SYNTAXSOFT LIMITED.

Jiffex File

SDX CONTROLLER

LISTING

FOOLSCAP

THE NORTHBRIDGE CENTRE, ELM STREET,
BURNLEY BB10 1PD
TELEPHONE (0282) 38596

```
                .Z80
0000'           CSEG
                ;
                ;Default Configuration Code
0003            DCFIG   EQU 3
                ;
                ;************ MTX SINGLE DISC SYSTEM *************
                ;
                PUBLIC  CCPSTART
                ;
                EXT     PCODE,EXCNFG,EXRD,EXWR,BLKRD,INITLZ,DISCROM
                EXT     SWROM,SWPAGE,RETBASIC,CALSUB
                EXT     CNFG,READ,WRITE,BBLKRD,INITLIZ,RWGO
                EXT     PTRKP,TRUST,CURDRV,DRVRQ,CFGBYT,SECRQ,TRKRQ,DMARQ
                EXT     CDDRV,BPNT,RETRY,PBASE,READ,DBUF,BFID,CFGTAB
                EXT     DUSER,USRJMP,DSCFLG,BDOS
                EXT     PAGEX,JPTABLE,SPARE,TYPEBO,JPLINK,KEYJP
                ;
                ;Basic Addresses
                ;
0000            PGPORT          EQU     0
0008            DEHL            EQU     8
0079            KBD             EQU     79H
00F4            SWITCHO         EQU     0F4H
0250            BASIC2          EQU     250H
0CAB            PRINTX          EQU     0CABH
FAD2            PAGE            EQU     0FAD2H
FA89            USER            EQU     0FA89H
FA92            STKLIM          EQU     0FA92H
FD51            USEFIO          EQU     0FD51H
FFD5            TYPTBL          EQU     0FFD5H
                ;
                ;Equates for Page 0 calls
                ;
3C45            AE              EQU     3C45H
3D84            EVALAB          EQU     3D84H
3E7E            EVALSE          EQU     3E7EH
3FE9            FIND1$          EQU     3FE9H
2927            GETINP          EQU     2927H
2087            GOTMINL         EQU     2087H
200A            INT             EQU     200AH
2AF3            SLOAD1          EQU     2AF3H
3FC6            STR$            EQU     3FC6H
0C4F            ADDVAL          EQU     0C4FH
                ;
                ;
                ;
                ;Bdos and Bios Addresses
                ;
D708            CRMUDC          EQU     0D708H
D7BA            DSKMDC          EQU     0D7BAH
D7C[illegible]  DSKERR          EQU     0D7C[illegible]H
D7E[illegible]  ERRFLG          EQU     0D7E[illegible]H
[illegible]     CURDSK          EQU     [illegible]
[illegible]     CPBASE          EQU     [illegible]
                ;
```

```
                        ;Ring Addresses
                        ;
2016                    POCO            EQU     2016H
0DFF                    LSTLOW          EQU     0DFFFH
D700                    LSTHGH          EQU     0D700H
F00F                    RNGFLG          EQU     0F00FH
F03D                    MBXCNT          EQU     0F03DH
F043                    NEWNAM          EQU     0F043H
F046                    RNGERR          EQU     0F046H
F052                    NLIST           EQU     0F052H
F054                    LSTBAS          EQU     0F054H
F072                    MBXTOP          EQU     0F072H
F074                    MBXBEG          EQU     0F074H
F076                    MBXEND          EQU     0F076H
F078                    MBXSIZE         EQU     0F078H
F07A                    MBXFREE         EQU     0F07AH
F0D1                    INTA            EQU     0F0D1H
F190                    RINGROM         EQU     0F190H
F2FE                    LSTTOP          EQU     0F2FEH
                        ;
                        ;Single Disc addresses
                        ;
D706                    JP59K           EQU     0D706H
E680                    DMADR           EQU     0E680H
                        ;
                        ;
                        ;Start of ROM code
                        ;
                        ;
                        ;ROM RECOGNITION CODE
                        ;
0000' 08 07 06 05       DB      8,7,6,5,4,3,2,1
0004' 04 03 02 01
0008' 02EA'             DW      ROMCODE
000A'                   DS      2
                        ;
                        ;ROM ENTRY POINT (200CH)    Called by ROM command.
                        ;
000C' C3 0014'          ROM6:   JP CPMLDR
000F' 00                        NOP
                        ;
                        ;POC ENTRY POINT (2010H)    Called by ROM system startup processing.
                        ;
0010' C3 01A0'          POC:    JP INITDSC
0013' 00                        NOP
                        ;
0014' CD 01A0'          CPMLDR: CALL INITDSC
0017' CD 001D'                  CALL LOADER
001A' C3 0000*                  JP RETBASIC
                        ;
                        ;
                        ;
001D' 21 D6FF           LOADER: LD HL,0D6FFH    } move STKLIM down.
0020' 22 FA92                   LD (STKLIM),HL
0023' CD 0221'                  CALL RNGCHK             ;Check for Ring ROM
0026' CD 024B'                  CALL I.MOVLST           ;Move mailbox and list
```

```
                        ;
0029'  21 02F5'         LD HL,?CNFG
002C'  11 0000*         LD DE,PCODE
002F'  01 00AF          LD BC,?PEND-?CNFG+1
0032'  ED B0            LDIR                    ;Load code and DPBs to RAM
                        ;
0034'  3A FAD2          LD A,(PAGE)
0037'  32 0001*         LD (SWROM+1),A          ;Insert ROM page in SWROM
                        ;
003A'  CD 008B'         CALL INDSC              ;Initialise DSC variables
003D'  CD 0000*         CALL RWGO               ;Read head move
0040'  21 0000          LD HL,0
0043'  22 0000*         LD (TRKRQ),HL
0046'  06 23            LD B,35
0048'  CD 0000*         CALL BLKRD              ;Load BDOS and BIOS
                        ;
004B'  CD 0072'         CALL DSCHK              ;Check for 59K system
004E'  CD 0102'         CALL RELOC              ;Change addresses in BDOS and BIOS
0051'  CD 00D4'         CALL INCPM              ;Initialise BDOS and BIOS
                        ;
0054'  21 D840          LD HL,0D840H
0057'  06 D2            LD B,210
0059'  AF               XOR A
005A'  77       CLRLP:  LD (HL),A
005B'  23               INC HL
005C'  10 FC            DJNZ CLRLP
                        ;
005E'  21 D706          LD HL,JP59K
0061'  22 0001*         LD (BDOS+1),HL
0064'  21 0000*         LD HL,BDOS
0067'  36 C3            LD (HL),0C3H            ;Insert BDOS jump
                        ;
0069'  CD 00FC'         CALL CCPSTART
                        ;
006C'  21 0000*         LD HL,DSCFLG
006F'  36 01            LD (HL),1
0071'  C9               RET
                        ;
                        ;
                        ;
0072'  3A D708  DSCHK:  LD A,(CPMLOC)
0075'  FE D7            CP 0D7H
0077'  C8               RET Z
0078'  D7               RST 10H
0079'  8E 0D 0A 44      DB 80H+14,0DH,0AH,'DISC ERROR',0DH,0AH
007D'  49 53 43 20
0081'  45 52 52 4F
0085'  52 0D 0A
0088'  C3 0000*         JP RETBASIC
                        ;
                        ;
                        ;
008B'  21 0000*  INDSC: LD HL,PTRKP
008E'  AF               XOR A
008F'  06 06            LD B,6
0091'  77        INDS1: LD (HL),A               ;PTRKP to SWUF = 0
```

```
0092'  23                      INC HL
0093'  10 FC                   DJNZ INDS1
0095'  3D                      DEC A
0096'  06 16                   LD B,22
0098'  77              INDS2:  LD (HL),A
0099'  23                      INC HL
009A'  10 FC                   DJNZ INDS2              ;CURDRV to TRUST = FFH
009C'  21 0000*                LD HL,DBUF
009F'  22 0006*                LD (BFID+6),HL
00A2'  AF                      XOR A
00A3'  32 0000*                LD (DRVRQ),A            ;Boot drive is B
00A6'  3E 03                   LD A,3
00A8'  32 0000*                LD (CFGBYT),A           ;Boot drive is type 3
00AB'  21 0014                 LD HL,20
00AE'  22 0000*                LD (TRKRQ),HL           ;Track 0
00B1'  21 0012                 LD HL,18
00B4'  22 0000*                LD (SECRQ),HL           ;Record 18  1st. record of BDOS
00B7'  21 D700                 LD HL,0D700H
00BA'  22 0000*                LD (DMARQ),HL           ;DMA address is start of BDOS
00BD'  CD 0000*                CALL INITLZ             ;initialise FDC
00C0'  21 0303                 LD HL,0303H
00C3'  22 0000*                LD (CFGTAB),HL          ;Replaces EXCNFG, drive types to table
                       ;
00C6'  AF                      XOR A
00C7'  32 0000*                LD (CDDRV),A            ;4   Current logged drive 0 -7
00CA'  3C                      INC A
00CB'  32 0000*                LD (BPNT),A             ;44H  Boot drive pointer 1 - 8
00CE'  3E 0A                   LD A,10
00D0'  32 0000*                LD (RETRY),A            ;45H  Retry counter
00D3'  C9                      RET
                       ;
                       ;Cold boot routine
                       ;
00D4'  11 E53D         INCPM:  LD DE,DPBASE+10
00D7'  21 0167'                LD HL,DPHTAB
00DA'  01 0006                 LD BC,6
00DD'  ED B0                   LDIR                    ;Load DPH for drive A
00DF'  11 E54D                 LD DE,DPBASE+26
00E2'  21 0167'                LD HL,DPHTAB
00E5'  01 0006                 LD BC,6
00E8'  ED B0                   LDIR                    ;Load DPH for drive B
00EA'  11 E55D                 LD DE,DPBASE+42
00ED'  21 016D'                LD HL,DPHTBC
00F0'  01 0006                 LD BC,6
00F3'  ED B0                   LDIR                    ;Load DPH for drive C
00F5'  21 E580                 LD HL,DMADR
00F8'  22 0000*                LD (DMARQ),HL
00FB'  C9                      RET
                       ;
                       ;
                       ;
00FC'                  CCPSTART:
00FC'  0E 0D                   LD C,0DH
00FE'  CD 0000*                CALL BDOS               ;Reset disc system
0101'  C9                      RET
                       ;
```

```
                        ;
                        ;
0102'  21 0127'         RELOC:  LD HL,RELTAB
0105'  06 10                    LD B,16
0107'  C5               RLOOP:  PUSH BC
0108'  5E                       LD E,(HL)
0109'  23                       INC HL
010A'  7E                       LD A,(HL)
010B'  C6 14                    ADD A,14H
010D'  57                       LD D,A                  ;Offset for 59K system
010E'  23                       INC HL
010F'  4E                       LD C,(HL)
0110'  23                       INC HL
0111'  46                       LD B,(HL)
0112'  23                       INC HL
0113'  EB                       EX DE,HL
0114'  71                       LD (HL),C
0115'  23                       INC HL
0116'  70                       LD (HL),B
0117'  EB                       EX DE,HL
0118'  C1                       POP BC
0119'  10 EC                    DJNZ RLOOP
                        ;
011B'  21 017D'                 LD HL,ROMERR
011E'  11 D7E5                  LD DE,ERRFLG
0121'  01 002D                  LD BC,ERREND-ROMERR+1
0124'  ED B0                    LDIR                    ;Load error routine to BDOS
0126'  C9                       RET
                        ;
                        ;
                        ;
0127'                   RELTAB:
0127'  CC25 DA00                DW 0CC25H, 0DA00H       ;Move BDOS stack below C601H
012B'  CCA2 0000*               DW 0CCA2H, RETBASIC
012F'  CCB8 0000*               DW 0CCB8H, RETBASIC
0133'  CF91 E680                DW 0CF91H, DMADR        ;Default DMARQ
0137'  DCF5 0000*               DW 0DCF5H, TRKRQ
013B'  DCFD 0000*               DW 0DCFDH, TRKRQ
013F'  D405 0000*               DW 0D405H, SECRQ
0143'  D40D 0000*               DW 0D40DH, DMARQ
0147'  D407 0000*               DW 0D407H, SFNT
014B'  D43B 0000*               DW 0D43BH, DRVRQ
014F'  D445 0000*               DW 0D445H, CDDRV
0153'  D44D 0000*               DW 0D44DH, CDDRV
0157'  D454 0000*               DW 0D454H, RETRY
015B'  D459 0000*               DW 0D459H, READ
015F'  D468 0000*               DW 0D468H, RETRY
0163'  D46D 0000*               DW 0D46DH, WRITE
                        ;
                        ;
                        ;
                        ;Disc parameter header table for drives A and B
                        ;
0167'  0001*            DPHTAB: DW PBASE+1              ;Address of DPB
0169'  E954                     DW 0E954H               ;CK vector
016B'  E940                     DW 0E940H               ;Allocation vector
```

```
                        ;
                        ;Disc parameter header table for drive C
                        ;
016D'  0001*            DPHT3C: DW PBASE+1              ;Address of DPH
016F'  E978                     DW 0E978H               ;CK vector
0171'  E984                     DW 0E984H               ;Allocation vector
                        ;
                        ;
                        ;
                        ;New BDOS error routine loaded to ERRFLG
                        ;
D80E                    CRLFO   EQU     CRLF-ROMERR+ERRFLG
D804                    PRINTO  EQU     PRINT-ROMERR+ERRFLG
                        ;
0173'  E5               ROMERR: PUSH HL
0174'  01 D80E                  LD BC,CRLFO
0177'  CD D804                  CALL PRINTO
017A'  3A DA42                  LD A,(CURDSK)
017D'  C6 41                    ADD A,'A'
017F'  32 D7C6                  LD (DSKERR),A
0182'  01 D7BA                  LD BC,DSKMSG
0185'  CD D804                  CALL PRINTO
0188'  C1                       POP BC
0189'  CD D804                  CALL PRINTO
                        ;
018C'  CD 0079          ERRFL1: CALL KBD
018F'  28 FB                    JR Z,ERRFL1             ;Jump if no key pressed
0191'  C9                       RET
                        ;
0192'  0A               PRINT:  LD A,(BC)
0193'  FE 24                    CP '$'
0195'  C8                       RET Z
0196'  CD 0CAB                  CALL PRINTX             ;Print character
0199'  03                       INC BC
019A'  18 F6                    JR PRINT
                        ;
019C'  0D 0A 24         CRLF:   DB 0DH,0AH, '$'
019F'  00               ERREND: NOP
                        ;
                        ;End of block copied to ERRFLG
                        ;
                        ;
                        ;
01A0'                   INITD3C:
01A0'  CD 0000*                 CALL INITL2             ;Reset FDC
                        ;
01A3'  21 0205'                 LD HL,USRCPY
01A6'  11 FA87                  LD DE,USER-2
01A9'  01 0006                  LD BC,6
01AC'  ED B0                    LDIR                    ;Copy user bytes to USER
                        ;
01AE'  21 020A'                 LD HL,USERCODE
01B1'  11 0000*                 LD DE,USRJMP
01B4'  01 0017                  LD BC,USREND-USERCODE+1
01B7'  ED B0                    LDIR                    ;Load USER jump routine
                        ;
```

```
01B9'  21 0000*                LD HL,USRJMP
01BC'  01 0005                 LD BC,NROM-USERCODE+1
01BF'  09                      ADD HL,BC
01C0'  3A FAD2                 LD A,(PAGE)
01C3'  77                      LD (HL),A              ;Insert correct page in code
                       ;
01C4'  11 0000*                LD DE,JPLINK
01C7'  06 02                   LD B,2
01C9'  3E C9                   LD A,0C9H
01CB'  12              LNKLP:  LD (DE),A              ;Insert RET for links
01CC'  13                      INC DE
01CD'  13                      INC DE
01CE'  13                      INC DE
01CF'  10 FA                   DJNZ LNKLP
                       ;
01D1'  AF                      XOR A
01D2'  32 0000*                LD (DSCFLG),A          ;Clear disc flag
                       ;
                       ;       LD A,(PAGE)
                       ;       AND 70H
                       ;       LD (TYPTBL+9),A
                       ;       LD HL,TYPE80
                       ;       LD (TYPTBL+10),HL
                       ;
                       ;The following code tests to see if the Ring is using USERIO. If it is not.
                       ;then it is diverted to MBXBUFF where the code at COPY rewrites the USER
                       ;bytes, in case they are have been overwritten by cassette load routine.
                       ;
01D5'  21 FD53                 LD HL,USERIO+2
01D8'  7E                      LD A,(HL)
01D9'  FE 36                   CP 36H
01DB'  C0                      RET NZ                 ;Return if USERIO in use
                       ;
01DC'  21 0000*                LD HL,KEYJP
01DF'  22 FD52                 LD (USERIO+1),HL       ;Jump MBXBUFF at USERIO
01E2'  11 01EC'                LD DE,COPY
01E5'  EB                      EX DE,HL
01E6'  01 001E                 LD BC,30
01E9'  ED B0                   LDIR                   ;Move COPY to MBXBUFF
01EB'  C9                      RET
                       ;
01EC'  21 0019*        COPY:   LD HL,KEYJP+25
01EF'  11 FA87                 LD DE,USER-2
01F2'  01 0005                 LD BC,5
01F5'  ED B0                   LDIR
                       ;
01F7'  21 FA93                 LD HL,STKLIM+1
01FA'  7E                      LD A,(HL)
01FB'  FE F3                   CP 0F3H
01FD'  DA 3622                 JP C,3622H
0200'  36 F4                   LD (HL),0F4H
                       ;
0202'  C3 3622                 JP 3622H
                       ;
                       ;User bytes
                       ;
```

```
0205'                           USRCPY:
0205'   C9 07 C3                       DB 0C9H,07,0C3H  - No eyeball check. Jump to interface code in RAM
0208'   0000*                          DW USRJMP
                                ;
                                ;This code is loaded to USRJMP
                                ;
020A'                           USERCODE:
020A'   3A FAD2                         LD A,(PAGE)      } Stash current page
020D'   F5                              PUSH AF
020E'   3E 70                   NROM:   LD A,70H                ;dummy value
0210'   32 FAD2                        LD (PAGE),A      } Select required Rom
0213'   D3 00                          OUT (PGPORT),A
0215'   D5                              PUSH DE          } Call Rom saving DE
0216'   CD 02DD'                        CALL USEROM
0219'   D1                              POP DE
021A'   F1                              POP AF           } Restore page
021B'   32 FAD2                        LD (PAGE),A
021E'   D3 00                          OUT (PGPORT),A
0220'   C9                      USREND: RET                     Exit
                                ;
                                ;
                                ;
                                ;RNGCHK returns with Z=1 if ring loaded
                                ;
0221'   06 04                  RNGCHK: LD B,4           }
0223'   11 F0D4                        LD DE,INTA+3     }
0226'   21 0247'                       LD HL,TSTBLK     } Test first 4 bytes of RINC code in RAM
0229'   1A                      RNGC1:  LD A,(DE)        }
022A'   BE                              CP (HL)          }
022B'   C0                              RET NZ           }       ;Return if code at INTA not present
022C'   13                              INC DE           }
022D'   23                              INC HL           }
022E'   10 F9                           DJNZ RNGC1       }
0230'   3A F00F                 RNGC3:  LD A,(RNGFLG)    } Jump if RNGFLG = 0
0233'   B7                              OR A             }
0234'   28 07                          JR Z,RNGC3       }
0236'   06 03                          LD B,3           }
0238'   3D                      RNGC2:  DEC A            }
0239'   C8                              RET Z            }       ;Return if (RNGFLG)=1,2,3 with Z=1
023A'   10 FC                          DJNZ RNGC2       }
023C'   C9                              RET              }       ;Return with Z=0
023D'   3A F046                 RNGC3:  LD A,(RNGERR)    }
0240'   B7                              OR A             }
0241'   C8                              RET Z            }
0242'   D6 37                          SUB 37H          }
0244'   C8                              RET Z            }
0245'   3D                              DEC A            }
0246'   C9                              RET              }       ;Return with Z=1 if (RNGERR)=37H or 38H
                                ;
0247'   D5                      TSTBLK: PUSH DE          }       ;Code at INTA+3
0248'   08                              EX AF,AF'        }
0249'   F5                              PUSH AF          }
024A'   08                              EX AF,AF'        }
                                ;
                                ;
                                ;
```

