H
         01CB C2DF01            JNZ     PWRUP   ; YES !
         01CE 3E80              MVI     A,080H  ; IS THIS USER RESET (COMMAND) ?
         01D0 D307              OUT     EXTOT
         01D2 DB01              IN      EXTIN
```

SMALL CHAR

```
06 01 0F 08
44 49 53 4B
20 42 41 53
49 43
10 16 20 56
38 2E 37 39
20 13 43 4F
50 59 52 49
47 48 54 20
28 43 29 20
31 39 37 39
16 20 50 45
54 45 52 20
47 45 1F 52
47 45 20 53
54 41 4E 44
15 16 30 20
0D 0A
```

CYAN → 16 20 50 45

my name

MODIFIED HEADER TO SMALL CHAR. & NAME ADDED 6/85. 8.79 [&]

```
01D4 E630              ANI     30H
01D6 2ACB00            LHLD    OUTBL-2 ; SETUP CRTMODE AS EXIT POINT
01D9 C20F02            JNZ     CRTSET  ; NO RESET TO CRT MODE
01DC C3EB01            JMP     INIBAS  ; YES, INITIALIZE BASIC ONLY

01DF 21B881  PWRUP:    LXI     H,HEX
01E2 0604              MVI     B,4
01E4 3600    PWRUP1:   MVI     M,0
01E6 23                INX     H
01E7 05                DCR     B
01E8 C2E401            JNZ     PWRUP1
01EB         INIBAS:                   ; <<8.79>> RESET CRT CHIP HERE

01EB D36E              OUT     STARTIT ; START CHIP IN CASE
01ED D36A              OUT     STOPIT  ; RESET CRT CHIP

01EF 3E5C              MVI     A,KAPO  ; COPY PROM PARAMETERS
01F1 D360              OUT     CRT0    ; INTO CRT REGISTERS
01F3 3E79              MVI     A,KAP1
01F5 D361              OUT     CRT1
01F7 3E3B              MVI     A,KAP2
01F9 D362              OUT     CRT2
01FB 3E9F              MVI     A,KAP3
01FD D363              OUT     CRT3
01FF 3E21              MVI     A,KAP4
0201 D364              OUT     CRT4
0203 3E0C              MVI     A,KAP5
0205 D365              OUT     CRT5

0207 AF                XRA     A
0208 D366              OUT     CRT6
020A D36E              OUT     STARTIT ; RESTART CRT CHIP

020C 2A0300            LHLD    INITAD

020F E5      CRTSET:   PUSH    H       ; SAVE EXIT POINT
0210 21CB81            LXI     H,VCRAD
0213 AF                XRA     A
0214 77      CRTST1:   MOV     M,A
0215 2C                INR     L
0216 C21402            JNZ     CRTST1
0219 2EE7              MVI     L,JUMP AND 255
021B 36C3              MVI     M,0C3H
021D 2ECF              MVI     L,VFILL+1 AND 255
021F 3607              MVI     M,7
0221 2ED1              MVI     L,BFILL+1 AND 255
0223 34                INR     M
0224 3A0700            LDA     PROLL
0227 CDDD04            CALL    PAGE
022A CD7F02            CALL    ERASE
022D CD3705            CALL    TIM4Y
0230 3A1600            LDA     RATEA
0233 D305              OUT     BAUD
0235 32E281            STA     CRATE
0238 D30B              OUT     TIME3
023A 3A0600            LDA     MODE
023D 32DD81            STA     DUPLX
0240 CDB004            CALL    BEGOT
```

```
0243 C3480B              JMP     RESET   ; DISPATCH TO EXIT POINT

0246 2ACE81              LHLD    VFILL
0249 EB                  XCHG
024A 2ACB81    GETBC:    LHLD    VCRAD
024D 4D                  MOV     C,L
024E 44                  MOV     B,H
024F 2AD281              LHLD    VHLAD
0252 C9                  RET

0253 CD2E0A    UPDATE:   CALL    SAVE    ; INTERRUPT ROUTINE
0256 2A1400              LHLD    DELTA   ; E <== DELTA, D <== TAU
0259 EB                  XCHG
025A 21B881              LXI     H,HEX   ; UPDATE FRACTION
025D 7E                  MOV     A,M     ; SUBTRACT DELTA
025E 93                  SUB     E
025F 77                  MOV     M,A     ; STORE NEW FRACTION
0260 D0                  RNC             ; RETURN IF NO CARRY
0261 82                  ADD     D       ; ADD TAU
0262 77                  MOV     M,A     ; REPLACE NEW FRACTION
0263 23                  INX     H       ; POINT AT SECONDS
0264 34                  INR     M       ; INCREMENT SECONDS
0265 7E                  MOV     A,M     ; AND LOAD THEM
0266 D63C                SUI     60      ; COMPARE TO 60
0268 D8                  RC              ; SECONDS ARE GOOD : EXIT
0269 77                  MOV     M,A     ; MAPS 60 TO 0
026A 23                  INX     H       ; POINT AT MINUTES
026B 34                  INR     M       ; INCREMENT MINUTES
026C 7E                  MOV     A,M     ; AND LOAD THEM
026D D63C                SUI     60      ; COMPARE TO 60
026F D8                  RC              ; MINUTES ARE GOOD : EXIT
0270 77                  MOV     M,A     ; MAPS 60 TO 0
0271 23                  INX     H       ; POINT AT HOURS
0272 34                  INR     M       ; INCREMENT HOURS
0273 7E                  MOV     A,M     ; AND LOAD THEM
0274 D618                SUI     24      ; COMPARE TO 24          <<8.79>>
0276 D8                  RC              ; HOURS ARE GOOD : EXIT
0277 00                  NOP             ;                        <<8.79>>
0278 77                  MOV     M,A     ; MAPS 24 TO 0
0279 C9                  RET             ; FIX HOURS AND EXIT ANYWAY

027A 77        TESTX:    MOV     M,A
027B 7B                  MOV     A,E
027C C38402              JMP     EARS+2
027F CDAD02    ERASE:    CALL    NOROL
0282 3E20      EARS:     MVI     A,20H   ;SPACE
0284 210070              LXI     H,X80
0287 0600                MVI     B,0     ;NOLIN*NOCHA/8
0289 E5                  PUSH    H
028A 2ACE81              LHLD    VFILL
028D EB                  XCHG
028E B3                  ORA     E
028F 5F                  MOV     E,A
0290 210000              LXI     H,0
0293 39                  DAD     SP
0294 22F281              SHLD    TEMPO
0297 E1                  POP     H
0298 F3                  DI
```

MEM LOCATIONS	20D	114D
0257	14	change to 72
0258	00	" " 10 ← 16D

WINDER ERR… MOD TO CHAN… 60HZ to 50H… 8-79 ⊕ ONLY 10/85

why?

```
0299 F9            SPHL
029A D5     ERALP: PUSH    D
029B D5            PUSH    D
029C D5            PUSH    D
029D D5            PUSH    D
029E D5            PUSH    D
029F D5            PUSH    D
02A0 D5            PUSH    D
02A1 D5            PUSH    D
02A2 05            DCR     B
02A3 C29A02        JNZ     ERALP
02A6 2AF281        LHLD    TEMPO
02A9 F9            SPHL
02AA FB            EI
02AB E1            POP     H
02AC C9            RET
02AD 210070 NOROL: LXI     H,X80
02B0 22D481        SHLD    BHLAD
02B3 AF            XRA     A
02B4 32CD81        STA     ROLLN
02B7 21CD81 HOME:  LXI     H,ROLLN
02BA 46            MOV     B,M
02BB 4F     CR:    MOV     C,A
02BC CDE002 SVCRS: CALL    LN5X
02BF 22D281        SHLD    VHLAD
02C2 21CB81        LXI     H,VCRAD
02C5 71            MOV     M,C
02C6 23            INX     H
02C7 70            MOV     M,B
02C8 23            INX     H
02C9 7E            MOV     A,M
02CA 3D            DCR     A
02CB E61F          ANI     31
02CD D376          OUT     ROLDA
02CF 78            MOV     A,B
02D0 D37D          OUT     CRYDA
02D2 79            MOV     A,C
02D3 D37C          OUT     CRXDA
02D5 C9            RET
02D6 CD9A04 CRSY:  CALL    TESTB-2
02D9 3ACD81        LDA     ROLLN
02DC 80            ADD     B
02DD CD9C04        CALL    TESTB
02E0 3ACE81 LN5X:  LDA     VFILL
02E3 07            RLC
02E4 B0            ORA     B
02E5 47            MOV     B,A
02E6 78            MOV     A,B
02E7 07     LN5Y:  RLC
02E8 07            RLC
                   ;       ADD     B       ;A=5B
02E9 218003        LXI     H,X80/(32)
02EC 85            ADD     L
02ED 6F            MOV     L,A
02EE 29            DAD     H
02EF 29            DAD     H
02F0 29            DAD     H
02F1 29            DAD     H
```

```
02F2 29              DAD     H
02F3 79              MOV     A,C
02F4 07              RLC
02F5 85              ADD     L
02F6 6F              MOV     L,A
02F7 D0              RNC
02F8 24              INR     H
02F9 C9              RET
02FA 3E07    TAB:    MVI     A,7
02FC B1              ORA     C
02FD 4F              MOV     C,A
02FE 0C      CRSRT:  INR     C
02FF 3EC0            MVI     A,-NOCHA
0301 81              ADD     C
0302 D2BC02          JNC     SVCRS
0305 4F              MOV     C,A
0306 04      LF:     INR     B
0307 CD9D04          CALL    TESTB+1
030A 3ADC81  IROLL:  LDA     ROLFL
030D A7              ANA     A
030E CABC02          JZ      SVCRS
0311 21CD81          LXI     H,ROLLN
0314 78              MOV     A,B
0315 AE              XRA     M
0316 C2BC02          JNZ     SVCRS
0319 34              INR     M
031A 34              INR     M
031B 3EE0            MVI     A,-NOLIN
031D CD7C04          CALL    TESTC+6
0320 23              INX     H
0321 4E              MOV     C,M
0322 C5              PUSH    B
0323 AF              XRA     A
0324 77              MOV     M,A
0325 CD4F05          CALL    EARLN
0328 018000          LXI     B,NOCHA*2
032B CD5805          CALL    EARLN+9
032E F1              POP     PSW
032F 47              MOV     B,A
0330 F0              RP
0331 21CE81  A7ON:   LXI     H,VFILL
0334 3680            MVI     M,80H
0336 C3BC02          JMP     SVCRS
0339 0D      CRSLT:  DCR     C
033A F2BC02          JP      SVCRS
033D 0E3F            MVI     C,NOCHA-1
033F 3ACE81  CRSUP:  LDA     VFILL
0342 A7              ANA     A
0343 F24703          JP      CRSUP1
0346 05              DCR     B
0347 05      CRSUP1: DCR     B
0348 F2BC02          JP      SVCRS
034B 061F            MVI     B,NOLIN-1
034D C3BC02          JMP     SVCRS
0350 E3      COLOR:  XTHL
0351 CD4A02          CALL    GETBC
0354 5F              MOV     E,A
0355 215D00          LXI     H,TBL00
```

```
0358 FE10              CPI     16
035A DADF03            JC      CODE
035D FE18              CPI     24
035F 214D00            LXI     H,TBL24-48
0362 D2DF03            JNC     CODE
0365 E1                POP     H
0366 2EE6              MVI     L,COLFL AND 255
0368 CD7503            CALL    COLW
036B CA7103            JZ      BKCOL
036E 3EF8              MVI     A,0F8H
0370 50                MOV     D,B
0371 A6        BKCOL:  ANA     M
0372 B2                ORA     D
0373 77                MOV     M,A
0374 C9                RET
0375 E607      COLW:   ANI     7
0377 47                MOV     B,A
0378 07                RLC
0379 07                RLC
037A 07                RLC
037B 57                MOV     D,A
037C 6E                MOV     L,M
037D 2D                DCR     L
037E 2ECF              MVI     L,VFILL+1 AND 255
0380 3EC7              MVI     A,0C7H
0382 C9                RET
0383 3C        COMON:  INR     A
0384 2EE6      COMOF:  MVI     L,COLFL AND 0FFH
0386 77                MOV     M,A
0387 C9                RET
0388 3E40      BLINK:  MVI     A,40H
038A 2ECF              MVI     L,VFILL+1 AND 255
038C B6                ORA     M
038D 77                MOV     M,A
038E C9                RET
038F 2ECE      BA70F:  MVI     L,VFILL AND 255
0391 77                MOV     M,A
0392 3EBF              MVI     A,0BFH
0394 2C                INR     L
0395 A6                ANA     M
0396 77                MOV     M,A
0397 C9                RET

0398 5F        KBREP:  MOV     E,A     ;PUT CHAR IN E
0399 3ADD81            LDA     DUPLX
039C A7                ANA     A       ;LOCAL=0
039D 3EDF              MVI     A,0DFH  ;T5,T4,-,RX,T3,RTC,T2,T1
039F 21F881            LXI     H,OUTFL
03A2 CAA903            JZ      KBR1    ;HALF=1
03A5 3601              MVI     M,1
03A7 3EF7              MVI     A,0F7H  ;T5,T4,TX,RX,-,RTC,T2,T1
03A9 D308      KBR1:   OUT     MASK
03AB 32E081            STA     CMASK   ;SAVE CURRENT MASK SETTING
03AE 2EDF              MVI     L,KBDFL AND 255
03B0 F8                RM              ;FULL=-1
03B1 C3DA03            JMP     ISERX

03B4 CD2E0A    CRTUBE: CALL    SAVE    ;SAVE ALL REGS.
```

```
03B7 21F181             LXI     H,BASFL
03BA C3D503             JMP     PROCES

                ;;      ISERL TO INPTB TIME = 103 US, NO WAIT
                ;       WITH PROM WAIT = 122.5 US

03BD CD2E0A     ISERL:  CALL    SAVE    ;SAVE ALL REGISTERS
03C0 21FF81             LXI     H,KBRDY
03C3 3650               MVI     M,50H
03C5 DB03               IN      STAT5
03C7 E601               ANI     1
03C9 C0                 RNZ
03CA 3680               MVI     M,80H
03CC 2B                 DCX     H
03CD DB00               IN      RXBUF
03CF 77                 MOV     M,A
03D0 DB00               IN      RXBUF
03D2 21E381             LXI     H,INPFL
03D5 5F         PROCES: MOV     E,A
03D6 3E01               MVI     A,1
03D8 D307               OUT     EXTOT
03DA 7E         ISERX:  MOV     A,M
03DB E5                 PUSH    H
03DC 21DD00             LXI     H,INPTB
03DF 07         CODE:   RLC
03E0 85                 ADD     L
03E1 6F                 MOV     L,A
03E2 7C                 MOV     A,H
03E3 CE00               ACI     0
03E5 67                 MOV     H,A
03E6 7E                 MOV     A,M
03E7 23                 INX     H
03E8 66                 MOV     H,M
03E9 6F                 MOV     L,A
03EA E3                 XTHL
03EB 3AD681     CODE2:  LDA     EXTBF
03EE D307               OUT     EXTOT
03F0 AF                 XRA     A
03F1 C9                 RET

03F2 7B         ZERFL:  MOV     A,E
03F3 E67F               ANI     7FH
03F5 FE20               CPI     32
03F7 DA5003             JC      COLOR
03FA FE60               CPI     96
03FC D8                 RC
03FD 2EE6       SP64C:  MVI     L,COLFL AND 255
03FF 6E                 MOV     L,M
0400 2D                 DCR     L
0401 C0                 RNZ
0402 D660               SUI     96
0404 C9                 RET

                ;;      SBCHA NO 2X CHA = 84 US ,NO WAIT
                ;       SBCHA WITH SP64 CHA ADD 18 US ,NO WAIT

0405 CDF203     SBCHA:  CALL    ZERFL   ;34 US MIN
0408 2AD081             LHLD    BFILL   ;41 US MIN WAIT
```

```
040B EB                 XCHG    ;52 US MAX
040C 2AD481             LHLD    BHLAD   ;63 US MAX WAIT
040F B3                 ORA     E
0410 77                 MOV     M,A
0411 23                 INX     H
0412 72                 MOV     M,D
0413 23                 INX     H
0414 FC3A04             CM      STOR2
0417 3E80       TESHI:  MVI     A,(X80+NC  IN*NOCHA*2) SHR 8
0419 BC                 CMP     H
041A 22D481             SHLD    BHLAD
041D C0                 RNZ
041E 210070             LXI     H,X80
0421 22D481             SHLD    BHLAD
0424 C9                 RET

0425 CDF203     SVCHA:  CALL    ZERFL
0428 CD3104             CALL    STOR1-3
042B FC3A04             CM      STOR2
042E C3E781             JMP     JUMP
0431 CD4602             CALL    GETBC-4
0434 B3         STOR1:  ORA     E
0435 77                 MOV     M,A
0436 23                 INX     H
0437 72                 MOV     M,D
0438 23                 INX     H
0439 C9                 RET
043A E5         STOR2:  PUSH    H
043B C5                 PUSH    B
043C 01?EFF             LXI     B,-NOCHA*2-2
043F 09                 DAD     B
0440 C1                 POP     B
0441 77                 MOV     M,A
0442 23                 INX     H
0443 72                 MOV     M,D
0444 E1                 POP     H
0445 C9                 RET

0446 1C         CHPLO:  INR     E
0447 1C         TMODE:  INR     E
0448 73                 MOV     M,E
0449 C9                 RET

                ;;      CCIX TRICKEY BYTE SAVER

044A 77         CCIX:   MOV     M,A
044B 2ECF               MVI     L,VFILL+1 AND 255
044D 0636               MVI     B,36H   ;TRICK
044F 012ECF             LXI     B,0CF2EH        ;TRICK
0452 (044E)             ORG     $-4

044E 3601       BCCIX:  MVI     M,1
0450 2ED1               MVI     L,BFILL+1 AND 255

0452            ESCAP:
0452            BRATE:
0452            BLIND:
0452            CRSXY:
```

```
0452            CCI:
0452            TEST:
0452            CHAIN:
0452 73                 MOV     M,E
0453            FREE:
0453            FREEX:
0453 C9         VISIB:  RET

0454 7B         CURSO:  MOV     A,E
0455 FE51               CPI     81
0457 3607               MVI     M,7
0459 D26204             JNC     B7ON
045C 3605               MVI     M,5
045E 2ECB               MVI     L,VCRAD AND 255
0460 77                 MOV     M,A
0461 C9                 RET
0462 E680       B7ON:   ANI     80H
0464 2ED0               MVI     L,BFILL AND 255
0466 77                 MOV     M,A
0467 C9                 RET

0468 77         VCRSY:  MOV     M,A
0469 2ECB               MVI     L,VCRAD AND 255
046B 4E                 MOV     C,M
046C 7B                 MOV     A,E
046D CDD602             CALL    CRSY
0470 C3BF02             JMP     SVCRS+3
0473 34         BCRSX:  INR     M
0474 2ED4               MVI     L,BHLAD AND 255
0476 7B         TESTC:  MOV     A,E
0477 E67F               ANI     127
0479 77                 MOV     M,A
047A 3EC0               MVI     A,-NOCHA
047C 86                 ADD     M
047D D0                 RNC
047E 77                 MOV     M,A
047F C9                 RET
0480 34         BCRSY:  INR     M
0481 2ED4               MVI     L,BHLAD AND 255
0483 4E                 MOV     C,M

0484 3ACE81             LDA     VFILL   ; FIX BLIND CURSOR A7 BUG  <<9/78>>
0487 F5                 PUSH    PSW
0488 3AD081             LDA     BFILL
048B 32CE81             STA     VFILL

048E 7B                 MOV     A,E
048F CDD602             CALL    CRSY

0492 F1                 POP     PSW     ; <<9/78>>
0493 32CE81             STA     VFILL

0496 22D481             SHLD    BHLAD
0499 C9                 RET

049A E63F               ANI     31 OR NOLIN
049C 47         TESTB:  MOV     B,A
049D 3EE0               MVI     A,-NOLIN
```

Should be 65 for CCII
(81 is for 80 chrs/line)

```
049F 80                 ADD     B
04A0 D0                 RNC
04A1 47                 MOV     B,A
04A2 C9                 RET

04A3 3E02       HALF:   MVI     A,2
04A5 3D         FULL:   DCR     A
04A6 32DD81     LOCAL:  STA     DUPLX
04A9 C9                 RET

04AA 77         KEYOT:  MOV     M,A
04AB 3AFE81             LDA     KBCHA
04AE D306               OUT     TXBUF
04B0 3EDF       BEQOT:  MVI     A,0DFH  ;T5,T4,-,RX,T3,RTC,T2,T1
04B2 D308               OUT     MASK
04B4 32E081             STA     CMASK   ;SAVE   CURRENT MASK SETTING
04B7 C9                 RET

04B8 CD2E0A     OSERL:  CALL    SAVE    ;SAVE ALL REGISTERS
04BB 21F881             LXI     H,OUTFL
04BE E5                 PUSH    H
04BF 7E                 MOV     A,M
04C0 21CD00             LXI     H,OUTBL
04C3 C3DF03             JMP     CODE

04C6 77         BRATX:  MOV     M,A
04C7 3E07               MVI     A,7
04C9 A3                 ANA     E
04CA C8                 RZ
04CB 5F                 MOV     E,A
04CC 3E80               MVI     A,80H
04CE 07         BRTX1:  RLC
04CF 1D                 DCR     E
04D0 C2CE04             JNZ     BRTX1
04D3 2ECE               MVI     L,VFILL AND 255
04D5 B6                 ORA     M
04D6 D305               OUT     BAUD
04D8 32E281             STA     CRATE   ;SAVE CURRENT BAUD RATE
04DB C9                 RET

04DC 3C         ROLL:   INR     A
04DD 21FE02     PAGE:   LXI     H,CRSRT
04E0 32DC81             STA     ROLFL
04E3 22E881             SHLD    JUMP+1
04E6 C9                 RET
04E7 210603     DOWN:   LXI     H,LF
04EA C3E004             JMP     PAGE+3

04ED 2EF8       PRINT:  MVI     L,OUTFL AND 255
04EF 3605               MVI     M,5
04F1 210180             LXI     H,X80+NOLIN*NOCHA*2+1   ;<<8,79>> PUT FF,00 PAST SCREEN
04F4 77                 MOV     M,A
04F5 2F                 CMA
04F6 2B                 DCX     H
04F7 77                 MOV     M,A
04F8 CD0108             CALL    NROLL
04FB 2AD281             LHLD    VHLAD
04FE C38905             JMP     TXOUT+2
```

```
0501 77        ESCAX:  MOV     M,A
0502 3E1F              MVI     A,31
0504 E5                PUSH    H
0505 218D00            LXI     H,ESCTB
0508 A3                ANA     E
0509 5F                MOV     E,A
050A C3DF03            JMP     CODE
050D CDA304    BRAKE:  CALL    HALF
0510 3A1700            LDA     RCOMD
0513 F602              ORI     2
0515 D304              OUT     COMND
0517 C31F05            JMP     BEL+5
051A 3E40      BEL:    MVI     A,40H
051C CD3D05            CALL    TIM4Z
051F 3A1B00            LDA     TPROG
0522 32FD81            STA     MS150
0525 C33205            JMP     TIM4X+10

0528 CD2E0A    TIM4X:  CALL    SAVE    ;SAVE ALL REGISTERS
052B 21FD81            LXI     H,MS150
052E 34                INR     M
052F CA3705            JZ      TIM4Y
0532 3EC3              MVI     A,195
0534 D30C              OUT     TIME4
0536 C9                RET
0537 3A1700    TIM4Y:  LDA     RCOMD
053A D304              OUT     COMND
053C AF                XRA     A
053D 32D681    TIM4Z:  STA     EXTBF
0540 D307              OUT     EXTOT
0542 C9                RET

0543 CD4F05    ELINE:  CALL    EARLN
0546 1C                INR     E
0547 1D                DCR     E
0548 0180FF            LXI     B,-NOCHA*2
054B FA5805            JM      EARLN+9
054E C9                RET

054F CDBB02    EARLN:  CALL    CR
0552 2AD281            LHLD    VHLAD
0555 0180F0            LXI     B,NOCHA*2-NOCHA*2*NOLIN
0558 09                DAD     B
0559 010008            LXI     B,NOCHA*32
055C 3E20              MVI     A,20H
055E C38902            JMP     EARS+7

0561 2AFB81    LTPEN:  LHLD    OUTHL
0564 3EF7              MVI     A,TEMP5 AND 255
                ;      CMP     L
0566 C30000            JMP     0       ; TWOUT  <<8.79>>  NEED 1 BYTE !

0569 2AFB81    XYMIT:  LHLD    OUTHL
056C 7D                MOV     A,L
056D E67F              ANI     127
056F CA9405            JZ      FEED
0572 AF                XRA     A
```

```
0573 2F        TXCONT: CMA             ; <<8.79>>  TEST FOR FF  EXTRA BYTE
0574 BE                CMP     M
0575 C27C05            JNZ     TWOUT
0578 23                INX     H       ; <<8.79>>  LOOK AHEAD 1 CHAR
0579 2F                CMA
057A BE                CMP     M
057B 2B                DCX     H       ; <<8.79>>  LOOK BACK 1 CHAR
057C 7E        TWOUT:  MOV     A,M
057D 23                INX     H
057E 23                INX     H
057F C28705            JNZ     TXOUT
0582 21F881            LXI     H,OUTFL
0585 3607              MVI     M,7
0587 D306      TXOUT:  OUT     TXBUF
0589 3EFF              MVI     A,0FFH  ;T5T4,TX,RX,T3,RTC,T2,T1
058B D308              OUT     MASK
058D 32E081            STA     CMASK   ;SAVE CURRENT MASK
0590 22FB81            SHLD    OUTHL
0593 C9                RET

0594 21F881    FEED:   LXI     H,OUTFL
0597 3E0A              MVI     A,10    ;LINEFEED
0599 C39E05            JMP     LINE+2
059C 3E0D      LINE:   MVI     A,13    ;CARRAGE RETURN
059E 34                INR     M       ;INC FLAG VALUE
059F 2AFB81            LHLD    OUTHL
05A2 C38705            JMP     TXOUT

05A5 35        CARR:   DCR     M
05A6 35                DCR     M
05A7 2AFB81            LHLD    OUTHL
05AA C37305            JMP     TXCONT

05AD 77        CRET:   MOV     M,A
05AE 3A0D00            LDA     CARET   ;CR
05B1 C3AE04            JMP     KEYOT+4

05B4 77        TXPEN:  MOV     M,A
05B5 CD4A02            CALL    GETBC
05B8 5E                MOV     E,M
05B9 23                INX     H
05BA 56                MOV     D,M
05BB 21F881            LXI     H,OUTFL
05BE 3602              MVI     M,2
05C0 2EF7              MVI     L,TEMP5 AND 255
05C2 73                MOV     M,E
05C3 2D                DCR     L
05C4 72                MOV     M,D
05C5 2D                DCR     L
05C6 3606              MVI     M,6
05C8 2D                DCR     L
05C9 3ACD81            LDA     ROLLN
05CC 2F                CMA
05CD 3C                INR     A
05CE 80                ADD     B
05CF F2D405            JP      TXPN1
05D2 C620              ADI     NOLIN
05D4 77        TXPN1:  MOV     M,A
```

```
05D5 2D              DCR     L
05D6 71              MOV     M,C
05D7 3E03            MVI     A,3
05D9 C38705          JMP     TXOUT

05DC 07      SHIFF:  RLC
05DD D2EB05          JNC     LS2
05E0 5F              MOV     E,A
05E1 2C              INR     L
05E2 7D              MOV     A,L
05E3 FE80            CPI     NOCHA*2
05E5 DAEA05          JC      LS1
05E8 AF              XRA     A
05E9 6F              MOV     L,A
05EA 7B      LS1:    MOV     A,E
05EB 07      LS2:    RLC
05EC D2F705          JNC     LS4
05EF 2D              DCR     L
05F0 2C              INR     L
05F1 C2F605          JNZ     LS3
05F4 2E80            MVI     L,NOCHA*2
05F6 2D      LS3:    DCR     L
05F7 07      LS4:    RLC
05F8 D20306          JNC     LS6
05FB 25              DCR     H
05FC 24              INR     H
05FD C20206          JNZ     LS5
0600 2680            MVI     H,NOLIN*4
0602 25      LS5:    DCR     H
0603 07      LS6:    RLC
0604 5F              MOV     E,A
0605 D21206          JNC     LS8
0608 24              INR     H
0609 7C              MOV     A,H
060A FE80            CPI     NOLIN*4
060C DA1106          JC      LS7
060F AF              XRA     A
0610 67              MOV     H,A
0611 7B      LS7:    MOV     A,E
0612 E6F0    LS8:    ANI     0F0H
0614 C9              RET

0615 CD3A06  VECTY:  CALL    INVY-2
0618 C3B808          JMP     VECTO
061B 2AEA81  INVEC:  LHLD    XTWO
061E CDDC05          CALL    SHIFF
0621 22EC81          SHLD    XDATA
0624 C8              RZ
0625 4D              MOV     C,L
0626 44              MOV     B,H
0627 2AEF81          LHLD    XZERO
062A 7B              MOV     A,E
062B CDDC05          CALL    SHIFF
062E 22EF81          SHLD    XZERO
0631 C8              RZ
0632 E5              PUSH    H
0633 CD4307          CALL    PUTYD-6
```

```
0636 C1              POP     B
0637 C38D08          JMP     VECTO+2

063A 2EF0            MVI     L,YZERO AND 255
063C 47        INVY: MOV     B,A
063D 3E7F            MVI     A,NOLIN*4-1
063F 90              SUB     B
0640 FE80      INVY1: CPI    NOLIN*4
0642 77              MOV     M,A
0643 47              MOV     B,A
0644 D8              RC
0645 C680            ADI     NOLIN*4
0647 C34006          JMP     INVY1

064A 2EEF      PXZER: MVI    L,XZERO AND 255
064C FE80            CPI     NOCHA*2
064E 77              MOV     M,A
064F D8              RC
0650 D680            SUI     NOCHA*2
0652 77              MOV     M,A
0653 C9              RET
0654 CD3C06    PLPTY: CALL   INVY
0657 1EC5            MVI     E,XYTAB AND 255
0659 C34907          JMP     PUTYD

065C CD8A07    PBINX: CALL   INCXY
065F CA6806          JZ      $+9
0662 CD6806          CALL    $+6
0665 CDA607          CALL    MOVXY
0668 79              MOV     A,C
0669 2646            MVI     H,46H    ;TRICK
066B (066A)          ORG     $-1

066A 46        BARXM: MOV    B,M
066B 21EF81          LXI     H,XZERO
066E BE              CMP     M
066F CA4407          JZ      PUTYD-5
0672 1EBD            MVI     E,BARTX AND 255
0674 CD4607          CALL    PUTYD-3
0677 3E0F            MVI     A,0FH
0679 A3              ANA     E
067A 07              RLC
067B 07              RLC
067C 07              RLC
067D 07              RLC
067E B3              ORA     E
067F 5F              MOV     E,A
0680 79              MOV     A,C
0681 07              RLC
0682 4F              MOV     C,A
0683 3AEF81          LDA     XZERO
0686 47              MOV     B,A
0687 AF        BXLOP: XRA    A
0688 B9              CMP     C
0689 C8              RZ
068A 0D              DCR     C
068B 2B              DCX     H
068C 79              MOV     A,C
```

```
068D B8                 CMP     B
068E D8                 RC
068F CA2408             JZ      PUTXZ
0692 0D                 DCR     C
0693 CD7107             CALL    PXYCH
0696 C38706             JMP     BXLOP
0699 CD3C06     BARYM:  CALL    INVY
069C 21F081             LXI     H,YZERO
069F AE                 XRA     M
06A0 E6FC               ANI     252
06A2 3AEC81             LDA     XDATA
06A5 CA0B08             JZ      PUTZZ
06A8 1EB5               MVI     E,BARTY AND 255
06AA CD4607             CALL    PUTYD-3
06AD 3AF081             LDA     YZERO
06B0 E6FC               ANI     252
06B2 0F                 RRC
06B3 0F                 RRC
06B4 4F                 MOV     C,A       ;C=YZERO SHR 2
06B5 7B                 MOV     A,E
06B6 07                 RLC
06B7 1E0F               MVI     E,0FH
06B9 D2BE06             JNC     $+5
06BC 1EF0               MVI     E,0F0H
06BE 79         BYLOP:  MOV     A,C
06BF 04                 INR     B
06C0 B8                 CMP     B
06C1 D8                 RC
06C2 CC2B08             CZ      PUTYZ
06C5 3E80               MVI     A,NOCHA*2
06C7 85                 ADD     L
06C8 6F                 MOV     L,A
06C9 D2CD06             JNC     $+4
06CC 24                 INR     H
06CD CD7007             CALL    PXYCH-1
06D0 C3BE06             JMP     BYLOP
06D3 CD8A07     PLINC:  CALL    INCXY
06D6 CA4307             JZ      PUTYD-6
06D9 CD4307             CALL    PUTYD-6
06DC CDA607             CALL    MOVXY
06DF C34307             JMP     PUTYD-6
06E2 CD8A07     PBINY:  CALL    INCXY
06E5 CAEE06             JZ      $+9
06E8 CDEE06             CALL    $+6
06EB CDA607             CALL    MOVXY
06EE 78                 MOV     A,B
06EF C39C06             JMP     BARYM+3
06F2 21EE81     PUTXD:  LXI     H,ODDFL
06F5 FE80               CPI     NOCHA*2
06F7 DAFC06             JC      $+5
06FA D680               SUI     NOCHA*2
06FC 4F                 MOV     C,A
06FD E601               ANI     1         ;ODD=1
06FF 77                 MOV     M,A       ;ODDFL
0700 2EEC               MVI     L,XDATA AND 255
0702 71                 MOV     M,C
0703 79                 MOV     A,C
0704 1F                 RAR
```

```
0705 2ED8              MVI     L,PCRAD AND 255
0707 77                MOV     M,A
0708 C9                RET
0709 CD4708   PLOTX:   CALL    PLOKB
070C 7B                MOV     A,E
070D EEFF              XRI     255
070F C21B07            JNZ     $+12
0712 77                MOV     M,A
0713 2ECF              MVI     L,VFILL+1 AND 255
0715 3E7F              MVI     A,7FH
0717 A6                ANA     M
0718 C30008            JMP     NROLL-1
071B 2EDA              MVI     L,PLOFL AND 255
071D 35                DCR     M
071E CA5908            JZ      ASCPL
0721 34                INR     M
0722 FE10              CPI     16
0724 DA8807            JC      INCXY-2
0727 7E                MOV     A,M
0728 07                RLC
0729 86                ADD     M
072A 21C707            LXI     H,PLTAB-6
072D 85                ADD     L
072E 6F                MOV     L,A
072F 3E00              MVI     A,0
0731 8C                ADC     H
0732 67                MOV     H,A
0733 7E                MOV     A,M     ;PLOFL NEW VALUE
0734 2C                INR     L
0735 4E                MOV     C,M     ;PLOT SUBR ADDR
0736 2C                INR     L
0737 46                MOV     B,M
0738 21DA81            LXI     H,PLOFL
073B 86                ADD     M
073C 77                MOV     M,A     ;PLOFL NEW VAL
073D 7B                MOV     A,E
073E C5                PUSH    B
073F 47                MOV     B,A
0740 2EED              MVI     L,YDATA AND 255
0742 C9                RET

0743 79                MOV     A,C
0744 1EC5              MVI     E,XYTAB AND 255
0746 CDF206            CALL    PUTXD
0749 2EEE     PUTYD:   MVI     L,ODDFL AND 255
074B 78                MOV     A,B     ;YDATA
074C FE80              CPI     NOLIN*4
074E DA5307            JC      $+5
0751 D680              SUI     NOLIN*4
0753 47                MOV     B,A
0754 E603              ANI     3       ;A=0,1,2,3
0756 07                RLC
0757 83                ADD     E
0758 86                ADD     M
[illegible] 6F         MOV     L,A
[illegible] 2607       MVI     H,DARTY SHR 8
075C 5E                MOV     E,M     ;PLOT CHARACTER
075D 21[illegible]     LXI     H,YDATA
```

```
0760 70                 MOV     M,B
0761 78                 MOV     A,B
0762 E6FC               ANI     252
0764 0F                 RRC
0765 0F                 RRC
0766 47                 MOV     B,A
0767 2ED8               MVI     L,PCRAD AND 255
0769 4E                 MOV     C,M
076A 2ECF               MVI     L,VFILL+1 AND 255
076C 56                 MOV     D,M
076D CDE702             CALL    LN5Y
0770 2C                 INR     L
0771 7E         PXYCH:  MOV     A,M     ;STATUS WORD
0772 72                 MOV     M,D
0773 2D                 DCR     L
0774 AA                 XRA     D
0775 7E                 MOV     A,M
0776 73                 MOV     M,E
0777 F8                 RM      ;OLD CHA NO PL
0778 CA7F07             JZ      ORCHA
077B EEFF               XRI     255
077D C8                 RZ      ;OLD CH DIF COL
077E 2F                 CMA
077F AB         ORCHA:  XRA     E
0780 77                 MOV     M,A
0781 3AE681             LDA     COLFL
0784 A7                 ANA     A
0785 C0                 RNZ
0786 7B                 MOV     A,E
0787 B6                 ORA     M
0788 77                 MOV     M,A
0789 C9                 RET

                ;;      RETURN FOR PXYCH

078A 2AEC81     INCXY:  LHLD    XDATA
078D CDDC05             CALL    SHIFF
0790 22EC81             SHLD    XDATA
0793 CA0209             JZ      LL10
0796 44                 MOV     B,H
0797 4D                 MOV     C,L
0798 7B                 MOV     A,E
0799 CDDC05             CALL    SHIFF
079C 22EA81             SHLD    XTWO
079F C0                 RNZ
07A0 22EC81             SHLD    XDATA
07A3 44                 MOV     B,H
07A4 4D                 MOV     C,L
07A5 C9                 RET
07A6 2AEA81     MOVXY:  LHLD    XTWO
07A9 22EC81             SHLD    XDATA
07AC 44                 MOV     B,H
07AD 4D                 MOV     C,L
07AE C9                 RET

07AF 07         BARTZ:  DB      07
07B0 70                 DB      70H
07B1 06                 DB      06
```

```
07B2 60                 DB      60H
07B3 03                 DB      03
07B4 30                 DB      30H
07B5 0F         BARTY:  DB      0FH
07B6 F0                 DB      0F0H
07B7 0E                 DB      0EH
07B8 E0                 DB      0E0H
07B9 0C                 DB      0CH
07BA C0                 DB      0C0H
07BB 08                 DB      8H
07BC 80                 DB      80H
07BD 01         BARTX:  DB      1
07BE 11                 DB      11H
07BF 02                 DB      2
07C0 22                 DB      22H
07C1 04                 DB      4H
07C2 44                 DB      44H
07C3 08                 DB      8H
07C4 88                 DB      88H
07C5 01         XYTAB:  DB      1
07C6 10                 DB      10H
07C7 02                 DB      2
07C8 20                 DB      20H
07C9 04                 DB      4
07CA 40                 DB      40H
07CB 08                 DB      8
07CC 80                 DB      80H
07CD 01         PLTAB:  DB      1
07CE F206               DW      PUTXD   ; (2) PLOT X
07D0 FF                 DB      -1
07D1 5406               DW      PLPTY   ; (3) PLOT Y
07D3 00                 DB      0
07D4 D306               DW      PLINC   ; (4) INC-XY
07D6 01                 DB      1
07D7 4A06               DW      PXZER   ; (5) BAR X
07D9 01                 DB      1
07DA 3C06               DW      INVY    ; (6) BAR X
07DC FF                 DB      -1
07DD 6A06               DW      BARXM   ; (7) BAR X
07DF 00                 DB      0
07E0 5C06               DW      PBINX   ; (8) INC BAR X
07E2 01                 DB      1
07E3 3A06               DW      INVY-2  ; (9) BAR Y
07E5 01                 DB      1
07E6 F206               DW      PUTXD   ;(10) BAR Y
07E8 FF                 DB      -1
07E9 9906               DW      BARYM   ;(11) BAR Y
07EB 00                 DB      0
07EC E206               DW      PBINY   ;(12) INC BAR Y
07EE 01                 DB      1
07EF 4A06               DW      PXZER   ;(13) VECT X0
07F1 FF                 DB      -1
07F2 1506               DW      VECTY   ;(14) VECT Y0
07F4 00                 DB      0
07F5 1B06               DW      INVEC   ;(15) INC VECT

07F7 73         POTON:  MOV     M,E
07F8 2EDA               MVI     L,PLOFL AND 255
```

```
07FA 73                 MOV     M,E
07FB 2ECF               MVI     L,VFILL+1 AND 255
07FD 3E80               MVI     A,80H
07FF B6                 ORA     M
0800 77                 MOV     M,A
0801 AF         NROLL:  XRA     A
0802 32CD81             STA     ROLLN
0805 CD4A02             CALL    GETBC
0808 C3C202             JMP     SVCRS+6

080B 4F         PUTZZ:  MOV     C,A
080C 78                 MOV     A,B
080D BE                 CMP     M
080E CA4307             JZ      PUTYD-6
0811 3E03               MVI     A,3
0813 A6                 ANA     M
0814 FE02               CPI     2
0816 79                 MOV     A,C
0817 1EAF               MVI     E,BARTZ AND 255
0819 CA4607             JZ      PUTYD-3
081C 1EB3               MVI     E,BARTZ+4 AND 255
081E FA4607             JM      PUTYD-3
0821 C3AB06             JMP     BARYM+15

0824 3EF0       PUTXZ:  MVI     A,0F0H
0826 A3                 ANA     E
0827 5F                 MOV     E,A
0828 C37107             JMP     PXYCH
082B 3AF081     PUTYZ:  LDA     YZERO
082E E603               ANI     3
0830 CA4208             JZ      PUTEZ-2
0833 FE02               CPI     2
0835 3E77               MVI     A,77H
0837 CA4408             JZ      PUTEZ
083A 3EFF               MVI     A,0FFH
083C F24408             JP      PUTEZ
083F 3E33               MVI     A,33H
0841 F24408             JP      PUTEZ    ;TRICK
0844 (0842)             ORG     $-2

0842 3E11               MVI     A,11H
0844 A3         PUTEZ:  ANA     E
0845 5F                 MOV     E,A
0846 C9                 RET

0847 3EDF       PLOKB:  MVI     A,KBDFL AND 255
0849 BD                 CMP     L
084A C0                 RNZ
084B 3EB0               MVI     A,0B0H
084D A3                 ANA     E
084E EEB0               XRI     0B0H
0850 C0                 RNZ
0851 7B                 MOV     A,E
0852 F6F0               ORI     0F0H
0854 5F                 MOV     E,A
0855 32FE81             STA     KBCHA
0858 C9                 RET
0859 34         ASCPL:  INR     M
```

```
085A 7B              MOV     A,E
085B CD3104          CALL    STOR1-3
085E C3E781          JMP     JUMP

0861 7B        CPLOX:MOV     A,E
0862 FE10            CPI     16          ;32 FOR#
0864 DA2504          JC      SVCHA
0867 FE18            CPI     24          ;31H FOR #
0869 D22504          JNC     SVCHA       ;RNC FOR #
086C E607            ANI     7
086E C6FC            ADI     0FCH
0870 17              RAL
0871 C6C5            ADI     XYTAB AND 255
0873 6F              MOV     L,A
0874 2607            MVI     H,XYTAB SHR 8
0876 5E              MOV     E,M
0877 2AD281          LHLD    VHLAD
087A 2C              INR     L
087B 3ACF81          LDA     VFILL+1
087E F680            ORI     80H
0880 57              MOV     D,A
0881 7E              MOV     A,M
0882 72              MOV     M,D
0883 2D              DCR     L
0884 AA              XRA     D
0885 7E              MOV     A,M
0886 73              MOV     M,E
0887 F8              RM
0888 AB              XRA     E
0889 77              MOV     M,A
088A C9              RET

                     ;;      START POINT:    X1=XDATA, Y1=YDATA
                     ;       END POINT:      X0=XZERO, Y0=YZERO

088B 2D        VECTO:DCR     L           ;POINT TO X0
088C 4E              MOV     C,M         ;GET X0
088D 2AEC81          LHLD    XDATA       ;L=X1, H=Y1
0890 22EA81          SHLD    XTWO        ;STORE PRESENT POSITION
0893 E5              PUSH    H           ;SAVE PRESENT POSITION
0894 78              MOV     A,B         ;GET NEW Y
0895 94              SUB     H           ;COMPUTE YCHANGE
0896 2601            MVI     H,1         ;ASSUME YSTEP OF +1
0898 D29F08          JNC     LL1         ;YCHANGE IS + OR 0
089B 26FF            MVI     H,-1        ;SET YSTEP OF -1
089D 2F              CMA                 ;COMPUTE MAGNITUDE ...
089E 3C              INR     A           ;... OF YCHANGE
089F 57        LL1:  MOV     D,A         ;SAVE YCHANGE MAGNITUDE
08A0 79              MOV     A,C         ;GET NEW X
08A1 95              SUB     L           ;COMPUTE XCHANGE
08A2 2E01            MVI     L,1         ;ASSUME XSTEP OF +1
08A4 D2AB08          JNC     LL2         ;XCHANGE IS + OR 0
08A7 2EFF            MVI     L,-1        ;SET XSTEP OF -1
08A9 2F              CMA                 ;COMPUTE MAGNITUDE ...
08AA 3C              INR     A           ;... OF XCHANGE
08AB 5F        LL2:  MOV     E,A         ;SAVE XCHANGE MAGNITUDE
08AC 22F281          SHLD    TEMPO       ;STORE XSTEP AND YSTEP
08AF C1              POP     B           ;GET PRESENT POSITION
```

```
08B0 BA              CMP     D         ;XCHANGE >= YCHANGE ?
08B1 D2DB08          JNC     LL6       ;YES
08B4 7A              MOV     A,D       ;GET YCHANGE
08B5 A7              ANA     A         ;CLEAR C-BIT
08B6 1F              RAR               ;HALVE YCHANGE
08B7 93        LL3:  SUB     E         ;SUBTRACT SMALL CHANGE
08B8 D2C308          JNC     LL4       ;DON'T STEP X THIS TIME
08BB 82              ADD     D         ;ADD LARGE CHANGE
08BC F5              PUSH    PSW       ;SAVE COUNTER
08BD 3AF281          LDA     TEMP0     ;GET XSTEP
08C0 81              ADD     C         ;ADD X
08C1 4F              MOV     C,A       ;SET NEW X
08C2 3EF5            MVI     A,0F5H    ;GO STEP Y
08C4 (08C3)          ORG     $-1       ;TRICK SKIP TO LL5
08C3 F5        LL4:  PUSH    PSW       ;SAVE COUNTER
08C4 3AF381    LL5:  LDA     TEMP1     ;GET YSTEP
08C7 80              ADD     B         ;ADD Y
08C8 47              MOV     B,A       ;SET NEW Y
08C9 D5              PUSH    D         ;SAVE D&E
08CA C5              PUSH    B         ;SAVE B&C
08CB CD4307          CALL    PUTYD-6   ;PLOT THIS POINT
08CE C1              POP     B         ;RESTORE B&C
08CF D1              POP     D         ;RESTORE D&E
08D0 3AF081          LDA     YZERO     ;GET END Y
08D3 B8              CMP     B         ;FINISHED ?
08D4 CA0209          JZ      LL10      ;YES
08D7 F1              POP     PSW       ;RESTORE COUNTER
08D8 C3B708          JMP     LL3       ;LOOP
08DB B7        LL6:  ORA     A         ;XCHANGE = 0 ?
08DC C8              RZ                ;YES: RETURN TO CALLER
08DD 1F              RAR               ;HALVE XCHANGE (C IS 0)
08DE 92        LL7:  SUB     D         ;SUBTRACT SMALL CHANGE
08DF D2EA08          JNC     LL8       ;DON'T STEP Y THIS TIME
08E2 83              ADD     E         ;ADD LARGE CHANGE
08E3 F5              PUSH    PSW       ;SAVE COUNTER
08E4 3AF381          LDA     TEMP1     ;GET YSTEP
08E7 80              ADD     B         ;ADD Y
08E8 47              MOV     B,A       ;SET NEW Y
08E9 3EF5            MVI     A,0F5H    ;GO STEP X
08EB (08EA)          ORG     $-1       ;TRICK SKIP TO LL9
08EA F5        LL8:  PUSH    PSW       ;SAVE COUNTER
08EB 3AF281    LL9:  LDA     TEMP0     ;GET XSTEP
08EE 81              ADD     C         ;ADD X
08EF 4F              MOV     C,A       ;SET NEW X
08F0 D5              PUSH    D         ;SAVE D&E
08F1 C5              PUSH    B         ;SAVE B&C
08F2 CD4307          CALL    PUTYD-6   ;PLOT THIS POINT
08F5 C1              POP     B         ;RESTORE B&C
08F6 D1              POP     D         ;RESTORE D&E
08F7 3AEF81          LDA     XZERO     ;GET END X
08FA B9              CMP     C         ;FINISHED ?
08FB CA0209          JZ      LL10      ;YES
08FE F1              POP     PSW       ;RESTORE COUNTER
08FF C3DE08          JMP     LL7       ;LOOP
0902 F1        LL10: POP     PSW       ;CLEAN STACK
0903 C9              RET               ;RETURN TO CALLER
```

```
0904 CD2E0A    TIM3X:  CALL    SAVE
0907 CD1109            CALL    KEYBD
090A D22B00            JNC     KEYCO
090D CA3B00            JZ      BREAK
0910 C9                RET

0911 3A1D00    KEYBD:  LDA     KEDEL
0914 D30B              OUT     TIME3
0916 01FF0F    G01:    LXI     B,0FFFH ;INIT. ROW COUNTER & 1ST KEYCODE
0919 59                MOV     E,C     ;INITIALIZE 2ND KEYCODE
091A 78        G02:    MOV     A,B     ;GET COMPLEMENT OF NEXT ROW NUMBER
091B D307              OUT     EXTOT   ;SEND
091D DB01              IN      EXTIN   ;READ
091F 3C                INR     A       ;GOT ANY ?
0920 CA4D09            JZ      G06     ;NO !
0923 1C                INR     E       ;ALREADY SEEN TWO KEYS ?
0924 C25C09            JNZ     ROT     ;YES: MORE THAN TWO KEYS: IGNORE !
0927 3D        G03:    DCR     A       ;RESTORE
0928 37                STC             ;SET <C> BIT
0929 1C        G04     INR     E
092A 1F                RAR             ;FOUND IT ?
092B DA2909            JC      G04     ;NO !
092E F5                PUSH    PSW
092F 7B                MOV     A,E     ;GET GROUP CODE
0930 87                ADD     A       ;POSITION IT ...
0931 87                ADD     A
0932 87                ADD     A
0933 87                ADD     A
0934 B0                ORA     B       ;MERGE COMPLEMENT OF ROW NUMBER
0935 EE0F              XRI     0FH     ;MAKE ROW NUMBER RIGHT
0937 0C                INR     C       ;IS THIS 1ST KEY SEEN ON THIS ROW ?
0938 CA4509            JZ      G05     ;YES !
093B 5F                MOV     E,A     ;COPY 2ND KEYCODE SEEN
093C F1                POP     PSW
093D 3C                INR     A       ;ANY MORE ON THIS ROW ?
093E C25C09            JNZ     ROT     ;YES: MORE THAN TWO KEYS !
0941 0D                DCR     C       ;RESTORE 1ST KEYCODE SEEN
0942 C34D09            JMP     G06
0945 4F        G05:    MOV     C,A     ;SAVE 1ST KEYCODE SEEN
0946 F1                POP     PSW
0947 3C                INR     A       ;ANY MORE ON THIS ROW ?
0948 C22709            JNZ     G03     ;YES !
094B 1EFF              MVI     E,-1    ;RE-INIT. 2ND KEYCODE
094D 05        G06:    DCR     B       ;SCANNED ALL ROWS ?
094E F21A09            JP      G02     ;NO: READ NEXT ROW !
0951 0C                INR     C       ;SEE ANY KEYS ?
0952 C26509            JNZ     G08     ;YES !
0955 AF        G07:    XRA     A
0956 32E481            STA     LKC     ;CLEAR "LAST" KEYCODE
0959 32E581            STA     NKC     ;CLEAR "NEW" KEYCODE

095C 3AD681    ROT:    LDA     EXTBF
095F D307              OUT     EXTOT   ;RESTORE OUTPUT PORT
0961 F601              ORI     1
0963 37                STC
0964 C9                RET             ;EXIT !

0965 0D        G08:    DCR     C       ;RESTORE 1ST KEYCODE
```

```
0966 1C                 INR     E       ;SEE TWO KEYS ?
0967 C28A09             JNZ     G09     ;YES !

                ;;      SAW ONE KEY :

096A AF                 XRA     A
096B 32E581             STA     NKC     ;CLEAR "NEW" KEYCODE
096E 3AE481             LDA     LKC     ;GET "LAST" KEYCODE
0971 B9                 CMP     C       ;MATCH ?
0972 C29D09             JNZ     G10     ;NO !
0975 3E80               MVI     A,80H
0977 D307               OUT     EXTOT   ;SELECT 'B' HIBITS
0979 DB01               IN      EXTIN   ;READ
097B E640               ANI     40H     ;REPEAT ?
097D C25C09             JNZ     ROT     ;NO !
0980 21D781             LXI     H,SEC15 ;POINT TO DELAY COUNTER
0983 35                 DCR     M       ;IS IT TIME FOR ANOTHER ?
0984 C25C09             JNZ     ROT     ;NO: NOT YET !
0987 C3A109             JMP     G11

                ;;      SAW TWO KEYS :

098A 1D         G09:    DCR     E       ;RESTORE 2ND KEYCODE
098B 3AE481             LDA     LKC     ;GET "LAST" KEYCODE
098E CD200A             CALL    G20     ;MATCH ?
0991 CA5C09             JZ      ROT     ;YES: DO NOTHING !
0994 3AE581             LDA     NKC     ;GET "NEW" KEYCODE
0997 CD200A             CALL    G20     ;MATCH ?
099A C25509             JNZ     G07     ;NO !
099D 79         G10:    MOV     A,C     ;GET KEYCODE
099E 32E481             STA     LKC     ;SAVE "LAST" KEYCODE
09A1 3A1C00     G11:    LDA     CHTIM   ;SCANS/CHAR. IN REPEAT MODE
09A4 32D781             STA     SEC15   ;SET REPEAT TIME COUNTER
09A7 79                 MOV     A,C     ;GET KEY CODE
09A8 FE50               CPI     50H     ;SPECIAL GROUP ?
09AA DAB209             JC      G12     ;NO: NORMAL !
09AD FE56               CPI     56H     ;SPECIAL GROUP ?
09AF DA0A0A             JC      G16     ;YES !
09B2 0F         G12:    RRC             ;ROTATE ...
09B3 0F                 RRC
09B4 0F                 RRC
09B5 0F                 RRC
09B6 E607               ANI     7       ;GET GROUP CODE
09B8 5F                 MOV     E,A     ;COPY GROUP CODE
09B9 3E80               MVI     A,80H
09BB D307               OUT     EXTOT   ;SELECT 'B' HIBITS
09BD DB01               IN      EXTIN   ;READ
09BF E630               ANI     30H     ;GET CONTROL & SHIFT BITS
09C1 07                 RLC
09C2 07                 RLC
09C3 B3                 ORA     E       ;MERGE WITH GROUP CODE
09C4 07                 RLC
09C5 07                 RLC
09C6 5F                 MOV     E,A
09C7 1600               MVI     D,0
09C9 21380A             LXI     H,KTAB-4 ;POINT TO TRANSLATE TABLE
09CC 19                 DAD     D       ;INDEX
09CD 3E0F               MVI     A,0FH
```

```
09CF A1                 ANA     C       ;GET ROW NUMBER
09D0 B6                 ORA     M       ;MERGE WITH TABLE ENTRY
09D1 5F                 MOV     E,A     ;SAVE
09D2 79                 MOV     A,C     ;GET KEYCODE
09D3 FE7C               CPI     7CH
09D5 D2DF09             JNC     Q13     ;WEIRD !
09D8 D61C               SUI     1CH
09DA FE04               CPI     20H-1CH
09DC D2E309             JNC     Q14     ;NOT WEIRD !
09DF 3E10       Q13:    MVI     A,10H
09E1 AB                 XRA     E       ;FIX WEIRD
09E2 5F                 MOV     E,A
09E3 DB01       Q14:    IN      EXTIN   ;READ 'B' HIBITS
09E5 A7                 ANA     A       ;CAPS LOCK ?
09E6 F2FC09             JP      Q15     ;YES: O.K. !
09E9 7B                 MOV     A,E     ;GET ASCII CODE
09EA D641               SUI     'A'
09EC FE3A               CPI     'Z'+20H+1-'A'
09EE D2FC09             JNC     Q15     ;NOT ALPHA !
09F1 D61A               SUI     'Z'+1-'A'
09F3 FE06               CPI     'A'+20H-'Z'-1
09F5 DAFC09             JC      Q15     ;NOT ALPHA !
09F8 3E20               MVI     A,20H
09FA AB                 XRA     E       ;FLIP SHIFT
09FB 5F                 MOV     E,A
09FC CD5C09     Q15:    CALL    ROT     ;RESTORE OUTPUT PORT
09FF 3E80               MVI     A,80H   ;SET UP READY FLAG
0A01 32FF81             STA     KBRDY
0A04 B7                 ORA     A       ;SET SIGN BIT = GOOD KEY
0A05 7B                 MOV     A,E     ;GET CHAR.
0A06 32FE81             STA     KBCHA   ;STORE IT
0A09 C9                 RET             ;  AND EXIT
                ;
0A0A 32FF81     Q16:    STA     KBRDY   ;SET READY = 50H IF BREAK
0A0D D650               SUI     50H     ;ADJUST CODE
0A0F 37                 STC             ;  CARRY AND ZERO IS BREAK
0A10 C8                 RZ              ;CODE IS BREAK !
0A11 1E7F               MVI     E,127   ;MAY BE DELETE
0A13 FE04               CPI     4       ;IS IT DELETE ?
0A15 CAFC09             JZ      Q15     ;YES: GO DO IT !
0A18 5F                 MOV     E,A
0A19 3E06               MVI     A,6     ;MAP 5==>1,3==>3,2==>4,1==>5
0A1B 93                 SUB     E
0A1C 5F                 MOV     E,A
0A1D C3FC09             JMP     Q15     ;PROCESS IT !
                ;
0A20 B9         Q20:    CMP     C       ;MATCH THIS ONE ?
0A21 CA290A             JZ      Q21     ;YES !
0A24 69                 MOV     L,C     ;SWITCH THEM ...
0A25 4B                 MOV     C,E
0A26 5D                 MOV     E,L
0A27 B9                 CMP     C       ;MATCH THIS ONE ?
0A28 C0                 RNZ             ;NO MATCH: EXIT <NZ> !
0A29 7B         Q21:    MOV     A,E     ;GET KEYCODE THAT DOES NOT MATCH
0A2A 32E581             STA     NKC     ;STORE IT AS "NEW" KEYCODE
0A2D C9                 RET             ;MATCH: EXIT <Z> !

0A2E E3         SAVE:   XTHL            ;SAVE HL WHILE LOADING ADDRESS
```

```
0A2F D5              PUSH    D       ; TO RESUME EXECUTION AT,
0A30 C5              PUSH    B       ;THEN SAVE ALL REGISTERS
0A31 F5              PUSH    PSW     ; AND STATUS
0A32 CD3B0A          CALL    JMPHL   ;PUSH $+3 AS RETURN ADDRESS
0A35 F1              POP     PSW     ; AND RESUME EXECUTION OF CALLER
0A36 C1              POP     B       ;THEN RESTORE ALL REGISTERS
0A37 D1              POP     D       ; AND STATUS
0A38 E1              POP     H
0A39 FB              EI              ;REENABLE INTERRUPTS
0A3A C9              RET             ; AND EXIT FROM INTERRUPT

0A3B E9      JMPHL:  PCHL            ; JUMP THRU HERE TO ROUTINE

0A3C A0      KTAB:   DB      0A0H    ; 1 CS
0A3D 20              DB      020H    ; 1 -S
0A3E B0              DB      0B0H    ; 1 C-
0A3F 30              DB      030H    ; 1 --
0A40 80              DB      080H    ; 2 CS
0A41 60              DB      060H    ; 2 -S
0A42 00              DB      000H    ; 2 C-
0A43 40              DB      040H    ; 2 --
0A44 90              DB      090H    ; 3 CS
0A45 70              DB      070H    ; 3 -S
0A46 10              DB      010H    ; 3 C-
0A47 50              DB      050H    ; 3 --
0A48 E0              DB      0E0H    ; 4 CS
0A49 D0              DB      0D0H    ; 4 -S
0A4A C0              DB      0C0H    ; 4 C-
0A4B F0              DB      0F0H    ; 4 --
0A4C 80              DB      080H    ; 5 CS
0A4D 80              DB      080H    ; 5 -S
0A4E 00              DB      000H    ; 5 C-
0A4F 00              DB      000H    ; 5 --
0A50 90              DB      090H    ; 6 CS
0A51 90              DB      090H    ; 6 -S
0A52 10              DB      010H    ; 6 C-
0A53 10              DB      010H    ; 6 --
0A54 20              DB      020H    ; 7 CS
0A55 20              DB      020H    ; 7 -S
0A56 20              DB      020H    ; 7 C-
0A57 20              DB      020H    ; 7 --

                ;*****************************************************
                ;;      [illegible]LE CONTROL SYSTEM - VERSION 3.0

                ;       COPYRIGHT (C) 1977,1978,1979,1980
                ;        BY INTELLIGENT SYSTEMS CORPORATION

                ;;      COMMAND TABLE:

0A58 494E49  CMTAB:  DB      'INI'
0A5B C40B            DW      INI00
0A5D 444952          DB      'DIR'
0A60 340C            DW      DIR00
0A62 534156          DB      'SAV'
```

```
0A65 D60C              DW      SAVOO
0A67 4C4F41            DB      'LOA'
0A6A 0C0D              DW      LOAOO
0A6C 52554E            DB      'RUN'
0A6F F90D              DW      RUNOO
0A71 44454C            DB      'DEL'
0A74 580E              DW      DELOO
0A76 52454E            DB      'REN'
0A79 640F              DW      RENOO
0A7B 434F50            DB      'COP'
0A7E C30F              DW      COPOO
0A80 444556            DB      'DEV'
0A83 390E              DW      DEVOO
0A85 524541            DB      'REA'
0A88 940B              DW      REAOO
0A8A 575249            DB      'WRI'
0A8D 910B              DW      WRIOO
0A8F 445550            DB      'DUP'
0A92 2001              DW      DUPOO

0A94 00                DB      0       ;TERMINATOR !


                ;;      OPEN TYPE CODE BIT DEFINITIONS :

     (0001)     FNEW    EQU     01H     ;0: OLD FILE, 1: NEW FILE


                ;;      DIRECTORY ENTRY TYPE CODE BIT DEFINITIONS :

     (0001)     TFREE   EQU     01H     ;"FREE SPACE" ENTRY - BYTE VALUE

     (0001)     TPROT   EQU     01H     ;PROTECTED FILE
     (0002)     TFILE   EQU     02H     ;PERMANENT FILE ENTRY


0A95 11580A    FCS:    LXI     D,CMTAB ;POINT TO COMMAND TABLE
0A98 E5        F1:     PUSH    H       ;SAVE COMMAND LINE PNTR
0A99 0603              MVI     B,3     ;SET COUNTER
0A9B 1A        F2:     LDAX    D       ;GET NEXT COMMAND CHARACTER
0A9C 13                INX     D
0A9D BE                CMP     M       ;MATCH ?
0A9E 23                INX     H
0A9F CAB00A            JZ      F4      ;YES!
0AA2 13        F3:     INX     D       ;SKIP TO ...
0AA3 05                DCR     B       ;... NEXT ...
0AA4 F2A20A            JP      F3      ;... COMMAND
0AA7 E1                POP     H       ;GET COMMAND LINE PNTR
0AA8 1A                LDAX    D       ;GET BYTE
0AA9 A7                ANA     A       ;END OF COMMAND TABLE ?
0AAA C2980A            JNZ     F1      ;NO: TRY NEXT COMMAND!
0AAD 0603      IVC:    MVI     B,EIVC  ;SET ERROR CODE
0AAF C9                RET             ;ERROR EXIT: INVALID COMMAND!
0AB0 05        F4:     DCR     B       ;ENOUGH ?
0AB1 C29B0A            JNZ     F2      ;NO: CHECK NEXT CHARACTER!
0AB4 E3                XTHL
0AB5 21CB0A            LXI     H,FCSEX ;POINT TO EXIT ROUTINE
```

```
0AB8 E3                 XTHL
0AB9 EB                 XCHG
0ABA 4E                 MOV     C,M     ;GET LOBYTE OF ROUTINE ADDRESS
0ABB 23                 INX     H
0ABC 46                 MOV     B,M     ;GET HIBYTE OF ROUTINE ADDRESS
0ABD C5                 PUSH    B       ;PUT ROUTINE ADDRESS ON STACK
0ABE 21F780             LXI     H,FPB   ;POINT TO FPB
0AC1 22F380             SHLD    FPBP    ;SAVE POINTER
0AC4 EB                 XCHG
0AC5 CDB418             CALL    LTNOR   ;SKIP TRAILING LETTERS
0AC8 C39618             JMP     SPNOR   ;SKIP SPACES & GO TO ROUTINE!