This code copied to RAM

```
                        ;MOVLST shifts the top of the list to LSTHGH from a higher address  and
                        ;sets the bottom of the mailbox to LSTLOW. If (RNGFLG) < 2, then the list
                        ;is undefined and the pointers can be set equal to LSTHGH.
                        ;
024B'  F3               MOVLST: DI
024C'  3A F00F                  LD A,(RNGFLG)
024F'  CB 4F                    BIT 1,A
0251'  20 14                    JR NZ,MOVLS1            ;Jump if list defined
0253'  21 D700                  LD HL,LSTHGH
0256'  22 F2FE                  LD (LSTTOP),HL
0259'  22 F054                  LD (LSTBAS),HL
025C'  22 F052                  LD (NLIST),HL
025F'  21 D3FF                  LD HL,LSTLOW
0262'  22 FA92                  LD (STKLIM),HL
0265'  FB                       EI
0266'  C9                       RET
                        ;
0267'  AF               MOVLS1: XOR A
0268'  32 F043                  LD (NEWNAM),A
026B'  2A F054                  LD HL,(LSTBAS)
026E'  E5                       PUSH HL                 ;old (LSTBAS) to stack
026F'  2A F052                  LD HL,(NLIST)
0272'  E5                       PUSH HL                 ;old (NLIST) to stack
0273'  FB                       EI
0274'  2A F2FE                  LD HL,(LSTTOP)
0277'  11 D700                  LD DE,LSTHGH
027A'  A7                       AND A
027B'  ED 52                    SBC HL,DE
027D'  44                       LD B,H
027E'  4D                       LD C,L                  ;BC = displacement  (LSTTOP) - LSTHGH
027F'  CA 02DA'                 JP Z,MOVLSX             ;Jump if list already at LSTHGH
0282'  2A F2FE                  LD HL,(LSTTOP)
0285'  D1                       POP DE                  ;HL = old (LSTTOP)  DE = old (NLIST)
0286'  A7                       AND A
0287'  ED 52                    SBC HL,DE               ;HL = number of bytes in list
0289'  E3                       EX (SP),HL
028A'  ED 42                    SBC HL,BC
028C'  E3                       EX (SP),HL              ;new (LSTBAS) to stack
028D'  D5                       PUSH DE                 ;old (NLIST) to stack
028E'  E5                       PUSH HL                 ;number of bytes to stack
028F'  EB                       EX DE,HL
0290'  A7                       AND A
0291'  ED 42                    SBC HL,BC               ;HL = new (NLIST)
0293'  C1                       POP BC                  ;BC = number of bytes
0294'  D1                       POP DE
0295'  E5                       PUSH HL                 ;new (NLIST) to stack
0296'  EB                       EX DE,HL                ;HL = old (NLIST)  DE = new (NLIST)
0297'  78                       LD A,B
0298'  B1                       OR C
0299'  28 02                    JR Z,MOVLS2             ;Jump if list has zero length
029B'  ED B0                    LDIR                    ;Move list
029D'  C1               MOVLS2: POP BC                  ;BC = new (NLIST)
029E'  D1                       POP DE                  ;DE = new (LSTBAS)
029F'  F3                       DI
02A0'  3A F043                  LD A,(NEWNAM)
02A3'  B7                       OR A                    ;Check for new name while list was moved
```

```
02A4'  C2 024B'                    JP NZ,MOVLST
02A7'  21 D700                     LD HL,LSTHGH
02AA'  22 F2FE                     LD (LSTTOP),HL
02AD'  ED 53 F054                  LD (LSTBAS),DE
02B1'  ED 53 F072                  LD (MBXTOP),DE
02B5'  ED 43 F052                  LD (NLIST),BC
02B9'  21 D3FF                     LD HL,LSTLOW
02BC'  22 FA92                     LD (STKLIM),HL
02BF'  23                          INC HL
02C0'  22 F074                     LD (MBXBEG),HL
02C3'  22 F076                     LD (MBXEND),HL
02C6'  FB                          EI
02C7'  EB                          EX DE,HL
02C8'  AF                          XOR A
02C9'  ED 52                       SBC HL,DE           ;(LSTBAS) - (STKLIM) - 1
02CB'  30 03                       JR NC,CLRMB
02CD'  21 0000                     LD HL,0
02D0'  32 F03D             CLRMB:  LD (MBXCNT),A
02D3'  22 F07A                     LD (MBXFREE),HL
02D6'  22 F078                     LD (MBXSIZE),HL
02D9'  C9                          RET
02DA'  E1                  MOVLSX: POP HL
02DB'  E1                          POP HL
02DC'  C9                          RET
                           ;
                           ;
                           ;
02DD'  3A 0000*            USEROM: LD A,(DSCFLG)
02E0'  FE 01                       CP 1
02E2'  D5                          PUSH DE
02E3'  C4 001D'                    CALL NZ,LOADER
02E6'  D1                          POP DE
02E7'  C3 0000*                    JP DUSER
                           ;
                           ;
                           ;
02EA'                      ROMCODE:
02EA'  46 57 31 37                 DB "FW17"
02EE'  D7                          RST 10H
02EF'  84                          DB 84H
02F0'  44 69 73 63                 DB "Disc"
02F4'  C9                          RET
                           ;
                       C   INCLUDE PCODE.ROM
                       C   ;Include file for DC and DV
                       C   ;
02F5'  CD 0000*        C   ?CNFG:: CALL DISCROM
02F8'  0000*           C           DW EXCNFG
02FA'  C9              C           RET
                       C   ;
02FB'  E5              C   ?READ:: PUSH HL
02FC'  CD 0000*        C           CALL DISCROM
02FF'  0000*           C           DW EXRD
0301'  E1              C           POP HL
0302'  C9              C           RET
                       C   ;
```

All this code copied into RAM.

```
0303'                   C       ?WRITE::
0303'  E5               C               PUSH HL
0304'  CD 0000*         C               CALL DISCROM
0307'  0000*            C               DW EXWR
0309'  E1               C               POP HL
030A'  C9               C               RET
                        C       ;
030B'                   C       ?BBLKRD::
030B'  CD 0000*         C               CALL DISCROM
030E'  0000*            C               DW BLKRD
0310'  C9               C               RET
                        C       ;
0311'                   C       ?INITLIZ::
0311'  CD 0000*         C               CALL DISCROM
0314'  0000*            C               DW INITLZ
0316'  C9               C               RET
                        C       ;
0317'  CD 0000*         C               CALL DISCROM
031A'  00FC'            C               DW CCPSTART
031C'  C9               C               RET
031D'  00               C               NOP
                        C       ;
                        C       ;
                        C       ;Switch in DISC ROM.
                        C       ;
031E'                   C       ?SWROM::
031E'  3E 70            C               LD A,70H                ;Dummy value
                        C       ;
                        C       ;Switch page.
                        C       ;
0320'                   C       ?SWPAGE::
0320'  32 FAD2          C               LD (PAGE),A
0323'  D3 00            C               OUT (PGPORT),A          } we must be working in RAM.
0325'  C9               C               RET
                        C       ;
                        C       ;
                        C       ;DISCROM entry point. return address on stack points to CALL address
                        C       ;Return address is adjusted and subroutine in ROM is called. DE and BC
                        C       ;preserved. Entry value of A preserved.
                        C       ;
0326'                   C       ?DISCROM::
0326'  E5               C               PUSH HL         } IY=HL
0327'  FD E1            C               POP IY                  ;Save HL, IY clobbered
0329'  E1               C               POP HL          - Return Addr
032A'  D5               C               PUSH DE         - Save DE
032B'  5E               C               LD E,(HL)       }
032C'  23               C               INC HL          } Get jump parameter
032D'  56               C               LD D,(HL)       }
032E'  23               C               INC HL          }
032F'  E3               C               EX (SP),HL              ;New return address to stack
0330'  EB               C               EX DE,HL                ;DE restored
0331'  E5               C               PUSH HL                 ;Call address to stack
0332'  6F               C               LD L,A          L=A
0333'  3A FAD2          C               LD A,(PAGE)
0336'  67               C               LD H,A          A=(Page)
0337'  CD 0000*         C               CALL SWROM              ;Select ROM  - Disc rom?
```

```
033A'  7D            C          LD A,L                  ;Restore A
033B'  E3            C          EX (SP),HL              ;Page to stack
033C'  E5            C          PUSH HL                 ;Call address to stack
033D'  21 0000*      C          LD HL,CALSUB
0340'  E3            C          EX (SP),HL              ;CALLSUB to stack
0341'  E5            C          PUSH HL                 ;Call address to stack
0342'  FD E5         C          PUSH IY
0344'  E1            C          POP HL                  ;Restore HL
0345'  C9            C          RET
                     C  ;
0346'                C  ?CALSUB::
0346'  E1            C          POP HL
0347'  F5            C          PUSH AF
0348'  7C            C          LD A,H
0349'  CD 0000*      C          CALL SWPAGE
034C'  F1            C          POP AF
034D'  C9            C          RET
                     C  ;
                     C  ;
                     C  ;
                     C  ;PAGE0 calls a specified routine in ROM page zero. Byte on top of machine
                     C  ;stack gives offset into jump table. Switches to page zero before jumping to
                     C  ;routine, and switches back to NODE ROM page on return.
                     C  ;Affects no registers, except AF'.
                     C  ;
034E'                C  ?PAGE0::
034E'  08            C          EX AF,AF'               ;Save true AF
034F'  E3            C          EX (SP),HL              ;HL -> data byte
0350'  7E            C          LD A,(HL)               ;A = offset
0351'  23            C          INC HL
0352'  E3            C          EX (SP),HL
                     C  ;
                     C  ;Registers can now be pushed onto stack.
                     C  ;
0353'  E5            C          PUSH HL                 ;Save HL
0354'  21 0000*      C          LD HL,PAGEX             ;Return address for page 0 routine
0357'  E3            C          EX (SP),HL              ;Restore HL
0358'  E5            C          PUSH HL
0359'  D5            C          PUSH DE
035A'  C5            C          PUSH BC                 ;Save HL,DE,BC
                     C  ;
035B'  4F            C          LD C,A
035C'  06 00         C          LD B,0                  ;BC = offset
035E'  08            C          EX AF,AF'               ;Restore true AF
                     C  ;
035F'  21 0000*      C          LD HL,JPTABLE
0362'  09            C          ADD HL,BC
0363'  09            C          ADD HL,BC               ;HL -> address of required routine
0364'  CF            C          RST DEHL
0365'  EB            C          EX DE,HL                ;HL = address of routine
                     C  ;
0366'  C1            C          POP BC
0367'  D1            C          POP DE                  ;Restore BC,DE
                     C  ;
                     C  ;Here HL -> address of routine. (SP) = true value of HL.
                     C  ;Switch in page 0 and call routine.
```

```
                        ;
036B'  CD 00F4          CALL SWITCH0            ;Switch in ROM page 0
036E'  E3               EX (SP),HL              ;HL = true HL. (SP) -> routine.
036F'  C9               RET                     ;'Call' routine
                        ;
                        ;The called routine returns to here.
                        ;
0370'                   ?PAGEX::
0370'  F5               PUSH AF
0371'  CD 0000*         CALL SWROM              ;Switch in RING ROM page
0374'  F1               POP AF
                        ;
0375'                   ?SPARE::
0375'  C9               RET
                        ;
                        ;
                        ;
0376'                   ?JPTABLE::
                        ;
0376'  3C45             ADD0:   DW      AE
0378'  3D84             ADD1:   DW      EVALAB
037A'  3E7E             ADD2:   DW      EVALSE
037C'  3FE9             ADD3:   DW      FIND1$
037E'  2927             ADD4:   DW      GETINP
0380'  20B7             ADD5:   DW      GOTMIN1
0382'  200A             ADD6:   DW      INT
0384'  2AF5             ADD7:   DW      SLOAD1
0386'  3FC6             ADD8:   DW      STR$
0388'  0C4F             ADD9:   DW      ADJVAL
038A'  0030             ADD10:  DW      30H             ;RESET GETRST
038C'  288F             ADD11:  DW      288FH           ;SGOTO
                        ;
                        ;
038E'                   ?RETBASIC::
038E'  3E 00            LD A,0
0390'  CD 0000*         CALL SWPAGE
0393'  C3 0250          JP BASIC2
                        ;
                        ;       DISC PARAMETER BLOCK SET
                        ;
0396'                   ?PBASE::
0396'  03               DB      3       ;       SIN S/T D/D D/S
0397'  001A             DW      26
0399'  04               DB      4
039A'  0F               DB      15
039B'  01               DB      1
039C'  009B             DW      155
039E'  003F             DW      63
03A0'  80               DB      10000000B
03A1'  00               DB      00000000B
03A2'  0010             DW      16
03A4'  0002             DW      2
                        ;
03A6'  FF               ?PEND:: DB      0FFH            ;TERMINATOR
                        ;
                        END
```

Select Rom a and jump to basic

End of RAM code

Macros:

Symbols:

?BBLKR	0308'	?CALSU	0346'	?CNFG	02F5'	?DISCR	0326'
?INITL	0311'	?JPTAB	0373'	?PAGE0	034E'	?PAGEX	036D'
?PBASE	0393'	?PEND	03A3'	?READ	02FB'	?RETBA	038B'
?SPARE	0372'	?SWPAG	0320'	?SWROM	031E'	?WRITE	0303'
ADD0	0373'	ADD1	0375'	ADD10	0387'	ADD11	0389'
ADD2	0377'	ADD3	0379'	ADD4	037B'	ADD5	037D'
ADD6	037F'	ADD7	0381'	ADD8	0383'	ADD9	0385'
ADJVAL	0C4F	AE	3C45	BASIC2	0250	BBLKRD	0000*
BDOS	00FF*	BFID	00A0*	BLKRD	030E*	BFNT	0149*
CALSUB	033E*	CCPSTA	00FC'	CDDRV	0155*	CFGBYT	00A9*
CFGTAB	00C4*	CLRLP	005A'	CLRMB	02D0'	CNFG	0000*
COPY	01EC'	CPMLDR	0014'	CPMLOC	D708	CRLF	019C'
CRLF0	D80E	CURDRV	0000*	CURDSK	DA42	DBUF	009D*
DCFIG	0003	DEHL	0008	DISCR0	0318*	DMADR	E580
DMARQ	0145*	DPBASE	E533	DPHTAB	0167'	DPHTBC	016D'
DRVRQ	014D*	DSCFLG	02DE*	DSCHK	0072'	DSKERR	D7C6
DSKMSG	D7BA	DUSER	02E8*	ERREND	019F'	ERRFL1	018C'
ERRFLG	D7E5	EVALAB	3D84	EVALSE	3E7E	EXCNFG	02F8*
EXRD	02FF*	EXWR	0307*	FIND1$	3FE9	GETINP	2927
GOTMIN	20B7	INCPM	00D4'	INDS1	0091'	INDS2	0098'
INDSC	008B'	INITDS	01A0'	INITLI	0000*	INITLZ	0314*
INT	200A	INTA	F0D1	JP59K	D706	JPLINK	01C5*
JPTABL	0360*	KBD	0079	KEYJP	01ED*	LNKLP	01CB'
LOADER	001D'	LSTBAS	F054	LSTHGH	D700	LSTLOW	03FF
LSTTOP	F2FE	MBXBEG	F074	MBXCNT	F03D	MBXEND	F076
MBXFRE	F07A	MBXSIZ	F078	MBXTOP	F072	MOVLS1	0267'
MOVLS2	029D'	MOVLST	024B'	MOVLSX	02DA'	NEWNAM	F043
NLIST	F052	NROM	020E'	PAGE	FAD2	PAGEX	0355*
PBASE	016D*	PCODE	002D*	PGPORT	0000	POC	0010'
POC0	2016	PRINT	0192'	PRINT0	D804	PRINTX	0CAB
PTRKP	008C*	READ	015D*	RELOC	0102'	RELTAB	0127'
RETBAS	0131*	RETRY	0161*	RINGR0	F190	RLOOP	0107'
RNGC1	0229'	RNGC2	0238'	RNGC3	023D'	RNGCHK	0221'
RNGERR	F046	RNGFLG	F00F	ROM6	000C'	ROMCOD	02EA'
ROMERR	0173'	RWGO	003E*	SECRQ	0141*	SLOAD1	2AF5
SPARE	0000*	STKLIM	FA92	STR$	3FC6	SWITCH	00F4
SWPAGE	033E*	SWROM	033F*	TRKRQ	013D*	TRUST	0000*
TSTBLK	0247'	TYPES0	0000*	TYPTBL	FFD5	USER	FA89
USERC0	020A'	USERI0	FD51	USEROM	02DD'	USRCPY	0205'
USREND	0220'	USRJMP	0208*	WRITE	0165*		

No No Fatal error(s) Fatal error(s)

```
0000'                   CSEG

                        ;       ****************************
                        ;       *                          *
                        ;       * DISC INTERFACE SOFTWARE *
                        ;       * FOR TB'S  FLOPPY CNTRLR *
                        ;       * By F Wallinger          *
                        ;       *                          *
                        ;       ****************************
                        ;
                        ;ROM version for MTX single disc system
                        ;With Tony Brewers 5" Only Board
                        ;
                        PUBLIC EXCNFG,EXRD,EXWR,BLKRD,INITLZ,RWGO
                        ;
                        EXT     TOAM,DBUF,TDBUF
                        EXT     PTRKP,LCA,EFLAG,LSTOUT,SWUF,CURDRV,TRACKS,BFID
                        EXT     PBASE
                        EXT     NSTK,SKEW6,CFGTAB
                        EXT     TRUST,DRVRQ,CFGBYT,TRKRQ,SECRQ,DMARQ
                        ;
0004                    SPEED   EQU     4       ; 4=4MHZ, 6=6MHZ
                        ;
B0ED                    LDIR    EQU 00EDH+08000H
0000                    FALSE   EQU     0
FFFF                    TRUE    EQU     NOT(FALSE)
                        ;
                        ;
                        ; ---------------------------------------------------
                        ;
                        ;Hardware Interface, Intermediate Code and Executives
                        ;
                        ;
                        ;
                        .Z80

                        ;
0010                    FDCPORT EQU     010H
                        ;
0010                    FDCCOM  EQU     FDCPORT         ;FDC command register port (OUT)
0010                    FDCSTA  EQU     FDCPORT         ;FDC status register port (IN)
0011                    FDCTRK  EQU     FDCPORT+1       ;FDC track register port (IN & OUT)
0012                    FDCSEC  EQU     FDCPORT+2       ;FDC sector register port (IN & OUT)
0013                    FDCDAT  EQU     FDCPORT+3       ;FDC data register port (IN & OUT)
                        ;
0014                    FDCTLI  EQU     FDCPORT+4       ;Controller board input port
0014                    FDCTLO  EQU     FDCPORT+4       ;Controller board output port
                        ;
0001                    DSLBIT  EQU     00000001B       ;Drive select: 0 - drive A, 1 - drive B
0002                    SSLBIT  EQU     00000010B       ;Side select: 0 - side 0, 1 - side 1
0004                    MONBIT  EQU     00000100B       ;Motor on: 1 - turns drive motor on
0008                    MRYBIT  EQU     00001000B       ;Motor ready: 1 - drive motor ready
0010                    DENBIT  EQU     00010000B       ;Density: 0 - FM, 1 - MFM
```