0ACB C5         FCSEX:  PUSH    B       ;SAVE STATUS CODE
0ACC CD480B             CALL    RESET   ;RESET DEVICES
0ACF C1                 POP     B       ;RESTORE STATUS CODE
0AD0 78                 MOV     A,B     ;GET STATUS CODE
0AD1 A7                 ANA     A       ;TEST IT
0AD2 C9                 RET             ;RETURN TO CALLER

0AD3 CD950A     FCSEM:  CALL    FCS     ;PROCESS COMMAND LINE
0AD6 78         EMESS:  MOV     A,B     ;COPY STATUS CODE
0AD7 A7                 ANA     A       ;TEST IT
0AD8 C8                 RZ              ;NO ERRORS!
0AD9 F5                 PUSH    PSW     ;SAVE ERROR CODE & FLAGS
0ADA 21F10A             LXI     H,FERS1
0ADD CD2A18             CALL    OSTR    ;PRINT MESSAGE
0AE0 21000B             LXI     H,FERS2-3
0AE3 48                 MOV     C,B
0AE4 0600               MVI     B,0
0AE6 09                 DAD     B
0AE7 0603               MVI     B,3
0AE9 CDF618             CALL    PSTR    ;PRINT ERROR CODE
0AEC CDC117             CALL    CRLF
0AEF F1                 POP     PSW     ;RESTORE ERROR CODE & FLAGS
0AF0 C9                 RET             ;RETURN TO CALLER

                ;;      ERROR CODE DEFINITIONS :

0AF1 FF06010B   FERS1:  DB      255,6,1,11,'FCS ERROR - E',239
0AF5 46435320
0AF9 4552524F
0AFD 52202D20
0B01 45EF

0B03            FERS2:
0B03 495643             DB      'IVC'   ; INVALID COMMAND
     (0003)     EIVC    EQU     $-FERS2
0B06 4D564E             DB      'MVN'   ; MISSING VOLUME NAME
     (0006)     EMVN    EQU     $-FERS2
0B09 53594E             DB      'SYN'   ; SYNTAX ERROR
     (0009)     ESYN    EQU     $-FERS2
0B0C 444952             DB      'DIR'   ; DIRECTORY ERROR
     (000C)     EDIR    EQU     $-FERS2
0B0F 495650             DB      'IVP'   ; INVALID PARAMETER(S)
     (000F)     EIVP    EQU     $-FERS2
0B12 4E5645             DB      'NVE'   ; NO VOLUME ENTRY IN DIRECTORY
     (0012)     ENVE    EQU     $-FERS2
```

```
OB15 4D464E             DB      'MFN'   ; MISSING FILE NAME
     (0015)   EMFN      EQU     $-FERS2
OB18 4D4456             DB      'MDV'   ; MISSING DEVICE NAME
     (0018)   EMDV      EQU     $-FERS2
OB1B 4D5652             DB      'MVR'   ; MISSING VERSION
     (001B)   EMVR      EQU     $-FERS2
OB1E 495644             DB      'IVD'   ; INVALID DEVICE
     (001E)   EIVD      EQU     $-FERS2
OB21 4E5341             DB      'NSA'   ; NO START ADDRESS
     (0021)   ENSA      EQU     $-FERS2
OB24 44464E             DB      'DFN'   ; DUPLICATE FILE NAME
     (0024)   EDFN      EQU     $-FERS2
OB27 564F56             DB      'VOV'   ; VERSION NUMBER OVERFLOW
     (0027)   EVOV      EQU     $-FERS2
OB2A 464E46             DB      'FNF'   ; FILE NOT FOUND
     (002A)   EFNF      EQU     $-FERS2
OB2D 445246             DB      'DRF'   ; DIRECTORY FULL
     (002D)   EDRF      EQU     $-FERS2
OB30 465244             DB      'FRD'   ; FILE READ ERROR
     (0030)   EFRD      EQU     $-FERS2
OB33 465752             DB      'FWR'   ; FILE WRITE ERROR
     (0033)   EFWR      EQU     $-FERS2
OB36 57535A             DB      'WSZ'   ; FILE TOO LARGE FOR WRITE
     (0036)   EWSZ      EQU     $-FERS2
OB39 52535A             DB      'RSZ'   ; FILE TOO LARGE FOR READ
     (0039)   ERSZ      EQU     $-FERS2
OB3C 44454C             DB      'DEL'   ; DELETE ERROR
     (003C)   EDEL      EQU     $-FERS2
OB3F 434F50             DB      'COP'   ; ERROR DURING COPY
     (003F)   ECOP      EQU     $-FERS2
OB42 495654             DB      'IVT'   ; INVALID FILE TYPE
     (0042)   EIVT      EQU     $-FERS2
OB45 424C46             DB      'BLF'   ; BAD '.LDA' FILE
     (0045)   EBLF      EQU     $-FERS2

OB48 010301  RESET:     LXI     B,103H  ;SET FLAG & COUNTER
OB4B 214300             LXI     H,HDVCT ;POINT TO HANDLER VECTOR AREA
OB4E 3EC3    RS01:      MVI     A,0C3H  ;SET 'JMP' BYTE
OB50 BE                 CMP     M       ;VECTOR EMPTY ?
OB51 C2710B             JNZ     RS02    ;YES: SKIP IT!
OB54 E5                 PUSH    H       ;SAVE HANDLER PNTR
OB55 C5                 PUSH    B       ;SAVE FLAG & COUNTER
OB56 EB                 XCHG            ;DE=HANDLER PNTR
OB57 21880B             LXI     H,OFFPB ;POINT TO HANDLER PB
OB5A CD3713             CALL    JMPD    ;CALL HANDLER
OB5D C1                 POP     B       ;RESTORE FLAG & COUNTER
OB5E E1                 POP     H       ;GET HANDLER PNTR
OB5F E5                 PUSH    H       ;SAVE HANDLER PNTR
OB60 23                 INX     H       ;POINT TO DEVICE NAME ...
OB61 23                 INX     H
OB62 23                 INX     H
OB63 5E                 MOV     E,M     ;GET 1ST CHAR OF NAME
OB64 23                 INX     H
OB65 56                 MOV     D,M     ;GET 2ND CHAR OF NAME
OB66 2AF080             LHLD    DFDV    ;GET DEFAULT DEVICE NAME
OB69 CD8918             CALL    CMPDH   ;MATCH ?
OB6C E1                 POP     H       ;RESTORE HANDLER PNTR
```

```
0B6D C2710B             JNZ     RS02    ;NO MATCH: STEP TO NEXT!
0B70 04                 INR     B       ;INDICATE DEFAULT DEVICE SEEN
0B71 110A00     RS02:   LXI     D,10
0B74 19                 DAD     D       ;STEP TO NEXT HANDLER
0B75 0D                 DCR     C       ;ANY MORE ?
0B76 C24E0B             JNZ     RS01    ;YES: LOOP!
0B79 05                 DCR     B       ;WAS DEFAULT DEVICE SEEN ?
0B7A C0                 RNZ             ;YES: EXIT!
0B7B 2A4000             LHLD    IDEV    ;GET INITIAL DEFAULT DEVICE
0B7E 22F080             SHLD    DFDV    ;SET DEFAULT DEVICE
0B81 3A4200             LDA     IUNT    ;GET INITIAL UNIT NO.
0B84 32F280             STA     DFUN    ;STORE IT
0B87 C9                 RET             ;EXIT!

0B88 FC00       OFFPB:  DB      -4,0    ;"TURN OFF" FUNCTION


0B8A CD9618     CKEND:  CALL    SPNOR   ;SKIP SPACES
0B8D C8                 RZ              ;RETURN <Z> IF END OF LINE
0B8E 0609       ERSYN:  MVI     B,ESYN  ;SET ERROR CODE
0B90 C9                 RET             ;RETURN <NZ> IF NOT END OF LINE


0B91 3E01       WRI00:  MVI     A,1     ;SET WRITE CODE
0B93 FE                 DB      0FEH    ;*** SKIP 1 BYTE ***
0B94 AF         REA00:  XRA     A       ;SET READ CODE
0B95 321381             STA     FFCN    ;SAVE FUNCTION CODE
0B98 CD1414             CALL    PDV     ;GET DEVICE NAME
0B9B D8                 RC              ;ERROR!
0B9C CD2C19             CALL    GN2Z    ;GET BLOCK NUMBER
0B9F D20611             JNC     GM02    ;NO NUMBER: ERROR!
0BA2 EB                 XCHG
0BA3 221581             SHLD    FBLK    ;STORE BLOCK NUMBER
0BA6 EB                 XCHG
0BA7 CDDA10             CALL    GMPRM   ;GET MEMORY PARAMS
0BAA D8                 RC              ;ERROR!
0BAB CD8A0B             CALL    CKEND   ;CHECK FOR END OF LINE
0BAE C0                 RNZ             ;NO: ERROR!
0BAF EB                 XCHG
0BB0 221781             SHLD    FBUF    ;STORE BUFFER ADDRESS
0BB3 60                 MOV     H,B
0BB4 69                 MOV     L,C
0BB5 221981             SHLD    FXBC    ;STORE BYTE COUNT
0BB8 3A1381             LDA     FFCN    ;GET FUNCTION CODE
0BBB 211181             LXI     H,FHAN
0BBE CD3213             CALL    CHDLR   ;CALL HANDLER
0BC1 0600       GDRET:  MVI     B,0
0BC3 C9                 RET


                ;;      INITIALIZE VOLUME DIRECTORY :

0BC4 CD1414     INI00:  CALL    PDV     ;PARSE DEVICE NAME
0BC7 D8                 RC              ;ERROR: INVALID DEVICE!
0BC8 C8                 RZ              ;ERROR: MISSING DEVICE NAME!
0BC9 CD9618             CALL    SPNOR   ;SKIP SPACES
0BCC 112081             LXI     D,DBF+3 ;POINT TO BUFFER
0BCF 060A               MVI     B,10    ;SET COUNTER
```

```
0BD1 CDCB18           CALL    MSTR      ;MOVE VOLUME NAME
0BD4 0606             MVI     B,EMVN    ;SET ERROR CODE
0BD6 C8               RZ                ;ERROR: MISSING VOLUME NAME!
0BD7 CDBE18           CALL    GCMA      ;SKIP COMMA IF IT'S THERE
0BDA CD2C19           CALL    GN2Z      ;GET NUMBER IF IT'S THERE
0BDD CD8A0B           CALL    CKEND     ;CHECK FOR END OF LINE
0BE0 C0               RNZ               ;NO: ERROR!
0BE1 D5               PUSH    D         ;SAVE USER'S NUMBER
0BE2 211181           LXI     H,FHAN    ;POINT TO HANDLER ADDRESS
0BE5 E5               PUSH    H         ;SAVE PNTR
0BE6 3EFE             MVI     A,-2      ;SET "ON" CODE
0BE8 CD3213           CALL    CHDLR     ;CALL HANDLER
0BEB E1               POP     H         ;RESTORE PNTR
0BEC 3EFF             MVI     A,-1      ;SET "GET SIZE" CODE
0BEE CD3213           CALL    CHDLR     ;CALL HANDLER
0BF1 D1               POP     D         ;RESTORE USER'S NUMBER
0BF2 060C             MVI     B,EDIR    ;SET ERROR CODE
0BF4 D8               RC                ;ERROR GETTING SIZE!
0BF5 E5               PUSH    H         ;SAVE DEVICE SIZE
0BF6 7B               MOV     A,E       ;GET USER-SPECIFIED DIR. SIZE
0BF7 A7               ANA     A         ;IS IT ZERO ?
0BF8 C2040C           JNZ     INI01     ;NO: USE IT!
0BFB 29               DAD     H
0BFC 7C               MOV     A,H       ;TOTAL BLKS /128
0BFD DA090C           JC      INI02     ;OVERFLOW: USE MAX
0C00 FE01             CPI     1         ;ENSURE AT LEAST ONE ...
0C02 CE00             ACI     0
0C04 FE21     INI01:  CPI     DSBUFS/128 + 1  ;VALID ?
0C06 DA0B0C           JC      INI03             ;YES: USE IT!
0C09 3E20     INI02:  MVI     A,DSBUFS/128      ;SET MAX. NO. OF DIR. BLOCKS
0C0B 5F       INI03:  MOV     E,A
0C0C 1600             MVI     D,0       ;DE=STRT BLK OF FREE SPACE
0C0E 210000           LXI     H,0
0C11 221581           SHLD    FBLK      ;SET BLOCK NUMBER
0C14 211D81           LXI     H,DBF     ;POINT TO BUFFER
0C17 221781           SHLD    FBUF      ;SET BUFFER PNTR
0C1A 72               MOV     M,D       ;SET DIR. BLK. NO. ZERO
0C1B 23               INX     H
0C1C 3D               DCR     A
0C1D 77               MOV     M,A       ;SET MAX. DIR. BLK. NO.
0C1E 23               INX     H
0C1F 3641             MVI     M,41H     ;SET "VOLUME" ATTRIBUTE BYTE
0C21 E1               POP     H         ;GET DEVICE SIZE
0C22 CD8F18           CALL    SUBHD     ;COMPUTE NO. OF FREE BLOCKS
0C25 44               MOV     B,H
0C26 4D               MOV     C,L
0C27 213481           LXI     H,DBF+2+21
0C2A CDCC13           CALL    SFR       ;SETUP FREE ENTRY
0C2D CDAB13           CALL    WRDIR     ;WRITE DIRECTORY BLOCK
0C30 D8               RC                ;ERROR!
0C31 C33C0C           JMP     DIR01     ;GO LIST DIRECTORY!

                ;;      LIST DIRECTORY

0C34 CD1414   DIR00:  CALL    PDV       ;GET DEVICE NAME
0C37 D8               RC                ;ERROR: INVALID DEVICE!
0C38 CD8A0B           CALL    CKEND     ;CHECK FOR END OF LINE
```

```
0C3B C0                 RNZ             ;NO: ERROR!
0C3C 11F780   DIR01:    LXI     D,FPB   ;POINT TO FPB
0C3F 212D11             LXI     H,MS1   ;POINT TO DIR. HEADER MSG.
0C42 CD2A18             CALL    OSTR    ;PRINT IT
0C45 CD0A11             CALL    PRDEV   ;PRINT DEVICE NAME
0C48 111181             LXI     D,FHAN  ;POINT TO FHAN
0C4B CD9611             CALL    OPDIR   ;'OPEN' DIRECTORY
0C4E D8                 RC              ;ERROR!
0C4F C5                 PUSH    B       ;SAVE ...
0C50 D5                 PUSH    D       ;... THINGS ...
0C51 E5                 PUSH    H       ;...
0C52 23                 INX     H       ;SKIP TYPE CODE
0C53 060A               MVI     B,10    ;SET COUNT
0C55 CDF318             CALL    PSSTR   ;PRINT VOLUME LABEL
0C58 CDE918             CALL    PSPAC   ;PRINT A SPACE
0C5B 3A1E81             LDA     DBF+1   ;GET MAX. DIR. BLK NO.
0C5E 3C                 INR     A       ;CHANGE TO NO. OF BLOCKS
0C5F CDD117             CALL    LBYT    ;PRINT IT
0C62 213A11             LXI     H,MS2   ;POINT TO HEADING
0C65 CD2A18             CALL    OSTR    ;PRINT HEADING
0C68 E1       DIR02:    POP     H       ;RESTORE ...
0C69 D1                 POP     D       ;... THINGS ...
0C6A C1                 POP     B       ;...
0C6B CDC117             CALL    CRLF    ;PRINT CR&LF
0C6E CD0B18             CALL    RTST    ;GOT A KEY ?
0C71 C27C0C             JNZ     DIR05   ;NO: PROCEED !
0C74 FE0A               CPI     10      ;IS IT LF ?
0C76 CCC817             CZ      LO      ;ECHO LF !
0C79 CAC10B             JZ      QDRET   ;YES: STOP !
0C7C CDBC11   DIR05:    CALL    QNDE    ;GET NEXT DIRECTORY ENTRY
0C7F D8                 RC              ;ERROR!
0C80 CACD0C             JZ      DIR04   ;FINISHED!
0C83 C5                 PUSH    B       ;SAVE ...
0C84 D5                 PUSH    D       ;... THINGS ...
0C85 E5                 PUSH    H       ;...
0C86 46                 MOV     B,M     ;GET TYPE CODE
0C87 CD0019             CALL    PSBYT   ;PRINT TYPE CODE
0C8A 3E01               MVI     A,TFREE ;GET 'FREE' TYPE CODE
0C8C B8                 CMP     B       ;IS THIS THE 'FREE' ENTRY ?
0C8D CAB90C             JZ      DIR03   ;YES!
0C90 0606               MVI     B,6     ;SET COUNTER
0C92 CDF318             CALL    PSSTR   ;PRINT NAME
0C95 3E2E               MVI     A,'.'   ;PRINT A ...
0C97 CDC817             CALL    LO      ;... PERIOD
0C9A 0603               MVI     B,3     ;SET COUNTER
0C9C CDF618             CALL    PSTR    ;PRINT TYPE
0C9F 3E3B               MVI     A,';'   ;PRINT A ...
0CA1 CDC817             CALL    LO      ;... SEMICOLON
0CA4 CD0319             CALL    PBYT    ;PRINT VERSION NUMBER
0CA7 CD0B19             CALL    PSNUM   ;PRINT STARTING BLOCK NUMBER
0CAA CD0B19             CALL    PSNUM   ;PRINT SIZE
0CAD CD0019             CALL    PSBYT   ;PRINT LAST BLK BYTE COUNT
0CB0 CD0819             CALL    P2SNUM  ;PRINT LOAD ADDRESS
0CB3 CD0B19             CALL    PSNUM   ;PRINT START ADDRESS
0CB6 C3680C             JMP     DIR02   ;STEP TO NEXT ENTRY!
0CB9 EB       DIR03:    XCHG
0CBA 216911             LXI     H,MS3   ;POINT TO STRING
0CBD CD2A18             CALL    OSTR    ;PRINT IT
```

```
0CC0 210A00             LXI     H,FSBK-FNAM
0CC3 19                 DAD     D           ;POINT TO FSBK
0CC4 CD0B19             CALL    PSNUM       ;PRINT STARTING BLOCK NUMBER
0CC7 CD0B19             CALL    PSNUM       ;PRINT SIZE
0CCA CDC117             CALL    CRLF
0CCD E1         DIR04:  POP     H           ;CLEAN ...
0CCE E1                 POP     H           ;... THE ...
0CCF E1                 POP     H           ;... STACK
0CD0 CDC117             CALL    CRLF
0CD3 0600               MVI     B,0         ;SET O.K. CODE
0CD5 C9                 RET                 ;THAT'S ALL!

                ;;      SAVE "IMAGE" TYPE FILE :

0CD6 CDAA14     SAV00:  CALL    PPFSP       ;GET FILE SPECIFICATION
0CD9 D8                 RC                  ;ERROR!
0CDA CDDA10             CALL    GMPRM       ;GET MEMORY PARAMS
0CDD D8                 RC                  ;ERROR!
0CDE EB                 XCHG
0CDF 220881             SHLD    FLAD        ;STORE LOAD ADDRESS
0CE2 EB                 XCHG
0CE3 CD1D19             CALL    GN1D        ;GET START ADDRESS
0CE6 EB                 XCHG
0CE7 220A81             SHLD    FSAD        ;STORE START ADDRESS
0CEA EB                 XCHG
0CEB CD1A19             CALL    GN1Z        ;GET AUX. MEM. ADDRESS
0CEE CD8A0B             CALL    CKEND       ;CHECK FOR END OF LINE
0CF1 C0                 RNZ                 ;NO: ERROR!
0CF2 3E01       WF2:    MVI     A,FNEW      ;SET OPEN TYPE CODE
0CF4 CDBC10             CALL    OPENX       ;OPEN THE FILE
0CF7 D8                 RC                  ;ERROR!
0CF8 CD0213             CALL    WRITE       ;WRITE THE FILE
0CFB D8                 RC                  ;ERROR!
0CFC CD5C13     CLX:    CALL    CLOSE       ;CLOSE THE FILE
0CFF D8                 RC                  ;ERROR!
0D00 111A00             LXI     D,FHAN-FPB
0D03 19                 DAD     D           ;POINT TO FHAN
0D04 3EFD               MVI     A,-3        ;SET "SUB USER" FUNCTION
0D06 CD3213             CALL    CHDLR       ;CALL THE HANDLER
0D09 AF                 XRA     A           ;CLEAR C-FLAG
0D0A 47                 MOV     B,A         ;SET GOOD STATUS
0D0B C9                 RET                 ;EXIT

                ;;      LOAD A FILE :

0D0C 019011     LOA00:  LXI     B,LTYP      ;SET DEFAULT TYPE
0D0F CDAD14             CALL    PFSPC       ;GET FILE SPECIFICATION
0D12 D8                 RC                  ;ERROR !
0D13 CD4E0F             CALL    QLDA        ;IS IT '.LDA' ?
0D16 CA320D             JZ      LOLDA       ;YES !

                ;;      LOAD AN "IMAGE" FILE :

0D19 CDBE18             CALL    GCMA        ;SKIP COMMA IF IT'S THERE
0D1C CD2C19             CALL    GN2Z        ;GET AUX. LOAD ADDRESS
0D1F CD8A0B             CALL    CKEND       ;CHECK FOR END OF LINE
```

```
0D22 C0                 RNZ             ;NO: ERROR!
0D23 01FFFF     RF1:    LXI     B,0FFFFH ;SET MAX. BYTE COUNT
0D26 AF         RF2:    XRA     A       ;SET OPEN TYPE CODE
0D27 CDBC10             CALL    OPENX   ;OPEN THE FILE
0D2A D8                 RC              ;ERROR!
0D2B CDD912             CALL    READ    ;READ THE FILE
0D2E D8                 RC              ;ERROR!
0D2F C3FC0C             JMP     CLX     ;CLOSE FILE & EXIT!

                ;;      LOAD A '.LDA' FILE :

0D32 CD290E     LOLDA:  CALL    SULD    ;SETUP FOR '.LDA' LOAD
0D35 CD8A0B             CALL    CKEND   ;END OF LINE ?
0D38 CA700D             JZ      LLDA    ;YES: LOAD ALL AT CORRECT ADDRESS !
0D3B CDBE18             CALL    GCMA    ;SKIP COMMA IF THERE
0D3E CD2C19             CALL    GN2Z    ;GET 'LOWEST ADDRESS TO LOAD'
0D41 D28E0B             JNC     ERSYN   ;SYNTAX ERROR !
0D44 EB                 XCHG
0D45 229D81             SHLD    XFHAN   ;SAVE 'LOWEST ADDRESS TO LOAD'
0D48 210082             LXI     H,08200H ; <<9/78>>  CHANGE DEFAULT LOWEST ADDRESS
0D4B 229F81             SHLD    XFFCN   ;SET DEFAULT 'LOWEST MEMORY ADDRESS'
0D4E EB                 XCHG
0D4F CD8A0B             CALL    CKEND   ;END OF LINE ?
0D52 CA700D             JZ      LLDA    ;YES: USE DEFAULT MEMORY SPECS !
0D55 CDDA10             CALL    GMPRM   ;GET MEMORY PARAMS
0D58 D8                 RC              ;ERROR !
0D59 CD8A0B             CALL    CKEND   ;END OF LINE ?
0D5C C0                 RNZ             ;NO: ERROR !
0D5D 78                 MOV     A,B     ;TEST BYTE COUNT ...
0D5E B1                 ORA     C
0D5F CA6C0D             JZ      LOL1    ;ZERO: USE DEFAULT 'HIGHEST' !
0D62 60                 MOV     H,B     ;COPY BYTE COUNT ...
0D63 69                 MOV     L,C
0D64 2B                 DCX     H       ;ADJUST IT
0D65 19                 DAD     D       ;COMPUTE END ADDRESS
0D66 060F               MVI     B,EIVP
0D68 D8                 RC              ;ERROR !
0D69 22A181             SHLD    XFBLK   ;SAVE 'HIGHEST MEMORY ADDRESS'
0D6C EB         LOL1:   XCHG
0D6D 229F81             SHLD    XFFCN   ;SAVE 'LOWEST MEMORY ADDRESS'

0D70 2A9D81     LLDA:   LHLD    XFHAN   ;GET 'LOWEST ADDRESS TO LOAD'
0D73 EB                 XCHG
0D74 2A9F81             LHLD    XFFCN   ;GET 'LOWEST MEMORY ADDRESS'
0D77 CD8F18             CALL    SUBHD   ;COMPUTE OFFSET
0D7A 229D81             SHLD    XFHAN   ;SAVE OFFSET
0D7D AF                 XRA     A       ;SET OPEN TYPE
0D7E CDBC10             CALL    OPENX   ;OPEN THE FILE
0D81 D8                 RC              ;ERROR !
0D82 CDFC14             CALL    RWSEQI  ;INITIALIZE FOR SEQUENTIAL INPUT
0D85 EB                 XCHG
0D86 210000             LXI     H,0     ;INITIAL START ADDRESS
0D89 220A81     LOL3:   SHLD    FSAD    ;STORE START ADDRESS
0D8C EB                 XCHG
0D8D CD6216     LOL4:   CALL    GTBYT   ;GET A BYTE
0D90 DAEA0D             JC      EOFOR   ;ERROR OR EOF !
0D93 4F                 MOV     C,A     ;LOBYTE OF BYTE COUNT
0D94 CD6216             CALL    GTBYT
```

```
0D97 DAF10D          JC      LER1      ;ERROR !
0D9A 47              MOV     B,A       ;HIBYTE OF BYTE COUNT
0D9B B1              ORA     C         ;IS IT ZERO ?
0D9C CA8D0D          JZ      LOL4      ;YES: EMPTY RECORD !
0D9F 0B              DCX     B
0DA0 78              MOV     A,B
0DA1 B1              ORA     C         ;WAS IT ONE ?
0DA2 CAF50D          JZ      LER2      ;YES: FILE FORMAT ERROR !
0DA5 CD6216          CALL    GTBYT
0DA8 DAF10D          JC      LER1      ;ERROR !
0DAB 5F              MOV     E,A       ;LOBYTE OF LOAD ADDRESS
0DAC CD6216          CALL    GTBYT
0DAF DAF10D          JC      LER1      ;ERROR !
0DB2 57              MOV     D,A       ;HIGH BYTE OF LOAD ADDRESS
0DB3 0B              DCX     B
0DB4 78              MOV     A,B
0DB5 B1              ORA     C         ;ANY MORE ?
0DB6 EB              XCHG
0DB7 CA890D          JZ      LOL3      ;NO: STORE START ADDRESS & LOOK FOR MORE !
0DBA EB              XCHG
0DBB E5              PUSH    H         ;SAVE FPB PTR
0DBC 2A9D81          LHLD    XFHAN     ;GET OFFSET
0DBF 19              DAD     D         ;COMPUTE ACTUAL LOAD ADDRESS
0DC0 EB              XCHG
0DC1 E1              POP     H         ;RESTORE FPB PTR
0DC2 CD6216  LOL5:   CALL    GTBYT     ;GET NEXT DATA BYTE
0DC5 DAF10D          JC      LER1      ;ERROR !
0DC8 E5              PUSH    H         ;SAVE FPB PTR
0DC9 F5              PUSH    PSW       ;SAVE THE BYTE
0DCA 2A9F81          LHLD    XFFCN     ;GET LOW LIMIT
0DCD CD8918          CALL    CMPDH     ;LOAD IT ?
0DD0 DAD90D          JC      LOL6      ;NO !
0DD3 2AA181          LHLD    XFBLK     ;GET HIGH LIMIT
0DD6 CD8318          CALL    CMPHD     ;LOAD IT ?
0DD9 E1      LOL6:   POP     H         ;GET BYTE BACK ...
0DDA 7C              MOV     A,H       ;    ... INTO A
0DDB E1              POP     H         ;RESTORE FPB PTR
0DDC DAE00D          JC      LOL7      ;DON'T LOAD THIS BYTE !
0DDF 12              STAX    D         ;LOAD THE BYTE INTO MEMORY
0DE0 13      LOL7    INX     D         ;INCREMENT LOAD ADDRESS
0DE1 0B              DCX     B         ;COUNT THE BYTE
0DE2 78              MOV     A,B
0DE3 B1              ORA     C         ;ANY MORE ?
0DE4 C2C20D          JNZ     LOL5      ;YES: LOOP !
0DE7 C38D0D          JMP     LOL4      ;NO: LOOK FOR NEXT RECORD !
0DEA 3F      EOFOK:  CMC               ;CLEAR <C> STATUS BIT
0DEB F5              PUSH    PSW       ;SAVE STATUS
0DEC CDFC0C          CALL    CLX       ;CLOSE & TURN MOTOR OFF
0DEF F1              POP     PSW       ;RESTORE STATUS
0DF0 C8              RZ                ;END OF FILE: O.K. !
0DF1 0630    LER1:   MVI     B,EFRD
0DF3 37              STC               ;INDICATE ERROR
0DF4 C0              RNZ               ;READ ERROR !
0DF5 0645    LER2:   MVI     B,EBLF
0DF7 37              STC
0DF8 C9              RET               ;FORMAT ERROR !
```

```
                ;;      RUN A PROGRAM :

0DF9 CDAA14     RUN00:  CALL    PPFSP   ;GET FILE SPECIFICATION
0DFC D8                 RC              ;ERROR!
0DFD CD8A0B             CALL    CKEND   ;CHECK FOR END OF LINE
0E00 C0                 RNZ             ;NO: ERROR!
0E01 CD4E0F             CALL    QLDA    ;IS IT '.LDA' ?
0E04 C2100E             JNZ     RUN01   ;NO !
0E07 CD290E             CALL    SULD    ;SETUP FOR '.LDA' LOAD
0E0A CD700D             CALL    LLDA    ;LOAD THE FILE
0E0D C31F0E             JMP     RUN02
0E10 018D11     RUN01:  LXI     B,PTYP
0E13 CD510F             CALL    QTYP    ;IS IT '.PRG' ?
0E16 0642               MVI     B,EIVT
0E18 C0                 RNZ             ;NO: INVALID TYPE !

                ;;      RUN AN "IMAGE" ( TYPE '.PRG' ) PROGRAM :

0E19 110000             LXI     D,0     ;SET AUX. LOAD ADDRESS
0E1C CD230D             CALL    RF1     ;READ THE FILE
0E1F D8         RUN02:  RC              ;ERROR!
0E20 2A0A81             LHLD    FSAD    ;GET START ADDRESS
0E23 7C                 MOV     A,H
0E24 B5                 ORA     L
0E25 0621               MVI     B,ENSA  ;SET ERROR CODE
0E27 C8                 RZ              ;NO START ADDRESS!
0E28 E9                 PCHL            ;GO TO THE PROGRAM!

0E29 E5         SULD:   PUSH    H
0E2A 210000             LXI     H,0
0E2D 229D81             SHLD    XFHAN   ;SET 'LOWEST ADDRESS TO LOAD'
0E30 229F81             SHLD    XFFCN   ;SET 'LOWEST MEMORY ADDRESS'
0E33 2B                 DCX     H
0E34 22A181             SHLD    XFBLK   ;SET 'HIGHEST MEMORY ADDRESS'
0E37 E1                 POP     H
0E38 C9                 RET


0E39 CD1414     DEV00:  CALL    PDV     ;PARSE DEVICE NAME
0E3C D8                 RC              ;ERROR!
0E3D CD8A0B             CALL    CKEND   ;CHECK FOR END OF LINE
0E40 C0                 RNZ             ;NO: ERROR!
0E41 217811             LXI     H,MS4
0E44 CD2A18             CALL    OSTR
0E47 CD0A11             CALL    PRDEV   ;PRINT DEVICE NAME
0E4A CDC117             CALL    CRLF
0E4D 21F080             LXI     H,DFDV  ;POINT TO DEFAULT DEVICE AREA
0E50 73                 MOV     M,E     ;STORE ...
0E51 23                 INX     H       ;... DEVICE ...
0E52 72                 MOV     M,D     ;... NAME
0E53 23                 INX     H
0E54 71                 MOV     M,C     ;STORE UNIT NUMBER
0E55 0600               MVI     B,0     ;INDICATE NO ERRORS
0E57 C9                 RET             ;EXIT!

0E58 018911     DEL00:  LXI     B,NTYP  ;SET DEFAULT TYPE        } CALL PNFSP
0E5B CDAD14             CALL    PFSPC   ;GET FILE SPECIFICATION  }
0E5E D8                 RC              ;ERROR!
```

```
0E5F C8                 RZ              ;NO VERSION: ERROR!
0E60 CD8A0B             CALL    CKEND   ;CHECK FOR END OF LINE
0E63 C0                 RNZ             ;NO: ERROR!
0E64 AF                 XRA     A       ;SET OPEN TYPE CODE <-- not necessary.
0E65 CDBC10             CALL    OPENX   ;OPEN THE FILE
0E68 D8                 RC              ;ERROR!
0E69 2A0D81             LHLD    FDBK    ;GET DIR. BLK. NUMBER ...
0E6C 2600               MVI     H,0     ;...
0E6E 221581             SHLD    FBLK    ;STORE BLOCK NUMBER
0E71 211F81             LXI     H,DBF+2 ;SETUP TO ...
0E74 111181             LXI     D,FHAN  ;... CONTINUE DIR ...
0E77 0606               MVI     B,6     ;... SCAN
0E79 CDBC11     DEL01:  CALL    GNDE    ;GET NEXT DIRECTORY ENTRY
0E7C D8                 RC              ;ERROR!
0E7D FE01               CPI     TFREE   ;IS THIS THE 'FREE' ENTRY ?
0E7F C2790E             JNZ     DEL01   ;NO: KEEP LOOKING FOR IT!
0E82 110B00             LXI     D,FSBK-FATR ;POINT TO FSBK ...
0E85 19                 DAD     D       ;... IN DIR ENTRY
0E86 5E                 MOV     E,M     ;GET STARTING BLOCK ...
0E87 23                 INX     H       ;... NUMBER OF ...
0E88 56                 MOV     D,M     ;... THE 'FREE'
0E89 D5                 PUSH    D       ;SAVE START BLOCK OF FREE
0E8A CDAD10             CALL    CPYDV   ;COPY DEVICE INFO
0E8D 2A0381             LHLD    FSBK    ;GET START BLK OF FILE
0E90 221581             SHLD    FBLK    ;SET DESTINATION START BLK
0E93 EB                 XCHG
0E94 2A0581             LHLD    FSIZ    ;GET FILE SIZE
0E97 19                 DAD     D       ;COMPUTE SOURCE START BLOCK
0E98 22A181             SHLD    XFBLK   ;SET SOURCE START BLOCK
0E9B EB                 XCHG            ;D&E = SOURCE START BLOCK
0E9C E1                 POP     H       ;GET START BLOCK OF FREE
0E9D CD8F18             CALL    SUBHD   ;COMPUTE NO. OF BLKS TO MOVE
0EA0 EB                 XCHG            ;PUT RESULT IN D&E
0EA1 CD5010             CALL    MCHNK   ;MOVE THE DATA
0EA4 DA3E0F             JC      DELER   ;ERROR DURING MOVE!
0EA7 2A0D81             LHLD    FDBK    ;GET DIR. BLK. NUMBER ...
0EAA 2600               MVI     H,0     ;...
0EAC 221581             SHLD    FBLK    ;SET BLOCK NO. FOR XFER
0EAF EB                 XCHG
0EB0 3A1E81             LDA     MDBLK   ;GET MAX. DIR BLK NO.
0EB3 3C                 INR     A       ;NO. OF DIR BLOCKS
0EB4 6F                 MOV     L,A
0EB5 62                 MOV     H,D
0EB6 CD8F18             CALL    SUBHD   ;COMPUTE BLOCKS TO XFER
0EB9 0E80               MVI     C,80H
0EBB CDF019             CALL    BK2BC   ;CHANGE TO BYTE COUNT
0EBE 221981             SHLD    FXBC    ;STORE BYTE COUNT
0EC1 211181             LXI     H,FHAN  ;POINT TO FHAN
0EC4 CD3113             CALL    RD      ;READ THE DIRECTORY
0EC7 DA3E0F             JC      DELER   ;DIR. READ ERROR!
0ECA 218060             LXI     H,DSBUF+2+6*21 ;PNT PAST 1ST BLOCK
0ECD CD410F             CALL    CENTR   ;COMPUTE ENTRY PNTR
0ED0 E5         DEL06:  PUSH    H       ;SAVE PNTR TO DESTINATION SLOT
0ED1 111500             LXI     D,21
0ED4 19                 DAD     D       ;POINT TO SOURCE SLOT
0ED5 EB                 XCHG
0ED6 E1                 POP     H       ;RESTORE DEST. SLOT PNTR
0ED7 E5                 PUSH    H       ;RESAVE IT
```

```
0ED8 78              MOV     A,B       ;GET SLOT NUMBER
0ED9 3D              DCR     A         ;IS DEST. SLOT LAST SLOT IN BLK ?
0EDA C2DF0E          JNZ     DEL07     ;NO: O.K. !
0EDD 13              INX     D         ;ADJUST SOURCE ...
0EDE 13              INX     D         ;... SLOT POINTER
0EDF C5      DEL07:  PUSH    B         ;SAVE B&C
0EE0 0615            MVI     B,21      ;SET COUNT
0EE2 CD7A18          CALL    MOVHD     ;MOVE THE ENTRY
0EE5 C1              POP     B         ;RESTORE B&C
0EE6 E1              POP     H         ;GET DEST. SLOT PNTR
0EE7 E5      DEL08:  PUSH    H         ;SAVE DEST. SLOT POINTER
0EE8 110B00          LXI     D,FSBK-FATR
0EEB 19              DAD     D         ;POINT TO FSBK IN SLOT
0EEC 5E              MOV     E,M       ;GET STARTING ...
0EED 23              INX     H         ;... BLOCK ...
0EEE 56              MOV     D,M       ;... NUMBER
0EEF E5              PUSH    H         ;SAVE POINTER
0EF0 2A0581          LHLD    FSIZ      ;GET SIZE OF DELETED FILE
0EF3 EB              XCHG
0EF4 CD8F18          CALL    SUBHD     ;COMPUTE NEW STARTING BLK NO.
0EF7 EB              XCHG
0EF8 E1              POP     H         ;RESTORE POINTER
0EF9 72              MOV     M,D       ;STORE ADJUSTED ...
0EFA 2B              DCX     H         ;... STARTING ...
0EFB 73              MOV     M,E       ;... BLOCK NUMBER
0EFC E1              POP     H         ;GET DEST. SLOT PNTR
0EFD 7E              MOV     A,M       ;GET TYPE CODE
0EFE FE01            CPI     TFREE     ;IS THIS THE 'FREE' ?
0F00 CA120F          JZ      DEL09     ;YES!
0F03 111500          LXI     D,21
0F06 19              DAD     D         ;STEP TO NEXT SLOT
0F07 05              DCR     B         ;WAS THIS LAST SLOT IN BLOCK ?
0F08 C2D00E          JNZ     DEL06     ;NO: GO DO NEXT SLOT!
0F0B 23              INX     H         ;ADJUST ...
0F0C 23              INX     H         ;... SLOT POINTER
0F0D 0606            MVI     B,6       ;RESET SLOT COUNTER
0F0F C3D00E          JMP     DEL06     ;GO DO NEXT SLOT!
0F12 110D00  DEL09:  LXI     D,FSIZ-FATR
0F15 19              DAD     D         ;POINT TO SIZE OF 'FREE'
0F16 5E              MOV     E,M       ;GET SIZE ...
0F17 23              INX     H         ;... OF ...
0F18 56              MOV     D,M       ;... 'FREE'
0F19 E5              PUSH    H         ;SAVE POINTER
0F1A 2A0581          LHLD    FSIZ      ;GET SIZE OF DELETED FILE
0F1D 19              DAD     D         ;COMPUTE NEW 'FREE' BLOCKS
0F1E EB              XCHG
0F1F E1              POP     H         ;RESTORE POINTER
0F20 72              MOV     M,D       ;STORE NEW ...
0F21 2B              DCX     H         ;... SIZE OF ...
0F22 73              MOV     M,E       ;... 'FREE'
0F23 110800          LXI     D,FDBK-FSIZ
0F26 19              DAD     D         ;POINT TO NEXT SLOT
0F27 05              DCR     B         ;WAS THIS LAST SLOT ?
0F28 C22D0F          JNZ     DEL10     ;NO: O.K. !
0F2B 23              INX     H         ;ADJUST ...
0F2C 23              INX     H         ;... SLOT POINTER
0F2D 3600    DEL10:  MVI     M,0       ;MARK END OF DIRECTORY
0F2F 211181          LXI     H,FHAN    ;POINT TO FHAN
```

```
0F32 CD2E13           CALL    WR        ;WRITE DIRECTORY
0F35 DA3E0F           JC      DELER     ;ERROR!
0F38 CD3010           CALL    ERAS      ;ERASE PAGE
0F3B C33C0C           JMP     DIR01     ;GO LIST DIRECTORY
0F3E 063C     DELER:  MVI     B,EDEL    ;SET ERROR CODE
0F40 C9               RET

0F41 11EBFF   CENTR:  LXI     D,-21
0F44 3A0E81           LDA     FDEN      ;GET ENTRY NO. OF FILE
0F47 47               MOV     B,A       ;COPY IT
0F48 19       CEN01:  DAD     D         ;BACK UP ONE ENTRY
0F49 3D               DCR     A         ;ENOUGH ?
0F4A C2480F           JNZ     CEN01     ;NO: DO IT AGAIN!
0F4D C9               RET               ;RETURN

0F4E 019011   GLDA:   LXI     B,LTYP
0F51 E5       GTYP:   PUSH    H         ;SAVE H
0F52 21FF80           LXI     H,FTYP
0F55 1E03             MVI     E,3       ;SET COUNTER
0F57 0A       CMT1:   LDAX    B         ;GET BYTE
0F58 03               INX     B
0F59 BE               CMP     M         ;COMPARE
0F5A 23               INX     H
0F5B C2620F           JNZ     CMT2      ;DOES NOT MATCH !
0F5E 1D               DCR     E         ;CHECKED ALL THREE ?
0F5F C2570F           JNZ     CMT1      ;NO: CHECK NEXT !
0F62 E1       CMT2:   POP     H         ;RESTORE H
0F63 C9               RET               ;<Z>: MATCH, <NZ>: NO MATCH !

0F64 22A181   REN00:  SHLD    XFBLK     ;SAVE STRING PNTR
0F67 CDA414           CALL    PNFSP     ;GET OLD NAME
0F6A D8               RC                ;ERROR!
0F6B CD4210           CALL    GETTO     ;GET "TO"
0F6E C0               RNZ               ;ERROR!
0F6F CDAD10           CALL    CPYDV     ;COPY DEVICE INFO
0F72 CDBD14           CALL    PCFSP     ;GET NEW NAME
0F75 D8               RC                ;ERROR!
0F76 CD8A0B           CALL    CKEND     ;CHECK FOR END OF LINE
0F79 C0               RNZ               ;NO: ERROR!
0F7A 2A9D81           LHLD    XFHAN     ;FORCE ...
0F7D 221181           SHLD    FHAN      ;... SAME ...
0F80 3AA081           LDA     XFDRV     ;... DEVICE ...
0F83 321481           STA     FDRV      ;...
0F86 CD3610           CALL    OPOX      ;OPEN THE FILE
0F89 D8               RC                ;ERROR!
0F8A 11F980           LXI     D,FNAM    ;POINT TO NEW NAME
0F8D 21A381           LXI     H,XFBUF   ;POINT TO TEMP AREA
0F90 060A             MVI     B,10      ;SET COUNT
0F92 CD7A18           CALL    MOVHD     ;COPY NEW NAME
0F95 2AA181           LHLD    XFBLK     ;GET STRNG PNTR TO OLD NAME
0F98 11F780           LXI     D,FPB     ;POINT TO FPB
0F9B CDA414           CALL    PNFSP     ;GET OLD NAME AGAIN
0F9E AF               XRA     A         ;SET OPEN TYPE CODE
0F9F CDBC10           CALL    OPENX     ;OPEN OLD FILE
0FA2 D8               RC                ;ERROR!
0FA3 CDEB13           CALL    GODBK     ;GET DIR BLK
```

```
0FA6 D8              RC              ;ERROR!
0FA7 21[illegible]   LXI   H,DBF+2+6*21+1
0FAA CD410F          CALL  CENTR     ;COMPUTE ENTRY PNTR
0FAD 11A381          LXI   D,XFBUF   ;POINT TO TEMP AREA
0FB0 060A            MVI   B,10      ;SET COUNT
0FB2 CD7A18          CALL  MOVHD     ;MOVE IN NEW NAME
0FB5 2A0D81          LHLD  FDBK      ;GET DIR BLK NO.
0FB8 2600            MVI   H,0
0FBA 221581          SHLD  FBLK      ;SET BLOCK NUMBER
0FBD 219D81          LXI   H,DBFE    ;SET END PNTR
0FC0 C3AB13          JMP   WRDIR     ;GO WRITE DIR BLK !

0FC3 22A181  COP00:  SHLD  XFBLK     ;SAVE STRING PNTR
0FC6 CDA414          CALL  PNFSP     ;GET SOURCE NAME
0FC9 D8              RC              ;ERROR!
0FCA CD4210          CALL  GETTO     ;GET "TO"
0FCD C0              RNZ             ;ERROR!
0FCE 22AB81          SHLD  TMP1      ;SAVE STRING PNTR
0FD1 CDBD14          CALL  PCFSP     ;GET DESTINATION NAME
0FD4 D2DB0F          JNC   COP01     ;O.K. !
0FD7 3E15            MVI   A,EMFN
0FD9 B8              CMP   B         ;MISSING FILE NAME ?
0FDA C0              RNZ             ;NO: ERROR!
0FDB CD8A0B  COP01:  CALL  CKEND     ;CHECK FOR END OF LINE
0FDE C0              RNZ             ;NO: ERROR!
0FDF 2AA181          LHLD  XFBLK     ;GET SOURCE PNTR
0FE2 CDA414          CALL  PNFSP     ;GET SOURCE NAME
0FE5 CDAD10          CALL  CPYDV     ;COPY DEVICE INFO
0FE8 AF              XRA   A         ;SET OPEN TYPE CODE
0FE9 CDBC10          CALL  OPENX     ;OPEN SOURCE FILE
0FEC D8              RC              ;ERROR!
0FED 110381          LXI   D,FSBK    ;SET SOURCE PNTR
0FF0 21A181          LXI   H,XFBLK   ;SET DESTINATION PNTR
0FF3 060A            MVI   B,FDBK-FSBK ;SET COUNT
0FF5 CD7A18          CALL  MOVHD     ;COPY FILE INFO
0FF8 11F780          LXI   D,FPB     ;POINT TO FPB
0FFB 2AAB81          LHLD  TMP1      ;GET STRING PNTR
0FFE CDBD14          CALL  PCFSP     ;GET DESTINATION NAME
1001 3E01            MVI   A,FNEW    ;SET OPEN TYPE CODE
1003 CDBC10          CALL  OPENX     ;OPEN DESTINATION FILE
1006 D8              RC              ;ERROR!
1007 2AA381          LHLD  XFBLK+FSIZ-FSBK ;GET SOURCE FILE SIZE
100A 220F81          SHLD  FAUX      ;SET SIZE FOR CLOSE
100D EB              XCHG
100E 2A0581          LHLD  FSIZ      ;GET "FREE" SIZE
1011 CD8318          CALL  CMPHD     ;SEE IF IT'LL FIT
1014 0636            MVI   B,EWSZ    ;SET ERROR CODE
1016 D8              RC              ;TOO BIG: ERROR!
1017 D5              PUSH  D         ;SAVE FILE SIZE
1018 11A581          LXI   D,XFBLK+FLBC-FSBK ;SET SOURCE PNTR
101B 210781          LXI   H,FLBC    ;SET DESTINATION PNTR
101E 0606            MVI   B,FDBK-FLBC ;SET COUNT
1020 CD7A18          CALL  MOVHD     ;COPY FILE INFO BACK
1023 D1              POP   D         ;GET FILE SIZE BACK
1024 CD5010          CALL  MCHNK     ;COPY THE FILE
1027 063F            MVI   B,ECOP    ;SET ERROR CODE
1029 D8              RC              ;ERROR!
```

UA4

UA5

```
102A 21F780          LXI     H,FPB     ;POINT TO FPB
102D C3FC0C          JMP     CLX       ;GO CLOSE FILE & EXIT!

1030 CD2E0A  ERAS:   CALL    SAVE      ;  SAVE REGISTERS
1033 C37F02          JMP     ERASE     ;  ERASE SCREEN AND RETURN

1036 3E01    OPOX:   MVI     A,FNEW    ;SET OPEN TYPE CODE
1038 CDBC10          CALL    OPENX     ;OPEN THE FILE
103B D0              RNC               ;NO ERRORS: RETURN
103C 3E2D            MVI     A,EDRF    ;IS IT ...
103E A8              XRA     B         ;... DIRECTORY FULL ?
103F C8              RZ                ;YES: O.K. RETURN!
1040 37              STC               ;INDICATE ERROR
1041 C9              RET               ;RETURN

1042 0609    GETTO:  MVI     B,ESYN    ;SET ERROR CODE
1044 CD9618          CALL    SPNOR     ;GET 1ST NON-SPACE
1047 FE54            CPI     'T'       ;IS IT 'T' ?
1049 C0              RNZ               ;NO: ERROR!
104A 23              INX     H
104B 7E              MOV     A,M       ;GET NEXT CHARACTER
104C 23              INX     H
104D FE4F            CPI     'O'       ;IS IT 'O' ?
104F C9              RET               ;RETURN

1050 210060  MCHNK:  LXI     H,DSBUF   ;POINT TO DISPLAY BUFFER
1053 22A381          SHLD    XFBUF     ;SET BUFFER ...
1056 221781          SHLD    FBUF      ;... POINTERS
1059 D5              PUSH    D         ;SAVE NO. OF BLOCKS
105A 3EFE            MVI     A,-2      ;SET "ADD USER" CODE
105C CD9F10          CALL    CTWO      ;CALL HANDLERS
105F D1              POP     D         ;GET NO. OF BLOCKS
1060 CD3010  MCH01:  CALL    ERAS      ;ERASE PAGE
1063 7B              MOV     A,E       ;ANY ...
1064 B2              ORA     D         ;... LEFT ?
1065 CA9D10          JZ      STWO      ;NO: FINISHED!
1068 EB              XCHG
1069 112000          LXI     D,DSBUFS/128    ;SET BUFFER SIZE
106C CD8F18          CALL    SUBHD     ;SUB. FROM NO. LEFT
106F EB              XCHG
1070 D27710          JNC     MCH02     ;DO FULL BUFFER!
1073 19              DAD     D         ;ADJUST COUNT TO ORIG.
1074 110000          LXI     D,0       ;SET NONE LEFT
1077 0E80    MCH02:  MVI     C,80H     ;SET LAST BLK B.C.
1079 CDF019          CALL    BK2BC     ;CONVERT TO BYTE COUNT
107C 22A581          SHLD    XFXBC     ;SET BYTE ...
107F 221981          SHLD    FXBC      ;... COUNTS
1082 D5              PUSH    D         ;SAVE REMAINING BLK COUNT
1083 219D81          LXI     H,XFHAN   ;POINT TO SOURCE HPB
1086 CD3113          CALL    RD        ;READ
1089 D1              POP     D         ;RESTORE REMAINING BLK COUNT
108A D8              RC                ;ERROR!
108B 22A181          SHLD    XFBLK     ;UPDATE THE BLOCK NUMBER
108E D5              PUSH    D         ;SAVE REMAINING BLK COUNT
108F 211181          LXI     H,FHAN    ;POINT TO DESTINATION HPB
1092 CD2E13          CALL    WR        ;WRITE
```

```
1095 D1              POP     D          ;RESTORE REMAINING BLK COUNT
1096 D8              RC                 ;ERROR!
1097 221581          SHLD    FBLK       ;* UPDATE THE BLOCK NUMBER
                                        ;* CAUTION *  RR
109A C36010          JMP     MCH01      ;LOOP!

109D 3EFD    STWO:   MVI     A,-3       ;SET "SUB USER" FUNCTION
109F F5      CTWO:   PUSH    PSW        ;SAVE FCN CODE
10A0 219D81          LXI     H,XFHAN    ;POINT TO SOURCE HPB
10A3 CD3213          CALL    CHDLR      ;CALL HANDLER
10A6 F1              POP     PSW        ;RESTORE FCN CODE
10A7 211181          LXI     H,FHAN     ;POINT TO DESTINATION HPB
10AA C33213          JMP     CHDLR      ;CALL HANDLER & RETURN

10AD E5      CPYDV:  PUSH    H          ;SAVE H&L
10AE 2A1181          LHLD    FHAN       ;GET HANDLER ADDRESS
10B1 229D81          SHLD    XFHAN      ;SAVE IN AUX. HPB
10B4 E1              POP     H          ;RESTORE H&L
10B5 3A1481          LDA     FDRV       ;GET DRIVE NO.
10B8 32A081          STA     XFDRV      ;SAVE IN AUX. HPB
10BB C9              RET                ;RETURN


10BC 21F780  OPENX:  LXI     H,FPB      ;POINT TO FPB
10BF C5      OPENY:  PUSH    B
10C0 D5              PUSH    D
10C1 F5              PUSH    PSW        ;SAVE OPEN TYPE CODE
10C2 E5              PUSH    H
10C3 111A00          LXI     D,FHAN-FPB
10C6 19              DAD     D          ;POINT TO FHAN
10C7 3EFE            MVI     A,-2       ;SET "ADD USER" FUNCTION
10C9 CD3213          CALL    CHDLR
10CC E1              POP     H          ;RESTORE FPB PNTR
10CD F1              POP     PSW        ;RESTORE OPEN TYPE CODE
10CE 77              MOV     M,A        ;STORE OPEN TYPE CODE
10CF CDE111          CALL    OPEN       ;OPEN THE FILE
10D2 D1              POP     D
10D3 D2D810          JNC     OPX01
10D6 D1              POP     D
10D7 C9              RET
10D8 C1      OPX01:  POP     B
10D9 C9              RET                ;RETURN

10DA CDBE18  GMPRM:  CALL    GCMA       ;SKIP COMMA IF ANY
10DD CD2C19          CALL    GN2Z       ;GET ONE NUMBER
10E0 D20611          JNC     GM02       ;NO NUMBER: ERROR!
10E3 D5              PUSH    D          ;SAVE 1ST NUMBER
10E4 CD9618          CALL    SPNOR      ;SKIP SPACES
10E7 D62D            SUI     '-'        ;CHECK FOR '-'
10E9 A7              ANA     A          ;CLEAR C-FLAG
10EA F5              PUSH    PSW        ;SAVE FLAGS
10EB 23              INX     H
10EC CAF..0          JZ      GM03
10EF 2B              DCX     H
10F0 CDBE18          CALL    GCMA       ;SKIP COMMA IF NOT '-'
10F3 CD2C19  GM03:   CALL    GN2Z       ;GET 2ND NUMBER
10F6 CDBE18          CALL    GCMA       ;SKIP COMMA IF THERE
10F9 F1              POP     PSW        ;RESTORE FLAGS
```

```
10FA E3                 XTHL
10FB EB                 XCHG
10FC C20311             JNZ     GM01    ;WAS NOT '-'
10FF CD8F18             CALL    SUBHD   ;COMPUTE BYTE COUNT
1102 23                 INX     H
1103 E3         GM01:   XTHL
1104 C1                 POP     B
1105 D0                 RNC             ;O.K.: EXIT!
1106 060F       GM02:   MVI     B,EIVP  ;SET ERROR CODE
1108 37                 STC             ;INDICATE ERROR
1109 C9                 RET             ;RETURN!

110A 211A00     PRDEV:  LXI     H,FHAN-FPB
110D 19                 DAD     D       ;POINT TO FHAN
110E 5E                 MOV     E,M     ;GET LOBYTE OF HANDLER PNTR
110F 23                 INX     H
1110 56                 MOV     D,M     ;GET HIBYTE OF HANDLER PNTR
1111 23                 INX     H
1112 23                 INX     H
1113 7E                 MOV     A,M     ;GET UNIT NUMBER
1114 C630               ADI     '0'     ;CONVERT TO ASCII
1116 4F                 MOV     C,A     ;COPY UNIT NO. TO C
1117 EB                 XCHG
1118 23                 INX     H       ;POINT TO DEVICE NAME
1119 23                 INX     H
111A 23                 INX     H
111B 5E                 MOV     E,M     ;GET 1ST CHARACTER
111C 23                 INX     H
111D 56                 MOV     D,M     ;GET 2ND CHARACTER
111E 7B                 MOV     A,E     ;PRINT...
111F CDC817             CALL    LO      ;... 1ST CHAR.
1122 7A                 MOV     A,D     ;PRINT...
1123 CDC817             CALL    LO      ;... 2ND CHAR.
1126 79                 MOV     A,C     ;PRINT...
1127 CDC817             CALL    LO      ;... UNIT NUMBER
112A C3EE18             JMP     PCOLN   ;PRINT ': ' AND RETURN!

112D 0A0A4449 MS1:      DB 10,10,'DIRECTORY ',239
1131 52454354
1135 4F525920
1139 EF
113A 0D0A0A     MS2:    DB 13,10,10
113D 41545220           DB 'ATR  NAME TYPE VR SBLK SIZE LBC'
1141 204E414D
1145 45205459
1149 50452056
114D 52205342
1151 4C4B2053
1155 495A4520
1159 4C4243
115C 204C4144           DB ' LADR SADR',13,10,239
1160 52205341
1164 44520D0A
1168 EF
1169 203C4652 MS3:      DB ' <FREE SPACE> ',239
116D 45452053
1171 50414345
```

```
1175 3E20EF
1178 06034445 MS4:     DB 6,3,'DEFAULT DEVICE='
117C 4641554C
1180 54204445
1184 56494345
1188 3D
1189 200602EF NTYP:    DB ' ',6,2,239

118D 505247   PTYP:    DB       'PRG'
1190 4C4441   LTYP:    DB       'LDA'
1193 535243   STYP:    DB       'SRC'


              ;;       DIRECTORY ACCESS ROUTINES :


              ;+
              ; OPDIR - "OPEN" DIRECTORY FOR A SCAN.
              ;
              ; INPUTS  - D&E: PNTR TO FHAN IN FPB
              ; OUTPUTS - <C>  : DIRECTORY READ ERROR,
              ;                  OR NO VOLUME ENTRY
              ;           <NC> : NO ERRORS AND :
              ;           A    : ATR BYTE OF THIS ENTRY (=41H)
              ;           B    : INTERNALLY MAINTAINED ENTRY COUNTER
              ;           D&E  : PNTR TO FHAN IN FPB (UNCHANGED)
              ;           H&L  : PNTR TO "VOLUME" ENTRY IN DIRECTORY
              ;-

1196 210400   OPDIR:   LXI      H,FBLK-FHAN
1199 19                DAD      D          ;POINT TO FBLK
119A D5                PUSH     D          ;SAVE PNTR TO FHAN
119B 11B611            LXI      D,DIP      ;POINT TO INITIAL PARAMETERS
119E 0606              MVI      B,DIPS     ;SET BYTE COUNT
11A0 CD7A18            CALL     MOVHD      ;MOVE PARAMETERS
11A3 E1                POP      H          ;GET FHAN PNTR
11A4 E5                PUSH     H          ;RESAVE IT
11A5 3EFE              MVI      A,-2       ;SET "ADD USER" FUNCTION
11A7 CD3213            CALL     CHDLR      ;CALL HANDLER
11AA D1                POP      D          ;RESTORE PNTR TO FHAN
11AB CDCA11            CALL     LOO1       ;GO READ 1ST DIRECTORY BLOCK
11AE D8                RC                  ;ERROR!
11AF FE41              CPI      41H        ;IS THIS VOLUME ENTRY ?
11B1 C8                RZ                  ;YES: O.K.: EXIT!
11B2 0612              MVI      B,ENVE     ;SET ERROR CODE
11B4 37                STC
11B5 C9                RET                 ;ERROR EXIT!

11B6 0000     DIP:     DW       0          ;BLOCK NUMBER
11B8 1D81              DW       DBF        ;BUFFER POINTER
11BA 8000              DW       128        ;BYTE COUNT
     (0006)   DIPS     EQU      $-DIP


              ;+
              ; GNDE  - GET NEXT DIRECTORY ENTRY
              ;
              ; INPUTS  - B     . INTERNALLY MAINTAINED ENTRY COUNTER
```

```
                ;             D&E  : PNTR TO FHAN IN FPB
                ;             H&L  : PNTR TO CURRENT DIRECTORY ENTRY
                ; OUTPUTS -   <C>  : DIRECTORY READ ERROR
                ;             <Z>  : END OF DIRECTORY
                ;          <NC,NZ> : NO ERRORS AND :
                ;             A    : ATTRIBUTE BYTE OF THIS ENTRY
                ;             B    : UPDATED, MUST BE PRESERVED
                ;             D&E  : PNTR TO FHAN IN FPB (UNCHANGED)
                ;             H&L  : PNTR TO NEXT DIRECTORY ENTRY
                ;-

11BC 05         GNDE:   DCR     B       ;ANY MORE IN THIS BLOCK ?
11BD C2D811             JNZ     L002    ;YES!
11C0 210400             LXI     H,FBLK-FHAN
11C3 19                 DAD     D       ;POINT TO FBLK
11C4 3A1E81             LDA     MDBLK   ;GET MAX. DIRECTORY BLOCK NUMBER
11C7 96                 SUB     M       ;SUBTRACT CURRENT BLOCK NUMBER
11C8 C8                 RZ              ;THIS WAS LAST BLOCK!
11C9 34                 INR     M       ;BUMP BLOCK NUMBER
11CA D5         L001:   PUSH    D       ;SAVE D&E
11CB EB                 XCHG            ;PUT PNTR IN H&L
11CC CD3113             CALL    RD      ;READ NEXT BLOCK
11CF D1                 POP     D       ;RESTORE D&E
11D0 060C               MVI     B,EDIR  ;SET ERROR CODE
11D2 D8                 RC              ;READ ERROR!
11D3 0606               MVI     B,6     ;SET ENTRY COUNTER
11D5 210A81             LXI     H,DBF+2-21 ;SET DIRECTORY PNTR
11D8 D5         L002:   PUSH    D       ;SAVE D&E
11D9 111500             LXI     D,21    ;ENTRY SIZE
11DC 19                 DAD     D       ;STEP TO NEXT ENTRY
11DD D1                 POP     D       ;RESTORE D&E
11DE 7E                 MOV     A,M     ;GET ATTRIBUTE BYTE
11DF A7                 ANA     A       ;TEST ATTRIBUTE BYTE
11E0 C9                 RET             ;RETURN, NO ERRORS


                ;+
                ; OPEN - OPEN A FILE FOR INPUT OR OUTPUT
                ;-

11E1 22F380     OPEN:   SHLD    FPBP    ;SAVE FPB PNTR
11E4 7E                 MOV     A,M     ;GET OPEN TYPE CODE
11E5 32F580             STA     OCODE   ;SAVE OPEN TYPE CODE
11E8 110B00             LXI     D,FVER-FPB
11EB 19                 DAD     D       ;POINT TO VERSION
11EC 7E                 MOV     A,M     ;GET VERSION
11ED 32F680             STA     OVERS   ;SAVE ORIGINAL VERSION
11F0 110B00             LXI     D,FHAN-FSIZ-1
11F3 23                 INX     H       ;POINT TO FSIZ ...
11F4 23                 INX     H       ;...
11F5 23                 INX     H       ;...
11F6 72                 MOV     M,D     ;SET FSIZ ...
11F7 23                 INX     H       ;... TO ...
11F8 72                 MOV     M,D     ;... ZERO
11F9 19                 DAD     D       ;POINT TO FHAN
11FA EB                 XCHG            ;PUT PNTR IN D&E
11FB CD9611             CALL    OPDIR   ;"OPEN" DIRECTORY
11FE D8                 RC              ;READ ERROR!
```

```
11FF CDBC11    L005:   CALL    GNDE        ;GET NEXT ENTRY
1202 D8                RC                  ;ERROR!
1203 CA9212            JZ      L011        ;END OF DIRECTORY!
1206 E5                PUSH    H           ;SAVE H&L
1207 D5                PUSH    D           ;SAVE D&E
1208 EB                XCHG                ;D&E = DIR PNTR
1209 2AF380            LHLD    FPBP        ;H&L = FPB PNTR
120C 0E0B              MVI     C,11        ;SET COUNTER
120E 1A                LDAX    D           ;GET ENTRY TYPE CODE
120F FE01              CPI     TFREE       ;IS IT "FREE" ?
1211 C24B12            JNZ     L106        ;NO!
1214 7E                MOV     A,M         ;GET OPEN TYPE CODE
1215 E601              ANI     FNEW        ;"NEW" FILE ?
1217 CA8012            JZ      L010        ;NO!
121A 23                INX     H           ;POINT TO ATR BYTE
121B 3603              MVI     M,TFILE+TPROT ;SET ATR BYTE
121D C5                PUSH    B           ;SAVE B&C
121E 010800            LXI     B,FSBK-FATR
1221 09                DAD     B           ;POINT TO FSBK IN FPB
1222 EB                XCHG                ;...
1223 09                DAD     B           ;POINT TO FSBK IN DIRECTORY
1224 EB                XCHG                ;...
1225 0605              MVI     B,FLAD-FSBK ;SET COUNT
1227 CD7A18            CALL    MOVHD       ;COPY PARAMETERS
122A 010500            LXI     B,FDBK-FLAD
122D 09                DAD     B           ;POINT TO FDBK
122E C1                POP     B           ;RESTORE B&C
122F 3A1D81            LDA     DBLK        ;GET DIR BLOCK NUMBER
1232 77                MOV     M,A         ;SAVE IT
1233 23                INX     H
1234 70                MOV     M,B         ;SAVE ENTRY NUMBER
1235 D1                POP     D           ;RESTORE D&E
1236 E1                POP     H           ;RESTORE H&L
1237 05                DCR     B           ;IS THIS LAST SLOT ?
1238 C29212            JNZ     L011        ;NO: O.K.!
123B 4F                MOV     C,A         ;COPY DIR BLK NO.
123C 3A1E81            LDA     MDBLK       ;GET MAX. DIR BLK NO.
123F B9                CMP     C           ;IS THIS LAST DIR BLOCK ?
1240 C29212            JNZ     L011        ;NO: O.K.!
1243 CD9212            CALL    L011        ;CHECK FOR OTHER ERRORS
1246 D8                RC                  ;ERROR: RETURN!
1247 062D              MVI     B,EDRF      ;SET ERROR CODE
1249 37                STC                 ;INDICATE ERROR
124A C9                RET                 ;DIR FULL: ERROR!
124B 23        L106:   INX     H
124C 0D        L006:   DCR     C           ;HAVE ALL MATCHED ?
124D CA6612            JZ      L007        ;YES!
1250 13                INX     D           ;BUMP DIR PNTR
1251 23                INX     H           ;BUMP FPB PNTR
1252 1A                LDAX    D           ;GET NEXT DIR BYTE
1253 BE                CMP     M           ;COMPARE WITH NEXT FPB BYTE
1254 CA4C12            JZ      L006        ;MATCH!
1257 DA8012            JC      L010        ;IF VERS. IT'S LESS!
125A 79                MOV     A,C         ;GET COUNTER
125B 3D                DCR     A           ;WAS THIS "VERSION" BYTE ?
125C C28012            JNZ     L010        ;NO!
125F 3AF680            LDA     OVERS       ;GET ORIGINAL VERSION
1262 A7                ANA     A           ;WAS IT ZERO ?
```

```
1263 C28012          JNZ     L010      ;NO: EXPLICIT VERSION!
1266 3AF580 L007:    LDA     OCODE     ;GET OPEN TYPE CODE
1269 E601            ANI     FNEW      ;"NEW" FILE ?
126B CA7312          JZ      L008      ;NO!
126E 1A              LDAX    D         ;GET VERS FROM DIRECTORY
126F 77              MOV     M,A       ;STORE IN FPB
1270 C38012          JMP     L010      ;
1273 C5      L008:   PUSH    B         ;SAVE B&C
1274 060B            MVI     B,FDBK-FVER ;SET COUNT
1276 CD7A18  L009:   CALL    MOVHD     ;MOVE PARAMETERS
1279 C1              POP     B         ;RESTORE B&C
127A 3A1D81          LDA     DBLK      ;GET DIR BLOCK NUMBER
127D 77              MOV     M,A       ;STORE IN FPB
127E 23              INX     H         ;...
127F 70              MOV     M,B       ;STORE ENTRY NUMBER
1280 D1      L010:   POP     D         ;RESTORE D&E
1281 E1              POP     H         ;RESTORE H&L
1282 0D              DCR     C         ;FIND EXPLICIT VERSION ?
1283 F2FF11          JP      L005      ;NO: CONTINUE SCAN
1286 3AF580          LDA     OCODE     ;GET OPEN TYPE CODE
1289 E601            ANI     FNEW      ;"NEW" FILE ?
128B CAB812          JZ      L013      ;NO: "OLD" FILE FOUND!
128E 0624            MVI     B,EDFN    ;SET ERROR CODE
1290 37              STC
1291 C9              RET               ;DUPLICATE FILENAME ERROR!
1292 2AF380  L011:   LHLD    FPBP      ;GET FPB PNTR
1295 7E              MOV     A,M       ;GET OPEN TYPE CODE
1296 110B00          LXI     D,FVER-FPB
1299 19              DAD     D         ;POINT TO FVER
129A E601            ANI     FNEW      ;"NEW" FILE ?
129C 3AF680          LDA     OVERS     ;GET ORIGINAL VERSION
129F CAAB12          JZ      L012      ;NO!
12A2 A7              ANA     A         ;TEST ORIGINAL VERSION
12A3 C2B812          JNZ     L013      ;EXPLICIT VERSION REQUESTED!
12A6 34              INR     M         ;INCREMENT VERSION
12A7 0627            MVI     B,EVOV    ;SET ERROR CODE
12A9 37              STC
12AA C8              RZ                ;VERSION OVERFLOW ERROR!
12AB A7      L012:   ANA     A         ;TEST ORIGINAL VERSION
12AC C2B412          JNZ     L023      ;EXPLICIT VERSION REQUESTED!
12AF 7E              MOV     A,M       ;GET "FOUND" VERSION
12B0 A7              ANA     A         ;IS IT STILL ZERO ?
12B1 C2B812          JNZ     L013      ;NO: O.K.!
12B4 062A    L023:   MVI     B,EFNF    ;SET ERROR CODE
12B6 37              STC
12B7 C9              RET               ;FILE NOT FOUND!
12B8 2AF380  L013:   LHLD    FPBP      ;GET FPB PNTR
12BB E5              PUSH    H         ;SAVE H&L
12BC 111A00          LXI     D,FHAN-FPB
12BF 19              DAD     D         ;POINT TO FHAN
12C0 3EFD            MVI     A,-3      ;SET "SUB USER" FUNCTION
12C2 CD3213          CALL    CHDLR     ;CALL HANDLER
12C5 E1              POP     H         ;GET FPB PNTR BACK
12C6 E5              PUSH    H         ;RESAVE IT
12C7 010C00          LXI     B,FSBK-FPB
12CA 09              DAD     B         ;POINT TO FSBK
12CB 5E              MOV     E,M       ;GET STARTING ...
12CC 23              INX     H         ;... BLOCK ...
```

```
12CD 56              MOV     D,M       ;... NUMBER
12CE 011100          LXI     B,FBLK-FSBK-1
12D1 09              DAD     B         ;POINT TO FBLK
12D2 73              MOV     M,E       ;STORE STARTING ...
12D3 23              INX     H         ;... BLOCK ...
12D4 72              MOV     M,D       ;... NUMBER
12D5 E1              POP     H         ;RESTORE FPB POINTER
12D6 AF              XRA     A         ;INDICATE NO ERRORS
12D7 47              MOV     B,A       ;SET GOOD STATUS CODE
12D8 C9              RET               ;RETURN TO CALLER


                ;+
                ; READ - READ FILE. CURRENTLY ONLY "IMAGE"
                ;        TYPE FILES ARE SUPPORTED.
                ;        "OPEN" MUST BE CALLED BEFORE CALLING "READ".
                ;
                ; INPUTS  - B&C: MAX. ALLOWABLE BYTE COUNT
                ;          D&E: IF ZERO, THEN FILE WILL BE READ IN TO
                ;               MEMORY AT LOAD ADDRESS SPECIFIED IN
                ;               DIRECTORY. IF NON-ZERO, THEN D&E IS
                ;               USED AS THE LOAD ADDRESS.
                ;          H&L: PNTR TO FPB.
                ;
                ;-

12D9 CD3913     READ:   CALL    LADSB     ;SETUP MEM. PNTR & GET SIZE
12DC C5                 PUSH    B         ;SAVE MAX. BYTE COUNT
12DD 4E                 MOV     C,M       ;GET LAST BLK BC
12DE EB                 XCHG              ;H&L=NUMBER OF BLOCKS
12DF CDF019             CALL    BK2BC     ;COMPUTE BYTE COUNT
12E2 EB                 XCHG              ;D&E=TOTAL BYTE COUNT
12E3 E3                 XTHL              ;GET MAX. BYTE COUNT
12E4 CD8318             CALL    CMPHD     ;COMPARE WITH FILE SIZE
12E7 E1                 POP     H         ;RESTORE H&L
12E8 0639               MVI     B,ERSZ    ;SET ERROR CODE
12EA D8                 RC                ;FILE TOO LARGE: ERROR!
12EB 011200             LXI     B,FXBC-FLBC
12EE 09                 DAD     B         ;POINT TO FXBC
12EF 73                 MOV     M,E       ;STORE ...
12F0 23                 INX     H         ;... BYTE ...
12F1 72                 MOV     M,D       ;... COUNT
12F2 01F7FF             LXI     B,FHAN-FXBC-1
12F5 09                 DAD     B         ;POINT TO FHAN
12F6 CD3113             CALL    RD        ;READ THE FILE
12F9 0630               MVI     B,EFRD    ;SET ERROR CODE
12FB 2AF380     RWE:    LHLD    FPBP      ;RESTORE FPB PNTR
12FE D8                 RC                ;ERROR EXIT!
12FF 0600               MVI     B,0       ;SET GOOD STATUS CODE
1301 C9                 RET               ;EXIT!