Control out byte

```
                        ;
0001                    HLDBIT  EQU     00000001B       ;Head load: 1 - head load on drive
0002                    DSDBIT  EQU     00000010B       ;Double-sided: 1 if drive double-sided
0004                    TPIBIT  EQU     00000100B       ;TPI: 0 - 48 TPI drive, 1 - 96 TPI drive
0008                    STPBIT  EQU     00001000B       ;Track stepping rate: 0 - 12 ms, 1 - 6 ms
0010                    NODBIT  EQU     00010000B       ;No. of drives: 0 - 1 drive, 1 - 2 drives
0020                    RDYBIT  EQU     00100000B       ;Ready: 1 - drive ready
0040                    INTBIT  EQU     01000000B       ;Interrupt: 1 - FDC interrupt request
0080                    DRQBIT  EQU     10000000B       ;Data request: 1 - FDC data request
                        ;
0001                    BUSYBIT EQU     00000001B
                        ;
                        ;
                        ;
                        ;
                        ;Initialisation routine.
                        ;
0000' AF                INITLZ: XOR A                   ;Zero control byte but
0001' 06 FE                     LD B,11111110B          ;leave drive select unchanged
0003' CD 03DE'                  CALL REPLACE            ;Update status byte
0006' 3E D0                     LD A,11010000B          ;'Force interrupt'
0008' D3 10                     OUT (FDCCOM),A          ;Terminate any FDC commands
000A' CD 03EE'                  CALL DELAY1
                        ;
                        ;Zero (SWUF). This indicates not waiting for any data.
                        ;
000D' AF                        XOR A
000E' 21 0000*                  LD HL,SWUF
0011' 77                        LD (HL),A
0012' 3D                        DEC A                   ;A = 255
0013' 06 06                     LD B,6
0015' 23                INILP:  INC HL
0016' 77                        LD (HL),A
0017' 10 FC                     DJNZ INILP
0019' C9                        RET



                        ;
                        ;
                        ;
                        ;???
                        ;
001A' C5                EXCNFG: PUSH BC
001B' D5                        PUSH DE
001C' 3A 0000*                  LD A,(CFGBYT)
001F' FE FF                     CP 255
0021' 28 07                     JR Z,CFGERR             ;INTERROGATE CFGTAB
0023' 3A 0000*                  LD A,(DRVRQ)
0026' E6 04                     AND 100B
0028' 28 0C                     JR Z,CFGSM
                        ;
002A' CD 0073'          CFGERR: CALL CFGCHK             ;REPLACE ERRONEOUS CONFIG
002D' 21 0000                   LD HL,0
0030' D1                        POP DE
0031' C1                        POP BC
```

```
0032'  3E 01                    LD A,1
0034'  A7                       AND A                   ;RESET ZERO FL
0035'  C9                       RET
                        ;
0036'  CD 0046'         CFGSM:  CALL DPBGET
0039'  7C                       LD A,H
003A'  B5                       OR L
003B'  28 ED                    JR Z,CFGERR             ;
003D'  E5                       PUSH HL
003E'  CD 0062'                 CALL CTUPD
0041'  E1                       POP HL
0042'  D1                       POP DE
0043'  C1                       POP BC
0044'  AF                       XOR A
0045'  C9                       RET
                        ;
                        ;DPBGET searches through PBASE for a match against CFGBYT
                        ;If a Match is not found then on Exit HL=0 Otherwise
                        ;HL<>0
                        ;
0046'  3A 0000*         DPBGET: LD A,(CFGBYT)
0049'  47                       LD B,A
004A'  FE FF                    CP 255                  ;Undefined
004C'  21 0000          NODPB:  LD HL,0
004F'  C8                       RET Z                   ;Z If Unconfigured
0050'  21 0000*                 LD HL,PBASE
0053'  11 000F                  LD DE,15
0056'  7E               DPBLP:  LD A,(HL)               ;Get configuration from table
0057'  FE FF                    CP 255                  ;Z If Unconfigured
0059'  CA 004C'                 JP Z,NODPB
005C'  B8                       CP B                    ;Z If same as before
005D'  23                       INC HL
005E'  C8                       RET Z
005F'  19                       ADD HL,DE               ;If not same then get next Item in table
0060'  18 F4                    JR DPBLP                ;Loop back and test again
                        ;
                        ;???
                        ;
0062'  3A 0000*         CTUPD:  LD A,(DRVRQ)
0065'  E6 07                    AND 111B
0067'  21 0000*                 LD HL,CFGTAB
006A'  5F                       LD E,A
006B'  16 00                    LD D,0
006D'  19                       ADD HL,DE
006E'  3A 0000*                 LD A,(CFGBYT)
0071'  77                       LD (HL),A
0072'  C9                       RET
                        ;
                        ;
                        ;
                        ;???
                        ;
                        ;CFGCHK stores configure byte for drive given by (DRVRQ)
                        ;at (CFGBYT). Returns Z if drive not configured.
                        ;
0073'  21 0000*         CFGCHK: LD HL,CFGTAB            ;HL -> configure table
```

```
0076'  3A 0000*                LD A,(DRVRQ)          ;A = drive to select
0079'  E6 07                   AND 111B              ;Consider only drive numbers 0-7
007B'  5F                      LD E,A
007C'  16 00                   LD D,0                ;DE = drive number
007E'  19                      ADD HL,DE             ;HL -> configure byte
007F'  7E                      LD A,(HL)
0080'  32 0000*                LD (CFGBYT),A         ;Store configure byte
0083'  FE FF                   CP 255
0085'  C9                      RET
                       ;
                       ;
                       ;
                       ;EXRD is the sector read routine called by the BIOS.
                       ;
0086'  C5              EXRD:   PUSH BC
0087'  D5                      PUSH DE               ;Save registers
0088'  E5                      PUSH HL
                       ;
0089'  CD 0073'                CALL CFGCHK
008C'  28 2B                   JR Z,EEXIT            ;Quit if drive not configured
                       ;
008E'  21 00BB'                LD HL,EXIT2
0091'  E5                      PUSH HL               ;Push return address
0092'  3A 0000*                LD A,(DRVRQ)
0095'  E6 04                   AND 100B              ;NZ if drive number > 3
0097'  C0                      RET NZ                ;Return if drive number out of range
0098'  C3 00FE'                JP RDSM               ;Jump to intermediate read routine
                       ;
                       ;
                       ;
                       ;EXWR is the sector write routine called by the BIOS.
                       ;
                       ;EXWR:  PUSH BC
                       ;       PUSH DE               ;Save registers
                       ;       PUSH HL
                       ;
                       ;       CALL CFGCHK
                       ;       JR Z,EEXIT            ;Quit if drive not configured
                       ;
                       ;       LD HL,EXIT2
                       ;       PUSH HL               ;Push return address
                       ;       LD A,(DRVRQ)
                       ;       AND 100B              ;NZ if drive number > 3
                       ;       RET NZ                ;Return if drive number out of range
                       ;
                       ;       PUSH BC               ;Save C = type of sector write
                       ;       CALL WRSM             ;Call intermediate write routine
                       ;       POP BC                ;Restore C
                       ;       RET NZ                ;Return if write unsuccessful
                       ;
                       ;Test whether C = 1 (directory write).
                       ;If so, write sector immediately.
                       ;
                       ;       DEC C
                       ;       RET NZ
                       ;       JP WTIDY              ;Jump if directory write
```

```
                        .8080
009B'  C5               EXWR:   PUSH    B
009C'  D5                       PUSH    D
009D'  E5                       PUSH    H
                        ;
009E'  CD 0073'                 CALL    CFGCHK
00A1'  CA 00B9'                 JZ      EEXIT
00A4'  21 00BB'                 LXI     H,EXIT
00A7'  E5                       PUSH    H               ; SETUP RETURN ADDR
00A8'  3A 0000*                 LDA     DRVRQ
00AB'  E6 04                    ANI     100B
00AD'  C0                       RNZ
00AE'  C5                       PUSH    B
00AF'  CD 012B'                 CALL    WRSM
00B2'  C1                       POP     B
00B3'  C0                       RNZ
00B4'  0D                       DCR     C
00B5'  CA 01CA'                 JZ      WTIDY           ; DIR UPDATE, SO DO IT NOW.
00B8'  C9                       RET
                        .Z80
                        ;
                        ;
                        ;
                        ;EEXIT is jumped to by EXRD and EXWR if drive is not configured.
                        ;
00B9'  3E 07            EEXIT:  LD A,7
                        ;
                        ;
                        ;Fall through to EXIT2.
                        ;
                        ;
                        ;EXIT2 is the exit point for EXRD and EXWR.
                        ;Returns A = 0, Z if operation successful.
                        ;
00BB'                   EXIT:
00BB'  F5               EXIT2:  PUSH AF                 ;Save error code
00BC'  AF                       XOR A
00BD'  06 04                    LD B,MONBIT             ;Disable motor on
00BF'  CD 03DE'                 CALL REPLACE            ;Update status
00C2'  F1                       POP AF                  ;Restore code
00C3'  A7                       AND A
00C4'  E1                       POP HL
00C5'  D1                       POP DE                  ;Restore registers
00C6'  C1                       POP BC
00C7'  C9                       RET
                        ;
                        ;
                        ;
                        ;BLKRD is the block sector read routine.
                        ;On entry, BC = number of sectors to read.
                        ;On exit, A = 0, Z if read successful, A = 1, NZ if read error.
                        ;
00C8'  C5               BLKRD:  PUSH BC
00C9'  06 0A                    LD B,10                 ;Allow up to 10 re-tries for each sector
                        ;
```

```
00CB'  05                BLKLP:  DEC B
00CC'  28 21                     JR Z,BLKRE              ;Jump if no more re-tries
                         ;
00CE'  CD 0086'                  CALL EXRD               ;Read sector
00D1'  A7                        AND A
00D2'  20 F7                     JR NZ,BLKLP             ;Jump if read error
                         ;
                         ;Here if last sector read successfully.
                         ;
00D4'  2A 0000*                  LD HL,(DMARQ)
00D7'  11 0080                   LD DE,128
00DA'  19                        ADD HL,DE
00DB'  22 0000*                  LD (DMARQ),HL           ;Store new DMA address
                         ;
00DE'  3A 0000*                  LD A,(SECRQ)
00E1'  3C                        INC A                   ;Increment sector number
00E2'  FE 1B                     CP 27
00E4'  CC 00F4'                  CALL Z,NTRK             ;Select next track if sector number > 26
00E7'  32 0000*                  LD (SECRQ),A            ;Store new sector number
                         ;
                         ;Test whether any more sectors to read.
                         ;
00EA'  C1                        POP BC
00EB'  10 DB                     DJNZ BLKRD              ;Jump if more sectors
                         ;
                         ;Here if block read successful.
                         ;
00ED'  AF                        XOR A
00EE'  C9                        RET
                         ;
                         ;Here if block read unsuccessful.
                         ;
00EF'  C1                BLKRE:  POP BC
00F0'  3E 01                     LD A,1
00F2'  A7                        AND A
00F3'  C9                        RET
                         ;
                         ;
                         ;
                         ;NTRK increments track number.
                         ;On exit, A = sector number = 1.
                         ;
00F4'  3A 0000*          NTRK:   LD A,(TRKRQ)
00F7'  3C                        INC A
00F8'  32 0000*                  LD (TRKRQ),A
00FB'  3E 01                     LD A,1
00FD'  C9                        RET
                         ;
                         ;
                         ;
                         ;               ****************************************
                         ;               *                                      *
                         ;               *       INTERMEDIATE DISC ROUTINES     *
                         ;               *                                      *
                         ;               ****************************************
                         ;
```

```
                        ;
                        ;
                        ;RDSM is the intermediate sector read routine.
                        ;
00FE'  CD 01CA'         RDSM:   CALL WTIDY              ;Flush write buffers
0101'  C0               RET NZ                          ;Return if error
                        ;
0102'  CD 0215'         CALL DDD                        ;NZ if D/D format
0105'  20 03            JR NZ,DRDSM
0107'  C3 0221'         JP READSM                       ;Jump to read sector routine
                        ;
                        ;
                        ;
                        ;DRDSM is the double-density intermediate sector read routine.
                        ;
010A'  CD 01A2'         DRDSM:  CALL NADITS
010D'  C4 0186'         CALL NZ,PREAD
0110'  C0               RET NZ
                        ;
0111'  CD 01BE'         CALL TX1
0114'  2A 0000*         LD HL,(DMARQ)                   ;HL = DMA address
0117'  EB               EX DE,HL                        ;DE = DMA address
0118'  CD 0198'         CALL QSIDE                      ;A = 1, NZ if sector even (?)
011B'  21 0000*         LD HL,DBUF
011E'  D5               PUSH DE
011F'  5F               LD E,A
0120'  16 00            LD D,0
0122'  19               ADD HL,DE
0123'  D1               POP DE
0124'  01 0080          LD BC,128
0127'  ED B0            LDIR
0129'  AF               XOR A
012A'  C9               RET
                        ;
                        ;
                        ;
                        ;WRSM is the intermediate sector write routine.
                        ;
                        .8080
012B'  CD 0215'         WRSM:   CALL    DDD
012E'  C2 0139'                 JNZ     DWRIT
0131'  CD 01CA'                 CALL    WTIDY
0134'  C0                       RNZ
0135'  CD 0257'                 CALL    WRITSM
0138'  C9                       RET
0139'  CD 0198'         DWRIT:  CALL    QSIDE
013C'  C2 0158'                 JNZ     DWR2            ; EVEN
013F'  CD 01CA'                 CALL    WTIDY
0142'  C0                       RNZ
0143'  2A 0000*                 LHLD    DMARQ
                        ;       CALL    DMACHK
0146'  11 0000*                 LXI     D,DBUF
0149'  01 0080                  LXI     B,128
014C'  B0ED                     DW      LDIR
014E'  CD 01BE'                 CALL    TX1
0151'  3E FF                    MVI     A,255
```

```
0153'  32 0000*                STA     SWUF
0156'  3C                      INR     A
0157'  C9                      RET
0158'  CD 01A2'          DWR2: CALL    NADITS
015B'  CA 0169'                JZ      DWR3
015E'  CD 01CA'                CALL    WTIDY
0161'  C0                      RNZ
0162'  CD 0186'                CALL    PREAD
0165'  C0                      RNZ
0166'  CD 01BE'                CALL    TX1
0169'  2A 0000*          DWR3: LHLD    DMARQ
                         ;     CALL    DMACHK
016C'  11 0080*                LXI     D,DBUF+128
016F'  01 0080                 LXI     B,128
0172'  E5                      PUSH    H
0173'  B0ED                    DW      LDIR
0175'  21 0000*                LXI     H,DBUF
0178'  22 0000*                SHLD    DMARQ
017B'  CD 0257'                CALL    WRITSM
017E'  E1                      POP     H
017F'  22 0000*                SHLD    DMARQ
0182'  32 0000*                STA     SWUF
0185'  C9                      RET
                        .Z80
                        ;WRSM:  CALL DDD
                        ;       JR NZ,DWRIT             ;Jump if D/D format
                        ;
                        ;       CALL WTIDY              ;Flush write buffers
                        ;       RET NZ                  ;Return if error
                        ;       JP WRITSM               ;Jump to sector write routine
                        ;
                        ;
                        ;DWRIT is the double-density intermediate sector write routine.
                        ;
                        ;DWRIT: CALL QSIDE
                        ;       JR NZ,DWR2              ;Jump if sector number even
                        ;
                        ;       CALL WTIDY              ;Flush write buffers
                        ;       RET NZ                  ;Return if error
                        ;
                        ;       LD HL,(DMARQ)
                        ;       LD DE,DBUF
                        ;       LD BC,128
                        ;       LDIR                    ;Copy record into DBUF
                        ;
                        ;       CALL TX1
                        ;       LD A,255
                        ;       LD (SWUF),A
                        ;       INC A
                        ;       RET
                        ;
                        ;DWR2:  CALL NADITS
                        ;       JR Z,DWR3
                        ;
                        ;       CALL WTIDY              ;Flush write buffers
```

```
                        ;       RET NZ                  ;Return if error
                        ;
                        ;       CALL PREAD
                        ;       RET NZ
                        ;       CALL TXI
                        ;;
                        ;DWR3:  LD HL,(DMARQ)           ;HL = DMA address
                        ;       LD DE,DBUF+128
                        ;       LD BC,128
                        ;       PUSH HL
                        ;       LDIR                    ;Copy record to DBUF + 128
                        ;       LD HL,DBUF
                        ;       LD (DMARQ),HL
                        ;       CALL WRITSM
                        ;       POP HL
                        ;       LD (DMARQ),HL
                        ;       LD (SWUF),A
                        ;       RET
                        ;
                        ;
                        .8080
0186' 2A 0000*          PREAD:  LHLD    DMARQ
0189' E5                        PUSH    H
018A' 21 0000*                  LXI     H,DBUF
018D' 22 0000*                  SHLD    DMARQ
0190' CD 0221'                  CALL    READSM
0193' E1                        POP     H
0194' 22 0000*                  SHLD    DMARQ
0197' C9                        RET

0198' 3A 0000*          QSIDE:  LDA     SECRQ
019B' 1F                        RAR
019C' 3F                        CMC
019D' 3E 00                     MVI     A,0
019F' 1F                        RAR
01A0' B7                        ORA     A
01A1' C9                        RET

01A2' 21 0000*          NADITS: LXI     H,DRVRQ         ; RQID
01A5' 11 0000*                  LXI     D,BFID
01A8' 06 04                     MVI     B,4
01AA' 1A                NADLP:  LDAX    D
01AB' BE                        CMP     M
01AC' C0                        RNZ
01AD' 23                        INX     H
01AE' 13                        INX     D
01AF' 05                        DCR     B
01B0' C2 01AA'                  JNZ     NADLP
01B3' 1A                        LDAX    D
01B4' 3D                        DCR     A
01B5' E6 FE                     ANI     11111110B
01B7' 47                        MOV     B,A
01B8' 7E                        MOV     A,M
01B9' 3D                        DCR     A
01BA' E6 FE                     ANI     11111110B
01BC' B8                        CMP     B
```

```
01BD'  C9                        RET

01BE'  21 0000*          TX1:    LXI     H,DRVRQ         ; RQID
01C1'  11 0000*                  LXI     D,BFID
01C4'  01 0006                   LXI     B,6
01C7'  B0ED                      DW      LDIR
01C9'  C9                        RET

01CA'  3A 0000*          WTIDY:  LDA     SWUF
01CD'  B7                        ORA     A
01CE'  C8                        RZ
01CF'  CD 0201'                  CALL    SWAP
01D2'  11 0000*                  LXI     D,TDBUF
01D5'  21 0000*                  LXI     H,DBUF
01D8'  01 0080                   LXI     B,128
01DB'  B0ED                      DW      LDIR
01DD'  CD 0186'                  CALL    PREAD
01E0'  11 0000*                  LXI     D,DBUF
01E3'  21 0000*                  LXI     H,TDBUF
01E6'  01 0080                   LXI     B,128
01E9'  B0ED                      DW      LDIR
01EB'  B7                        ORA     A
01EC'  C2 01FB'                  JNZ     WTY1
01EF'  21 0000*                  LXI     H,DBUF
01F2'  22 0000*                  SHLD    DMARQ
01F5'  CD 0257'                  CALL    WRITSM
01F8'  32 0000*                  STA     SWUF
01FB'  F5                WTY1:   PUSH    PSW
01FC'  CD 0201'                  CALL    SWAP
01FF'  F1                        POP     PSW
0200'  C9                        RET

0201'  21 0000*          SWAP:   LXI     H,BFID
0204'  11 0000*                  LXI     D,DRVRQ         ; RQID
0207'  06 08                     MVI     B,8
0209'  4E                SWP1:   MOV     C,M
020A'  1A                        LDAX    D
020B'  77                        MOV     M,A
020C'  79                        MOV     A,C
020D'  12                        STAX    D
020E'  13                        INX     D
020F'  23                        INX     H
0210'  05                        DCR     B
0211'  C2 0209'                  JNZ     SWP1
0214'  C9                        RET

0215'  3A 0000*          DDD:    LDA     CFGBYT
0218'  E6 02                     ANI     010B
021A'  C8                        RZ
021B'  CD 03CB'                  CALL    D1213
021E'  C0                        RNZ
021F'  B7                        ORA     A
0220'  C9                        RET
                         .Z80
                         ;
                         ; -----------------------------------------
```

```
                        ;
                        ;
                        ;
                        ;HARDWARE INTERFACE
                        ;
                        ;
                        ;
                        ;READSM reads sector. Drive. side, track, sector
                        ;and DMA address already specified.
                        ;Returns Z if read successful, else NZ.
                        ;
0221'  CD 0283'         READSM: CALL RWGO
0224'  C0                       RET NZ                  ;Return if error
                        ;
                        ;
0225'  3A 0000*                 LD A,(EFLAG)            ;A = E flag for read or write command
0228'  F6 80                    OR 10000000B            ;Read command
022A'  D3 10                    OUT (FDCCOM),A          ;Issue command
                        ;
022C'  CD 0240'                 CALL DISCRD             ;Read bytes from disc
                        ;
022F'  DB 10                    IN A,(FDCSTA)           ;A = FDC status register
0231'  E6 9C                    AND 10011100B           ;Possible error bits
0233'  C8                       RET Z                   ;Return if no error
                        ;
                        ;Here if error in reading or writing a sector.
                        ;
0234'  32 0000*         RWEF:   LD (CFGBYT),A           ;Store error byte
0237'  E6 10                    AND 00010000B           ;NZ if record not found
0239'  C4 029B'                 CALL NZ,RCBRQ           ;Re-calibrate if record not found
023C'  3E 06                    LD A,6
023E'  A7                       AND A
023F'  C9                       RET
                        ;
                        ;
                        ;
                        ;DISCRD reads bytes from disc, and stores them at address given by (DMARQ).
                        ;
0240'  F3               DISCRD: DI                      ;Ensure no interruptions
0241'  2A 0000*                 LD HL,(DMARQ)           ;HL -> destination address for bytes read
0244'  0E 13                    LD C,FDCDAT             ;C = FDC data register port
                        ;
                        ;Main loop for reading bytes from disc.
                        ;Time taken to read each byte = 73 T-states.
                        ;
0246'  DB 14            DISCR1: IN A,(FDCTLI)           ;11. A = control input byte
0248'  E6 C0                    AND INTBIT+DRQBIT       ;7. NZ if interrupt or data request
024A'  28 FA                    JR Z,DISCR1             ;7/12. Jump if no request
                        ;
                        ;Here if data byte read or command finished.
                        ;
024C'  CB 77                    BIT 6,A                 ;8. NZ if command finished
024E'  20 05                    JR NZ,DISCR2            ;7/12. Jump if command finished
                        ;
                        ;Here if data byte in FDC data register.
                        ;
```

```
0250'  ED A2                    INI                     ;16. Store byte and increment pointer
0252'  C3 0246'                 JP DISCR1               ;10. Get next byte
                        ;
                        ;Here if read command finished.
                        ;
0255'  FB               DISCR2: EI
0256'  C9                       RET
                        ;
                        ;
                        ;
                        ;WRITSM writes sector. Drive, side, track, sector
                        ;and DMA address already specified.
                        ;Returns Z if read successful, else NZ.
                        ;
0257'  CD 0283'         WRITSM: CALL RWGO
025A'  C0                       RET NZ                  ;Return if error
                        ;
                        ;
025B'  3A 0000*                 LD A,(EFLAG)            ;A = E flag for read or write command
025E'  F6 A0                    OR 10100000B            ;Write command
0260'  D3 10                    OUT (FDCCOM),A          ;Issue command
                        ;
0262'  CD 026C'                 CALL DISCWR             ;Write bytes to disc
                        ;
0265'  DB 10                    IN A,(FDCSTA)           ;A = FDC status register
0267'  E6 FC                    AND 11111100B           ;Possible error bits
0269'  20 C9                    JR NZ,RWEF              ;Jump if error
026B'  C9                       RET
                        ;
                        ;
                        ;
                        ;DISCWR writes bytes to disc, from address given by (DMARQ).
                        ;
026C'  F3               DISCWR: DI                      ;Ensure no interruptions
026D'  2A 0000*                 LD HL,(DMARQ)           ;HL -> start address for bytes to write
0270'  0E 13                    LD C,FDCDAT             ;C = FDC data register port
                        ;
                        ;Main loop for writing bytes from disc.
                        ;Time taken to write each byte = 73 T-states.
                        ;
0272'  DB 14            DISCW1: IN A,(FDCTLI)           ;11. A = control input byte
0274'  E6 C0                    AND INTBIT+DRQBIT       ;7. NZ if interrupt or data request
0276'  28 FA                    JR Z,DISCW1             ;7/12. Jump if no request
                        ;
                        ;Here if data byte needed or command finished.
                        ;
0278'  CB 77                    BIT 6,A                 ;8. NZ if command finished
027A'  20 05                    JR NZ,DISCW2            ;7/12. Jump if command finished
                        ;
                        ;Here if data byte needed for FDC data register.
                        ;
027C'  ED A3                    OUTI                    ;16. Output byte and increment pointer
027E'  C3 0272'                 JP DISCW1               ;10. Get next byte
                        ;
                        ;Here if write command finished.
                        ;
```

```
0281'  FB                  DISCW2: EI
0282'  C9                          RET
                           ;
                           ;
                           ;RWGO is called by the READSM and WRITSM routines.
                           ;Returns Z if successful, else NZ.
                           ;
0283'  CD 02A1'            RWGO:   CALL DRVSET             ;Select drive given by (DRVRQ)
0286'  C0                          RET NZ                  ;Return if drive cannot be selected
0287'  CD 040D'                   CALL WAIT2              ;NZ if drive not ready
028A'  C0                          RET NZ
                           ;
028B'  DB 11                       IN A,(FDCTRK)           ;A = contents of FDC track register
028D'  2A 0000*                    LD HL,(PTRKP)
0290'  77                          LD (HL),A               ;Store old track number
0291'  3A 0000*                    LD A,(LCA)
0294'  D3 11                       OUT (FDCTRK),A          ;Output new track number
                           ;
                           ;
0296'  DB 14                       IN A,(FDCTLI)           ;A = input control byte
0298'  E6 80                       AND DRQBIT              ;NZ if DR full (read) or DR empty (write)
029A'  C9                          RET
                           ;
                           ;
                           ;
                           ;RCBRQ
                           ;
029B'  2A 0000*            RCBRQ:  LD HL,(PTRKP)
029E'  36 FF                       LD (HL),255
02A0'  C9                          RET
                           ;
                           ;
                           ;
                           ;DRVSET selects drive given by (DRVRQ).
                           ;Returns Z if select successful, else NZ.
                           ;
02A1'  CD 03F5'            DRVSET: CALL WAIT               ;Wait until FDC not busy
                           ;
02A4'  AF                          XOR A
02A5'  32 0000*                    LD (EFLAG),A            ;Zero E flag
02A8'  3A 0000*                    LD A,(DRVRQ)            ;A = drive number to select
02AB'  F6 0C                       OR MONBIT+MRYBIT
02AD'  06 0D                       LD B,DSLBIT+MONBIT+MRYBIT       ;Drive select, drive enable
02AF'  CD 03DE'                    CALL REPLACE            ;Update status
                           ;
02B2'  3A 0000*                    LD A,(DRVRQ)            ;A = drive number to select
02B5'  47                          LD B,A
02B6'  3A 0000*                    LD A,(CURDRV)           ;A = current drive
02B9'  FE FF                       CP 255
02BB'  28 0E                       JR Z,SKIP1              ;Jump if no drive selected
                           ;
02BD'  B8                          CP B
02BE'  28 0B                       JR Z,SKIP1              ;Jump if selecting current drive
                           ;
                           ;Here if drive change required.
                           ;
```

```
02C0'  DB 11                    IN A,(FDCTRK)          ;A = current track number
02C2'  D3 13                    OUT (FDCDAT),A         ;Load track number into DR
02C4'  3E 10                    LD A,00010000B         ;Seek current track with head unloaded
02C6'  D3 10                    OUT (FDCCOM),A         ;Issue command ('Unload head')
                        ;
02C8'  CD 03FF'                 CALL WAIT1             ;Wait until FDC has finished command
                        ;
02CB'  78               SKIP1:  LD A,B
02CC'  32 0000*                 LD (CURDRV),A          ;Store new drive number
02CF'  5F                       LD E,A
02D0'  16 00                    LD D,0
02D2'  21 0000*                 LD HL,TRACKS
02D5'  19                       ADD HL,DE              ;HL -> track variable for this drive
02D6'  22 0000*                 LD (PTRKP),HL          ;Store address of track variable
02D9'  7E                       LD A,(HL)              ;A = track required for this drive
02DA'  D3 11                    OUT (FDCTRK),A         ;Load FDC track register
                        ;
                        ;Test whether drive is double-sided.
                        ;
02DC'  3A 0000*                 LD A,(CFGBYT)          ;A = configure byte
02DF'  E6 01                    AND 01B                ;NZ if drive configured as D/S
02E1'  28 09                    JR Z,SKIP2             ;Jump if drive configured S/S
                        ;
02E3'  DB 14                    IN A,(FDCTLI)          ;A = input control byte
02E5'  EE 0F                    XOR 0FH                ;INVERT SWITCHES
02E7'  E6 02                    AND DSDBIT             ;NZ if drive D/S
02E9'  CA 0409'                 JP Z,DRVSE5            ;Jump if drive select error
                        ;
                        ;Test whether drive is 96 TPI.
                        ;
02EC'  3A 0000*         SKIP2:  LD A,(CFGBYT)          ;A = configure byte
02EF'  E6 04                    AND 0100B              ;NZ if drive configured 96 TPI
02F1'  28 09                    JR Z,SKIP3             ;Jump if drive configured 48 TPI
                        ;
02F3'  DB 14                    IN A,(FDCTLI)          ;A = input control byte
02F5'  EE 0F                    XOR 0FH                ;INVERT SWITCHES
02F7'  E6 04                    AND TPIBIT             ;NZ if drive 96 TPI
02F9'  CA 0409'                 JP Z,DRVSE5            ;Jump if drive select error
                        ;
02FC'  CD 0215'         SKIP3:  CALL DDD               ;NZ if drive configured D/D
02FF'  3E 00                    LD A,0
0301'  28 01                    JR Z,SKIP4             ;Jump if drive configured S/D
0303'  3D                       DEC A                  ;A = 255
                        ;
0304'  06 10            SKIP4:  LD B,DENBIT            ;Select single or double density
0306'  CD 03DE'                 CALL REPLACE           ;Update status
                        ;
0309'  DB 11                    IN A,(FDCTRK)
030B'  FE FF                    CP 255
030D'  20 09                    JR NZ,SKIP6            ;Jump if current track number not 255
                        ;
030F'  CD 035C'                 CALL RECAL3            ;Move disc head to track 00
0312'  C0                       RET NZ                 ;Return if seek error
                        ;
0313'  3E 04                    LD A,0100B
0315'  32 0000*                 LD (EFLAG),A           ;Load 'E' bit flag
```

```
                        ;
0318'  2A 0000*         SKIP6:  LD HL,(TRKRQ)           ;HL = track to select
031B'  3A 0000*                 LD A,(SECRQ)            ;A = sector to select
031E'  5F                       LD E,A
031F'  16 00                    LD D,0                  ;DE = required sector
0321'  3E 1A                    LD A,26
0323'  32 03B0'                 LD (SECMAX+1),A         ;Store maximum sector number
0326'  3A 0000*                 LD A,(CFGBYT)           ;A = configure byte
0329'  4F                       LD C,A
032A'  E6 10                    AND 00010000B           ;NZ if 8" drive
032C'  CC 037C'                 CALL Z,CALC5
032F'  79                       LD A,C
0330'  E6 02                    AND 00000010B           ;NZ if D/D
0332'  C4 03A1'                 CALL NZ,CALCD
0335'  79                       LD A,C
0336'  E6 01                    AND 00000001B           ;CHECK SIDE BIT
0338'  C4 03B9'                 CALL NZ,CALCS
033B'  7D                       LD A,L
033C'  32 0000*                 LD (LCA),A              ;SETUP LCA
                        ;
                        ;       Deleted 96 TPI Check Here
                        ;
033F'  7A               SKIPC:  LD A,D                  ;SIDE IN A
0340'  5D                       LD D,L                  ;CYL  IN D
                                                        ;SEC  IN E
0341'  0F                       RRCA
0342'  06 02                    LD B,SSLBIT             ;Select side
0344'  CD 03DE'                 CALL REPLACE            ;Update status
                        ;
0347'  7B                       LD A,E
0348'  D3 12                    OUT (FDCSEC),A          ;Load FDC sector register
                        ;
                        ;       LD HL,(DMARQ)           ;HL = DMA address
                        ;       LD A,L
                        ;       OUT (DMALO),A           ;Set low (DMA address)
                        ;       LD A,H
                        ;       OUT (DMAHI),A           ;Set high (DMA address)
                        ;
034A'  DB 11                    IN A,(FDCTRK)           ;A = FDC track register
034C'  BA                       CP D
034D'  3E 00                    LD A,0
034F'  C8                       RET Z                   ;Return if head at desired track
                        ;
0350'  3E 04                    LD A,0100B
0352'  32 0000*                 LD (EFLAG),A            ;Load E bit flag
                        ;
0355'  7A                       LD A,D
0356'  D3 13                    OUT (FDCDAT),A          ;Load FDC DR with desired track
                        ;
                        ;
                        ;
                        ;SEEK moves disc head to track given by FDC track register.
                        ;Returns A = 0, Z if seek successful.
                        ;
0358'  3E 18            SEEK:   LD A,00011000B          ;Seek command, head loaded
035A'  18 02                    JR SKTRK
```

```
                        ;
                        ;
                        ;
                        ;RECALB moves disc head to track 00.
                        ;Returns A = 0, Z if seek track 00 successful.
                        ;
035C'  3E 08            RECALB: LD A,00001000B          ;Restore command, head loaded
                        ;
035E'  47               SKTRK:  LD B,A                  ;B = command byte
035F'  DB 14                    IN A,(FDCTLI)           ;A = input status byte
0361'  EE 0F                    XOR 0FH                 ;INVERT SWITCHES
0363'  E6 08                    AND STPBIT              ;A = track stepping rate
0365'  3E 00                    LD A,0                  ;0 = 6mS
0367'  20 02                    JR NZ,SKTR1             ;Z If Step '20 ms'
0369'  3E 02                    LD A,00000010B          ;10 = 20mS
036B'  B0               SKTR1:  OR B                    ;A = seek command byte
036C'  D3 10                    OUT (FDCSTA),A          ;Issue command
                        ;
036E'  CD 03FF'                 CALL WAIT1              ;Wait until command has finished
                        ;
0371'  CD 03EE'                 CALL DELAY1             ;Must wait 54 microseconds
0374'  DB 10                    IN A,(FDCSTA)           ;A = FDC status byte
0376'  E6 10                    AND 00010000B           ;NZ if seek error
0378'  C8                       RET Z
0379'  3E 04                    LD A,4                  ;KQ1
037B'  C9                       RET
                        ;
                        ;
                        ;
037C'  D5               CALC5:  PUSH DE
037D'  29                       ADD HL,HL               ;*2
037E'  E5                       PUSH HL
037F'  29                       ADD HL,HL
0380'  29                       ADD HL,HL               ;*8
0381'  E5                       PUSH HL
0382'  29                       ADD HL,HL               ;*16
0383'  D1                       POP DE
0384'  19                       ADD HL,DE               ;*24
0385'  D1                       POP DE
0386'  19                       ADD HL,DE               ;*26
0387'  D1                       POP DE
0388'  19                       ADD HL,DE
0389'  2B                       DEC HL                  ;SECTOR OFFSET
038A'  7D                       LD A,L
038B'  E6 0F                    AND 1111B
038D'  3C                       INC A                   ;SECTOR OFFSET
038E'  5F                       LD E,A                  ;NEW SECTOR
038F'  CD 03C2'                 CALL CALDIV
0392'  CD 03C2'                 CALL CALDIV
0395'  CD 03C2'                 CALL CALDIV
0398'  CD 03C2'                 CALL CALDIV             ;HL = HL/16, NEW CYL
039B'  3E 10                    LD A,16
039D'  32 03B0'                 LD (SECMAX+1),A
03A0'  C9                       RET
                        ;
```

```
                                ;
                                ;
03A1' CD 03CB'          CALCD:  CALL D1213
03A4' 20 03                     JR NZ,CALD0
03A6' A7                        AND A
03A7' C8                        RET Z
03A8' 23                        INC HL
                                ;
03A9' CD 03C2'          CALD0:  CALL CALDIV
03AC' 7B                        LD A,E
03AD' 30 03                     JR NC,CALD1
                                ;
03AF' 3E 10             SECMAX: LD A,16
03B1' 83                        ADD A,E
                                ;
03B2' 3D                CALD1:  DEC A
03B3' 37                        SCF
03B4' 3F                        CCF
03B5' 1F                        RRA
03B6' 5F                        LD E,A
03B7' 1C                        INC E
03B8' C9                        RET
                                ;
                                ;
                                ;
03B9' CD 03C2'          CALCS:  CALL CALDIV
03BC' D0                        RET NC
03BD' 16 05                     LD D,101B
03BF' C9                        RET
                                ;
                                ;
                                ;
03C0' 29                CALCT:  ADD HL,HL
03C1' C9                        RET
                                ;
                                ;
                                ;
03C2' 37                CALDIV: SCF
03C3' 3F                        CCF
03C4' 7C                        LD A,H
03C5' 1F                        RRA
03C6' 67                        LD H,A
03C7' 7D                        LD A,L
03C8' 1F                        RRA
03C9' 6F                        LD L,A
03CA' C9                        RET
                                ;
                                ;
                                ;
03CB' 3A 0000*          D1213:  LD A,(CFGBYT)
03CE' FE 12                     CP 12H
03D0' 28 03                     JR Z,D12131
03D2' FE 13                     CP 13H
03D4' C0                        RET NZ
                                ;
03D5' E5                D12131: PUSH HL
```

```
03D6'  2A 0000*          LD HL,(TRKRQ)
03D9'  7C                LD A,H
03DA'  B5                OR L
03DB'  BF                CP A                    ;SET ZERO
03DC'  E1                POP HL
03DD'  C9                RET
                         ;
                         ;
                         ;
                         ;REPLACE updates hardware status byte.
                         ;On entry, A = new value of status byte,
                         ;B = mask for old status byte.
                         ;
                         ;N.B.   Those bits which are zero in mask
                         ;       will remain unchanged in status byte.
                         ;
03DE'                    REPLACE:
03DE'  A0                        AND B
03DF'  4F                        LD C,A             ;C = masked new value
03E0'  78                        LD A,B
03E1'  2F                        CPL
03E2'  47                        LD B,A             ;B = complemented mask
03E3'  3A 0000*                  LD A,(LSTOUT)      ;Get old value of status byte
03E6'  A0                        AND B
03E7'  B1                        OR C
03E8'  32 0000*                  LD (LSTOUT),A      ;Store new value of status byte
03EB'  D3 14                     OUT (FDCTLO),A     ;Update status byte
03ED'  C9                        RET
                         ;
                         ;
                         ;
03EE'  3E 32             DELAY1: LD A,50
                         ;
03F0'  3D                DELY11: DEC A
03F1'  C2 03F0'                  JP NZ,DELY11
03F4'  C9                        RET
                         ;
                         ;
                         ;
                         ;WAIT calls DELAY1, then waits until FDC is not busy before returning.
                         ;
03F5'  CD 03EE'          WAIT:   CALL DELAY1
03F8'  DB 10                     IN A,(FDCSTA)      ;A = FDC status register
03FA'  E6 01                     AND BUSYBIT        ;NZ if FDC busy (bit 0)
03FC'  20 F7                     JR NZ,WAIT
03FE'  C9                        RET
                         ;
                         ;
                         ;
                         ;WAIT1 calls DELAY1, then waits until FDC has finished command.
                         ;
03FF'  CD 03EE'          WAIT1:  CALL DELAY1
0402'  DB 14                     IN A,(FDCTLI)      ;A = hardware status byte
0404'  E6 40                     AND INTBIT         ;NZ if INTRQ from FDC (bit 4)
0406'  28 F7                     JR Z,WAIT1
0408'  C9                        RET
```

```
                        ;
                        ;
                        ;
                        ;WAIT3: CALL WAIT2
                        ;       RET NZ
                        ;       LD A,(CFGBYT)
                        ;       AND 00010000B           ;Jump if 8" drive
                        ;       IN A,(FDCTLI)           ;A = hardware status byte
                        ;       JR Z,DRVT5
                        ;
                        ;       AND RY5BIT              ;NZ if 5" drive ready (bit 7)
                        ;       JR NZ,SKIP5             ;Jump if 5" drive ready
                        ;
                        ;Here if drive (5" or 8") not ready.
                        ;
0409'  3E 05            DRVSE5: LD A,5
040B'  A7                       AND A                   ;NZ
040C'  C9                       RET
                        ;
                        ;DRVT5: AND RY8BIT              ;NZ if 8" drive ready
                        ;       JR Z,DRVSE5             ;Jump if 8" drive not ready
                        ;
                        ;Here if drive (5" or 8") ready.
                        ;
                        ;SKIP5: XOR A
                        ;       RET
                        ;
                        ;
                        ;
                        ;WAIT2 waits until only one drive ready, or no drives ready.
                        ;
040D'                   WAIT2:
040D'  CD 0433'                 CALL TEST
0410'  C8                       RET Z           ;DRIVE IS READY
                        ;
                        ;HERE IF DRIVE IS NOT RAEDY
                        ;
0411'  06 08                    LD B,MRYBIT
0413'  AF                       XOR A
0414'  CD 03DE'                 CALL REPLACE    ;TURN OFF MOTOR READY

0417'  3E 0C                    LD A,MRYBIT+MONBIT
0419'  47                       LD B,A
041A'  CD 03DE'                 CALL REPLACE    ;ENSURE MOTOR ON & MOTOR READY
                        ;
041D'  CD 0427'                 CALL DELAY2
                        ;
0420'  CD 0433'                 CALL TEST
0423'  C8                       RET Z
0424'  3E 09                    LD A,9
0426'  C9                       RET

0427'  01 0320          DELAY2: LD BC,800
042A'  CD 03EE'         DEL22:  CALL DELAY1
042D'  0B                       DEC BC
042E'  79                       LD A,C
```

```
042F'  B0                      OR B
0430'  C8                      RET Z
0431'  18 F7                   JR DEL22


                               ;;
                               ;
0433'  DB 14                   TEST:   IN A,(FDCTLI)
0435'  CB 6F                           BIT 5,A
0437'  28 02                           JR Z,TEST1
0439'  AF                              XOR A
043A'  C9                              RET
                               ;
043B'  3C                      TEST1:  INC A
043C'  C9                              RET


                               .8080

                               END
```