                ;+
                ; WRITE - FILE WRITE. CURRENTLY ONLY "IMAGE" TYPE
                ;         FILES ARE SUPPORTED. "OPEN" MUST BE
                ;         CALLED BEFORE CALLING "WRITE".
                ;
                ;-
```

```
1302 E5        WRITE:  PUSH    H           ;SAVE FPB PNTR
1303 D5                PUSH    D           ;SAVE AUX. LOAD ADDRESS
1304 112200            LXI     D,FXBC-FPB
1307 19                DAD     D           ;POINT TO FXBC
1308 D1                POP     D           ;RESTORE AUX. LOAD ADDRESS
1309 71                MOV     M,C         ;STORE ...
130A 23                INX     H           ;... BYTE ...
130B 70                MOV     M,B         ;... COUNT
130C E1                POP     H           ;RESTORE FPB POINTER
130D CD3913            CALL    LADSB       ;SETUP MEM. PNTR & GET SIZE
1310 C5                PUSH    B           ;PUT BYTE COUNT ON STACK
1311 E3                XTHL                ;NOW HL=BYTE COUNT
1312 CDDE19            CALL    BC2BK       ;CONVERT TO BLOCKS & BYTE COUNT
1315 CD8918            CALL    CMPDH       ;COMPARE H&L WITH D&E
1318 EB                XCHG                ;D&E=NUMBER OF BLOCKS
1319 E1                POP     H           ;H&L=PNTR TO FLBC
131A 0636              MVI     B,EWSZ      ;SET ERROR CODE
131C D8                RC                  ;ERROR EXIT!
131D 71                MOV     M,C         ;STORE LAST BLOCK BYTE COUNT
131E 010800            LXI     B,FAUX-FLBC
1321 09                DAD     B           ;POINT TO FAUX
1322 73                MOV     M,E         ;STORE NUMBER ...
1323 23                INX     H           ;... OF BLOCKS ...
1324 72                MOV     M,D         ;... USED
1325 23                INX     H           ;POINT TO FHAN
1326 CD2E13            CALL    WR          ;WRITE THE FILE
1329 0633              MVI     B,EFWR      ;SET ERROR CODE
132B C3FB12            JMP     RWE         ;TAKE COMMON EXIT!

132E 3E01      WR:     MVI     A,1         ;SET "WRITE" FUNCTION CODE
1330 FE                DB      OFEH        ;** SKIP ONE BYTE **
1331 AF        RD:     XRA     A           ;SET "READ" FUNCTION CODE
1332 5E        CHDLR:  MOV     E,M         ;GET ...
1333 23                INX     H           ;... HANDLER ...
1334 56                MOV     D,M         ;... ADDRESS
1335 23                INX     H           ;POINT TO PARAMETER BLOCK
1336 77                MOV     M,A         ;STORE FUNCTION CODE
1337 D5        JMPD:   PUSH    D           ;JUMP INDIRECT ...
1338 C9                RET                 ;... D&E

1339 22F380    LADSB:  SHLD    FPBP        ;SAVE FPB PNTR
133C 7A                MOV     A,D         ;TEST AUX. ...
133D B3                ORA     E           ;... LOAD ADDRESS
133E C24A13            JNZ     LO14        ;NON-ZERO: USE IT!
1341 E5                PUSH    H           ;SAVE H&L
1342 111100            LXI     D,FLAD-FPB
1345 19                DAD     D           ;POINT TO FLAD
1346 5E                MOV     E,M         ;GET LOBYTE
1347 23                INX     H           ;...
1348 56                MOV     D,M         ;GET HIBYTE
1349 E1                POP     H           ;RESTORE H&L
134A D5        LO14:   PUSH    D           ;SAVE D&E
134B 112000            LXI     D,FBUF-FPB
134E 19                DAD     D           ;POINT TO FBUF
134F D1                POP     D           ;RESTORE D&E
1350 73                MOV     M,E         ;STORE LOAD ...
```

```
1351 23              INX     H         ;... ADDRESS IN ...
1352 72              MOV     M,D       ;... FBUF
1353 11EDFF          LXI     D,FSIZ-FBUF-1
1356 19              DAD     D         ;POINT TO FSIZ
1357 5E              MOV     E,M       ;GET NUMBER ...
1358 23              INX     H         ;... OF ...
1359 56              MOV     D,M       ;... BLOCKS
135A 23              INX     H         ;POINT TO FLBC
135B C9              RET               ;RETURN

                     ;+
                     ; CLOSE - FILE CLOSE ROUTINE.
                     ;
                     ;-

135C 0600    CLOSE:  MVI     B,0       ;SET GOOD STATUS CODE
135E 7E              MOV     A,M       ;GET OPEN TYPE CODE
135F E601            ANI     FNEW      ;"NEW" FILE ?
1361 C8              RZ                ;NO: RETURN!
1362 22F380          SHLD    FPBP      ;SAVE FPB PNTR
1365 CDEB13          CALL    GODBK     ;GET DIRECTORY BLOCK
1368 D8              RC                ;ERROR!
1369 111800          LXI     D,FAUX-FPB
136C 19              DAD     D         ;POINT TO FAUX
136D 5E              MOV     E,M       ;GET NUMBER ...
136E 23              INX     H         ;... OF BLOCKS ...
136F 56              MOV     D,M       ;... IN FILE
1370 D5              PUSH    D         ;SAVE FILE BLOCKS
1371 11F3FF          LXI     D,FSBK-FAUX-1
1374 19              DAD     D         ;POINT TO FSBK
1375 5E              MOV     E,M       ;GET FILE ...
1376 23              INX     H         ;... STARTING
1377 56              MOV     D,M       ;... BLOCK NUMBER
1378 23              INX     H         ;POINT TO FSIZ
1379 E3              XTHL              ;SAVE PNTR & GET FILE BLOCKS
137A EB              XCHG              ;D&E=FILE BLOCKS
137B 19              DAD     D         ;H&L=START BLK OF FREE SPACE
137C E3              XTHL              ;SAVE H&L & GET PNTR
137D D5              PUSH    D         ;SAVE FILE BLOCKS
137E 5E              MOV     E,M       ;GET TOTAL ...
137F 23              INX     H         ;... BLOCKS IN ...
1380 56              MOV     D,M       ;... "FREE" SLOT
1381 E3              XTHL              ;SAVE PNTR & GET FILE BLOCKS
1382 EB              XCHG              ;H&L=OLD "FREE" BLOCKS
1383 CD8F18          CALL    SUBHD     ;COMPUTE NEW "FREE" BLOCKS
1386 E3              XTHL              ;SAVE NEW "FREE" & GET PNTR
1387 72              MOV     M,D       ;STORE NUMBER ...
1388 2B              DCX     H         ;... OF BLOCKS ...
1389 73              MOV     M,E       ;... IN FILE
138A 11F3FF          LXI     D,FATR-FSIZ
138D 19              DAD     D         ;POINT TO FATR
138E C5              PUSH    B         ;SAVE B&C
138F E5              PUSH    H         ;SAVE PNTR
1390 219D81          LXI     H,DBF+2+6*21 ;POINT PAST LAST DIR ENTRY
1393 11EBFF          LXI     D,-21     ;ENTRY SIZE
1396 19      LO17:   DAD     D         ;STEP BACK ONE ENTRY
1397 05              DCR     B         ;AT CORRECT ENTRY ?
```

```
1398 C29613           JNZ     LO17      ;NO: STEP AGAIN!
139B D1               POP     D         ;GET PNTR TO FATR
139C 0615             MVI     B,21      ;SET COUNT
139E CD7A18           CALL    MOVHD     ;MOVE DIRECTORY ENTRY
13A1 C1               POP     B         ;GET OLD B&C
13A2 05               DCR     B         ;WAS THIS LAST DIR. SLOT ?
13A3 CCE013           CZ      LO25      ;YES: STORE BLK NO., ETC.
13A6 C1               POP     B         ;GET SIZE
13A7 D1               POP     D         ;GET START BLK. NO.
13A8 CDCC13           CALL    SFR       ;STORE SIZE & START BLK. NO.
13AB 111D81   WRDIR:  LXI     D,DBF     ;POINT TO DIRECTORY BUFFER
13AE CD8F18           CALL    SUBHD     ;COMPUTE DIRECTORY BYTE COUNT
13B1 EB               XCHG              ;D&E = BYTE COUNT
13B2 2AF380           LHLD    FPBP      ;GET FPB PNTR
13B5 012200           LXI     B,FXBC-FPB
13B8 09               DAD     B         ;POINT TO FXBC
13B9 73               MOV     M,E       ;STORE ...
13BA 23               INX     H         ;... BYTE ...
13BB 72               MOV     M,D       ;... COUNT
13BC 11F7FF           LXI     D,FHAN-FXBC-1
13BF 19               DAD     D         ;POINT TO FHAN
13C0 CD2E13           CALL    WR        ;WRITE DIRECTORY
13C3 0600             MVI     B,0       ;SET GOOD STATUS CODE
13C5 2AF380           LHLD    FPBP      ;GET FPB PNTR
13C8 D0               RNC               ;O.K.: EXIT!
13C9 060C             MVI     B,EDIR    ;SET ERROR CODE
13CB C9               RET               ;ERROR EXIT!

13CC 3601     SFR:    MVI     M,TFREE   ;SET 'FREE' ATTRIBUTE BYTE
13CE D5               PUSH    D
13CF 110B00           LXI     D,FSBK-FATR
13D2 19               DAD     D         ;POINT TO FSBK
13D3 D1               POP     D
13D4 73               MOV     M,E       ;STORE START ...
13D5 23               INX     H         ;... BLOCK NUMBER ...
13D6 72               MOV     M,D       ;... OF FREE SPACE
13D7 23               INX     H
13D8 71               MOV     M,C       ;STORE ...
13D9 23               INX     H         ;... SIZE OF ...
13DA 70               MOV     M,B       ;... FREE SPACE ...
13DB 23               INX     H
13DC 3680             MVI     M,128     ;STORE LBC
13DE 23               INX     H
13DF C9               RET               ;RETURN

13E0 3A1E81   LO25:   LDA     MDBLK     ;GET MAX. DIR. BLK. NO.
13E3 B9               CMP     C         ;IS THIS THE LAST BLOCK ?
13E4 C8               RZ                ;YES: RETURN!
13E5 0C               INR     C         ;BUMP BLOCK NUMBER
13E6 71               MOV     M,C       ;STORE IT
13E7 23               INX     H
13E8 77               MOV     M,A       ;STORE MAX. BLK. NO.
13E9 23               INX     H
13EA C9               RET               ;RETURN TO CALLER


13EB 2AF380   GODBK:  LHLD    FPBP      ;GET FPB PNTR
13EE E5               PUSH    H         ;SAVE IT
```

```
13EF 111600             LXI     D,FDBK-FPB
13F2 19                 DAD     D       ;POINT TO FDBK
13F3 4E                 MOV     C,M     ;GET DIRECTORY BLOCK NUMBER
13F4 23                 INX     H       ;POINT TO FDEN
13F5 46                 MOV     B,M     ;GET DIRECTORY ENTRY NUMBER
13F6 110700             LXI     D,FBLK-FDEN
13F9 19                 DAD     D       ;POINT TO FBLK
13FA 71                 MOV     M,C     ;STORE BLOCK NUMBER
13FB 23                 INX     H
13FC 11B711             LXI     D,DIP+1 ;POINT TO REST OF PARAMETERS
13FF C5                 PUSH    B       ;SAVE B&C
1400 0605               MVI     B,DIPS-1 ;SET COUNT
1402 CD7A18             CALL    MOVHD   ;STORE REST OF PARAMETERS
1405 C1                 POP     B       ;RESTORE B&C
1406 C5                 PUSH    B       ;RESAVE B&C
1407 11F6FF             LXI     D,FHAN-FXBC-2
140A 19                 DAD     D       ;POINT TO FHAN
140B CD3113             CALL    RD      ;READ DIR BLOCK
140E C1                 POP     B       ;RESTORE B&C
140F E1                 POP     H       ;RESTORE FPB PNTR
1410 D0                 RNC             ;NO ERRORS!
1411 060C               MVI     B,EDIR  ;SET ERROR CODE
1413 C9                 RET             ;RETURN TO CALLER

                ;+
                ; PDV - PARSE DEVICE NAME
                ; <C> : INVALID DEVICE
                ; <Z> : NO DEVICE, DEFAULT USED
                ; OTHERWISE: GOT VALID DEVICE
                ;-

1414 CD9618     PDV:    CALL    SPNOR   ;SKIP SPACES
1417 D5                 PUSH    D       ;SAVE FPB PNTR
1418 E5                 PUSH    H       ;SAVE STRING PNTR
1419 EB                 XCHG
141A 2AF080             LHLD    DFDV    ;GET DEFAULT DEVICE
141D EB                 XCHG            ;PUT IN D&E
141E 3AF280             LDA     DFUN    ;GET DEFAULT UNIT NUMBER
1421 4F                 MOV     C,A     ;PUT IN C
1422 7E                 MOV     A,M     ;TEST IF ...
1423 A7                 ANA     A       ;... END OF LINE
1424 CA4F14             JZ      PDV05   ;END OF LINE!
1427 0604               MVI     B,4     ;SET COUNTER
1429 05         PDV01:  DCR     B       ;ENOUGH ?
142A CA4F14             JZ      PDV05   ;YES: NO DEVICE!
142D 23                 INX     H
142E 7E                 MOV     A,M     ;GET NEXT CHARACTER
142F A7                 ANA     A       ;TEST FOR END OF LINE
1430 CA4F14             JZ      PDV05   ;END OF LINE: NO DEVICE!
1433 FE3A               CPI     ':'     ;CHECK FOR COLON
1435 C22914             JNZ     PDV01   ;NO: LOOP!
1438 E1         PDV02:  POP     H       ;RESTORE STRING PNTR
1439 3E03               MVI     A,3
143B B8                 CMP     B
143C CA4514             JZ      PDV03   ;UNIT NUMBER ONLY!
143F 5E                 MOV     E,M     ;GET ...
1440 23                 INX     H       ;... DEVICE ...
```

```
1441 56                MOV     D,M       ;... NAME
1442 23                INX     H
1443 0E30              MVI     C,'0'     ;SET UNIT ZERO
1445 05        PDV03:  DCR     B
1446 05                DCR     B
1447 CA4C14            JZ      PDV04     ;NO UNIT NUMBER!
144A 4E                MOV     C,M       ;GET UNIT NUMBER
144B 23                INX     H
144C 23        PDV04:  INX     H         ;SKIP OVER COLON
144D E5                PUSH    H         ;SAVE STRING POINTER
144E A7                ANA     A         ;SET FLAGS TO <NZ>
144F E1        PDV05:  POP     H         ;GET STRING PNTR
1450 E3                XTHL              ;SWAP
1451 F5                PUSH    PSW       ;SAVE FLAGS
1452 79                MOV     A,C       ;COPY UNIT NUMBER
1453 D630              SUI     '0'       ;CONVERT FROM ASCII      — CONVERT FROM ASCII
1455 F5                PUSH    PSW       ;SAVE UNIT NUMBER
1456 E5                PUSH    H         ;SAVE ORIGINAL D&E
1457 214300            LXI     H,HDVCT   ;POINT TO DEVICE TABLE
145A 0E03              MVI     C,3       ;SET NUMBER OF SLOTS
145C E5        PDV06:  PUSH    H         ;SAVE HANDLER PNTR
145D 3EC3              MVI     A,0C3H    ;SET 'JMP' BYTE
145F AE                XRA     M         ;VECTOR EMPTY ?
1460 C28B14            JNZ     PDV07     ;YES: SKIP IT!
1463 23                INX     H         ;POINT TO ASCII DEVICE NAME ...
1464 23                INX     H
1465 23                INX     H
1466 7E                MOV     A,M       ;GET 1ST CHAR. OF NAME
1467 BB                CMP     E         ;MATCH ?
1468 C28B14            JNZ     PDV07     ;NO: GO TRY NEXT DEVICE!
146B 23                INX     H
146C 7E                MOV     A,M       ;GET 2ND CHAR. OF NAME
146D BA                CMP     D         ;MATCH ?
146E C28B14            JNZ     PDV07     ;NO: GO TRY NEXT DEVICE!
1471 23                INX     H         ;POINT TO NUMBER OF UNITS
1472 66                MOV     H,M       ;GET NUMBER OF UNITS
1473 C1                POP     B         ;GET HANDLER PNTR
1474 D1                POP     D         ;RESTORE D&E
1475 F1                POP     PSW       ;GET UNIT NUMBER
1476 BC                CMP     H         ;VALID UNIT NUMBER ?
1477 D29814            JNC     PDV08     ;NO: ERROR!
147A 211A00            LXI     H,FHAN-FPB
147D 19                DAD     D         ;POINT TO FHAN IN FPB
147E 71                MOV     M,C       ;STORE ...
147F 23                INX     H         ;... HANDLER ...
1480 70                MOV     M,B       ;... ADDRESS
1481 23                INX     H
1482 23                INX     H
1483 77                MOV     M,A       ;STORE UNIT NUMBER
1484 F1                POP     PSW       ;RESTORE FLAGS
1485 E1                POP     H         ;RESTORE STRING PNTR
1486 37                STC               ;CLEAR ...
1487 3F                CMC               ;... C-FLAG
1488 0618              MVI     B,EMDV    ;SET ERROR CODE
148A C9                RET               ;EXIT!
148B E1        PDV07:  POP     H         ;GET HANDLER PNTR
148C 79                MOV     A,C       ;SAVE COUNTER
148D 010A00            LXI     B,10
```

DEVICE NAME CHECK

UNIT NUMBER CHECK

```
1490 09                 DAD     B       ;STEP TO NEXT HANDLER
1491 4F                 MOV     C,A     ;COPY COUNTER BACK
1492 0D                 DCR     C       ;ANY MORE DEVICES ?
1493 C25C14             JNZ     PDV06   ;YES:  TRY NEXT DEVICE!
1496 D1                 POP     D       ;RESTORE D&E
1497 F1                 POP     PSW     ;DISCARD UNIT NUMBER
1498 F1         PDV08:  POP     PSW     ;RESTORE FLAGS
1499 E1                 POP     H       ;RESTORE STRING PNTR
149A 061E               MVI     B,EIVD  ;SET ERROR CODE
149C 37                 STC             ;INDICATE ENVALID DEVICE
149D C9                 RET             ;EXIT!

149E 019311     PSFSP:  LXI     B,STYP  ;SET DEFAULT TYPE 'SRC'   } Probably Necessary — LDA may be more useful.
14A1 C3AD14             JMP     PFSPC
14A4 018911     PNFSP:  LXI     B,NTYP  ;SET BLANK DEFAULT TYPE
14A7 C3AD14             JMP     PFSPC
14AA 018D11     PPFSP:  LXI     B,PTYP  ;SET DEFAULT TYPE 'PRG'
14AD E5         PFSPC:  PUSH    H       ;SAVE STRING PNTR
14AE D5                 PUSH    D       ;SAVE FPB PNTR
14AF C5                 PUSH    B       ;SAVE DEFAULT TYPE PNTR
14B0 210800             LXI     H,FTYP-FPB
14B3 19                 DAD     D       ;POINT TO TYPE
14B4 EB                 XCHG            ;PNTR IN D&E
14B5 E1                 POP     H       ;GET DEFAULT TYPE PNTR
14B6 0603               MVI     B,3     ;SET COUNT
14B8 CDCB18             CALL    MSTR    ;MOVE DEFAULT TYPE IN
14BB D1                 POP     D       ;RESTORE FPB PNTR
14BC E1                 POP     H       ;RESTORE STRING PNTR
14BD CD1414     PCFSP:  CALL    PDV     ;PARSE DEVICE NAME
14C0 D8                 RC              ;ERROR!
14C1 D5                 PUSH    D       ;SAVE FPB PNTR
14C2 13                 INX     D
14C3 13                 INX     D       ;POINT TO NAME
14C4 CDA818             CALL    LODG    ;LETTER OR DIGIT ?
14C7 D2F714             JNC     PFS04   ;NO:  NO NAME!
14CA 0606               MVI     B,6     ;SET COUNT
14CC CDCB18             CALL    MSTR    ;MOVE NAME IN
14CF 7E                 MOV     A,M     ;GET NEXT CHAR.
14D0 FE2E               CPI     '.'     ;IS THERE A TYPE ?
14D2 CADB14             JZ      PFS01   ;YES!
14D5 13                 INX     D       ;SKIP ...
14D6 13                 INX     D       ;... OVER ...
14D7 13                 INX     D       ;... TYPE
14D8 C3E114             JMP     PFS02   ;
14DB 23         PFS01:  INX     H       ;SKIP OVER '.'
14DC 0603               MVI     B,3     ;SET COUNT
14DE CDCB18             CALL    MSTR    ;MOVE TYPE IN
14E1 AF         PFS02:  XRA     A       ;CLEAR A
14E2 12                 STAX    D       ;STORE ZERO VERSION NUMBER
14E3 7E                 MOV     A,M     ;GET NEXT CHAR.
14E4 FE3B               CPI     ';'     ;IS THERE A VERSION ?
14E6 C2F114             JNZ     PFS03   ;NO!
14E9 23                 INX     H       ;SKIP OVER ';'
14EA D5                 PUSH    D       ;SAVE PNTR TO VERSION
14EB CD2C19             CALL    GN2Z    ;GET NUMBER
14EE 7B                 MOV     A,E     ;USE LOBYTE OF NUMBER
14EF D1                 POP     D       ;RESTORE PNTR TO VERSION
```

```
14F0 12                 STAX    D       ;STORE VERSION NUMBER
14F1 1A         PFS03:  LDAX    D       ;GET VERSION NUMBER
14F2 D1                 POP     D       ;RESTORE FPB PNTR
14F3 061B               MVI     B,EMVR  ;SET STATUS CODE
14F5 A7                 ANA     A       ;TEST VERSION NUMBER
14F6 C9                 RET             ;RETURN TO CALLER
14F7 D1         PFS04:  POP     D       ;RESTORE FPB PNTR
14F8 0615               MVI     B,EMFN  ;SET ERROR CODE
14FA 37                 STC             ;INDICATE ERROR
14FB C9                 RET             ;RETURN TO CALLER
```

```
;+
;    THIS IS AN INITIAL SET OF ROUTINES TO FACILITATE FILE
; OPERATIONS WITH THE FCS SYSTEM.  ROUTINES ARE PROVIDED
; FOR SEQUENTIAL BYTE ACCESS, SEQUENTIAL RECORD ACCESS,
; AND BLOCK ACCESS.
;
;    CERTAIN PARAMETERS IN THE FILE PARAMETER BLOCK (FPB)
; MUST BE SET UP BEFORE USING ANY OF THESE ROUTINES.  THESE
; PARAMETERS SHOULD BE SET UP AFTER CALLING 'OPEN' TO OPEN
; THE FILE, BUT BEFORE ANY CALL TO ANY OF THESE ACCESS
; ROUTINES.
;
;    THE PARAMETERS ARE :
;
; FBUF - SHOULD BE SET TO THE ADDRESS OF THE USER-PROVIDED
;        BLOCK BUFFER.
;
; FXBC - FOR SEQUENTIAL ACCESS, SHOULD BE SET TO THE SIZE
;        (NUMBER OF BYTES) OF THE USER-PROVIDED BLOCK BUFFER.
;        THE SIZE SHOULD BE A MULTIPLE OF THE SYSTEM STANDARD
;        BLOCK SIZE, 128 DECIMAL.
;       FOR DIRECT BLOCK ACCESS, SHOULD BE SET TO THE NUMBER
;        OF BYTES TO BE TRANSFERRED.
;
;    WHEN USING SEQUENTIAL BYTE OR RECORD ACCESS, THESE
; PARAMETERS SHOULD NOT BE CHANGED DURING THE I/O OPERATIONS.
;    WHEN USING DIRECT BLOCK ACCESS, THESE PARAMETERS MAY
; BE SET TO THE DESIRED VALUES FOR THE TRANSFER, PRIOR TO
; EACH CALL TO RBLK, RBLKI, WBLK, OR WBLKI.  THEY'RE VALUES
; ARE PRESERVED BY THE BLOCK I/O ROUTINES, SO THEY
; DO NOT NEED TO BE RESET PRIOR TO EACH CALL UNLESS
; YOU SPECIFICALLY WANT TO CHANGE EITHER THE LOCATION
; OR THE SIZE OF THE BUFFER.
;
;   SEE THE INDIVIDUAL DESCRIPTION FOR DETAILS ON EACH
; OF THE FOLLOWING ACCESS ROUTINES.
;-


;+
; *** RWSEQI *** - "REWIND SEQUENTIAL INPUT" ROUTINE
;
;  RWSEQI IS USED TO "REWIND" A SEQUENTIAL INPUT FILE.
;
;  RWSEQI MUST BE CALLED BEFORE THE FIRST CALL TO ANY
;  OF THE SEQUENTIAL BYTE OR RECORD INPUT ROUTINES !
```

```
                        ;
                        ;  INPUTS:   HL => FPB
                        ;
                        ;  OUTPUTS:  A - LOST
                        ;            BC,DE - UNCHANGED
                        ;            HL => FPB
                        ;
                        ;  STATUS:   NONE
                        ;-

14FC E5         RWSEQI: PUSH    H       ;SAVE FPB PTR
14FD D5                 PUSH    D
14FE AF                 XRA     A       ;CLEAR A REGISTER
14FF 111E00             LXI     D,FBLK-FPB
1502 19                 DAD     D       ;POINT TO FBLK
1503 77                 MOV     M,A     ;SET VIRTUAL BLOCK = 0 ...
1504 23                 INX     H
1505 77                 MOV     M,A
1506 11F9FF             LXI     D,FAUX-FBLK-1
1509 19                 DAD     D       ;POINT TO AUX. BYTE COUNT
150A 77                 MOV     M,A     ;INITIALIZE AUX. BYTE COUNT ...
150B 23                 INX     H
150C 77                 MOV     M,A
150D D1                 POP     D
150E E1                 POP     H       ;RESTORE FPB PTR
                        ;
150F E5         ZPTR:   PUSH    H       ;SAVE FPB PTR
1510 D5                 PUSH    D
1511 112400             LXI     D,FPTR-FPB
1514 19                 DAD     D       ;POINT TO RELATIVE BUFFER POINTER
1515 3600               MVI     M,0     ;ZERO IT ...
1517 23                 INX     H
1518 3600               MVI     M,0
151A D1                 POP     D
151B E1                 POP     H       ;RESTORE FPB PTR
151C C9                 RET


                        ;+
                        ;  *** INSEQO *** "INITIALIZE SEQUENTIAL OUTPUT" ROUTINE
                        ;
                        ;   INSEQO IS USED TO INITIALIZE A NEWLY CREATED OPEN FILE
                        ;   FOR SEQUENTIAL BYTE OR RECORD OUTPUT OPERATIONS.
                        ;
                        ;   INSEQO MUST BE CALLED BEFORE THE FIRST CALL TO ANY OF
                        ;   THE SEQUENTIAL BYTE OR RECORD OUTPUT ROUTINES !
                        ;
                        ;  INPUTS:   HL => FPB
                        ;
                        ;  OUTPUTS:  A - LOST
                        ;            BC,DE - UNCHANGED
                        ;            HL => FPB
                        ;
                        ;  STATUS:   NONE
                        ;-

151D CDFC14     INSEQO: CALL    RWSEQI  ;INITIALIZE THINGS
1520 D5                 PUSH    D
```

```
1521 C5              PUSH    B
1522 110000          LXI     D,0         ;VIRTUAL BLOCK ZERO
1525 CD2B15          CALL    CBC         ;SETUP THINGS
1528 C1              POP     B
1529 D1              POP     D
152A C9              RET                 ;EXIT !
                     ;
152B E5      CBC:    PUSH    H           ;SAVE FPB PTR
152C 010E00          LXI     B,FSIZ-FPB
152F 09              DAD     B           ;POINT TO FSIZ
1530 7E              MOV     A,M         ;GET IT ...
1531 23              INX     H
1532 66              MOV     H,M
1533 6F              MOV     L,A
1534 CD8918          CALL    CMPDH       ;COMPARE WITH VIRTUAL BLOCK NUMBER
1537 DA3E15          JC      CBC1        ;O.K.: VALID BLOCK NUMBER !
153A E1              POP     H           ;RESTORE FPB PTR
153B AF              XRA     A
153C 37              STC
153D C9              RET                 ;<C><Z>: INVALID BLOCK NUMBER !
153E CD8F18  CBC1:   CALL    SUBHD       ;COMPUTE NUMBER OF REMAINING BLOCKS
1541 EB              XCHG
1542 E1              POP     H           ;GET FPB PTR
1543 E5              PUSH    H           ;SAVE FPB PTR
1544 012200          LXI     B,FXBC-FPB
1547 09              DAD     B           ;POINT TO FXBC
1548 7E              MOV     A,M         ;GET BUFFER SIZE ...
1549 23              INX     H
154A 66              MOV     H,M
154B 6F              MOV     L,A
154C CDDE19          CALL    BC2BK       ;CONVERT TO NUMBER OF BLOCKS
154F CD8318          CALL    CMPHD       ;ENTIRE BUFFER O.K. ?
1552 DA5D15          JC      CBC2        ;YES !
1555 E1              POP     H           ;GET FPB PTR
1556 E5              PUSH    H           ;SAVE FPB PTR
1557 011000          LXI     B,FLBC-FPB
155A 09              DAD     B           ;POINT TO LAST BLOCK BYTE COUNT
155B 4E              MOV     C,M         ;GET IT
155C EB              XCHG                ;HL = REMAINING BLOCKS
155D CDF019  CBC2:   CALL    BK2BC       ;CONVERT TO BYTE COUNT
1560 EB              XCHG
1561 E1              POP     H           ;GET FPB PTR
1562 E5              PUSH    H           ;SAVE FPB PTR
1563 011800          LXI     B,FAUX-FPB
1566 09              DAD     B           ;POINT TO AUX. BYTE COUNT
1567 73              MOV     M,E         ;SET IT ...
1568 23              INX     H
1569 72              MOV     M,D
156A E1              POP     H           ;RESTORE FPB PTR
156B C9              RET                 ;<NC>: O.K. !