Macros:

Symbols:

BFID	0202*	BLKLP	00CB'	BLKRD	00C8I'	BLKRE	00EF'
BUSYBI	0001	CALC5	037C'	CALCD	03A1'	CALCS	03B9'
CALCT	03C0'	CALD0	03A9'	CALD1	03B2'	CALDIV	03C2'
CFGBYT	03CC*	CFGCHK	0073'	CFGERR	002A'	CFGSM	0036'
CFGTAB	0074*	CTUPD	0062'	CURDRV	02CD*	D1213	03CB'
D12131	03D5'	DBUF	01F0*	DDD	0215'	DEL22	042A'
DELAY1	03EE'	DELAY2	0427'	DELY11	03F0'	DENBIT	0010
DISCR1	0246'	DISCR2	0255'	DISCRD	0240'	DISCW1	0272'
DISCW2	0281'	DISCWR	026C'	DMARQ	026E*	DPBGET	0046'
DPBLP	0056'	DRDSM	010A'	DRQBIT	0080	DRVRQ	02B3*
DRVSEL	0409'	DRVSET	02A1'	DSDBIT	0002	DSLBIT	0001
DWR2	0158'	DWR3	0169'	DWRIT	0139'	EEXIT	00B9'
EFLAG	0353*	EXCNFG	001AI'	EXIT	00BB'	EXIT2	00BB'
EXRD	0086I'	EXWR	009BI'	FALSE	0000	FDCCOM	0010
FDCDAT	0013	FDCFOR	0010	FDCSEC	0012	FDCSTA	0010
FDCTLI	0014	FDCTLO	0014	FDCTRK	0011	HLDBIT	0001
INILP	0015'	INITLZ	0000I'	INTBIT	0040	LCA	033D*
LDIR	B0ED	LSTOUT	03E9*	MONBIT	0004	MRYBIT	0008
NADITS	01A2'	NADLP	01AA'	NODBIT	0010	NODPB	004C'
NSTK	0000*	NTRK	00F4'	PBASE	0051*	PREAD	0186'
PTRKP	02D7*	QSIDE	0198'	RCBRQ	029B'	RDSM	00FE'
RDYBIT	0020	READSM	0221'	RECALB	035C'	REPLAC	03DE'
RWEF	0234'	RWGO	0283I'	SECMAX	03AF'	SECRQ	031C*
SEEK	0358'	SKEW6	0000*	SKIP1	02CB'	SKIP2	02EC'
SKIP3	02FC'	SKIP4	0304'	SKIP6	0318'	SKIPC	033F'
SKTR1	036B'	SKTRK	035E'	SPEED	0004	SSLBIT	0002
STPBIT	0008	SWAP	0201'	SWP1	0209'	SWUF	01F9*
TDBUF	01E4*	TEST	0433'	TEST1	043B'	TOAM	0000*
TPIBIT	0004	TRACKS	02D3*	TRKRQ	03D7*	TRUE	FFFF
TRUST	0000*	TX1	01BE'	WAIT	03F5'	WAIT1	03FF'
WAIT2	040D'	WRITSM	0257'	WRSM	012B'	WTIDY	01CA'
WTY1	01FB'						

No Fatal error(s)

```
0000'                   CSEG
                        .Z80
                        ;
                        ;
                                EXT             BDOS.PAGE0          BDOS = 6348
                        ;
                        ;
                        ;
D840                    CHNL1           EQU     0D840H              >F03
D86A                    CHNL2           EQU     CHNL1+42            >F28
D894                    CHNL3           EQU     CHNL2+42            >F53
D8BE                    CHNL4           EQU     CHNL3+42            >F78
D8E8                    CHNL5           EQU     CHNL4+42            >FA3
                        ;
D912                    USERSAVE        EQU     CHNL5+42                >FCB
D8F6                    IXTEMP          EQU     CHNL5+14
D8F4                    LINPUT          EQU     CHNL5+12
D894                    FLAG            EQU     CHNL3
8000                    BUFFER          EQU     8000H
                        ;
                        ;
                        ;
0010                    .RADIX 16
                        ;
                        ;
                        ;ROM calls and System varables.
                        ;
FDCC                    ACC1            EQU     0FDCC
C4F4                    ADJVAL          EQU     0C4F4
3C45                    AE              EQU     3C45    ;0
FACC                    ARRTOP          EQU     0FACC
0250                    BASIC2          EQU     0250
09F2                    BREAKMON        EQU     09F2
FA7F                    CALCBOT         EQU     0FA7F
FA81                    CALCST          EQU     0FA81
1B43                    CALLEN          EQU     1B43
FB49                    DESAVE          EQU     0FB49
                        ;EVALAB         EQU     3D84    ;!!
                        ;EVALSE         EQU     3E7E    ;!!
                        ;FIND1$         EQU     3FE9    ;!!!
0689                    FINDJP          EQU     0689
                        ;GETINP         EQU     2927    ;!!!
                        ;GOTMIN1        EQU     20B7    ;!!!
10A6                    GOTZER          EQU     10A6
                        ;INT            EQU     200A    ;!!!
07B8                    JPHL            EQU     07B8
0079                    KBD             EQU     0079
FA83                    KBDBUF          EQU     0FA83
FA7A                    LSTPG           EQU     0FA7A
0224                    NEWINT          EQU     221+3   ;Skips IJINIT
0CAB                    PRINTX          EQU     0CAB
26F7                    RWTAB           EQU     26F7    ;!!!
10A9                    SETZER          EQU     10A9
                        ;SLOAD1         EQU     2AF5    ;!!!
                        ;SRUN           EQU     2CAF    ;!!!
```

```
FA92                    STKLIM          EQU             0FA92
0744                    STACKS          EQU             0744
                        ;STR$           EQU             3FC6    ;!!!
FA94                    SYSTOP          EQU             0FA94
FA89                    USER            EQU             0FA89
FA7B                    VARNAM          EQU             0FA7B
FD65                    VAZERO          EQU             0FD65
FAD2                    PAGE            EQU             0FAD2H
                        ;
000A                    .RADIX 10
                        ;
                        ;
                        ;RST numbers.
                        ;
0000                    PGPORT          EQU             0
0008                    DEHL            EQU             08H
0028                    ERRRST          EQU             28H
0030                    GETRST          EQU             30H
0028                    JT              EQU             28H
0018                    OSPRST          EQU             18H
0010                    SCRRST          EQU             10H
                        ;
                        ;
                        ;Control Characters.
                        ;
000D                    CR              EQU             13
000A                    LF              EQU             10
000C                    FF              EQU             12
                        ;
                        ;
                        ;BDOS call codes.
                        ;
000F                    DOPEN           EQU             15
0010                    DCLOSE          EQU             16
0011                    SEARCH          EQU             17
0012                    NEXT            EQU             18
0013                    DELETEF         EQU             19
0014                    SREAD           EQU             20
0015                    SWRITE          EQU             21
0016                    MAKE            EQU             22
0017                    DOSREN          EQU             23
001A                    DMASET          EQU             26
0021                    RREAD           EQU             33
0022                    RWRITE          EQU             34
0023                    COMPUT          EQU             35
                        ;
                        ;
00D3                    EQ              EQU             0D3H
E680                    DMA             EQU             0E680H
                        ;
                        ;
                        EXT     SFORMAT,STAT,EXWR,BLKRD
                        EXT     DRVRQ,SECRQ,DMARQ,TRKRQ
                        EXT     JPLINK
                        ;
                         PUBLIC GETUFN,NOFILE,SETDMA,GETFNAM
```