;+
;  *** CLSEQO *** "CLOSE SEQUENTIAL OUTPUT" ROUTINE
;
;    CLSEQO IS USED TO CLOSE A NEWLY CREATED SEQUENTIAL
;    OUTPUT FILE. THE REMAINING UNWRITTEN PART OF THE BLOCK
;    BUFFER IS WRITTEN OUT, IF ANY, AND THE FINAL FILE SIZE
```

```
                ;   IS CALCULATED AND APPROPRIATE INFORMATION UPDATED IN
                ;   THE FPB.  THEN 'CLOSE' IS CALLED TO ENTER THE FILE INTO
                ;   THE DIRECTORY.
                ;
                ;  INPUTS:   HL => FPB
                ;
                ;  OUTPUTS:  A,BC,DE - LOST
                ;            HL  => FPB IF NO ERRORS, ELSE HL LOST
                ;
                ;  STATUS:   <NC> - NO ERRORS, B=0
                ;            <C>  - FILE WRITE ERROR OR DIRECTORY WRITE ERROR,
                ;                   WITH B = SYSTEM ERROR CODE
                ;-

156C E5         CLSEQO: PUSH    H           ;SAVE FPB PTR
156D 112400             LXI     D,FPTR-FPB
1570 19                 DAD     D           ;POINT TO RELATIVE BUFFER POINTER
1571 5E                 MOV     E,M         ;GET IT ...
1572 23                 INX     H
1573 56                 MOV     D,M
1574 01EBFF             LXI     B,FLBC-FPTR-1
1577 09                 DAD     B           ;POINT TO LAST BLOCK BYTE COUNT
1578 EB                 XCHG
1579 CDDE19             CALL    BC2BK       ;CONVERT TO BLOCKS & LAST BLOCK BYTE COUNT
157C EB                 XCHG
157D 71                 MOV     M,C         ;SET LAST BLOCK BYTE COUNT
157E 010E00             LXI     B,FBLK-FLBC
1581 09                 DAD     B           ;POINT TO NEXT VIRTUAL BLOCK
1582 7E                 MOV     A,M         ;GET IT ...
1583 23                 INX     H
1584 66                 MOV     H,M
1585 6F                 MOV     L,A
1586 19                 DAD     D           ;COMPUTE ACTUAL FILE SIZE
1587 EB                 XCHG
1588 E1                 POP     H           ;GET FPB PTR
1589 E5                 PUSH    H           ;SAVE FPB PTR
158A 010E00             LXI     B,FSIZ-FPB
158D 09                 DAD     B           ;POINT TO FILE SIZE
158E 4E                 MOV     C,M         ;GET IT ...
158F 23                 INX     H
1590 46                 MOV     B,M
1591 72                 MOV     M,D         ;SET ACTUAL FILE SIZE ...
1592 2B                 DCX     H
1593 73                 MOV     M,E
1594 E1                 POP     H           ;RESTORE FPB PTR
1595 C5                 PUSH    B           ;SAVE ORIGINAL FILE SIZE
1596 CD3816             CALL    IWBKI       ;WRITE REMAINING PART OF BUFFER
1599 C1                 POP     B           ;GET ORIGINAL FILE SIZE
159A F5                 PUSH    PSW         ;SAVE STATUS
159B E5                 PUSH    H           ;SAVE FPB PTR
159C 110E00             LXI     D,FSIZ-FPB
159F 19                 DAD     D           ;POINT TO FILE SIZE
15A0 5E                 MOV     E,M         ;GET ACTUAL FILE SIZE ...
15A1 23                 INX     H
15A2 56                 MOV     D,M
15A3 70                 MOV     M,B         ;RESTORE ORIGINAL FILE SIZE ...
15A4 2B                 DCX     H
15A5 71                 MOV     M,C
```

```
15A6 010A00            LXI     B,FAUX-FSIZ
15A9 09                DAD     B       ;POINT TO ACTUAL FILE SIZE
15AA 73                MOV     M,E     ;SET ACTUAL FILE SIZE FOR CLOSE ...
15AB 23                INX     H
15AC 72                MOV     M,D
15AD E1                POP     H       ;RESTORE FPB PTR
15AE F1                POP     PSW     ;RESTORE STATUS
15AF 0633              MVI     B,EFWR
15B1 D8                RC              ;<C>: FILE WRITE ERROR !
15B2 C35C13            JMP     CLOSE   ;CLOSE THE FILE AND EXIT !


                ;+
                ; *** RBLK *** "READ BLOCK" ROUTINE
                ; *** WBLK *** "WRITE BLOCK" ROUTINE
                ;
                ;  RBLK AND WBLK ARE USED TO READ/WRITE TO/FROM A
                ;  SPECIFIED VIRTUAL BLOCK NUMBER IN A FILE.
                ;
                ; INPUTS:  HL => FPB
                ;        FBLK =  DESIRED STARTING VIRTUAL BLOCK NUMBER
                ;        FBUF => BLOCK BUFFER
                ;        FXBC = NUMBER OF BYTES TO READ/WRITE
                ;
                ; OUTPUTS: A - LOST
                ;          BC,DE - UNCHANGED
                ;          HL => FPB
                ;          FBLK,FBUF,FXBC - UNCHANGED
                ;
                ; STATUS:  <NC><Z> - NO ERRORS:
                ;            FAUX= NUMBER OF BYTES TRANSFERRED = (FXBC).
                ;          <NC><NZ> - TRANSFER TRUNCATED BY END-OF-FILE
                ;            FAUX= NUMBER OF BYTES TRANSFERRED < (FXBC).
                ;          <C><Z> - VIRTUAL BLOCK NOT WITHIN FILE:
                ;            FAUX UNCHANGED.
                ;          <C><NZ><M> - READ/WRITE ERROR:
                ;            FAUX = NUMBER OF BYTES ATTEMPTED.
                ;-

15B5 3E01       WBLK:   MVI     A,1     ;SET WRITE FUNCTION CODE
15B7 FE                 DB      OFEH    ;*** SKIP ONE BYTE *** (CPI)
15B8 AF         RBLK:   XRA     A       ;SET READ FUNCTION CODE
15B9 D5                 PUSH    D
15BA C5                 PUSH    B
15BB CDC515             CALL    RWBCM   ;DO THE TRANSFER
15BE C1                 POP     B
15BF D1                 POP     D
15C0 C9                 RET             ;EXIT !

15C1 3E01       IWBK:   MVI     A,1     ;SET WRITE FUNCTION CODE
15C3 FE                 DB      OFEH    ;*** SKIP ONE BYTE *** (CPI)
15C4 AF         IRBK:   XRA     A       ;SET READ FUNCTION CODE
15C5 E5         RWBCM:  PUSH    H       ;SAVE FPB PTR
15C6 111C00             LXI     D,FFCN-FPB
15C9 19                 DAD     D       ;POINT TO HANDLER FUNCTION CODE
15CA 77                 MOV     M,A     ;SET FUNCTION CODE
15CB 110200             LXI     D,FBLK-FFCN
15CE 19                 DAD     D       ;POINT TO VIRTUAL BLOCK NUMBER
```

```
15CF 5E          MOV     E,M     ;GET IT ...
15D0 23          INX     H
15D1 56          MOV     D,M
15D2 E1          POP     H       ;GET FPB PTR
15D3 CD2B15      CALL    CBC     ;CALL COMMON ROUTINE
15D6 D8          RC              ;<C><Z>: INVALID BLOCK NUMBER !
15D7 012200      LXI     B,FXBC-FPB
15DA 09          DAD     B       ;POINT TO FXBC
15DB 4E          MOV     C,M     ;GET REQUESTED BYTE COUNT ...
15DC 23          INX     H
15DD 46          MOV     B,M
15DE C5          PUSH    B       ;SAVE REQUESTED BYTE COUNT
15DF 72          MOV     M,D     ;SET BYTE COUNT FOR XFER
15E0 2B          DCX     H
15E1 73          MOV     M,E
15E2 01EAFF      LXI     B,FSBK-FXBC
15E5 09          DAD     B       ;POINT TO STARTING BLOCK NUMBER
15E6 5E          MOV     E,M     ;GET IT ...
15E7 23          INX     H
15E8 56          MOV     D,M
15E9 011100      LXI     B,FBLK-FSBK-1
15EC 09          DAD     B       ;POINT TO VIRTUAL BLOCK NUMBER
15ED 4E          MOV     C,M     ;GET IT ...
15EE 23          INX     H
15EF 46          MOV     B,M
15F0 C5          PUSH    B       ;SAVE VIRTUAL BLOCK NUMBER
15F1 EB          XCHG
15F2 09          DAD     B       ;COMPUTE ABSOLUTE BLOCK NUMBER
15F3 EB          XCHG
15F4 72          MOV     M,D     ;SET ABSOLUTE BLOCK NUMBER ...
15F5 2B          DCX     H
15F6 73          MOV     M,E
15F7 E5          PUSH    H       ;SAVE PTR TO FBLK
15F8 01FCFF      LXI     B,FHAN-FBLK
15FB 09          DAD     B       ;POINT TO FHAN
15FC 5E          MOV     E,M     ;GET HANDLER ADDRESS ...
15FD 23          INX     H
15FE 56          MOV     D,M
15FF 23          INX     H
1600 CD3713      CALL    JMPD    ;PERFORM THE READ/WRITE
1603 E1          POP     H       ;GET PTR TO FBLK
1604 D1          POP     D       ;GET VIRTUAL BLOCK NUMBER
1605 73          MOV     M,E     ;RESTORE VIRTUAL BLOCK NUMBER ...
1606 23          INX     H
1607 72          MOV     M,D
1608 F5          PUSH    PSW     ;SAVE HANDLER STATUS
1609 010300      LXI     B,FXBC-FBLK-1
160C 09          DAD     B       ;POINT TO FXBC
160D F1          POP     PSW     ;RESTORE HANDLER STATUS
160E C1          POP     B       ;GET REQUESTED BYTE COUNT
160F 5E          MOV     E,M     ;GET ATTEMPTED BYTE COUNT ...
1610 23          INX     H
1611 56          MOV     D,M
1612 70          MOV     M,B     ;RESTORE REQUESTED BYTE COUNT ...
1613 2B          DCX     H
1614 71          MOV     M,C
1615 DA2216      JC      RWB1    ;ERROR DURING TRANSFER !
1618 E5          PUSH    H       ;SAVE POINTER TO FXBC
```

```
1619 60                 MOV     H,B
161A 69                 MOV     L,C
161B CD8318             CALL    CMPHD   ;CHECK IF ATTEMPTED = REQUESTED
161E E1                 POP     H       ;RESTORE POINTER TO FXBC
161F C32516             JMP     RWB2    ;FINISH CLEANUP & EXIT
1622 F6FF       RWB1:   ORI     -1
1624 37                 STC             ;<C><NZ><M>:  I/O ERROR STATUS
1625 F5         RWB2:   PUSH    PSW     ;SAVE STATUS
1626 01DEFF             LXI     B,FPB-FXBC
1629 09                 DAD     B       ;RESTORE FPB PTR
162A F1                 POP     PSW     ;RESTORE CONDITION CODES
162B C9                 RET             ;EXIT !

                ;+
                ;  *** RBLKI *** "READ BLOCK & INCREMENT" ROUTINE
                ;  *** WBLKI *** "WRITE BLOCK & INCREMENT" ROUTINE
                ;
                ;   RBLKI AND WBLKI ARE IDENTICAL IN FUNCTION TO
                ;   RBLK AND WBLK EXCEPT:  IF NO ERRORS OCCURRED, THEN
                ;   FBLK IS SET TO THE NEXT VIRTUAL BLOCK NUMBER
                ;   FOR SEQUENTIAL ACCESS.  IF ANY ERRORS OCCURRED,
                ;   THEN FBLK IS UNCHANGED.
                ;-

162C 3E01       WBLKI:  MVI     A,1     ;SET WRITE FUNCTION CODE
162E FE                 DB      0FEH    ;*** SKIP ONE BYTE *** (CPI)
162F AF         RBLKI:  XRA     A       ;SET READ FUNCTION CODE
1630 D5                 PUSH    D
1631 C5                 PUSH    B
1632 CD3C16             CALL    RWICM   ;DO THE TRANSFER
1635 C1                 POP     B
1636 D1                 POP     D
1637 C9                 RET             ;EXIT !

1638 3E01       IWBKI:  MVI     A,1     ;SET WRITE FUNCTION CODE
163A FE                 DB      0FEH    ;*** SKIP ONE BYTE *** (CPI)
163B AF         IRBKI:  XRA     A       ;SET READ FUNCTION CODE
163C CDC515     RWICM:  CALL    RWBCM   ;DO THE TRANSFER
163F D8                 RC              ;INVALID BLOCK OR READ/WRITE ERROR !
1640 F5                 PUSH    PSW     ;SAVE CONDITION CODES
1641 011E00             LXI     B,FBLK-FPB
1644 09                 DAD     B       ;POINT TO VIRTUAL BLOCK NUMBER
1645 E5                 PUSH    H       ;SAVE PTR TO FBLK
1646 7E                 MOV     A,M     ;GET VIRTUAL BLOCK NUMBER
1647 23                 INX     H
1648 66                 MOV     H,M
1649 6F                 MOV     L,A
164A EB                 XCHG            ;HL = BYTES TRANSFERED
164B CDDE19             CALL    BC2BK   ;CONVERT TO BLOCKS
164E 19                 DAD     D       ;COMPUTE NEXT VIRTUAL BLOCK NUMBER
164F EB                 XCHG
1650 E1                 POP     H       ;GET PTR TO FBLK
1651 73                 MOV     M,E     ;SET NEXT VIRTUAL BLOCK NUMBER ...
1652 23                 INX     H
1653 72                 MOV     M,D
1654 01E1FF             LXI     B,FPB-FBLK-1
1657 09                 DAD     B       ;RESTORE FPB PTR
```

```
1658 F1                 POP     PSW     ;RESTORE CONDITION CODES
1659 C9                 RET             ;EXIT: EITHER <NC><Z> OR <NC><NZ> !

                ;+
                ; *** GTBYT *** "GET BYTE" ROUTINE
                ;
                ;  GTBYT IS USED TO READ SEQUENTIAL BYTES FROM AN OPEN
                ;  FILE.
                ;
                ;  RWSEQI MUST HAVE BEEN CALLED BEFORE THE FIRST CALL
                ;  TO GTBYT !
                ;
                ; INPUTS:   HL => FPB
                ;
                ; OUTPUTS:  A = THE BYTE, IF NO ERRORS
                ;           BC,DE - UNCHANGED
                ;           HL => FPB
                ;
                ; STATUS:   <NC> - NO ERRORS, A= THE BYTE
                ;           <C><Z> - END OF FILE
                ;           <C><NZ><M> - READ ERROR
                ;-

165A E1         GB1:    POP     H       ;RESTORE FPB PTR
165B CD2F16             CALL    RBLKI   ;READ NEXT BUFFERFULL
165E D8                 RC              ;END-OF-FILE OR READ ERROR !
165F CD0F15             CALL    ZPTR    ;RESET RELATIVE BUFFER POINTER
1662 E5         GTBYT:  PUSH    H       ;SAVE FPB PNTR
1663 CDC816             CALL    BCHK    ;NEED TO READ NEXT BUFFERFULL ?
1666 D25A16             JNC     GB1     ;YES !
1669 AF                 XRA     A       ;SET GOOD CODES
166A 7E                 MOV     A,M     ;GET THE BYTE
166B E1                 POP     H       ;RESTORE FPB PTR
166C C9                 RET             ;<NC>: NO ERROR !

                ;+
                ; *** PTBYT *** "PUT BYTE" ROUTINE
                ;
                ;  PTBYT IS USED TO WRITE SEQUENTIAL BYTES TO AN OPEN FILE.
                ;
                ;  INSEQO MUST BE CALLED AFTER OPEN AND BEFORE THE FIRST
                ;  CALL TO PTBYT !
                ;
                ; INPUTS:   A = THE BYTE
                ;           HL => FPB
                ;
                ; OUTPUTS:  A = THE BYTE
                ;           BC,DE - UNCHANGED
                ;           HL => FPB
                ;
                ; STATUS:   <NC> - NO ERRORS
                ;           <C><Z> - FILE FULL, BYTE NOT WRITTEN
                ;           <C><NZ><M> - WRITE ERROR
                ;-

166D E1         PB1:    POP     H       ;GET THE BYTE
```

```
166E EB                 XCHG                ;D = THE BYTE
166F E3                 XTHL                ;GET FPB PTR, SAVE ORIG. DE
1670 C5                 PUSH    B           ;SAVE BC
1671 D5                 PUSH    D           ;SAVE THE BYTE
1672 CD3816             CALL    IWBKI       ;WRITE THE BUFFER OUT
1675 D42B15             CNC     CBC         ;CALL COMMON ROUTINE IF NO ERROR
1678 D1                 POP     D           ;GET THE BYTE BACK
1679 7A                 MOV     A,D         ;... INTO A
167A C1                 POP     B           ;RESTORE BC
167B D1                 POP     D           ;RESTORE DE
167C D8                 RC                  ;WRITE ERROR OR FILE FULL !
167D CD0F15             CALL    ZPTR        ;RESET RELATIVE BUFFER POINTER
1680 E5         PTBYT:  PUSH    H           ;SAVE FPB PTR
1681 F5                 PUSH    PSW         ;SAVE THE BYTE
1682 CDC816             CALL    BCHK        ;NEED TO WRITE OUT BLOCK BUFFER ?
1685 D26D16             JNC     PB1         ;YES !
1688 F1                 POP     PSW         ;GET THE BYTE
1689 77                 MOV     M,A         ;PUT BYTE INTO BUFFER
168A E1                 POP     H           ;RESTORE FPB PTR
168B A7                 ANA     A           ;SET GOOD CODES
168C C9                 RET                 ;<NC>: NO ERRORS !


                ;+
                ; *** GAREC *** "GET ASCII RECORD" ROUTINE
                ;
                ;  GAREC IS USED TO READ SEQUENTIAL RECORDS FROM AN
                ;  ASCII FILE.  A RECORD IS A STRING OF ASCII CHARACTERS
                ;  TERMINATED BY EITHER A LINE FEED (10.) OR A
                ;  FORM FEED (12.) .
                ;
                ;  RWSEGI MUST HAVE BEEN CALLED BEFORE THE FIRST CALL
                ;  TO GAREC !
                ;
                ; INPUTS:   HL => FPB
                ;           BC => RECORD BUFFER
                ;           DE =  RECORD BUFFER LENGTH (BYTES)
                ;
                ; OUTPUTS:  HL => FPB
                ;           BC => PAST LAST BYTE STORED
                ;           DE = NUMBER OF BYTES READ
                ;           A - LOST
                ;
                ; STATUS:   <NC> - NO ERRORS
                ;           <C><Z> - END OF FILE
                ;           <C><NZ><M> - READ ERROR
                ;           <C><NZ><P> - BUFFER WAS FILLED, BUT A VALID
                ;                        TERMINATOR WAS NOT SEEN.  THE NEXT
                ;                        CALL TO GAREC WILL START WITH THE
                ;                        NEXT SEQUENTIAL BYTE.
                ;-

168D D5         GAREC:  PUSH    D           ;SAVE BUFFER LENGTH
168E CD6216     GAR1:   CALL    GTBYT       ;GET NEXT BYTE
1691 DAA816             JC      GAR2        ;ERROR OR EOF !
1694 02                 STAX    B           ;STORE THE BYTE
1695 03                 INX     B
1696 1B                 DCX     D           ;COUNT IT
```

```
16[illegible] FE0A              CPI     10      ;LINE FEED ?
1699 CAA816             JZ      GAR2    ;YES: <NC>: NO ERRORS !
169C FE0C               CPI     12      ;FORM FEED ?
169E CAA816             JZ      GAR2    ;YES: <NC>: NO ERRORS !
16A1 7A                 MOV     A,D
16A2 B3                 ORA     E       ;RECORD BUFFER FULL ?
16A3 C28E16             JNZ     GAR1    ;NO: GET ANOTHER BYTE !
16A6 3C                 INR     A
16A7 37                 STC             ;<C><NZ><P>: LINE TOO LONG
16A8 E3         GAR2:   XTHL            ;GET BUFFER LENGTH
16A9 F5                 PUSH    PSW     ;SAVE STATUS
16AA CD8F18             CALL    SUBHD   ;COMPUTE BYTE COUNT
16AD EB                 XCHG            ;DE = BYTE COUNT
16AE F1                 POP     PSW     ;RESTORE STATUS
16AF E1                 POP     H       ;RESTORE FPB PTR
16B0 C9                 RET             ;EXIT !

;+
; *** PVREC *** "PUT VARIABLE LENGTH RECORD" ROUTINE
;
;  PVREC IS USED TO PUT A VARIABLE LENGTH RECORD INTO A
;  SYSTEM STANDARD 'VARIABLE LENGTH RECORD, SEQUENTIAL' FILE.
;  WITHIN THE FILE, EACH RECORD CONSISTS OF A TWO BYTE BYTE
;  COUNT, LOW BYTE FIRST, FOLLOWED BY THAT NUMBER OF DATA
;  BYTES.
;
; INPUTS:   HL => FPB
;           BC => RECORD BUFFER
;           DE = RECORD LENGTH (BYTES)
;
; OUTPUTS:  HL => FPB
;           BC => PAST LAST BYTE IN RECORD BUFFER
;           DE = 0 IF NO ERRORS
;           A - LOST
;
; STATUS:  <NC> - NO ERRORS
;          <C><Z> - TRANSFER TERMINATED BY END OF FILE - FILE FULL
;          <C><NZ><M> - WRITE ERROR
;-

16B1 7B         PVREC:  MOV     A,E
16B2 CD8016             CALL    PTBYT   ;PUT LOBYTE OF BYTE COUNT
16B5 D8                 RC              ;ERROR OR EOF !
16B6 7A                 MOV     A,D
16B7 CD8016             CALL    PTBYT   ;PUT HIBYTE OF BYTE COUNT
16BA D8                 RC              ;ERROR OF EOF !
; FALL INTO PTREC !

;+
; *** PTREC *** "PUT UNFORMATTED RECORD" ROUTINE
;
;  PTREC IS USED TO PUT AN 'UNFORMATTED' RECORD INTO A
;  SEQUENTIAL FILE. OPERATION OF PTREC IS IDENTICAL TO
;  PVREC, ABOVE, EXCEPT THAT THE TWO BYTE BYTE COUNT
;  IS NOT WRITTEN INTO THE FILE.
;-
```

```
16BB 0A        PTREC:  LDAX    B       ;GET NEXT BYTE
16BC CD8016            CALL    PTBYT   ;PUT THE BYTE
16BF D8                RC              ;ERROR OR EOF !
16C0 03                INX     B
16C1 1B                DCX     D       ;COUNT THE BYTE
16C2 7A                MOV     A,D
16C3 B3                ORA     E       ;RECORD ALL DONE ?
16C4 C2BB16            JNZ     PTREC   ;NO: PUT ANOTHER BYTE !
16C7 C9                RET             ;<NC>: NO ERRORS !

               ;+
               ; * BCHK *
               ;-

16C8 D5        BCHK:   PUSH    D
16C9 C5                PUSH    B
16CA 111800            LXI     D,FAUX-FPB
16CD 19                DAD     D       ;POINT TO AUX. BYTE COUNT
16CE 5E                MOV     E,M     ;GET IT ...
16CF 23                INX     H
16D0 56                MOV     D,M
16D1 010B00            LXI     B,FPTR-FAUX-1
16D4 09                DAD     B       ;POINT TO RELATIVE BUFFER POINTER
16D5 E5                PUSH    H       ;SAVE POINTER TO FPTR
16D6 7E                MOV     A,M     ;GET IT ...
16D7 23                INX     H
16D8 66                MOV     H,M
16D9 6F                MOV     L,A
16DA EB                XCHG
16DB CD8918            CALL    CMPDH   ;NEED TO DO A TRANSFER ?
16DE E1                POP     H       ;GET POINTER TO FPTR
16DF D2F116            JNC     BCHK1   ;<NC>: NEED A TRANSFER !
16E2 13                INX     D       ;INCREMENT RELATIVE BUFFER POINTER
16E3 73                MOV     M,E     ;SAVE UPDATED PTR ...
16E4 23                INX     H
16E5 72                MOV     M,D
16E6 1B                DCX     D       ;RESTORE RELATIVE BUFFER POINTER
16E7 01FBFF            LXI     B,FBUF-FPTR-1
16EA 09                DAD     B       ;POINT TO BUFFER PTR
16EB 7E                MOV     A,M     ;GET IT ...
16EC 23                INX     H
16ED 66                MOV     H,M
16EE 6F                MOV     L,A
16EF 19                DAD     D       ;INDEX INTO BUFFER
16F0 37                STC             ;<C>: NO TRANSFER NEEDED
16F1 C1        BCHK1:  POP     B
16F2 D1                POP     D
16F3 C9                RET             ;<C>: NO TRANSFER , <NC>: TRANSFER !

               ;;      FCS COMMAND INPUT ROUTINE

16F4 360D      ESCQ:   MVI     M,13    ;SETUP INPUT FLAG
16F6 7D                MOV     A,L
16F7 21F981            LXI     H,LOFL  ;SET UP FCS OUTPUT TO PORT
16FA 360E              MVI     M,14
```

```
16FC C30717             JMP     ESCDG   ; CONTINUE WITH COMMON

16FF 360D       ESCD:   MVI     M,13    ;SETUP INPUT FLAG
1701 7D                 MOV     A,L
1702 21F981             LXI     H,LOFL  ;SET UP FCS OUTPUT FLAG
1705 3600               MVI     M,0     ; TO BE CRT
1707 FEDF       ESCDG:  CPI     KBDFL AND 255
1709 21E181             LXI     H,FCSFL ;SET UP FCS ECHO FLAG
170C 3600               MVI     M,0     ;SET FOR ECHO TO SCREEN
170E CA1F17             JZ      L1      ;YES KEYBOARD!
1711 FEF1               CPI     BASFL AND 255
1713 360C               MVI     M,12    ;DON'T ECHO IF BASIC INPUT
1715 CA1F17             JZ      L1      ;YES BASIC !
1718 360E               MVI     M,14    ; ECHO TO OUTPUT PORT
171A 21F981             LXI     H,LOFL  ;SET LO TO OUTPUT PORT
171D 360E               MVI     M,14
171F 214780     L1:     LXI     H,BUFP  ;SET BUFFER PNTR
1722 22F681             SHLD    TEMP4   ;SAVE IT
1725 CD480B             CALL    RESET   ;
1728 3E0B               MVI     A,11
172A 21B817             LXI     H,PRMPT ;POINT TO PROMPT
172D CDAF17     L2:     CALL    FCSOUT
1730 7E                 MOV     A,M
1731 A7                 ANA     A
1732 C8                 RZ              ;FINISHED WITH PROMPT!
1733 23                 INX     H
1734 C32D17             JMP     L2      ;LOOP!

1737 7B         FCSX:   MOV     A,E     ;GET CHARACTER
1738 FE1B               CPI     27      ;IS IT ESCAPE ?
173A CAA817             JZ      EFCS    ;YES!
173D FE7F               CPI     127     ;RUB OUT ?
173F CA6A17             JZ      L4      ;YES !
1742 FE20               CPI     32      ;CONTROL CHARACTER ?
1744 DA6017             JC      L3      ;YES!
1747 D6FF               SUI     255     ; IS IT BASIC EXIT ?
1749 CAA817             JZ      EFCS    ; YES, EXIT
174C 2AF681             LHLD    TEMP4
174F 3E81               MVI     A,(BUFP AND 255)+58
1751 BD                 CMP     L       ;BUFFER FULL
1752 3E07               MVI     A,7
1754 CAAF17             JZ      FCSOUT  ;YES: BEEP !
1757 73                 MOV     M,E     ;STORE CHARACTER IN BUFFER
1758 23                 INX     H       ;BUMP POINTER
1759 22F681             SHLD    TEMP4   ;SAVE BUFFER PNTR
175C 7B                 MOV     A,E     ;GET CHAR.
175D C3AF17             JMP     FCSOUT  ;GO ECHO IT
1760 FE0B       L3:     CPI     11      ;ERASE LINE ?
1762 CA1F17             JZ      L1      ;YES!
1765 FE1A               CPI     26      ;CURSOR LEFT ?
1767 C28417             JNZ     EL[illegible]   ;NO: TERMINATE [illegible] LINE!
176A 2AF681     L4:     LHLD    TEMP4
176D 3E47               MVI     A,BUFP AND 255
176F BD                 CMP     L       ;BUFFER EMPTY ?
1770 CA1F17             JZ      L1      ;YES: ERASE LINE!
1773 2B                 DCX     H       ;BUMP PNTR BACK
1774 22F681             SHLD    TEMP4   ;SAVE BUFFER PNTR
1777 CD7F17             CALL    L5      ;PRINT CURSOR LEFT
```

```
177A 3E20              MVI     A,32     ;SPACE
177C CDAF17            CALL    FCSOUT   ;PRINT IT
177F 3E1A      L5:     MVI     A,26     ;CURSOR LEFT
1781 C3AF17            JMP     FCSOUT   ;PRINT IT & RETURN

1784 CDAF17    ELIN:   CALL    FCSOUT   ;ECHO
1787 3E0A              MVI     A,10
1789 CDAF17            CALL    FCSOUT
178C 7E                MOV     A,M
178D F5                PUSH    PSW
178E 3600              MVI     M,0

               ;       SHLD    TEMPHL   ;* CHANGE 'SHLD' AND 'LHLD' OF
                                        ;* 'TEMPHL' TO 2 'NOP'S AND A
1790 E5                PUSH    H        ;* 'PUSH' AND 'POP' RESPECTIVELY.  THE
1791 00                NOP              ;* NUMBER OF BYTES USED IS THE SAME.
1792 00                NOP              ;* THIS IS TO FREE UP TWO BYTES OF RAM
                                        ;* FOR FUTURE USE.  THE SUBSTITUTE CODE
                                        ;* IS FUNCTIONALLY IDENTICAL.
                                        ;*  1/15/80  JKP III

1793 2AF681            LHLD    TEMP4
1796 3600              MVI     M,0
1798 214780            LXI     H,BUFP   ;SET BUFFER PNTR
179B 7E                MOV     A,M
179C A7                ANA     A        ;EMPTY ?
179D C4D30A            CNZ     FCSEM    ;NO:  GO TO FCS!

               ;       LHLD    TEMPHL   ;* SEE COMMENT ABOVE
                                        ;*  1/15/80  JKP III
17A0 E1                POP     H        ;*
17A1 00                NOP              ;*
17A2 00                NOP              ;*

17A3 F1                POP     PSW
17A4 77                MOV     M,A
17A5 C31F17            JMP     L1

17A8 77        EFCS:   MOV     M,A      ;SETUP INPUT FLAG
17A9 21F981            LXI     H,LOFL   ;  CLEAR LO FLAG
17AC 3600              MVI     M,0
17AE C9                RET

17AF E5        FCSOUT: PUSH    H        ;SAVE MEMORY POINTER
17B0 21E181            LXI     H,FCSFL  ;SETUP FOR FCS OUTPUT
17B3 CDD503            CALL    PROCES   ;ECHO CHAR IN A REG.
17B6 E1                POP     H        ;RESTORE MEMORY POINTER
17B7 C9                RET

17B8 06034643  PRMPT:  DB      6,3,'FCS>',6,2,0
17BC 533E0602
17C0 00
```


[redacted]

```
               ;;
               ;

     (0008)    SSIDA   EQU     08H
```

```
     (0010)    SSOBE    EQU     10H

17C1 3E0D      CRLF:    MVI     A,13      ;ASCII CR
17C3 CDC817             CALL    LO
17C6 3E0A               MVI     A,10      ;ASCII LF

17C8 CD2E0A    LO:      CALL    SAVE      ;SAVE ALL REGS.
17CB 21F981             LXI     H,LOFL
17CE C3D503             JMP     PROCES

17D1 F5        LBYT:    PUSH    PSW       ;SAVE BYTE
17D2 0F                 RRC
17D3 0F                 RRC
17D4 0F                 RRC
17D5 0F                 RRC
17D6 CDDA17             CALL    LHXD      ;LIST 1ST HEX DIGIT
17D9 F1                 POP     PSW       ;RETRIEVE BYTE
17DA CDE017    LHXD:    CALL    B2HEX     ;CONVERT VALUE TO HEX DIGIT
17DD C3C817             JMP     LO        ;LIST DIGIT AND RETURN!

17E0 E60F      B2HEX:   ANI     0FH       ;CLEANSE VALUE
17E2 C690               ADI     90H
17E4 27                 DAA
17E5 CE40               ACI     40H
17E7 27                 DAA
17E8 C9                 RET               ;RETURN: A=ASCII HEX DIGIT

17E9 D630      NIBL:    SUI     '0'
17EB D8                 RC                ;NOT HEX: RETURN <C>
17EC FE0A               CPI     10
17EE 3F                 CMC
17EF D0                 RNC               ;VALID: RETURN <NC>
17F0 FE11               CPI     'A'-'0'
17F2 D8                 RC                ;NOT HEX: RETURN <C>
17F3 D607               SUI     'A'-'9'-1
17F5 FE10               CPI     16
17F7 3F                 CMC
17F8 C9                 RET               ;HEX: <NC>, NOT HEX: <C>

17F9 DB03      S1OUT:   IN      STAT5     ;GET STATUS
17FB E610               ANI     SSOBE     ;CAN WE SEND ?
17FD CAF917             JZ      S1OUT     ;NO: WAIT!
1800 DB01               IN      EXTIN     ;IS CLEAR TO SEND HI ?
1802 E680               ANI     80H
1804 CAF917             JZ      S1OUT
1807 7B                 MOV     A,E       ;GET CHARACTER TO SEND
1808 D306               OUT     TXBUF     ;SEND THE CHARACTER
180A C9                 RET               ;RETURN

180B E5        RTST:    PUSH    H
180C D5                 PUSH    D
180D C5                 PUSH    B
180E 3AFF81    RTST2:   LDA     KBRDY
1811 FE50               CPI     50H
1813 CA0E18             JZ      RTST2
1816 FE80               CPI     80H
1818 3AFE81             LDA     KBCHA
```

```
181B C1                 POP     B
181C D1                 POP     D
181D E1                 POP     H
181E C9                 RET

                ; SEND A MESSAGE STRING TO "LO" DEVICE.  BYTE VALUE OF
                ; 239 DECIMAL TERMINATES THE STRING.
                ; ALSO SUPPORTS REPEAT LOOPS OF THE FORM :
                ;       ...,237,N,D1,D2,...,DM,238,...
                ; WHERE N IS THE REPEAT COUNT FOR THE STRING OF
                ; BYTES:  D1,D2,...,DM .

181F 7E         BSTR:   MOV     A,M
1820 23                 INX     H
1821 FEEF               CPI     239
1823 C8                 RZ
1824 CDB403             CALL    CRTUBE
1827 C31F18             JMP     BSTR

182A 7E         OSTR:   MOV     A,M     ;GET NEXT BYTE
182B 23                 INX     H
182C FEED               CPI     237     ;SPECIAL CODE ?
182E DA3A18             JC      OSTR1   ;A<237:  GO SEND BYTE
1831 CA4018             JZ      OSTR2   ;A=237:  START REPEAT LOOP
1834 FEEF               CPI     239     ;SPECIAL CODE ?
1836 C8                 RZ              ;A=239:  END:  RETURN
1837 DA4618             JC      OSTR4   ;A=238:  END OF REPEAT LOOP
183A CDC817     OSTR1:  CALL    LO      ;SEND BYTE
183D C32A18             JMP     OSTR    ;GET NEXT BYTE
1840 56         OSTR2:  MOV     D,M     ;GET REPEAT COUNT
1841 23                 INX     H
1842 E5         OSTR3:  PUSH    H       ;SAVE START POINTER
1843 C32A18             JMP     OSTR    ;GET NEXT BYTE
1846 15         OSTR4:  DCR     D       ;FINISHED REPEAT ?
1847 CA4E18             JZ      OSTR5   ;YES!
184A E1                 POP     H       ;RESTORE START POINTER
184B C34218             JMP     OSTR3   ;REPEAT AGAIN!
184E F1         OSTR5:  POP     PSW     ;CLEAN THE STACK
184F C32A18             JMP     OSTR    ;GET NEXT BYTE!