DMA = F900 - Indirect via 773C

```
                        ;
                        ;
                        ; returns HL = default buffer + A * 32 +1.
                        ;
7700    0000'           FILEADDR:
0000'   21 E681                 LD HL,DMA+1     { LD HL,(773C)
0003'   11 0020                 LD DE,32        { INC HL
0006'   A7                      AND A
0007'   C8                      RET Z
0008'   47                      LD B,A
0009'   19             FILEA2:  ADD HL,DE
000A'   10 FD                   DJNZ FILEA2
000C'   C9                      RET
                        ;
                        ;
                        ; QUIT either loads and executes NCPM.COM at 8000h. else jumps to BASIC.
                        ;
                        ;
770E    000D'   11 D8BE QUIT::  LD DE,CHNL4     Major Difference
0010'   D5                      PUSH DE
0011'   21 0037'               LD HL,DATA
0014'   01 000B                LD BC,11
0017'   ED B0                  LDIR                    ;Move name into common ram.
0019'   D1                      POP DE
001A'   DD 21 D8E8              LD IX,CHNL5
001E'   CD 0756'                CALL GETFNAM            ;Read in name.
0021'   CD 02EC'                CALL FOPEN
0024'   CA 0259                 JP Z,259H               ;JP to BASICS if NCPM not on disk.
                        ;
0027'   DD 36 27 05   QUIT1:  LD (IX+39),5            ;Read in NCPM and pass control.
002B'   CD 06A1'                CALL STARW
002E'   21 8000                 LD HL,8000H
0031'   CD 0351'                CALL RAWREAD
0034'   C3 8003                 JP 8003H
                        ;
7734    0037'   22 4E 43 50 DATA:   DB      '"NCPM.COM"',0FFH
003B'   4D 2E 43 4F
003F'   4D 22 FF
                        ;
                        ;
                        ;
                        ;               ***********************************
                        ;               *                                 *
                        ;               *       DISC BASIC COMMANDS       *
                        ;               *                                 *
                        ;               ***********************************
                        ;
                        ;
                        ;
                        ; Routine to set up the DMA.
                        ;
773E    0042'   11 E680 SETDMA: LD DE,DMA       LD DE,(773C)
0045'   E5                      PUSH HL
0046'   0E 1A                  LD C,DMASET
0048'   CD 0000*                CALL BDOS
```

```
                                        POP HL
004C'  C9                               RET
                                ;
                                ;
                                ;DIR lists specified file(s) on specified drive.
                                ;Syntax: DIR (drivename:)(filename).
                                ;
004D'  01                               DB 1
                                ;
7748  004E'  CD 0B43'           DIR:    CALL USRCHK
0051'  13                               INC DE
0052'  DD 21 D8E3                      LD IX,CHNL5
                                ;
0056'  1A                               LD A,(DE)               ;Allows USER DIR  with no string
0057'  FE FF                            CP 0FFH
0059'  20 05                            JR NZ,DIR$
005B'  CD 0782'                         CALL GETNUL
005E'  18 03                            JR DIR0
                                ;
0060'  CD 0756'                DIR$:   CALL GETFNAM            ;CHNL5 contains filename
0063'  1B                       DIR0:   DEC DE
0064'  D5                              PUSH DE
0065'  CD 0042'                        CALL SETDMA             ;Preserves HL
0068'  EB                              EX DE,HL                ;DE -> FCB
0069'  0E 11                           LD C,SEARCH
006B'  CD 066B'                        CALL DOS
006E'  FE FF                           CP 0FFH
0070'  28 25                           JR Z,NOFILE
0072'  CD 0000'                DIR1:   CALL FILEADDR           ;HL -> filename
0075'  06 08                           LD B,8
0077'  CD 071D'                        CALL PRNTCHRS
007A'  3E 2E                           LD A,'.'
007C'  CD 0CAB                         CALL PRINTX
007F'  06 03                           LD B,3
0081'  CD 071D'                        CALL PRNTCHRS
0084'  3E 20                    DIRPAD::        LD A,' '
0086'  CD 0CAB                         CALL PRINTX
0089'  0E 12                           LD C,NEXT
008B'  CD 066B'                        CALL DOS
008E'  FE FF                           CP 0FFH
0090'  20 E0                           JR NZ,DIR1
0092'  D7                      DIR2:   RST SCRRST
0093'  2D 0A                           DB 2DH,0AH
0095'  D1                              POP DE
0096'  C9                              RET
                                ;
                                ;
0097'  D7                      NOFILE: RST SCRRST
0098'  87 4E 6F 20                    DB 87H,'No File'
009C'  46 69 6C 65
00A0'  18 F0                           JR DIR2
                                ;
                                ;
                                ;
                                ;ERASE erases file(s) on specified drive.
                                ;Syntax: ERASE (drivename:)(filename).
```

```
                                ;
00A2' C9                        RET
00A3' 01                        DB 1
                                ;
7797 00A4'                      ERASE:
00A4' CD 0B43'                  CALL USRCHK
00A7' 13                        INC DE
00A8' DD 21 D8E8                LD IX,CHNL5
00AC' CD 0756'                  CALL GETFNAM
00AF' 1B                        DEC DE
00B0' D5                        PUSH DE                 ;Save text pointer
00B1' 3E 3F                     LD A,'?'
00B3' CD 08A0'                  CALL TESTCHR
00B6' 20 1B                     JR NZ,ERASE2            ;Jump if file name not all '?'s
                                ;
                                ;Make sure that all files are to be deleted.
                                ;
00B8' D7                        RST SCRRST
00B9' 8A 41 6C 6C               DB 80H+10,'All files?'
00BD' 20 66 69 6C
00C1' 65 73 3F
00C4' CD 0079                   ERASE1: CALL KBD
00C7' 28 FB                     JR Z,ERASE1
00C9' FE 03                     CP 3
00CB' 28 C5                     JR Z,DIR2               ;Quit if BREAK pressed
00CD' CB AF                     RES 5,A                 ;Upper and lower checked.
00CF' FE 59                     CP 'Y'
00D1' 20 BF                     JR NZ,DIR2              ;Jump if 'Y' not pressed
                                ;
                                ;Now erase file(s).
                                ;
00D3' 0E 13                     ERASE2: LD C,DELETE
00D5' CD 066B'                  CALL DOS
                                ;
                                ;File(s) deleted or command abandoned.
                                ;
00D8' 18 B8                     JR DIR2
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
00DA' 21 D8F1                   PUT$:   LD HL,CHNL5+9
00DD' 06 03                     LD B,3
00DF' 3E 24                     PUT$1:  LD A,'$'
00E1' 77                        LD (HL),A
00E2' 23                        INC HL
00E3' 10 FA                     DJNZ PUT$1
00E5' C9                        RET
                                ;
                                ;
                                ;
00E6' D9                        MOVNAM: EXX
00E7' 11 D8F8                   LD DE,CHNL5+16  ;Move name up by 16.
00EA' 21 D8E8                   LD HL,CHNL5
```

```
00ED'  01 000C                    LD BC,12
00F0'  ED B0                      LDIR
00F2'  D9                         EXX
00F3'  C9                         RET
                                  ;
                                  ;
                                  ;
                                  ;
00F4'  11 D912                    SETREG: LD DE,USERSAV
00F7'  21 D8F1                    LD HL,CHNL5+9
00FA'  01 0003                    LD BC,3
00FD'  C9                         RET
                                  ;
                                  ;
00FE'                             LOADTYP:
00FE'  D9                         EXX
00FF'  CD 00F4'                   CALL SETREG
0102'  EB                         EX DE,HL
0103'  ED B0                      LDIR
0105'  D9                         EXX
0106'  C9                         RET
                                  ;
                                  ;
0107'                             SAVETYP:
0107'  D9                         EXX
0108'  CD 00F4'                   CALL SETREG
010B'  ED B0                      LDIR
010D'  D9                         EXX
010E'  C9                         RET
                                  ;
                                  ;
                                  ;SAVE . Saves basic program on disk.
                                  ;syntax: SAVE <filename>
                                  ;
010F'  01                         DB 1
                                  ;
77D8 0110'  DD 21 D8E8            DSAVE:  LD IX,CHNL5     ;FCB @ CHNL5
0114'  13                         INC DE
0115'  CD 078F'                   CALL GETUFN
                                  ;
                                  ;
0118'  1B                         DEC DE
0119'  ED 53 FB49                LD (DESAVE),DE
011D'  CD 0107'                   CALL SAVETYP
0120'  CD 00DA'                   CALL PUT$
                                  ;       CALL KILL1      ;Kill $$$ TYPE IF PRESENT.
0123'  DD 36 27 06                LD (IX+39),6
0127'  CD 0283'                   CALL OPENc      ; OPEN file.
                                  ;
                                  ; Save system varables.
                                  ;
012A'                             DBSTKLIM::
012A'  21 F8F2                    LD HL,0F8F2H    ; Bottom of system variables.
012D'  44                         LD B,H
012E'  4D                         LD C,L
012F'  2A FA94                    LD HL,(SYSTOP)  ; Top of system varables.
```

```
0132'  CD 017E'          CALL CALSIZ
0135'  CD 0176'          CALL PUTHL      ; Put Start of SV's.
0138'  E5                PUSH HL
0139'  60                LD H,B
013A'  69                LD L,C
013B'  CD 0176'          CALL PUTHL      ; Put length.
013E'  E1                POP HL
013F'  CD 05EB'          CALL STRPUT     ; Put SV's.
                         ;
                         ; Save basic program.
                         ;
0142'  CD 021E'          CALL SETVA
0145'  7E                PUTNXT: LD A,(HL)
0146'  D5                PUSH DE
0147'  CD 05BE'          CALL BPUT
014A'  D1                POP DE
014B'  DD E5             PUSH IX
014D'  CD 0242'          CALL INCLRA
0150'  DD E1             POP IX
0152'  CD 0239'          CALL DECVA
0155'  20 EE             JR NZ,PUTNXT
                         ;
                         ; Save basic variables.
                         ;
0157'  2A FA7F           LD HL,(CALCBOT)
015A'  ED 4B FA7B        LD BC,(VARNAM)
015E'  CD 017E'          CALL CALSIZ
0161'  CD 05EB'          CALL STRPUT
                         ;
0164'  CD 0321'          CALL CLOSEa     ; Write out last record, and close file.
                         ;
0167'  CD 00FE'          SAVEc:  CALL LOADTYP
016A'  CD 034C'          CALL KILL1      ;KILL ORIGINAL
016D'  CD 00E6'          CALL MOVNAM     ;MOVNAM UP 16 PLACES
0170'  CD 00DA'          CALL PUT$       ;RENAME $$$ FILE.
0173'  C3 03BD'          JP REN2
                         ;
                         ;
                         ;Saves the contents of HL.
                         ;
0176'  7D                PUTHL:  LD A,L
0177'  CD 05BE'          CALL BPUT
017A'  7C                LD A,H
017B'  C3 05BE'          JP BPUT
                         ;
                         ;
                         ;Start of data in BC, end in HL, returns with length in BC and start in HL.
                         ;
017E'  C5                CALSIZ: PUSH BC
017F'  A7                AND A
0180'  ED 42             SBC HL,BC
0182'  44                LD B,H
0183'  4D                LD C,L
0184'  E1                POP HL
0185'  C9                RET
                         ;
```

```
                                ;
                                ;
                                ;
                                ; Routine to load in programs. The programs are stored: system varables,
                                ;basic program, and basic varables. Each section starts with the number
                                ;of bytes it contains and the address at which they are to be stored.
                                ;Syntax LOAD <FILENAME>
                                ;
0186'  01                               DB 1
                                ;
7858  0187'                      DLOAD:
0187'  13                               INC DE
                                ;
0188'  DD 21 D8E8                       LD IX,CHNL5
018C'  CD 078F'                         CALL GETUFN
018F'  1B                               DEC DE
                                ;
                                ;
0190'                           AUTORUN::
                                ;
                                ;
0190'  DD 21 D8E8               ADLOAD::LD IX,CHNL5
0194'  D9                               EXX
0195'  21 FA85                         LD HL,USER-4
0198'  11 D912                         LD DE,USERSAV
019B'  01 0007                         LD BC,7
019E'  C5                               PUSH BC
019F'  D5                               PUSH DE
01A0'  E5                               PUSH HL
01A1'  ED B0                            LDIR
01A3'  D9                               EXX
01A4'  DD 36 27 05                      LD (IX+39),5
01A8'  CD 02E6'                         CALL SIOPEN
01AB'  CD 06A1'                         CALL STARW
                                ;
                                ; Load in the system variables.
                                ;
01AE'  21 FA7A                          LD HL,LSTPG     ;Save LSTPG
01B1'  7E                               LD A,(HL)
01B2'  F5                               PUSH AF
01B3'  E5                               PUSH HL
                                ;
01B4'  CD 0215'                         CALL GETBC
01B7'  60                               LD H,B
01B8'  69                               LD L,C
01B9'  CD 0215'                         CALL GETBC
01BC'  CD 01FE'                         CALL READIN
                                ;
01BF'  E1                               POP HL          ;Save disc LSTPG
01C0'  46                               LD B,(HL)
01C1'  F1                               POP AF          ;Restore old LSTPG
01C2'  77                               LD (HL),A
01C3'  C5                               PUSH BC
                                ;
                                ; Load in the Basic program.
                                ;
```

```
01C4'  CD 021E'                    CALL SETVA
01C7'  D5                  LDNXT:  PUSH DE
01C8'  CD 05F6'                    CALL BGET
01CB'  D1                          POP DE
01CC'  77                          LD (HL),A
01CD'  3A FA9J                     LD A,(STKLIM+1)
01D0'  BC                          CP H
01D1'  30 02                       JR NC,LDNX1
01D3'  EF                          RST ERRRST
01D4'  23                          DB 35                ;No space
01D5'  DD E5               LDNX1:  PUSH IX
01D7'  CD 0242'                    CALL INCLRA
01DA'  DD E1                       POP IX
01DC'  CD 0239'                    CALL DECVA
01DF'  20 E6                       JR NZ,LDNXT
                           ;
01E1'  C1                          POP BC               ;disc LSTPG
01E2'  CD 0000*                    CALL PAGE0      ;    CALL ADJVAL
01E5'  09                          DB 9
                           ;
                           ; Load in the basic variables.
                           ;
01E6'  2A FA7F                     LD HL,(CALCBOT)
01E9'  ED 4B FA7B                  LD BC,(VARNAM)
01ED'  CD 017E'                    CALL CALSIZ
01F0'  CD 01FE'                    CALL READIN
                           ;
01F3'  D1                          POP DE
01F4'  E1                          POP HL
01F5'  C1                          POP BC
01F6'  ED B0                       LDIR
01F8'  D1                          POP DE
01F9'  CD 0000*                    CALL PAGE0
01FC'  07                          DB 7             ;JP SLOAD1
01FD'  C9                          RET
                           ;
                           ; Routine to read in a number of bytes strored in the first two btes past the
                           ;file pointer, to an address stored in the second two bytes after the file
                           ;pionter
                           ;
                           ;
                           ; Routine to read in BC bytes to HL.
                           ;
01FE'  78                  READIN: LD A,B
01FF'  B1                          OR C
0200'  C8                          RET Z
0201'  CD 05F6'                    CALL BGET
                           ;
0204'  C5                          PUSH BC
0205'  E5                          PUSH HL
0206'  01 FAD2                     LD BC,0FAD2H
0209'  B7                          OR A
020A'  ED 42                       SBC HL,BC
020C'  E1                          POP HL
020D'  C1                          POP BC
                           ;
```

```
020E'  28 01                    JR Z,READI1
0210'  77                       LD (HL),A
0211'  23               READI1: INC HL
0212'  0B                       DEC BC
0213'  18 E9                    JR READIN
                        ;
                        ; Routine to read in HL.
                        ;
0215'  CD 05F6'         GETBC:  CALL BGET
0218'  4F                       LD C,A
0219'  CD 05F6'                 CALL BGET
021C'  47                       LD B,A
021D'  C9                       RET
                        ;
                        ;
                        ;
                        ;SETVA  (B,D,E) = size of program, (C,H,L) = start of program
                        ;
021E'  ED 5B FACC       SETVA:  LD DE,(ARRTOP)
0222'  3A FACE                  LD A,(ARRTOP+2)
0225'  47                       LD B,A
0226'  2A FD65                  LD HL,(VAZERO)
0229'  AF                       XOR A
022A'  E6 0F            SRAMPG: AND 0FH
022C'  4F                       LD C,A
022D'  3A FAD2                  LD A,(PAGE)
0230'  E6 F0                    AND 0F0H
0232'  B1                       OR C
0233'  32 FAD2                  LD (PAGE),A
0236'  D3 00                    OUT (PGPORT),A
0238'  C9                       RET
                        ;
                        ;DECVA  Z set when VA decremented to 0
                        ;
0239'  1B               DECVA:  DEC DE
023A'  7A                       LD A,D
023B'  B3                       OR E
023C'  C0                       RET NZ
023D'  B0                       OR B
023E'  C8                       RET Z
023F'  05                       DEC B
0240'  B7                       OR A
0241'  C9                       RET
                        ;
                        ;INCLRA  Increment LRA (in C,H,L), select page
                        ;
0242'  79               INCLRA: LD A,C
0243'  23                       INC HL
0244'  DD 21 FA7A               LD IX,LSTPG
0248'  DD BE 00                 CP (IX)
024B'  C8                       RET Z
024C'  D5                       PUSH DE
024D'  11 4000                  LD DE,4000H
0250'  19                       ADD HL,DE
0251'  38 04                    JR C,INCL1              ;End of page reached?
0253'  ED 52                    SBC HL,DE               ;Restore offset
```

```
0255'  D1                      POP DE
0256'  C9                      RET
0257'  3C              INCL1:  INC A
0258'  19                      ADD HL,DE              ;Adjust offset
0259'  DD BE 00                CP (IX)
025C'  20 01                   JR NZ,INCL2            ;Just moved to last page?
025E'  19                      ADD HL,DE
025F'  D1              INCL2:  POP DE
0260'  18 C8                   JR SRAMPG
                       ;
                       ;
                       ;
                       ;OPEN opens a file and initialises channel.
                       ;On entry, DE -> channel no.
                       ;Syntax: OPEN £<channel no.>,<filename>,<type>(,record length).
                       ; DE is looked after.
                       ;
0262'  07 01 2C 01             DB 7,1,',',1,',',2,'£'
0266'  2C 02 23
                       ;
78E7 0269'  CD 0B43'   OPEN:   CALL USRCHK
026C'  13                      INC DE
026D'  CD 0725'                CALL GETCHAN    ; Set IX -> FCB, DE -> <type>, A=0.
0270'  DD B6 27                OR (IX+39)      ; Is chanel in use?
0273'  C2 0741'                JP NZ,CHANERR
                       ;
0276'  CD 078F'                CALL GETUFN
                       ;
0279'  CD 0795'                CALL GETTYPE    ; A = type.
027C'  DD 77 27                LD (IX+39),A
                       ;
                       ; FCB contains: Drive,filename (8),file type (3), rest is zero.
                       ;
027F'  CD 08BC'                CALL TESTRND
0282'  28 25                   JR Z,OPEN2      ;Jump if not a random file
0284'  1A                      LD A,(DE)
0285'  FE 2C                   CP ','
0287'  C2 049A'                JP NZ,ERROR
028A'  13                      INC DE
028B'  CD 07BC'                CALL GETNXT     ;BC = record length
028E'  DD 71 25                LD (IX+37),C
0291'  DD 70 26                LD (IX+38),B    ;Store record length
0294'  0E 23                   LD C,COMPUT
0296'  CD 066B'                CALL DOS
0299'  DD 7E 21                LD A,(IX+33)
029C'  DD B6 22                OR (IX+34)
029F'  28 0D                   JR Z,SOPEN
02A1'  CD 0693'                CALL DECRR
02A4'  CD 08D3'                CALL SWOP
02A7'  18 05                   JR SOPEN        ;Open file and return
                       ;
                       ;
02A9'  1B              OPEN2:  DEC DE
02AA'  CB 47                   BIT 0,A
02AC'  20 38                   JR NZ,SIOPEN    ;Jump if sequential read file
                       ;
```

```
                        ;Here if sequential write file.
                        ;
02AE'  CD 02EC'         SOPEN:  CALL FOPEN              ;NZ if file opens OK else file not exist.
02B1'  20 09                    JR NZ,OLD               ;Jump if file exists.
                        ;
                        ; Create new file.
                        ;
7931    02B3'  0E 16    OPENc:  LD C,MAKE
02B5'  CD 066B'                 CALL DOS
02B8'  3C                       INC A
02B9'  C0                       RET NZ
                        ;
02BA'  EF                       RST ERRRST
02BB'  23                       DB 35                   ;No space error.
                        ;
02BC'  0E 23            OLD:    LD C,COMPUT
02BE'  CD 066B'                 CALL DOS
                        ;
                        ;Set random pointers to last record.
                        ;
02C1'  CD 0693'                 CALL DECRR
                        ;
                        ;Now read in last record to get last record offset.
                        ;
02C4'  CD 06A1'                 CALL STARW              ;Read record
02C7'  DD CB 27 7E              BIT 7,(IX+39)           ;???????
02CB'  C0                       RET NZ                  ;Return if file empty.
                        ;
                        ;Find end of data in record.
                        ;
02CC'  01 0080          OK3:    LD BC,128
02CF'  21 E680                  LD HL,DMA               ;HL -> First byte in record.
02D2'  3E 1A                    LD A,1AH
02D4'  ED B1                    CPIR
02D6'  3E 7F                    LD A,127
02D8'  91                       SUB C
02D9'  DD 77 24                 LD (IX+36),A
02DC'  3C                       INC A
02DD'  F0                       RET P                   ;Return if ^Z found
                        ;
                        ;Record is full, so increment record count.
                        ;
02DE'  CD 0685'                 CALL INCRR
02E1'  DD 36 24 00              LD (IX+36),0
02E5'  C9                       RET
                        ;
                        ;
                        ;Sequental input file open routine.
                        ;
7964    02E6'           SIOPEN::
02E6'  CD 02EC'                 CALL FOPEN
02E9'  C0                       RET NZ                  ;Return if file found.
                        ;
                        ;File not found error.
                        ;
02EA'  EF               NFERR:  RST ERRRST
```

```
02EB'  26                      DB 38                ;'UNDEFINED',File not found
                               ;
                               ;
02EC'  0E 0F           FOPEN:  LD C,DOPEN
02EE'  CD 066B'                CALL DOS
02F1'  3C                      INC A
02F2'  C9                      RET
                               ;
                               ;
                               ;
                               ;CLOSE closes file. On entry, DE -> channel no.
                               ;Syntax: CLOSE £<channel no.>.
                               ;
02F3'  02 23                   DB 2,'£'
                               ;
02F5'  CD 0B43'        CLOSE:  CALL USRCHK
02F8'  13                      INC DE
02F9'  1A                      LD A,(DE)
02FA'  FE FF                   CP 0FFH
02FC'  18 2F                   JR CLOSALL
02FE'  FE 3A                   CP ':'
0300'  18 2B                   JR CLOSALL
                               ;
0302'  CD 0725'                CALL GETCHAN         ;IX -> channel FCB
0305'  DD 7E 27        CLOSEb: LD A,(IX+39)
0308'  CB 47                   BIT 0,A
030A'  20 18                   JR NZ,IRCLOSE
030C'  B7                      OR A
030D'  C8                      RET Z
                               ;
030E'  CD 06A1'        FCLOSE: CALL STARW           ;Read in last record
0311'  DD 7E 24                LD A,(IX+36)
0314'  A7                      AND A                ;Is record empty?
0315'  28 0D                   JR Z,FCLOS2          ;Jump if last record full
                               ;
0317'  CD 0679'                CALL CALPOS          ;HL -> first free byte
031A'  36 1A           FCLOS1: LD (HL),1AH
031C'  23                      INC HL
031D'  3C                      INC A
031E'  F2 031A'                JP P,FCLOS1          ;Jump if MSB not set.
0321'  CD 06C2'        CLOSEa: CALL STOPRW          ;Write out last record.
                               ;
                               ;Now close file. Entry point for sequential input and random close.
                               ;
0324'                  IRCLOSE:
0324'                  FCLOS2:
0324'  DD 36 27 00            LD (IX+39),0         ;Zero type byte.
0328'  0E 10                   LD C,DCLOSE
032A'  C3 066B'                JP DOS
                               ;
                               ;
                               ; Close all channels.
                               ;
032D'                  CLOSALL::
032D'  11 0028                 LD DE,40
0330'  06 04                   LD B,4          ;5!!!!
```

```
0332'  DD 21 D840          LD IX,CHNL1
0336'  C5                  CLSA1: PUSH BC
0337'  CD 0305'                   CALL CLOSEb
033A'  C1                         POP BC
033B'  DD 19                      ADD IX,DE
033D'  10 F7                      DJNZ CLSA1
033F'  C9                         RET
                           ;
                           ;
                           ;
                           ;KILL closes and deletes file for specified channel.
                           ;On entry, DE -> channel no.
                           ;Syntax: KILL £<channel no.>.
                           ;
0340'  02 23                      DB 2,'£'
                           ;
7987 0342'  CD 0B43'       KILL:  CALL USRCHK
0345'  13                         INC DE
0346'  CD 0725'                   CALL GETCHAN
0349'  CD 0324'                   CALL IRCLOSE
034C'  0E 13               KILL1: LD C,DELETEF
034E'  C3 066B'                   JP DOS

                           ;
                           ;
                           ;
0351'                      RAWREAD:
0351'  CD 05F6'                   CALL BGET
0354'  77                         LD (HL),A
0355'  03                         INC BC
0356'  23                         INC HL
0357'  DD CB 27 7E                BIT 7,(IX+39)
035B'  28 F4                      JR Z,RAWREAD
035D'  C9                         RET
                           ;
                           ;
                           ;
                           ;TYPE, types out a file to the screen, up an end of file.
                           ;syntax: TYPE <filename>
                           ;
035E'  01                         DB 1
                           ;
7968 035F'  CD 0B43'       ETYPE:: CALL USRCHK
0362'  13                         INC DE
0363'  DD 21 D8E8                 LD IX,CHNL5
0367'  CD 078F'                   CALL GETUFN
036A'  1B                         DEC DE
036B'  CD 02E6'                   CALL SIOPEN
036E'  CD 06A1'                   CALL STARW
0371'  CD 05F6'            TYPE1: CALL BGET
0374'  DD CB 27 7E                BIT 7,(IX+39)
0378'  C0                         RET NZ
0379'  CD 0CAB                    CALL PRINTX
037C'  CD 09F2                    CALL BREAKMON
037F'  28 F0                      JR Z,TYPE1
```

```
0381'  C9                       RET
                        ;
                        ;
                        ;
                        ;RENAM changes then filenaame of a file.
                        ;svntax: REN <new filename>=<old filename>
                        ;
0382'  C9                       RET
0383'  01 D3 01                 DB 1,0D3H,1
                        ;
79FD 0386'  CD 0B43'    REN::   CALL USRCHK
0389'  13                       INC DE
038A'  DD 21 D8E8               LD IX,CHNL5
038E'  CD 078F'                 CALL GETUFN             ;Read in new filename.
                        ;
0391'  0E 11                    LD C,SEARCH
0393'  CD 066B'                 CALL DOS
0396'  FE FF                    CP 0FFH
0398'  28 02                    JR Z,REN0
                        ;
                        ;Here if file already exists
                        ;
039A'  EF                       RST ERRRST
039B'  30                       DB 48+0
                        ;
039C'  D5               REN0:   PUSH DE
039D'  11 D8F8                  LD DE,CHNL5+16  ;Move name up by 16.
03A0'  21 D8E8                  LD HL,CHNL5
03A3'  01 000C                  LD BC,12
03A6'  ED B0                    LDIR
03A8'  21 D8E9                  LD HL,CHNL5+1
03AB'  06 0B                    LD B,11
03AD'  36 20            REN1:   LD (HL),' '
03AF'  23                       INC HL
03B0'  10 FB                    DJNZ REN1
03B2'  D1                       POP DE
                        ;
03B3'  21 D8E8                  LD HL,CHNL5
03B6'  CD 075C'                 CALL GETFNp             ;Read old file name into FCB.
03B9'  1B                       DEC DE
03BA'  CA 0793'                 JP Z,FNAMERR
                        ;
03BD'  0E 17            REN2:   LD C,DOSREN
03BF'  CD 066B'                 CALL DOS
03C2'  3C                       INC A
03C3'  C0                       RET NZ
                        ;
03C4'  EF                       RST ERRRST
03C5'  26                       DB 38                   ;UNDIFINED

                        ;
                        ;
                        ;RUN loads the first record of a utillt program and passes control to it.
                        ;svntax: RUN <filename>
                        ;
03C6'  C9                       RET
```

```
                03C7'  07 01                    DB 7,1
                                                ;
7A38            03C9'  13               RUN::   INC DE
                03CA'  DD 21 D8E8               LD IX,CHNLS
                03CE'  CD 078F'                 CALL GETUFN
                03D1'  DD 36 27 05              LD (IX+39),5
                03D5'  1B                       DEC DE
                03D6'  CD 02E6'                 CALL SIOPEN
                03D9'  CD 06A1'                 CALL STARW
                03DC'  CD 0215'                 CALL GETBC
                03DF'  60                       LD H,B
                03E0'  69                       LD L,C
                03E1'  E5                       PUSH HL
                03E2'  CD 0215'                 CALL GETBC
                03E5'  CD 01FE'                 CALL READIN
                03E8'  E1                       POP HL
                03E9'  D5                       PUSH DE
                03EA'  CD 07B8                  CALL JPHL
                03ED'  D1                       POP DE
                03EE'  C9                       RET
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;RECORD sets record number for specified channel.
                                        ;Sets up both the random record count in the FCB and the record offset.
                                        ;On entry, DE -> channel no.
                                        ;Syntax: RECORD £<channel no.>,<record No.>.
                                        ;
                03EF'  C9                       RET
                03F0'  02 2C 02 23              DB 2,',',2,'£'
                                        ;
7A64            03F4'                   RECORD::
                03F4'  CD 0B43'                 CALL USRCHK
                03F7'  13                       INC DE
                03F8'  CD 0725'                 CALL GETCHAN            ;IX -> channel FCB
                03FB'  CD 08BC'                 CALL TESTRND
                03FE'  C8                       RET Z
                03FF'  DD E5                    PUSH IX
                0401'  CD 07C5'                 CALL GETREC             ;ABC = record number
                0404'  DD E1                    POP IX
                0406'  D5                       PUSH DE
                                        ;
                                        ; Routine to set the ex-FCB up for a random disk acces. The logical rec. No.
                                        ; is passed in ABC this is then multiplyed by the logical record length
                                        ; and divided by the phisical Rec. length (128). This gives the phisical
                                        ; Rec. No. The offset for the record is also calculated, ie
                                        ; MOD ( No. char's into file , 128).
                                        ;
                0407'  DD CB 27 BE      SETREC: RES 7,(IX+39)
                040B'  DD CB 27 B6              RES 6,(IX+39)
                040F'  57                       LD D,A                  ;DBC = record no.
                0410'  AF                       XOR A
                0411'  67                       LD H,A
                0412'  6F                       LD L,A                  ;AHL = 0 =  record. size * logical record no.
                0413'  DD 5E 26                 LD E,(IX+38)            ;E = High byte of record size
```

```
0416'  CD 0434'                    CALL MULT
0419'  DD 5E 25                    LD E,(IX+37)          ;E = Low byte of record size
041C'  CD 0434'                    CALL MULT
                                   ;
                                   ;
                                   ;Here, AH = physical record no., L/2 = record offset.
                                   ;
041F'  DD 74 21                    LD (IX+33),H
0422'  DD 77 22                    LD (IX+34),A
0425'  CB 3D                       SRL L
0427'  DD 75 24                    LD (IX+36),L          ;Store record offset
                                   ;
042A'  D1                          POP DE
042B'  CD 08C5'                    CALL STEST
042E'  D0                          RET NC
042F'  DD CB 27 FE                 SET 7,(IX+39)
0433'  C9                          RET
                                   ;
                                   ;
                                   ;MULT multiplies DBC * E, adding the result to AHL.
                                   ;Error if AHL > 256*64K
                                   ;NB the result is multiplyed by two.
                                   ;
0434'  08                          MULT:   EX AF,AF'
0435'  3E 08                       LD A,8
0437'  08                          MULT1:  EX AF,AF'
0438'  CB 23                       SLA E
043A'  30 02                       JR NC,SKIP
                                   ;
043C'  09                          ADD HL,BC
043D'  8A                          ADC A,D
                                   ;
043E'  29                          SKIP:   ADD HL,HL
043F'  17                          RLA
0440'  DA 07C3'                    JP C,TOOBIG

0443'  08                          EX AF,AF'
0444'  3D                          DEC A
0445'  20 F0                       JR NZ,MULT1
0447'  08                          EX AF,AF'
0448'  C9                          RET
                                   ;
                                   ;
                                   ;
                                   ;Entry point for line input, ie ignor ","'s in input lines.
                                   ;
0449'  08 2C 02 23                 DB 8,',',2,'£',98H
044D'  98

7ABE 044E'                         ;
044E'  13                          DLINPUT::
044F'  3E FF                       INC DE
0451'  18 06                       LD A,0FFH
                                   JR DINPT0
                                   ;
                                   ;
                                   ;INPUT assigns variable(s) with values read from disc file.
```

```
                        ;Syntax: DINPUT.<channel no.>,<list of variables>.
                        ;
                        ;
0453'  C9                       RET
0454'  08 2C 02 23              DB 8,',',2,'£'
7AC8 0458'              DINPUT::
0458'  AF                       XOR A
0459'  32 D8F4          DINPT0: LD (LINPUT),A
045C'  13                       INC DE
045D'  CD 06CD'                 CALL LINE$
0460'  CD 0725'                 CALL GETCHAN
0463'  DD CB 27 46              BIT 0,(IX+39)
0467'  28 31                    JR Z,ERROR              ;Jump if not read enabled.
0469'  CD 06A1'                 CALL STARW              ;Read in current record.
046C'  CD 08BC'                 CALL TESTRND
046F'  20 2B                    JR NZ,RANDIN
0471'  1A               DINPT1: LD A,(DE)
0472'  FE FF                    CP 0FFH
0474'  C8                      RET Z
                        ;
0475'  FE 2C                    CP ','
0477'  20 01                    JR NZ,DINPT2
0479'  13                      INC DE
                        ;
                        ;DE -> varable name. Variable value (as string) on stack.
                        ;
047A'  2A FA81          DINPT2: LD HL,(CALCST)  ;Put field onto stack.
047D'  CD 0632'                 CALL STRGET
0480'  71               DINPT3: LD (HL),C
0481'  23                       INC HL
0482'  70                       LD (HL),B
0483'  23                       INC HL
0484'  22 FA81                  LD (CALCST),HL
                        ;
0487'  DD E5            DINPT4: PUSH IX
0489'  D5                       PUSH DE
048A'  EB                       EX DE,HL
048B'  CD 0000*                 CALL PAGE0
048E'  00                       DB 0                    ;CALL AE
048F'  D1                       POP DE
0490'  CD 0000*                 CALL PAGE0              ;CALL GETINP    ; Preserves DE.
0493'  04                       DB 4
0494'  DD E1                    POP IX
0496'  28 D9                    JR Z,DINPT1
                        ;
0498'  EF                       RST ERRRST                      ;'Not Numeric'
0499'  36                       DB 48+6
                        ;
                        ;
                        ;
049A'  EF               ERROR:  RST ERRRST                      ;'Mistake',Invalid file access.
049B'  3B                       DB 48+11
                        ;
                        ;
                        ;
                        ;RANDOM record input bit.
```

```
                                ;
049C' 2A FA81                   RANDIN: LD HL,(CALCST)  ;Put field onto stack.
049F' EB                                EX DE,HL
04A0' CB 76                             BIT 6,(HL)
04A2' EB                                EX DE,HL
04A3' C2 049A'                          JP NZ,ERROR
                                ;
04A6' DD 46 26                          LD B,(IX+38)
04A9' DD 4E 25                          LD C,(IX+37)
04AC' C5                                PUSH BC
04AD' CD 01FE'                          CALL READIN
04B0' C1                                POP BC
04B1' 71                                LD (HL),C
04B2' 23                                INC HL
04B3' 70                                LD (HL),B
04B4' 23                                INC HL
04B5' 22 FA81                           LD (CALCST),HL
04B8' CD 0000*                          CALL PAGE0
04BB' 04                                DB 4                    ;CALL GETINP
04BC' C9                                RET
                                ;
                                ;
                                ;
                                ;
                                ;PRINT writes values to disc file.
                                ;Syntax DPRINT,<channel no.>,<list of variables>.
                                ;
04BD' C9                                RET
04BE' 03 2C 02 23               DB 3,',',2,'£'
                                ;
7838 04C2'                      DPRINT:
04C2' 13                                INC DE
04C3' CD 06CD'                          CALL LINE$
04C6' CD 0725'                          CALL GETCHAN
04C9' DD 22 D8F6                        LD (IXTEMP),IX
04CD' CD 06A1'                          CALL STARW
04D0' CD 08C5'                          CALL STEST
04D3' 30 03                             JR NC.DPRIN2
04D5' CD 08D3'                          CALL SWOP
04D8' DD CB 27 4E               DPRIN2: BIT 1,(IX+39)
04DC' 28 BC                             JR Z,ERROR              ;Jump if file read only
04DE' CD 08BC'                          CALL TESTRND
04E1' 20 61                             JR NZ.RNDOUT
04E3' 1B                                DEC DE
04E4' 13                        DPRIN0: INC DE
04E5' 1A                        DPRIN1: LD A,(DE)
04E6' FE FF                             CP 0FFH
04E8' 20 0E                             JR NZ.DPRIN2
04EA' 1B                                DEC DE
04EB' 1A                                LD A,(DE)
04EC' 13                                INC DE
04ED' FE 3B                             CP ';'          ; Return if ';' at end of line.
04EF' CA 06C2'                          JP Z.STOPRW
04F2' CD 0509'                          CALL WRCRLF     ; Write CR/LF
04F5' C3 06C2'                          JP STOPRW
                                ;
```

```
04F8'  FE 3B          DPRIN2: CP ';'
04FA'  28 E8                  JR Z,DPRINO
04FC'  FE 2C                  CP ','
04FE'  20 17                  JR NZ,DPRIN3
0500'  DD 2A D8F6             LD IX,(IXTEMP)
0504'  CD 05BE'               CALL BPUT       ;Write A=',' to file.
0507'  18 DB                  JR DPRINO
                       ;
                       ;
                       ;
0509'  DD 2A D8F6     WRCRLF: LD IX,(IXTEMP)
050D'  3E 0D                  LD A,CR
050F'  CD 05BE'               CALL BPUT
0512'  3E 0A                  LD A,LF
0514'  C3 05BE'               JP BPUT
                       ;
                       ; Evaluate expression and write to file.
                       ;
0517'  D5              DPRIN3: PUSH DE
0518'  EB                    EX DE,HL
0519'  CD 0000*               CALL PAGE0      ;CALL AE
051C'  00                     DB 0
051D'  D1                     POP DE
051E'  28 06                  JR Z,DPRIN4
0520'  CD 0000*               CALL PAGE0
0523'  02                     DB 2            ;CALL EVALSE
0524'  18 0A                  JR DPRIN5
                       ;
0526'  CD 0000*        DPRIN4: CALL PAGE0      ;CALL EVALAB
0529'  01                     DB 1
052A'  D5                     PUSH DE
052B'  CD 0000*               CALL PAGE0
052E'  08                     DB 8            ;CALL STR$
052F'  D1                     POP DE
                       ;
0530'  D5              DPRIN5: PUSH DE
0531'  CD 0000*               CALL PAGE0      ;CALL FIND1$
0534'  03                     DB 3
0535'  ED 53 FA81             LD (CALCST),DE
0539'  EB                     EX DE,HL        ; HL -> String.
053A'  DD 2A D8F6             LD IX,(IXTEMP)
                       ;
053E'  CD 05EB'        DPRIN6: CALL STRPUT
0541'  D1                     POP DE
0542'  18 A1                  JR DPRIN1
                       ;
                       ;
                       ;If Random file then only strings allowed, whos length must be less than or
                       ;equal to the length of the record.
                       ;
0544'  CD 0000*        RNDOUT: CALL PAGE0              ;CALL EVALSE
0547'  02                      DB 2
0548'  D5                      PUSH DE
0549'  CD 0000*                CALL PAGE0              ;CALL FIND1$
054C'  03                      DB 3
054D'  ED 53 FA81              LD (CALCST),DE
```

```
0551'  DD 2A D8F6          LD IX,(IXTEMP)
0555'  DD 66 26            LD H,(IX+38)
0558'  DD 6E 25            LD L,(IX+37)
055B'  B7                  OR A
055C'  ED 42               SBC HL,BC
055E'  E5                  PUSH HL
055F'  DA 07C3'            JP C,TOOBIG
0562'  EB                  EX DE,HL
0563'  CD 05EB'            CALL STRPUT
0566'  C1                  POP BC
                       ;
0567'  78              PUT0:  LD A,B          ;FILL REST OF REC WITH ZEROS.
0568'  B1                  OR C
0569'  28 07               JR Z,PUT01
056B'  AF                  XOR A
056C'  CD 05BE'            CALL BPUT
056F'  0B                  DEC BC
0570'  18 F5               JR PUT0
                       ;
0572'  D1              PUT01: POP DE
0573'  1A                  LD A,(DE)
0574'  FE FF               CP 0FFH
0576'  CA 06C2'            JP Z,STOPRW
                       ;
0579'  EF                  RST ERRRST
057A'  3B                  DB 48+11
                       ;
                       ;
                       ;
                       ; Routine to read from disk to location HL raw data until an end of file
                       ; is signified by reading in the last record of the file.
                       ;syntax: READ <file name>,<start>
                       ;
057B'  C9                  RET
057C'  02 2C 01            DB 2,',',1                      ;1,',',2
                       ;
7BE8 057F'  13          READ:  INC DE
0580'  DD 21 D8E8          LD IX,CHNLS
0584'  CD 078F'            CALL GETUFN
0587'  DD 36 27 05          LD (IX+39),5
058B'  CD 02E6'            CALL SIOPEN
058E'  CD 06A1'            CALL STARW
0591'  CD 07BC'            CALL GETNXT     ; BC = Base address.
0594'  60                  LD H,B
0595'  69                  LD L,C
0596'  C3 0351'            JP RAWREAD
                       ;
                       ;
                       ;
                       ; Routine to write out a number of bytes to a file
                       ;syntax: WRITE <filename>,<start address>,<number of bytes>
                       ;
0599'  C9                  RET
059A'  02 2C 02 2C         DB 2,',',2,',',1                ; 1,',',2,',',2
059E'  01
                       ;
```

7C08

```
059F'  CD 0B43'          WRITE:  CALL USRCHK
05A2'  13                        INC DE
05A3'  DD 21 D8E8                LD IX.CHNL5     ; FCB at CHNL5
05A7'  CD 078F'                  CALL GETUFN
05AA'  CD 034C'                  CALL KILL1      ; Erases current copy.
05AD'  CD 02B3'                  CALL OPENc      ;Makes new file.
05B0'  CD 07BC'                  CALL GEtNXT     ; Base address => BC.
05B3'  C5                        PUSH BC
05B4'  CD 07BC'                  CALL GETNXT     ; Number of bytes =>BC.
05B7'  E1                        POP HL
05B8'  CD 05EB'                  CALL STRPUT     ; Write out.
05BB'  C3 0321'                  JP CLOSEa       ; Close file and write out last record.
                                 ;
                                 ;
                                 ;
                                 ; NOTE: STARW MUST HAVE BEEN CALLED BEFORE IO TO A DIFFERENT FILE.
                                 ; Routine to write away one byte to a file whoes extended FCB is pointed
                                 ; to by IX. BC DE, and HL preserved.
                                 ;
7C2A 05BE'  C5                   BPUT:   PUSH BC
05BF'  E5                                PUSH HL
05C0'  CD 0679'                          CALL CALPOS
05C3'  77                                LD (HL),A
05C4'  DD 34 24                          INC (IX+36)
                                 ;
                                 ; Is DMA full ?
                                 ;
05C7'  F2 05E8'                          JP P,BPUT1              ;Jump if high bit reset.
                                 ;
                                 ; Yes.
                                 ;
05CA'  0E 22                             LD C,RWRITE
05CC'  CD 066B'                          CALL DOS
                                 ;
05CF'  B7                                OR A
05D0'  28 02                             JR Z,BPUTQ
05D2'  EF                                RST ERRRST
05D3'  23                                DB 35
                                 ;
05D4'  CD 0685'          BPUTQ:  CALL INCRR
05D7'  0E 21                     LD C,RREAD
05D9'  CD 066B'                  CALL DOS
05DC'  DD 36 24 00               LD (IX+36),0
05E0'  CD 08C5'                  CALL STEST
05E3'  30 03                     JR NC,BPUT1
                                 ;
05E5'  CD 08D3'                  CALL SWOP
05E8'  E1                BPUT1:  POP HL
05E9'  C1                        POP BC
05EA'  C9                        RET
                                 ;
                                 ;
                                 ; Routine to write out BC bytes  starting at (HL) to a file whos
                                 ;FCB is pointed to by IX.
                                 ;
7C55 05EB'  78           STRPUT: LD A,B
```

```
05EC'  B1                      OR C
05ED'  C8                      RET Z
05EE'  7E                      LD A,(HL)
05EF'  23                      INC HL
05F0'  0B                      DEC BC
05F1'  CD 05BE'                CALL BPUT
05F4'  18 F5                   JR STRPUT
                               ;
                               ;
                               ; Routine to get one byte from current disk file.
                               ;
7C64   05F6'  C5               BGET:   PUSH BC
05F7'  E5                      PUSH HL
                               ;
05F8'  DD CB 27 7E             BIT 7,(IX+39)
05FC'  28 02                   JR Z,BGET1
                               ;
05FE'  EF                      RST ERRRST
05FF'  3F                      DB 48+15
                               ;
0600'  CD 0679'                BGET1:  CALL CALPOS
0603'  7E                      LD A,(HL)
0604'  F5                      PUSH AF
                               ;
0605'  DD 34 24                INC (IX+36)
0608'  F2 061C'                JP P,BGET2
                               ;
060B'  E5                      PUSH HL
060C'  CD 0685'                CALL INCRR
060F'  E1                      POP HL
0610'  0E 21                   LD C,RREAD
0612'  CD 066B'                CALL DOS
0615'  DD 36 24 00             LD (IX+36),0
0619'  B7                      OR A
061A'  20 10                   JR NZ,BGET4
                               ;
061C'  DD CB 27 56             BGET2:  BIT 2,(IX+39)
0620'  20 06                   JR NZ,BGET3
0622'  23                      INC HL
0623'  7E                      LD A,(HL)
0624'  FE 1A                   CP 1AH
0626'  28 04                   JR Z,BGET4
                               ;
0628'  F1                      BGET3:  POP AF
0629'  E1                      POP HL
062A'  C1                      POP BC
062B'  C9                      RET
                               ;
062C'  DD CB 27 FE             BGET4:  SET 7,(IX+39)
0630'  18 F6                   JR BGET3
```