                ;;      SHORT WAIT:  0.5 MS.  / COUNT :

                ;STIM   EQU     50      ;FOR 'WAIT' STATE IN ROM
     (003F)     STIM    EQU     63      ;FOR NO 'WAIT' STATE IN ROM

1852 F5         WATS:   PUSH    PSW
1853 3E3F               MVI     A,STIM
1855 3D         WS1:    DCR     A
1856 C25518             JNZ     WS1
1859 F1                 POP     PSW
185A 3D                 DCR     A
185B C25218             JNZ     WATS
185E C9                 RET

                ;;      LONG WAIT:  20 MS.  / COUNT :
```

```
                ;LTIM   EQU     2347    ;FOR 'WAIT' STATE IN ROM
     (0B5A)     LTIM    EQU     2906    ;FOR NO 'WAIT' STATE IN ROM

185F E5         WATL:   PUSH    H
1860 215A0B     WL1:    LXI     H,LTIM
1863 2D         WL2:    DCR     L
1864 C26318             JNZ     WL2
1867 25                 DCR     H
1868 C26318             JNZ     WL2
186B 3D                 DCR     A
186C C26018             JNZ     WL1
186F E1                 POP     H
1870 C9                 RET

1871 7E         MOVDH:  MOV     A,M
1872 12                 STAX    D
1873 23                 INX     H
1874 13                 INX     D
1875 05                 DCR     B
1876 C27118             JNZ     MOVDH
1879 C9                 RET

187A 1A         MOVHD:  LDAX    D
187B 77                 MOV     M,A
187C 13                 INX     D
187D 23                 INX     H
187E 05                 DCR     B
187F C27A18             JNZ     MOVHD
1882 C9                 RET

1883 7C         CMPHD:  MOV     A,H
1884 BA                 CMP     D
1885 C0                 RNZ
1886 7D                 MOV     A,L
1887 BB                 CMP     E
1888 C9                 RET

1889 7A         CMPDH:  MOV     A,D
188A BC                 CMP     H
188B C0                 RNZ
188C 7B                 MOV     A,E
188D BD                 CMP     L
188E C9                 RET

188F 7D         SUBHD:  MOV     A,L
1890 93                 SUB     E
1891 6F                 MOV     L,A
1892 7C                 MOV     A,H
1893 9A                 SBB     D
1894 67                 MOV     H,A
1895 C9                 RET

1896 7E         SPNOR:  MOV     A,M     ;GET NEXT CHAR.
1897 23                 INX     H
1898 FE20               CPI     32      ;IS IT A SPACE ?
189A CA9618             JZ      SPNOR   ;YES:  IGNORE IT!
```

```
189D 2B             DCX     H
189E A7             ANA     A           ;TEST FOR END OF LINE
189F C9             RET                 ;RETURN TO CALLER

18A0 7E       LET:  MOV     A,M         ;GET CHAR.
18A1 FE5B           CPI     'Z'+1       ;IS IT A LETTER ?
18A3 D0             RNC                 ;NO!
18A4 FE41           CPI     'A'         ;CHECK OTHER END
18A6 3F             CMC                 ;SET C CORRECTLY
18A7 C9             RET                 ;RETURN: C SET IF LETTER

18A8 CDA018   LODQ: CALL    LET         ;IS IT A LETTER ?
18AB D8             RC                  ;YES: RETURN!

18AC 7E       DIQ:  MOV     A,M         ;GET CHAR.
18AD FE3A           CPI     '9'+1       ;IS IT A DIGIT ?
18AF D0             RNC                 ;NO!
18B0 FE30           CPI     '0'         ;CHECK OTHER END
18B2 3F             CMC                 ;SET C CORRECTLY
18B3 C9             RET                 ;RETURN: C SET IF DIGIT

18B4 CDA018   LTNOR: CALL   LET         ;IS NEXT CHAR. A LETTER ?
18B7 23             INX     H
18B8 DAB418         JC      LTNOR       ;YES: IGNORE IT!
18BB 2B             DCX     H
18BC A7             ANA     A           ;TEST FOR END OF LINE
18BD C9             RET                 ;RETURN

18BE CD9618   GCMA: CALL    SPNOR       ;GET FIRST NON-SPACE
18C1 D62C           SUI     ','         ;CHECK FOR COMMA
18C3 A7             ANA     A           ;CLEAR C-BIT
18C4 C0             RNZ                 ;NO COMMA: EXIT <NC>
18C5 23             INX     H           ;SKIP OVER COMMA
18C6 CD9618         CALL    SPNOR       ;GET NEXT NON-SPACE
18C9 37             STC                 ;INDICATE COMMA SEEN
18CA C9             RET                 ;COMMA: EXIT <C>

18CB 48       MSTR: MOV     C,B         ;COPY ORIG. COUNT
18CC CDA818   MST01: CALL   LODQ        ;LETTER OR DIGIT ?
18CF D2D918         JNC     MST02       ;NO!
18D2 12             STAX    D           ;STORE CHAR.
18D3 23             INX     H
18D4 13             INX     D
18D5 05             DCR     B           ;ENOUGH ?
18D6 C2CC18         JNZ     MST01       ;NO: LOOP!
18D9 79       MST02: MOV    A,C         ;GET ORIG. COUNT
18DA 90             SUB     B           ;COMPUTE BYTE COUNT
18DB F5             PUSH    PSW         ;SAVE IT
18DC 3E20           MVI     A,32        ;SET SPACE
18DE C3E318         JMP     MST04
18E1 12       MST03: STAX   D           ;STORE A SPACE
18E2 13             INX     D
18E3 05       MST04: DCR    B           ;ANY MORE TO DO ?
18E4 F2E118         JP      MST03       ;YES: LOOP!
18E7 F1             POP     PSW         ;RESTORE BYTE COUNT
18E8 C9             RET                 ;RETURN

18E9 3E20     PSPAC: MVI    A,32        ;SET SPACE
```

```
18EB C3C817             JMP     LO          ;PRINT A SPACE

18EE 3E3A       PCOLN:  MVI     A,':'       ;SET COLON
18F0 C3C817             JMP     LO          ;PRINT A COLON

18F3 CDE918     PSSTR:  CALL    PSPAC       ;PRINT A SPACE
18F6 7E         PSTR:   MOV     A,M         ;GET NEXT CHARACTER
18F7 23                 INX     H
18F8 CDC817             CALL    LO          ;PRINT IT
18FB 05                 DCR     B           ;ENOUGH ?
18FC C2F618             JNZ     PSTR        ;NO: LOOP!
18FF C9                 RET                 ;RETURN

1900 CDE918     PSBYT:  CALL    PSPAC       ;PRINT A SPACE
1903 7E         PBYT:   MOV     A,M         ;GET NEXT BYTE
1904 23                 INX     H
1905 C3D117             JMP     LBYT        ;PRINT THE BYTE

1908 CDE918     P2SNUM: CALL    PSPAC       ;PRINT A SPACE
190B CDE918     PSNUM:  CALL    PSPAC       ;PRINT A SPACE
190E 7E         PNUM:   MOV     A,M         ;GET LOBYTE
190F F5                 PUSH    PSW         ;SAVE LOBYTE
1910 23                 INX     H
1911 7E                 MOV     A,M         ;GET HIBYTE
1912 23                 INX     H
1913 CDD117             CALL    LBYT        ;PRINT HIBYTE
1916 F1                 POP     PSW         ;GET LOBYTE
1917 C3D117             JMP     LBYT        ;PRINT LOBYTE

191A 110000     GN1Z:   LXI     D,0         ;SET ZERO
191D CDBE18     GN1D:   CALL    GCMA        ;LOOK FOR COMMA
1920 3F                 CMC
1921 D0                 RNC                 ;NO NUMBER: EXIT <NC>
1922 CD2F19             CALL    GN2D        ;LOOK FOR NUMBER
1925 D0                 RNC                 ;NO NUMBER: EXIT <NC>
1926 F5                 PUSH    PSW         ;SAVE FLAGS
1927 CDBE18             CALL    GCMA        ;LOOK FOR TRAILING COMMA
192A F1                 POP     PSW         ;RESTORE CODES
192B C9                 RET                 ;GOT NUMBER: EXIT <C>

192C 110000     GN2Z:   LXI     D,0         ;SET ZERO
192F CD9618     GN2D:   CALL    SPNOR       ;GET 1ST NON-SPACE
1932 CDE917             CALL    NIBL        ;CHECK IF HEX
1935 3F                 CMC
1936 D0                 RNC                 ;NO NUMBER: EXIT <NC>
1937 110000             LXI     D,0         ;INITIALIZE NUMBER TO ZERO
193A 23         G01:    INX     H           ;SKIP CHARACTER
193B EB                 XCHG
193C 29                 DAD     H           ;MULTIPLY ...
193D 29                 DAD     H           ;... BY ...
193E 29                 DAD     H           ;... SIXTEEN ...
193F 29                 DAD     H
1940 B5                 ORA     L           ;MERGE NEW NIBBLE
1941 6F                 MOV     L,A
1942 EB                 XCHG
1943 7E                 MOV     A,M         ;GET NEXT CHARACTER
1944 CDE917             CALL    NIBL        ;CHECK IF HEX
1947 D23A19             JNC     G01         ;YES: LOOP!
```

```
194A 7A             MOV     A,D     ;TEST ...
194B B3             ORA     E       ;... NUMBER
194C 37             STC             ;INDICATE NUMBER SEEN
194D C9             RET             ;GOT NUMBER: EXIT <C>

                ;;  GENERAL MATH ROUTINES :

194E 85         ADHLA:  ADD     L
194F 6F                 MOV     L,A
1950 D0                 RNC
1951 24                 INR     H
1952 C9                 RET

1953 7B         ANHD:   MOV     A,E
1954 A5                 ANA     L
1955 6F                 MOV     L,A
1956 7A                 MOV     A,D
1957 A4                 ANA     H
1958 67                 MOV     H,A
1959 C9                 RET

195A 2B         NEGH:   DCX     H
195B 7D         NOTH:   MOV     A,L
195C 2F                 CMA
195D 6F                 MOV     L,A
195E 7C                 MOV     A,H
195F 2F                 CMA
1960 67                 MOV     H,A
1961 C9                 RET

1962 7B         ORHD:   MOV     A,E
1963 B5                 ORA     L
1964 6F                 MOV     L,A
1965 7A                 MOV     A,D
1966 B4                 ORA     H
1967 67                 MOV     H,A
1968 C9                 RET

1969 7B         XORHD:  MOV     A,E
196A AD                 XRA     L
196B 6F                 MOV     L,A
196C 7A                 MOV     A,D
196D AC                 XRA     H
196E 67                 MOV     H,A
196F C9                 RET

1970 CD8A19     SHLHD:  CALL    SCMN
1973 C8                 RZ
1974 29         SL1:    DAD     H
1975 1D                 DCR     E
1976 C27419             JNZ     SL1
1979 C9                 RET

197A CD8A19     SHRHD:  CALL    SCMN
197D C8                 RZ
197E AF         SR1:    XRA     A
197F 7C                 MOV     A,H
```

```
1980 1F                 RAR
1981 67                 MOV     H,A
1982 7D                 MOV     A,L
1983 1F                 RAR
1984 6F                 MOV     L,A
1985 1D                 DCR     E
1986 C27E19             JNZ     SR1
1989 C9                 RET

198A EB         SCMN:   XCHG
198B 3EF0               MVI     A,0F0H
198D A3                 ANA     E
198E B2                 ORA     D
198F C29419             JNZ     ZH
1992 AB                 XRA     E
1993 C9                 RET
1994 AF         ZH:     XRA     A
1995 67                 MOV     H,A
1996 6F                 MOV     L,A
1997 C9                 RET

1998 C5         MULHD:  PUSH    B
1999 44                 MOV     B,H
199A 4D                 MOV     C,L
199B 210000             LXI     H,0
199E 3E10               MVI     A,16
19A0 F5         M1:     PUSH    PSW
19A1 29                 DAD     H
19A2 7B                 MOV     A,E
19A3 17                 RAL
19A4 5F                 MOV     E,A
19A5 7A                 MOV     A,D
19A6 17                 RAL
19A7 57                 MOV     D,A
19A8 D2B019             JNC     M2
19AB 09                 DAD     B
19AC D2B019             JNC     M2
19AF 13                 INX     D
19B0 F1         M2:     POP     PSW
19B1 3D                 DCR     A
19B2 C2A019             JNZ     M1
19B5 C1                 POP     B
19B6 C9                 RET

19B7 C5         DIVHD:  PUSH    B
19B8 EB                 XCHG
19B9 42                 MOV     B,D
19BA 4B                 MOV     C,E
19BB 110000             LXI     D,0
19BE 3E10               MVI     A,16
19C0 F5         D1:     PUSH    PSW
19C1 29                 DAD     H
19C2 7B                 MOV     A,E
19C3 17                 RAL
19C4 5F                 MOV     E,A
19C5 7A                 MOV     A,D
19C6 17                 RAL
19C7 57                 MOV     D,A
```

```
19C8 7B                 MOV     A,E
19C9 91                 SUB     C
19CA 5F                 MOV     E,A
19CB 7A                 MOV     A,D
19CC 98                 SBB     B
19CD 57                 MOV     D,A
19CE D2D519             JNC     D2
19D1 EB                 XCHG
19D2 09                 DAD     B
19D3 EB                 XCHG
19D4 23                 INX     H
19D5 F1         D2:     POP     PSW
19D6 3D                 DCR     A
19D7 C2C019             JNZ     D1
19DA C1                 POP     B
19DB C35B19             JMP     NOTH

                ;;      BLOCK STRUCTURED DEVICE HANDLER UTILITY
                ;         SUBROUTINES - VERSION 1.0
                ;
                ;       COPYRIGHT (C) 1977,1980
                ;         BY INTELLIGENT SYSTEMS CORPORATION

                ;;      ************************************************
19DE            BSDHU:  ; IF THIS ADDRESS IS NOT '19DE', THEN THERE
                        ; MAY BE ADDRESS RESOLUTION PROBLEMS WITH RAM
                        ; BASED SOFTWARE PACKAGES.
                ;       ************************************************


                ;+
                ; BC2BK - CONVERT TOTAL BYTE COUNT TO NUMBER OF
                ;         BLOCKS AND LAST BLOCK BYTE COUNT.
                ; INPUTS  - H&L:  TOTAL BYTE COUNT
                ; OUTPUTS - H&L:  NUMBER OF BLOCKS
                ;           C  :  LAST BLOCK BYTE COUNT
                ;-

19DE 7C         BC2BK:  MOV     A,H
19DF B5                 ORA     L
19E0 CAE819             JZ      BC01
19E3 2B                 DCX     H
19E4 3E7F               MVI     A,7FH
19E6 A5                 ANA     L
19E7 3C                 INR     A
19E8 4F         BC01:   MOV     C,A
19E9 AF                 XRA     A
19EA 29                 DAD     H
19EB 17                 RAL
19EC 6C                 MOV     L,H
19ED 67                 MOV     H,A
19EE 23                 INX     H
19EF C9                 RET


                ;+
                ; BK2BC - CONVERT NUMBER OF BLOCKS AND LAST BLOCK
```

```
                ;          BYTE COUNT TO TOTAL BYTE COUNT.
                ; INPUTS  - H&L:  NUMBER OF BLOCKS
                ;           C   :  LAST BLOCK BYTE COUNT
                ; OUTPUTS - H&L:  TOTAL BYTE COUNT
                ;-
19F0 2B         BK2BC:  DCX     H
19F1 7C                 MOV     A,H
19F2 1F                 RAR
19F3 7D                 MOV     A,L
19F4 1F                 RAR
19F5 47                 MOV     B,A
19F6 3E00               MVI     A,0
19F8 67                 MOV     H,A
19F9 1F                 RAR
19FA 6F                 MOV     L,A
19FB 09                 DAD     B
19FC C9                 RET

19FD C1         BSB01:  POP     B       ;GET CALLER'S RETURN ADDR.
19FE E5                 PUSH    H       ;SAVE PBLK PNTR
19FF D5                 PUSH    D       ;SAVE DEVICE NAME
1A00 11271A             LXI     D,BS02
1A03 D5                 PUSH    D       ;SET RETURN ADDRESS
1A04 C5                 PUSH    B       ;SAVE CALLER'S RETURN ADDR.
1A05 4F                 MOV     C,A     ;COPY NUMBER OF UNITS
1A06 11E580             LXI     D,TFCN  ;POINT TO TEMP STORAGE AREA
1A09 0608               MVI     B,8     ;SET COUNT
1A0B CD7118             CALL    MOVDH   ;MOVE PARAMETERS
1A0E E1                 POP     H       ;GET CALLER'S RETURN ADDR.
1A0F 3AE680             LDA     TDRV    ;GET DRIVE NUMBER
1A12 0606               MVI     B,EIVU  ;SET ERROR CODE
1A14 B9                 CMP     C       ;VALID UNIT NUMBER ?
1A15 D0                 RNC             ;NO: ERROR EXIT!
1A16 3AE580             LDA     TFCN    ;GET FUNCTION CODE
1A19 FE02               CPI     2       ;IS IT VALID ?
1A1B DA231A             JC      BS01    ;YES!
1A1E 0603               MVI     B,EIVF  ;SET ERROR CODE
1A20 FEFC               CPI     -4      ;IS IT VALID ?
1A22 D8                 RC              ;NO: ERROR EXIT!
1A23 0600       BS01:   MVI     B,SOK   ;SET GOOD STATUS CODE
1A25 A7                 ANA     A       ;TEST FUNCTION CODE
1A26 E9                 PCHL            ;RETURN TO CALLER!

1A27 78         BS02:   MOV     A,B     ;GET STATUS CODE
1A28 A7                 ANA     A       ;ANY ERRORS ?
1A29 C2321A             JNZ     HANER   ;YES!
1A2C D1                 POP     D       ;DISCARD DEVICE NAME
1A2D D1                 POP     D       ;DISCARD PBLK PNTR
1A2E 2AE780             LHLD    TBLK    ;GET BLOCK NUMBER
1A31 C9                 RET             ;EXIT!

                        HANDLER ERROR OUTPUT ROUTINE
                ;;      ***[redacted]***
1A32 21671A     HANER:  LXI     H,ERS1
1A35 CD2A18             CALL    OSTR    ;PRINT ERROR STRING
```

```
1A38 216A1A            LXI     H,ERS2-3
1A3B 48                MOV     C,B       ;COPY ERROR CODE
1A3C 0600              MVI     B,0
1A3E 09                DAD     B         ;POINT TO ERROR STRING
1A3F 0603              MVI     B,3       ;SET COUNT
1A41 CDF618            CALL    PSTR      ;PRINT 3 ERROR CHARACTERS
1A44 210000            LXI     H,0
1A47 39                DAD     SP        ;POINT TO DEV. NAME ON STACK
1A48 0602              MVI     B,2       ;SET COUNT
1A4A CDF318            CALL    PSSTR     ;PRINT DEVICE NAME
1A4D E1                POP     H         ;DISCARD DEVICE NAME
1A4E E1                POP     H         ;GET PBLK PNTR
1A4F 46                MOV     B,M       ;GET FUNCTION CODE
1A50 23                INX     H
1A51 7E                MOV     A,M       ;GET UNIT NUMBER
1A52 CDD117            CALL    LBYT      ;PRINT UNIT NUMBER
1A55 CDEE18            CALL    PCOLN     ;PRINT A COLON
1A58 78                MOV     A,B       ;GET FUNCTION CODE
1A59 CDD117            CALL    LBYT      ;PRINT FUNCTION CODE
1A5C 21E780            LXI     H,TBLK    ;POINT TO BLOCK NUMBER
1A5F CD0B19            CALL    PSNUM     ;PRINT BLOCK NUMBER
1A62 CDC117            CALL    CRLF      ;PRINT CR & LF
1A65 37                STC               ;INDICATE ERROR
1A66 C9                RET               ;EXIT!

                ;;     HANDLER ERROR CODES

1A67 FF06010B ERS1:    DB      255,6,1,11,'E',239
1A6B 45EF
     (0000)   SOK      EQU     0         ; "GOOD" STATUS CODE
1A6D 495646   ERS2:    DB      'IVF'
     (0003)   EIVF     EQU     $-ERS2    ; INVALID FUNCTION
1A70 495655            DB      'IVU'
     (0006)   EIVU     EQU     $-ERS2    ; INVALID UNIT NUMBER
1A73 424C4B            DB      'BLK'
     (0009)   EBLK     EQU     $-ERS2    ; BLOCK NUMBER TOO LARGE
1A76 534B46            DB      'SKF'
     (000C)   ESKF     EQU     $-ERS2    ; SEEK FAILURE
1A79 434642            DB      'CFB'
     (000F)   ECFB     EQU     $-ERS2    ; CANNOT FIND BLOCK
1A7C 445359            DB      'DSY'
     (0012)   EDSY     EQU     $-ERS2    ; DATA SYNC
1A7F 444353            DB      'DCS'
     (0015)   EDCS     EQU     $-ERS2    ; DATA CRC
1A82 4D454D            DB      'MEM'
     (0018)   EMEM     EQU     $-ERS2    ; MEMORY
1A85 564659            DB      'VFY'
     (001B)   EVFY     EQU     $-ERS2    ; VERIFY

1A88 3AE580   BSB08:   LDA     TFCN      ;GET FUNCTION CODE
1A8B 3D                DCR     A         ;TEST IT
1A8C A7                ANA     A         ;CLEAR C STATUS BIT
1A8D F8                RM                ;READ: RETURN!
1A8E CA951A            JZ      BS04      ;WRITE: PROCESS!
1A91 32E580            STA     TFCN      ;WAS VERIFY: SET WRITE
1A94 C9                RET               ;RETURN
1A95 2AEB80   BS04:    LHLD    TBC       ;GET CURRENT BYTE COUNT
```

```
1A98 EB              XCHG
1A99 2AE380          LHLD    OBC       ;GET OLD BYTE COUNT
1A9C 22EB80          SHLD    TBC       ;RESET BYTE COUNT TO OLD VALUE
1A9F CD8F18          CALL    SUBHD     ;COMPUTE DIFFERENCE
1AA2 EB              XCHG
1AA3 2AE980          LHLD    TMEM      ;GET CURRENT MEMORY PNTR
1AA6 CD8F18          CALL    SUBHD     ;ADJUST IT
1AA9 22E980          SHLD    TMEM      ;RESET MEM PNTR TO OLD VALUE
1AAC EB              XCHG
1AAD CDDE19          CALL    BC2BK     ;CONVERT TO NUMBER OF BLOCKS
1AB0 EB              XCHG
1AB1 2AE780          LHLD    TBLK      ;GET CURRENT BLOCK NUMBER
1AB4 CD8F18          CALL    SUBHD     ;ADJUST IT
1AB7 22E780          SHLD    TBLK      ;RESET BLK NUMBER TO OLD VALUE
1ABA 3E02            MVI     A,2       ;SET VERIFY FUNCTION CODE
1ABC 32E580          STA     TFCN      ;STORE IT
1ABF 37              STC               ;INDICATE "RESET"
1AC0 C9              RET               ;RETURN

                ;  ****************************************************************
                ;
                ;;      MICRO DISK HANDLER (4 PHASE) V1.20
                ;
                ;        FOR INTECOLOR 3621/COMPUCOLOR II
                ;
                ;       COPYRIGHT (C) 1978,1979,1980
                ;                BY INTELLIGENT SYSTEMS CORPORATION
                ;
                ;       Last Revised:   January 15, 1980   JKP III
                ;
                ;
                ;  INPUTS:      HL => PBLK (PARAMETER BLOCK)
                ;               DE =  HAMDLER ADDRESS (VECTOR TABLE 1)
                ;
                ;  OUTPUTS:     HL =  SIZE (RESPONSE TO 'GETSIZE')
                ;               HL =  LAST BLOCK TRANSFERRED + 1 (R/W) RR
                ;
                ;  STATUS:      <NC> - NO ERROR
                ;               <C>  - ERROR ('HANER' GENERATES ERROR MESSAGE)
                ;
                ;
                ;;      *******  FUNCTION CODES  *******
                ;
                ;         0 = READ       -1 = GET SIZE
                ;         1 = WRITE      -2 = ADD USER
                ;       * 2 = VERIFY     -3 = SUB USER
                ;                        -4 = RESET
                ;
                ;---------------------------------------------------

                ;;      MICRO DISK CONSTANTS :

(000A)  NSEC    EQU     10        ; NUMBER OF SECTORS / TRACK
(0029)  NTRK    EQU     41        ; NUMBER OF TRACKS
(0014)  STPTIM  EQU     20        ; STEP TIME / 0.5 MS.
(0640)  UPTIM   EQU     1600      ; MOTOR UP-TO-SPEED TIME / 0.5 MS.
(00FF)  FILL    EQU     0FFH      ; "FILL" BYTE
```

```
                (0055)  IDM     EQU     55H             ; ID MARK
                (005A)  DATAM   EQU     5AH             ; DATA MARK

                (0003)  RGAPS   EQU     3               ; BYTE TIMES / GAP CONFIRMATION

                (0003)  CRTR    EQU     3               ; 'CHUNK' RETRY COUNT
                (003C)  HRTR    EQU     6*NSEC          ; HEADER RETRY COUNT (WAS 3* 1/15/80)
                (000A)  RRTR    EQU     10              ; READ RETRY COUNT
                (001E)  SRTR    EQU     3*NSEC          ; SEARCH TRY COUNT
                (0002)  VRTR    EQU     2               ; VERIFY RETRY COUNT
                (0004)  WRTR    EQU     4               ; WRITE/VERIFY RETRY COUNT

                        ;;      MICRO DISK HANDLER (4 PHASE) :

1AC1 EB         CDHD:   XCHG
1AC2 2A4600             LHLD    CDNM            ; GET NAME
1AC5 EB                 XCHG
1AC6 3A4800             LDA     CDNU            ; GET NUMBER OF UNITS
1AC9 CDFD19             CALL    BSBO1           ; COPY PARAMETERS & SETUP RETURN
1ACC F5                 PUSH    PSW             ; SAVE FLAGS
1ACD 210000             LXI     H,0
1AD0 54                 MOV     D,H
1AD1 3A4900             LDA     CDSEC           ; GET NUMBER OF SECTORS
1AD4 5F                 MOV     E,A             ; E=NUMBER OF SECTORS / TRACK
1AD5 3E28               MVI     A,NTRK-1        ;  NO. OF TRACKS USED FOR DATA
1AD7 19         DX10:   DAD     D               ; ADD SECTORS AGAIN
1AD8 3D                 DCR     A               ; ENOUGH ?
1AD9 C2D71A             JNZ     DX10            ; NO: LOOP!
1ADC F1                 POP     PSW             ; RESTORE FLAGS
1ADD F25D1B             JP      XFER            ; IT IS READ OF WRITE!
1AE0 22E780             SHLD    TBLK            ; SAVE TOTAL NUMBER OF BLOCKS
1AE3 3C                 INR     A               ; "GET SIZE" FUNCTION ?
1AE4 C8                 RZ                      ; YES: EXIT!

1AE5 3C                 INR     A               ; "ADD USER" FUNCTION ?
1AE6 C2EE1A             JNZ     DX11            ; NO!
1AE9 CD141F     ADDU:   CALL    GCUCNT          ; HL => CUCNT0 OR CUCNT1
1AEC 34                 INR     M               ; INCREMENT USER COUNT
1AED C9                 RET                     ; EXIT

1AEE 3C         DX11:   INR     A               ; "SUBTRACT USER" FUNCTION ?
1AEF C2F91A             JNZ     CDRSET          ; NO: MUST BE "TURN OFF" FUNCTION!
1AF2 CD141F     SUBU:   CALL    GCUCNT          ; HL => CUCNT0 OR CUCNT1
1AF5 35                 DCR     M               ; SUBTRACT A USER
1AF6 F2171B             JP      DX12            ; POSITIVE: O.K. !
                                                ; IF EITHER USER COUNT IS NEGATIVE,
                                                ; THEN FULL RESET

                        ;;      ** RESET DISK DRIVES **

1AF9            CDRSET:                         ;*  1/15/80  JKP III
1AF9 CD511B             CALL    CALFLG          ;* GET CALCULATED FLAG VALUE
1AFC BE                 CMP     M               ;* COMPARE TO FLAG
1AFD CA061B             JZ      CDRST1          ;* <Z> = GOOD FLAG: LOGICAL RESET ONLY
1B00 3EEE       FDRST:  MVI     A,0EEH          ;* BAD TRACK NUMBER
1B02 2B                 DCX     H               ;* 'CTRK1'
1B03 77                 MOV     M,A             ;* SET FOR POUND
```

```
1B04 2B                 DCX     H       ;* 'CTRK0'
1B05 77                 MOV     M,A     ;* SET FOR POUND
                                        ;*  1/15/80  JKP III
1B06 AF         CDRST1: XRA     A
1B07 32E680             STA     TDRV    ; SET DRIVE ZERO
1B0A CD121B             CALL    TOFF    ; PARK DRIVE ZERO
1B0D 3E01               MVI     A,1
1B0F 32E680             STA     TDRV    ; SET DRIVE ONE
                                        ;* PARK DRIVE ONE AND EXIT

1B12 CD141F     TOFF:   CALL    GCUCNT  ; HL => CUCNT0 OR CUCNT1
1B15 3600               MVI     M,0     ; SET ZERO USERS
1B17 C5         DX12:   PUSH    B
1B18 7E                 MOV     A,M     ; GET USER COUNT
1B19 A7                 ANA     A       ; TEST IT
1B1A C22B1B             JNZ     DX13    ; NOT ZERO !
1B1D CD8A1B             CALL    DX14A   ; WAIT FOR BUFFERS TO EMPTY
1B20 CD941E             CALL    VTP     ; ARE TRACK & PHASE VALID ?

1B23            DX12A:                  ;*  1/15/80  JKP III
1B23 CD0E1F             CALL    GCTRK   ;* POINT TO 'CURRENT TRACK'
1B26 AF                 XRA     A       ;*
1B27 BE                 CMP     M       ;* TEST FOR ZERO
1B28 C4C01E             CNZ     STEPS   ;* IF NOT ZERO, MOVE HEAD TO HOME
                                        ;*  1/15/80  JKP III

1B2B AF         DX13:   XRA     A       ; NO REASON
1B2C D307               OUT     EXTOT   ; DESELECT DRIVE
1B2E 3E10               MVI     A,16    ; WAIT ONE CHARACTER TIME ...
1B30 3D         DX13A:  DCR     A
1B31 C2301B             JNZ     DX13A
1B34 F3                 DI
1B35 3E10               MVI     A,10H
1B37 D308               OUT     MASK    ; MASK SERIAL IN ONLY
1B39 DB02               IN      RSTBF   ; "CLEAR" RECVR INTERRUPT
1B3B 3A1700             LDA     RCOMD   ; PICK UP HI/LO RATE COMND
1B3E D304               OUT     COMND   ; ISSUE
1B40 3AE281             LDA     CRATE   ; PICK UP CURRENT RATE
1B43 D305               OUT     BAUD    ; ISSUE
1B45 3AE081             LDA     CMASK
1B48 D308               OUT     MASK    ; RESTORE 5501 MASK
1B4A CD511B             CALL    CALFLG  ;* CALCULATE 'RESET' FLAG
1B4D 77                 MOV     M,A     ;* STORE IT IN 'DDRFLG'  1/15/80 JKP
1B4E C1                 POP     B
1B4F FB                 EI              ; 5501 RESTORED - ENABLE INTERRUPTS
1B50 C9                 RET             ; EXIT !!


                ;;      *** CALCULATE 'RESET' FLAG ***
                ;               1/15/80  JKP III
                ;
                ;       OUTPUT: A  =  CALCULATED 'RESET' FLAG
                ;               HL => 'DDRFLG'

1B51 21B181     CALFLG: LXI     H,CTRK0 ; ADDRESS OF CURRENT TRACK FOR DRIVE 0
1B54 7E                 MOV     A,M     ; GET 'CTRK0'
1B55 07                 RLC             ; TIMES 2
1B56 23                 INX     H       ; POINT TO 'CTRK1'
```

```
1B57 86                 ADD     M       ; ADD 'CTRK1' TO 2*'CTRK0'
1B58 07                 RLC             ; TIMES 2
1B59 C607               ADI     7       ; CONSTANT FOR NON-ZERO RESULT
1B5B 23                 INX     H       ; POINT TO 'DDRFLG'
1B5C C9                 RET

1B5D AF         XFER:   XRA     A       ;*
1B5E 32B381             STA     DDRFLG  ;* CLEAR RESET FLAG (NOT RESET)  1/15/80 JKP
1B61 7B                 MOV     A,E     ; NO. OF SECTORS
1B62 2F                 CMA
1B63 3C                 INR     A
1B64 4F                 MOV     C,A     ; C= - NO. OF SECTORS
1B65 EB                 XCHG            ; D&E=TOTAL NUMBER OF BLOCKS
1B66 2AE780             LHLD    TBLK    ; GET BLOCK NUMBER
1B69 CD8318             CALL    CMPHD   ; IS IT VALID ?
1B6C 0609               MVI     B,EBLK  ; SET ERROR CODE
1B6E D0                 RNC             ; ERROR: INVALID BLOCK!
1B6F 06FF               MVI     B,-1    ; SET HI BYTE OF B&C
1B71 AF                 XRA     A       ; CLEAR A
1B72 3C         DX14:   INR     A       ; COUNT TRACK
1B73 09                 DAD     B       ; SUBTRACT SECTORS  TRACK
1B74 DA721B             JC      DX14    ; LOOP!
1B77 67                 MOV     H,A     ; MOVE TRACK NUMBER TO H
1B78 7D                 MOV     A,L     ; GET LO BYTE
1B79 91                 SUB     C       ; ADJUST
1B7A 6F                 MOV     L,A     ; SET SECTOR NUMBER
1B7B 22EE80             SHLD    SEC     ; SAVE TRACK & SECTOR
1B7E CDE91A             CALL    ADDU    ; "ADD A USER"
1B81 CD8A1B             CALL    DX14A   ; WAIT TIL CHAR FINISHED
1B84 CDC81B             CALL    DXX     ; PERFORM READ OR WRITE
1B87 C3F21A             JMP     SUBU    ; SUBTRACT USER AND EXIT

1B8A 3A2E00     DX14A:  LDA     CDMK
1B8D D308               OUT     MASK    ; SET 5501 MASK FOR EXTERNAL SENSOR ONLY
1B8F 3AE281             LDA     CRATE   ; GET 5501 CURRENT RATE
1B92 07                 RLC             ; GET RID OF STOP BIT BIT
1B93 210100             LXI     H,1     ; INITIALIZE DELAY NEEDED
1B96 0604               MVI     B,4     ; INITIALIZE COUNTER
1B98 07         DX15:   RLC             ; GET NEXT BIT
1B99 DAAD1B             JC      DX16    ; FOUND THE RATE !
1B9C 29                 DAD     H       ; DOUBLE THE DELAY NEEDED
1B9D 05                 DCR     B       ; AGAIN ?
1B9E C2981B             JNZ     DX15    ; YES !
1BA1 29                 DAD     H       ; DOUBLE THE DELAY NEEDED
1BA2 07                 RLC             ; GET NEXT BIT
1BA3 DAAD1B             JC      DX16    ; FOUND THE RATE !
1BA6 29                 DAD     H       ; DOUBLE THE DELAY NEEDED
1BA7 07                 RLC             ; GET NEXT BIT
1BA8 DAAD1B             JC      DX16    ; FOUND THE RATE !
1BAB 2E60               MVI     L,96    ; IT MUST BE 110 BAUD: SET DELAY NEEDED

1BAD            DX16:                   ; WAIT FOR BUFFERS TO EMPTY :

1BAD 3A1700     DX17:   LDA     RCOMD   ; GET RATE COMMAND
1BB0 E610               ANI     10H     ; MASK OFF HI-RATE BIT
1BB2 CAB71B             JZ      DX18    ; LO-RATE: USE ZERO COUNTER !
1BB5 3E20               MVI     A,32    ; HI-RATE: USE PROPER COUNTER
```

```
1BB7 3D         DX18:   DCR     A       ; ENOUGH ?
1BB8 C2B71B             JNZ     DX18    ; NO: LOOP !
1BBB 2D                 DCR     L       ; ENOUGH ?
1BBC C2AD1B             JNZ     DX17    ; NO: LOOP !

1BBF 3E18               MVI     A,18H
1BC1 D304               OUT     COMND   ; SET HIGH SPEED
1BC3 3EC0               MVI     A,0C0H
1BC5 D305               OUT     BAUD    ; SET RATE = 76.8 KBAUD
1BC7 C9                 RET

1BC8 CD941E     DXX:    CALL    VTP     ; POUND HOME IF NEEDED
1BCB CD851E             CALL    SX0     ;* SELECT DRIVE  1/15/80  JKP

1BCE 3AE580             LDA     TFCN    ; LOAD FUNCTION TYPE
1BD1 A7                 ANA     A       ; TEST FOR READ = 0
1BD2 CAF31B             JZ      SEEK    ;   YES - DO NOT WAIT  <<9/78>>

1BD5 CD0E1F             CALL    GCTRK   ; HL => CTRK0 OR CTRK1
1BD8 66                 MOV     H,M     ; GET CURRENT TRACK NUMBER
1BD9 3A4A00             LDA     CDK     ; GET UPTIM/STPTIM
1BDC 6F                 MOV     L,A     ; SAVE IT
1BDD 3AEF80             LDA     TRK     ; GET DESIRED TRACK NUMBER
1BE0 94                 SUB     H       ; COMPUTE POSITIVE DIFFERENCE ...
1BE1 F2E61B             JP      CD03
1BE4 2F                 CMA
1BE5 3C                 INR     A
1BE6 95         CD03:   SUB     L       ; COMPUTE -(ADDITIONAL STEP DELAYS NEEDED)
1BE7 F2F31B             JP      SEEK    ; NO ADDITIONAL DELAY NEEDED !
1BEA F5         CD04:   PUSH    PSW     ; SAVE COUNT
1BEB CD021F             CALL    STPWAT  ; DELAY ONE STEP TIME
1BEE F1                 POP     PSW     ; RESTORE COUNT
1BEF 3C                 INR     A       ; ENOUGH ?
1BF0 C2EA1B             JNZ     CD04    ; NO: WAIT AGAIN !


1BF3 3E03       SEEK:   MVI     A,3     ;
1BF5 32E180             STA     RFLG    ;
1BF8            STEP1:
1BF8 3AEF80             LDA     TRK
1BFB CDC01E             CALL    STEPS   ; MOVE HEAD TO DESIRED TRACK

1BFE 3AEE80             LDA     SEC     ; GET CURRENT SECTOR NUMBER
1C01 32ED80             STA     OSEC    ; SAVE IT FOR VERIFY PASS
1C04 2AEB80             LHLD    TBC     ; GET CURRENT BYTE COUNT
1C07 22E380             SHLD    OBC     ; SAVE IT FOR VERIFY PASS

1C0A FB         ISEC:   EI              ; *** ENABLE INTERRUPTS ***
                ;       LXI     H,RFLG  ; HL => RESTORE FLAG / RETRY COUNTER
                ;       MOV     A,M     ; GET FLAG
                ;       ANA     A       ; HAVE WE DONE A RESTORE ?
                ;       JNZ     IS1     ; NO: LEAVE FLAG ALONE !
                ;       MVI     M,5     ; INITIALIZE WIGGLE COUNTER (3)
1C0B            IS1:

1C0B 2AEB80             LHLD    TBC     ; GET REMAINING BYTE COUNT
1C0E 3E80               MVI     A,80H   ; DETERMINE SECTOR BYTE COUNT ...
1C10 A5                 ANA     L
```

```
1C11 B4              ORA     H
1C12 3E80            MVI     A,80H
1C14 C2181C          JNZ     IS2         ; FULL SECTOR (128.) REQUIRED !
1C17 7D              MOV     A,L         ; PARTIAL SECTOR BYTE COUNT
1C18 324280    IS2:  STA     SBC         ; SAVE SECTOR BYTE COUNT
1C1B 3AE580          LDA     TFCN        ; GET FUNCTION CODE
1C1E 3D              DCR     A           ; TEST IT
1C1F 3E0A            MVI     A,RRTR      ; SET RETRY COUNT
1C21 FA321C          JM      IS4         ; FUNCTION IS READ !
1C24 CD471E          CALL    CRCX        ; COMPUTE CRC FOR SECTOR
1C27 EB              XCHG
1C28 224380          SHLD    CRC1        ; SAVE CRC BYTES
1C2B 3E04            MVI     A,WRTR      ; SET WRITE RETRY COUNT
1C2D 32E280          STA     CRTRY
1C30 3E02      IS3:  MVI     A,VRTR      ; SET VERIFY RETRY COUNT ...
1C32 32E080    IS4:  STA     BRTRY       ; INITIALIZE RETRY COUNT

1C35 0E1E      DSEC: MVI     C,SRTR      ; INITIALIZE SEARCH TRY COUNT
1C37 063C      FQ0:  MVI     B,HRTR      ; INITIALIZE HEADER RETRY COUNT
1C39 DB00      FQ1:  IN      RXBUF       ; CLEAR RECEIVER BUFFER
1C3B 2E04            MVI     L,RGAPS+1   ; SET NO. BYTE TIMES / GAP
1C3D DB03      FQ2:  IN      STAT5       ; GET 5501 STATUS
1C3F E608            ANI     8           ; HAS A BYTE BEEN RECEIVED ?
1C41 C2391C          JNZ     FQ1         ; YES: RESET & TRY AGAIN !
1C44 2D              DCR     L           ; HAS GAP BEEN CONFIRMED ?
1C45 CA511C          JZ      FQ4         ; YES !
1C48 3E10            MVI     A,16        ; WAIT ONE BYTE TIME ....
1C4A 3D        FQ3:  DCR     A
1C4B C24A1C          JNZ     FQ3
1C4E C33D1C          JMP     FQ2         ; GO CHECK AGAIN !

1C51           FQ4:                      ; GAP HAS BEEN FOUND

1C51 F3              DI                  ; *** DISABLE INTERRUPTS ***
1C52 11FFFF          LXI     D,0FFFFH    ; INITIALIZE CRC RESIDUE
1C55 CD1C1E          CALL    RBYTEC      ; READ ID MARK
1C58 FE55            CPI     IDM         ; IS IT CORRECT ?
1C5A C2701C          JNZ     HER1        ; NO !
1C5D CD1C1E          CALL    RBYTEC      ; READ TRACK NUMBER
1C60 67              MOV     H,A         ; SAVE IT
1C61 CD1C1E          CALL    RBYTEC      ; READ SECTOR NUMBER
1C64 6F              MOV     L,A         ; SAVE IT
1C65 CD1C1E          CALL    RBYTEC      ; READ 1ST CRC BYTE
1C68 CD1C1E          CALL    RBYTEC      ; READ 2ND CRC BYTE
1C6B 7A              MOV     A,D
1C6C B3              ORA     E           ; IS CRC CORRECT ?
1C6D CA831C          JZ      GH1         ; YES: GOT A GOOD HEADER !
1C70 FB        HER1: EI                  ; *** ENABLE INTERRUPTS ***
1C71 05              DCR     B           ; HEADER RETRIES EXHAUSTED ?
1C72 C2391C          JNZ     FQ1         ; NO: TRY READING HEADER AGAIN !

1C75 FB        SEEKF: EI                 ; *** ENABLE INTERRUPTS ***
1C76 060C            MVI     B,ESKF      ; SET ERROR CODE
1C78           WIQ1:
1C78 21E180          LXI     H,RFLG      ; GET "RESTORE" FLAG
1C7B 35              DCR     M
1C7C C8              RZ
1C7D CDB11E          CALL    POUND       ; POUND TO HOME
```

```
1C80 C3F81B             JMP     STEP1   ; NOW SEEK DESIRED TRACK !

1C83 3AEF80   GH1:      LDA     TRK     ; GET DESIRED TRACK
1C86 BC                 CMP     H       ; CORRECT TRACK ?
1C87 C2751C             JNZ     SEEKF   ; NO !
1C8A 3AEE80   GH2:      LDA     SEC     ; GET DESIRED SECTOR
1C8D BD                 CMP     L       ; CORRECT SECTOR ?
1C8E CA9B1C             JZ      GH4     ; YES: DO IT !
1C91 FB       GH3:      EI              ; *** ENABLE INTERRUPTS ***
1C92 0D                 DCR     C       ; SEARCH TRIES EXHAUSTED ?
1C93 C2371C             JNZ     FQO     ; NO: TRY AGAIN !
1C96 060F               MVI     B,ECFB  ; SET POSSIBLE ERROR CODE
1C98 C3781C             JMP     WIG1    ; GO WIGGLE OR GIVE UP !

1C9B 3A4280   GH4:      LDA     SBC     ; GET SECTOR BYTE COUNT
1C9E 47                 MOV     B,A
1C9F 2AE980             LHLD    TMEM    ; GET BUFFER POINTER
1CA2 3AE580             LDA     TFCN    ; GET FUNCTION CODE
1CA5 3D                 DCR     A       ; TEST FUNCTION CODE
1CA6 FAB91D             JM      RDOO    ; COMMAND IS READ !
1CA9 3D                 DCR     A
1CAA C24E1D             JNZ     WROO    ; COMMAND IS WRITE !

1CAD CD701E   VEOO:     CALL    GDATAM  ; CONFIRM DATA MARK
1CB0 CAD21C             JZ      VEO2    ; GOT IT !
1CB3 FB       VERR:     EI              ; *** ENABLE INTERRUPTS ***
1CB4 21E080             LXI     H,BRTRY ; POINT TO VERIFY RETRY COUNTER
1CB7 35                 DCR     M       ; VERIFY RETRIES EXHAUSTED ?
1CB8 C2351C             JNZ     DSEC    ; NO: TRY AGAIN !
1CBB 3E03               MVI     A,3
1CBD 32E580             STA     TFCN    ; SET REWRITE FUNCTION CODE
1CC0 21E280             LXI     H,CRTRY ; POINT TO WRITE RETRY COUNTER
1CC3 35                 DCR     M       ; WRITE RETRIES EXHAUSTED ?
1CC4 C2301C             JNZ     IG3     ; NO: GO WRITE SECTOR AGAIN !
1CC7 061B               MVI     B,EVFY  ; SET ERROR CODE
1CC9 C9                 RET             ; ERROR: VERIFY FAILURE !

1CCA CD121E   VEO1:     CALL    RBYTE   ; READ NEXT DATA BYTE
1CCD AE                 XRA     M       ; "XOR" WITH MEMORY BYTE
1CCE C2B31C             JNZ     VERR    ; VERIFY ERROR !
1CD1 23                 INX     H       ; BUMP MEMORY PNTR
1CD2 05       VEO2:     DCR     B       ; ANY DATA LEFT ?
1CD3 F2CA1C             JP      VEO1    ; YES: LOOP !
1CD6 3A4280             LDA     SBC     ; GET SECTOR BYTE COUNT
1CD9 3D                 DCR     A       ; ADJUST
1CDA 47                 MOV     B,A     ; B=TRAILER COUNTER (^ TO 80H)
1CDB C3E51C             JMP     VEO4    ; GO CHECK TRAILERS IF ANY !

1CDE CD121E   VEO3:     CALL    RBYTE   ; READ NEXT TRAILER BYTE
1CE1 A7                 ANA     A       ; IT SHOULD BE ZERO
1CE2 C2B31C             JNZ     VERR    ; VERIFY ERROR !
1CE5 04       VEO4:     INR     B       ; ANY TRAILERS LEFT ?
1CE6 F2DE1C             JP      VEO3    ; YES: LOOP !

1CE9 E5                 PUSH    H       ; SAVE MEMORY POINTER
1CEA CD121E             CALL    RBYTE   ; READ 1ST CRC BYTE
1CED 5F                 MOV     E,A
1CEE CD121E             CALL    RBYTE   ; READ 2ND CRC BYTE
```

```
1CF1 57              MOV     D,A
1CF2 2A4380          LHLD    CRC1      ; GET CORRECT CRC BYTES
1CF5 CD8318          CALL    CMPHD     ; COMPARE CRC BYTES
1CF8 E1              POP     H         ; RESTORE MEMORY POINTER
1CF9 C2B31C          JNZ     VERR      ; VERIFY ERROR !

1CFC 22E980  ESEC:   SHLD    TMEM      ; SAVE MEMORY POINTER
1CFF 2AE780          LHLD    TBLK      ; GET BLOCK NUMBER
1D02 23              INX     H         ; INCREMENT IT
1D03 22E780          SHLD    TBLK      ; SAVE UPDATED BLOCK NUMBER
1D06 2AE880          LHLD    TBC       ; GET BYTE COUNT
1D09 1180FF          LXI     D,-128
1D0C 19              DAD     D         ; SUBTRACT ONE SECTOR
1D0D DA131D          JC      BS13      ; LAST SECTOR WAS NOT PARTIAL !
1D10 210000          LXI     H,0       ; SET BYTE COUNT = 0
1D13 22E880  BS13:   SHLD    TBC       ; SAVE UPDATED BYTE COUNT
1D16 7C              MOV     A,H       ; TEST NEW BYTE COUNT ...
1D17 B5              ORA     L
1D18 C2261D          JNZ     BS10      ; MORE TO DO!
1D1B 3AE580          LDA     TFCN      ; GET FUNCTION CODE
1D1E 3D              DCR     A         ; "WRITE" FUNCTION ?
1D1F CA311D          JZ      BS11      ; YES: GO VERIFY!
1D22 0600            MVI     B,0       ; SET "GOOD" STATUS CODE
1D24 FB              EI                ; *** ENABLE INTERRUPTS ***
1D25 C9              RET               ; FINISHED!

1D26 21EE80  BS10:   LXI     H,SEC     ; POINT TO CURRENT SECTOR
1D29 34              INR     M         ; INCREMENT TO NEXT SECTOR
1D2A 3A4900          LDA     CDSEC     ; GET NO. OF SECTORS
1D2D BE              CMP     M         ; OVERFLOW TO NEXT TRACK ?
1D2E C20A1C          JNZ     ISEC      ; NO: GO DO NEXT SECTOR!
1D31 CD881A  BS11:   CALL    BSBO8     ; SETUP
1D34 DA451D          JC      BS12      ; RESET SECTOR!
1D37 3600            MVI     M,0       ; SET SECTOR = ZERO
1D39 23              INX     H         ; POINT TO TRACK
1D3A 34              INR     M         ; INCREMENT TRACK
1D3B 3E29            MVI     A,NTRK    ; GET HIGHEST TRACK NUMBER PLUS ONE
1D3D BE              CMP     M         ; IS IT VALID ?
1D3E C2F31B          JNZ     SEEK      ; YES: GO START NEXT TRACK !
1D41 0609            MVI     B,EBLK    ; SET ERROR CODE
1D43 FB              EI                ; *** ENABLE INTERRUPTS ***
1D44 C9              RET               ; ERROR EXIT: INVALID BLOCK NUMBER !

1D45 3AED80  BS12:   LDA     OSEC      ; GET OLD SECTOR
1D48 32EE80          STA     SEC       ; RESET SECTOR
1D4B C30A1C          JMP     ISEC      ; GO DO NEXT SECTOR!

1D4E E5      WR00:   PUSH    H
1D4F CD081F          CALL    GXOUT     ; HL => XOUT0 OR XOUT1
1D52 7E              MOV     A,M       ; GET PHASE
1D53 F608            ORI     08H       ; SET WRITE BIT
1D55 5F              MOV     E,A       ; COPY IT
1D56 CD851E          CALL    BX0       ; SEND TO DRIVE
1D59 E1              POP     H
1D5A 3EFF            MVI     A,FILL
1D5C CD061E          CALL    WBYTE     ; WRITE A FILL BYTE
1D5F 3E5A            MVI     A,DATAM
1D61 CD061E          CALL    WBYTE     ; WRITE DATA MARK
```

```
1D64 C36C1D             JMP     WR02    ; GO WRITE DATA IF ANY !

1D67 7E         WR01:   MOV     A,M     ; GET NEXT BYTE FROM MEMORY
1D68 23                 INX     H       ; BUMP BUFFER POINTER
1D69 CD061E             CALL    WBYTE   ; WRITE DATA BYTE
1D6C 05         WR02:   DCR     B       ; ANY DATA LEFT ?
1D6D F2671D             JP      WR01    ; YES: LOOP!
1D70 3A4280             LDA     SBC     ; GET SECTOR BYTE COUNT
1D73 3D                 DCR     A       ; ADJUST
1D74 47                 MOV     B,A     ; B=TRAILER COUNTER (^ TO 80H)
1D75 C37C1D             JMP     WR04    ; GO WRITE TRAILERS IF ANY !

1D78 AF         WR03:   XRA     A       ; CLEAR A REGISTER
1D79 CD061E             CALL    WBYTE   ; WRITE A ZERO TRAILER BYTE
1D7C 04         WR04:   INR     B       ; ANY TRAILERS LEFT TO DO ?
1D7D F2781D             JP      WR03    ; YES: LOOP !

1D80 3A4380             LDA     CRC1
1D83 CD061E             CALL    WBYTE   ; WRITE 1ST CRC BYTE
1D86 3A4480             LDA     CRC2
1D89 CD061E             CALL    WBYTE   ; WRITE 2ND CRC BYTE

1D8C 3E20               MVI     A,32    ; WAIT FOR BUFFERS TO EMPTY (2 BYTE TIMES) ....
1D8E 3D         WR05:   DCR     A
1D8F C28E1D             JNZ     WR05
1D92 3E1A               MVI     A,1AH   ; INTERUPT ACKNOWLEDGE BIT SET !!!!!
1D94 D304               OUT     COMND   ; SET TX LINE TO "SPACING"
1D96 3E02               MVI     A,2     ; DELAY A FEW BIT TIMES ....
1D98 3D         WR06:   DCR     A
1D99 C2981D             JNZ     WR06
1D9C E5                 PUSH    H
1D9D CD081F             CALL    QXOUT   ; HL => XOUT0 OR XOUT1
1DA0 5E                 MOV     E,M     ; GET PHASE
1DA1 CD851E             CALL    SX0     ; DISABLE WRITE CIRCUITRY
1DA4 E1                 POP     H
1DA5 3E18               MVI     A,18H   ; INTERUPT ACKNOWLEDGE BIT SET !!!!!
1DA7 D304               OUT     COMND   ; SET TX LINE TO "MARKING"

1DA9 3AE580             LDA     TFCN    ; GET FUNCTION CODE
1DAC 3D                 DCR     A       ; TEST IT
1DAD CAFC1C             JZ      ESEC    ; NORMAL WRITE: WRAP-UP !
1DB0 3E02               MVI     A,2
1DB2 32E580             STA     TFCN    ; SET FUNCTION BACK TO VERIFY
1DB5 FB                 EI              ; *** ENABLE INTERRUPTS ***
1DB6 C3351C             JMP     DSEC    ; NOW GO VERIFY AGAIN !

1DB9 CD701E     RD00:   CALL    QDATAM  ; CONFIRM DATA MARK
1DBC CAD11D             JZ      RD02    ; GOT IT !
1DBF 0612               MVI     B,EDSY  ; SET POSSIBLE ERROR CODE
1DC1 C3FB1D             JMP     RERR

1DC4 CD121E     RD01:   CALL    RBYTE   ; READ NEXT DATA BYTE
1DC7 77                 MOV     M,A     ; STORE THE BYTE
1DC8 AE                 XRA     M       ; "XOR" BACK FROM MEMORY
1DC9 23                 INX     H       ; BUMP MEMORY PNTR
1DCA CAD11D             JZ      RD02    ; O.K. !
1DCD 0618               MVI     B,EMEM  ; SET ERROR CODE
1DCF FB                 EI              ; *** ENABLE INTERRUPTS ***
```

```
1DD0 C9              RET                 ; MEMORY ERROR !
1DD1 05       RD02:  DCR     B           ; ANY DATA LEFT ?
1DD2 F2C41D          JP      RD01        ; YES: LOOP !
1DD5 3A4280          LDA     SBC         ; GET SECTOR BYTE COUNT
1DD8 3D              DCR     A           ; ADJUST
1DD9 47              MOV     B,A         ; B=TRAILER COUNT (^ TO 80H)
1DDA C3E01D          JMP     RD04        ; GO CHECK TRAILERS IF ANY !

1DDD CD121E   RD03:  CALL    RBYTE       ; READ NEXT TRAILER BYTE
1DE0 04       RD04:  INR     B           ; ANY TRAILERS LEFT ?
1DE1 F2DD1D          JP      RD03        ; YES: LOOP !

1DE4 E5              PUSH    H           ; SAVE MEMORY POINTER
1DE5 CD121E          CALL    RBYTE       ; READ 1ST CRC BYTE
1DE8 6F              MOV     L,A
1DE9 CD121E          CALL    RBYTE       ; READ 2ND CRC BYTE
1DEC 67              MOV     H,A
1DED E5              PUSH    H           ; SAVE CRC BYTES
1DEE CD471E          CALL    CRCX        ; COMPUTE CRC FOR SECTOR
1DF1 E1              POP     H           ; RESTORE CRC BYTES
1DF2 CD8318          CALL    CMPHD       ; COMPARE CRC BYTES
1DF5 E1              POP     H           ; RESTORE MEMORY POINTER
1DF6 CAFC1C          JZ      ESEC        ; NO ERRORS !
1DF9 0615            MVI     B,EDCS      ; SET POSSIBLE ERROR CODE
1DFB FB       RERR:  EI                  ; *** ENABLE INTERRUPTS ***
1DFC 21E080          LXI     H,BRTRY     ; POINT TO READ RETRY COUNTER
1DFF 35              DCR     M           ; RETRIES EXHAUSTED ?
1E00 C2351C          JNZ     DSEC        ; NO: TRY AGAIN !
1E03 C3781C          JMP     WIG1        ; GO WIGGLE MAYBE !


1E06 4F       WBYTE: MOV     C,A
1E07 DB03     WB1:   IN      STAT5       ; GET 5501 STATUS
1E09 E610            ANI     10H         ; READY ?
1E0B CA071E          JZ      WB1         ; NO: WAIT !
1E0E 79              MOV     A,C
1E0F D306            OUT     TXBUF       ; SEND THE BYTE0
1E11 C9              RET                 ; EXIT !

1E12 DB03     RBYTE: IN      STAT5       ; GET 5501 STATUS
1E14 E608            ANI     08H         ; GOT A BYTE ?
1E16 CA121E          JZ      RBYTE       ; NO: WAIT !
1E19 DB00            IN      RXBUF       ; READ THE BYTE
1E1B C9              RET                 ; EXIT !

1E1C DB03     RBYTEC: IN     STAT5       ; GET 5501 STATUS
1E1E E608            ANI     08H         ; GOT A BYTE ?
1E20 CA1C1E          JZ      RBYTEC      ; NO: WAIT !
1E23 DB00            IN      RXBUF       ; READ THE BYTE

1E25 F5       CRC:   PUSH    PSW
1E26 E5              PUSH    H
1E27 AB              XRA     E
1E28 67              MOV     H,A
1E29 0F              RRC
1E2A 0F              RRC
1E2B 0F              RRC
1E2C 0F              RRC
```

```
1E2D E60F              ANI     0FH
1E2F AC                XRA     H
1E30 6F                MOV     L,A
1E31 0F                RRC
1E32 0F                RRC
1E33 0F                RRC
1E34 67                MOV     H,A
1E35 E61F              ANI     1FH
1E37 AA                XRA     D
1E38 5F                MOV     E,A
1E39 7C                MOV     A,H
1E3A E6E0              ANI     0E0H
1E3C AD                XRA     L
1E3D 57                MOV     D,A
1E3E 7C                MOV     A,H
1E3F 0F                RRC
1E40 E6F0              ANI     0F0H
1E42 AB                XRA     E
1E43 5F                MOV     E,A
1E44 E1                POP     H
1E45 F1                POP     PSW
1E46 C9                RET

1E47 3A4280   CRCX:     LDA     SBC     ; GET SECTOR BYTE COUNT
1E4A 47                MOV     B,A     ; B=DATA BYTE COUNTER
1E4B 3D                DCR     A
1E4C 4F                MOV     C,A     ; C="TRAILER" COUNTER (^ TO 80H)
1E4D 11FFFF            LXI     D,0FFFFH ; INITIALIZE CRC RESIDUE
1E50 2AE980            LHLD    TMEM    ; GET MEMORY POINTER
1E53 3E5A              MVI     A,DATAM ; DATA MARK
1E55 CD251E            CALL    CRC     ; UPDATE CRC RESIDUE
1E58 C3601E            JMP     CR2

1E5B 7E       CR1:     MOV     A,M     ; GET NEXT BYTE
1E5C 23                INX     H       ; BUMP MEMORY POINTER
1E5D CD251E            CALL    CRC     ; UPDATE CRC RESIDUE
1E60 05       CR2:     DCR     B       ; ANY MORE DATA ?
1E61 F25B1E            JP      CR1     ; YES: LOOP !
1E64 C36B1E            JMP     CR4

1E67 AF       CR3:     XRA     A
1E68 CD251E            CALL    CRC     ; UPDATE CRC RESIDUE
1E6B 0C       CR4:     INR     C       ; ANY TRAILERS LEFT ?
1E6C F2671E            JP      CR3     ; YES: LOOP !
1E6F C9                RET             ; DE = CRC !


1E70 CD121E   GDATAM:  CALL    RBYTE   ; READ GARBAGE BYTE
1E73 CD121E            CALL    RBYTE   ; READ NEXT BYTE
1E76 FE5A              CPI     DATAM   ; DATA MARK ?
1E78 C8                RZ              ; YES !
1E79 CD121E            CALL    RBYTE   ; READ NEXT BYTE
1E7C FE5A              CPI     DATAM   ; DATA MARK ?
1E7E C8                RZ              ; YES !
1E7F CD121E            CALL    RBYTE   ; READ NEXT BYTE
1E82 FE5A              CPI     DATAM   ; DATA MARK ?
1E84 C9                RET             ; <Z>: O.K., <NZ>: ERROR !
```

```
                ;;      ** SELECT DRIVE AND PHASE **

1E85 3AE680     SXO:    LDA     TDRV    ; GET DRIVE NUMBER
1E88 A7                 ANA     A       ; TEST DRIVE NUMBER
1E89 3E10               MVI     A,10H   ; SELECT BIT FOR DRIVE ZERO
1E8B CA8F1E             JZ      SXO1    ; SELECT BIT IS CORRECT
1E8E 07                 RLC             ; CHANGE TO SELECT BIT FOR DRIVE ONE
1E8F B3         SXO1:   ORA     E       ; MERGE WITH PHASE
1E90 D307               OUT     EXTOT   ; SELECT DRIVE AND MOTOR ON
1E92 37                 STC             ; SET CARRY
1E93 C9                 RET             ; RETURN <C>: NO POUND

                ;;      ** VERIFY TRACK AND PHASE **

1E94 AF         VTP:    XRA     A       ; NO COMMENT
                ;       STA     RFLG    ; CLEAR RESTORE FLAG (RFLG)
1E95 CD0E1F             CALL    GCTRK   ; HL => CTRK0 OR CRTK1
1E98 7E                 MOV     A,M     ; GET SUPPOSED CURRENT TRACK
1E99 FE29               CPI     NTRK    ; IS IT VALID ?
1E9B D2B11E             JNC     POUND   ; NO: GO POUND !
1E9E E601               ANI     1       ; SELECT EVEN OR ODD TRACK
1EA0 07                 RLC             ; DOUBLE FOR ZERO OR TWO
1EA1 3C                 INR     A       ; INCREMENT FOR 1 OR 3
1EA2 5F                 MOV     E,A     ; SET ROTATION COUNT IN E
1EA3 3E80               MVI     A,10000000B
1EA5 07         VTP2:   RLC             ; SHIFT SET BIT
1EA6 1D                 DCR     E       ; LEFT E TIMES
1EA7 C2A51E             JNZ     VTP2
1EAA 5F                 MOV     E,A     ; E=CORRECT PHASE FOR SUPPOSED TRACK
1EAB CD081F             CALL    GXOUT   ; HL => XOUT0 OR XOUT1
1EAE 7E                 MOV     A,M     ; GET SUPPOSED PHASE
1EAF BB                 CMP     E       ; CHECK IF IT'S CORRECT
1EB0 C8                 RZ              ;* TRACK GOOD - RETURN   1/15/80 JKP
                                        ; ELSE NOT CORRECT: GO POUND !

1EB1 CD081F     POUND:  CALL    GXOUT   ; HL => XOUT0 OR XOUT1
1EB4 1E01               MVI     E,1     ; SET PHASE TO 1
1EB6 73                 MOV     M,E     ; STORE IT
1EB7 CD851E             CALL    SXO     ; SEND TO DRIVE
1EBA CD0E1F             CALL    GCTRK   ; HL => CTRK0 OR CTRK1
1EBD 362A               MVI     M,42    ; SET CURRENT TRACK NUMBER
1EBF AF                 XRA     A       ; SET TRACK ZERO

1EC0 47         STEPS:  MOV     B,A     ; B=DESIRED TRACK
1EC1 CD0E1F     STP1:   CALL    GCTRK   ; HL => CTRK0 OR CTRK1
1EC4 78                 MOV     A,B     ; GET DESIRED TRACK
1EC5 BE                 CMP     M       ; COMPARE WITH CURRENT TRACK
1EC6 CADA1E             JZ      STP3    ; WE ARE THERE: EXIT !
1EC9 DAD31E             JC      STP2    ; NEED TO STEP OUT !
1ECC 34                 INR     M       ; UPDATE CURRENT TRACK
1ECD CDDD1E             CALL    STPIN   ; STEP IN ONE TRACK
1ED0 C3C11E             JMP     STP1    ; SEE IF WE'RE THERE NOW !

1ED3 35         STP2:   DCR     M       ; UPDATE CURRENT TRACK
1ED4 CDEB1E             CALL    STPOUT  ; STEP OUT ONE TRACK
1ED7 C3C11E             JMP     STP1    ; SEE IF WE'RE THERE NOW !

1EDA CDEB1E     STP3:   CALL    STPOUT  ;* ALWAYS STEP IN TO A TRACK 11/29/79 MAM
```

```
1EDD CD081F   STPIN:   CALL    QXOUT    ; HL => XOUTO OR XOUT1
1EE0 CDE31E            CALL    STPITN   ; STEP IT IN
1EE3 7E       STPITN:  MOV     A,M      ; GET PHASE
1EE4 D601              SUI     1        ; FORCE CARRY IF ZERO
1EE6 3C                INR     A        ; GET OLD PHASE
1EE7 17                RAL              ; SHIFT OLD PHASE
1EE8 C3FB1E            JMP     STP4     ; APPLY STEPPING PULSE

1EEB CD081F   STPOUT:  CALL    QXOUT    ; HL => XOUTO OR XOUT1
1EEE CDF11E            CALL    STPITO   ; STEP IT OUT
1EF1 7E       STPITO:  MOV     A,M      ; GET PHASE
1EF2 0F                RRC              ; B7 <== XO
1EF3 0F                RRC              ; B6 <== XO, B7 <== X1
1EF4 0F                RRC              ; B5 <== XO, B6 <== X1, B7 <== X2
1EF5 D601              SUI     1        ; SET CARRY IF ZERO
1EF7 3C                INR     A        ; GET OLD PHASE
1EF8 17                RAL              ; B0 <== CARRY
1EF9 17                RAL              ; B1 <== CARRY, B0 <== X2
1EFA 17                RAL              ; B2 <== CARRY, B1 <== X2, B0 <== X1
1EFB E607     STP4:    ANI     0111B    ; CLEAR NON PHASE BITS
1EFD 77                MOV     M,A      ; STORE PHASE BYTE
1EFE 5F                MOV     E,A      ; COPY PHASE BYTE
1EFF CD851E            CALL    SXO      ; SEND TO DRIVE

1F02 3A2700   STPWAT:  LDA     STEPCD   ; STEP RATE FOR MICRO DISK
1F05 C35218            JMP     WATS     ; DELAY ONE STEP TIME & RETURN

1F08 21AF81   QXOUT:   LXI     H,XOUTO
1F0B C3171F            JMP     WXYZ

1F0E 21B181   QCTRK:   LXI     H,CTRKO
1F11 C3171F            JMP     WXYZ

1F14 21B581   QCUCNT:  LXI     H,CUCNTO
1F17 3AE680   WXYZ:    LDA     TDRV
1F1A A7                ANA     A
1F1B C8                RZ
1F1C 23                INX     H
1F1D C9                RET

E 1F1E (FFF4)          END     ********** END OF SYSTEM SOFTWARE **********

1 ERRORS
FCS>
```

* UA5
ENDS AT
1FFF