Extra code

```
                               ;
                               ;
                               ;
                               ; Routine to read bytes to (HL) upto , cr,lf or "," if linput <>0 and ^z if
                               ;raw data flag not set and put the number of bytes into in BC.
                               ;
```

7CA9

```
0632'  01 0000      STRGET: LD BC,0          ; Zero count.
0635'  CD 05F6'     STRGT1: CALL BGET        ;Read in one character.
0638'  C5           EOL1:   PUSH BC
0639'  ED 4B FA92           LD BC,(STKLIM)
063D'  B7                   OR A
063E'  ED 42                SBC HL,BC
0640'  38 02                JR C,RAWSKIP
                    ;
0642'  EF                   RST ERRRST
0643'  23                   DB 35            ;No space.
                    ;
0644'               RAWSKIP:
0644'  09                   ADD HL,BC
0645'  47                   LD B,A
0646'  3A D8F4              LD A,(LINPUT)    ;Line input mode ?
0649'  B7                   OR A
064A'  78                   LD A,B
064B'  C1                   POP BC
064C'  20 03                JR NZ,STRGT2     ;Jump if Line input mode.
064E'  FE 2C                CP ','           ;Comma
0650'  C8                   RET Z
                    ;       OR A             ;End of valid data. (Null character).
                    ;       JR Z,NULDAT
0651'  FE 0D        STRGT2: CP CR            ;Carage return line feed ?
0653'  28 0A                JR Z,EOL
0655'  DD CB 27 7E  STRGT3: BIT 7,(IX+39)    ;End of file.
0659'  C0                   RET NZ
065A'  77                   LD (HL),A
065B'  23                   INC HL
065C'  03                   INC BC
065D'  18 D6                JR STRGT1
                    ;
                    ;
                    ;
                    ;This section tests for CR LF combination.
                    ;NB if LF CR is found this will not act as an end of field delimeter.
                    ;
065F'  CD 05F6'     EOL:    CALL BGET
0662'  FE 0A                CP 0AH
0664'  C8                   RET Z
0665'  36 0D                LD (HL),0DH
0667'  03                   INC BC
0668'  23                   INC HL
0669'  18 CD                JR EOL1
                    ;
                    ;
                    ;
                    ;
                    ; Routine to make BDOS calls, preserve IX and load DE with the FCB address
                    ; ie IX.
                    ;
                    ; B is loaded with image of IFF2, 0 if interupts disabled, else FF.
066B'               ;
                    DOS:
                    ;       LD A,I           ;PE if interrupts enabled.
                    ;       PUSH AF
```

```
                                ;       DI
                                ;
7CE2    066B'  D5               ;       PUSH DE
        066C'  E5               ;       PUSH HL
        066D'  DD E5            ;       PUSH IX
        066F'  D1               ;       POP DE
        0670'  D5               ;       PUSH DE
        0671'  CD 0000*         ;       CALL BDOS
        0674'  DD E1            ;       POP IX
        0676'  E1               ;       POP HL
        0677'  D1               ;       POP DE
                                ;
                                ;       POP BC          ;If IFF2 was set, then re-enable inerupts.
                                ;       BIT 2,C         ;NZ if interrupts were enabled
                                ;       RET Z
                                ;       EI
        0678'  C9               ;       RET
                                ;
                                ; Routine to calculat absolute address in DMA.
                                ;
7CF1    0679'  D5               CALPOS: PUSH DE
        067A'  21 E680          ;       LD HL,DMA
        067D'  16 00            ;       LD D,0
        067F'  DD 5E 24         ;       LD E,(IX+36)
        0682'  19               ;       ADD HL,DE
        0683'  D1               ;       POP DE
        0684'  C9               ;       RET
                                ;
                                ; Routine to increment random record pionter.
                                ;
        0685'  DD 6E 21         INCRR:  LD L,(IX+33)
        0688'  DD 66 22         ;       LD H,(IX+34)
        068B'  23               ;       INC HL
        068C'  DD 75 21         ;       LD (IX+33),L
        068F'  DD 74 22         ;       LD (IX+34),H
        0692'  C9               ;       RET
                                ;
                                ;Decrement random record Number.
                                ;
        0693'  DD 6E 21         DECRR:  LD L,(IX+33)
        0696'  DD 66 22         ;       LD H,(IX+34)
        0699'  2B               ;       DEC HL
        069A'  DD 75 21         ;       LD (IX+33),L
        069D'  DD 74 22         ;       LD (IX+34),H            ;Store last record no
        06A0'  C9               ;       RET
                                ;
                                ; THIS ROUTINE MUSH BE CALLED BEFORE I/O TO A PARTICULAR FILE STARTS.
                                ; The routine sets up the DMA, fetches the relevant record from the
                                ; disk to load the DMA.
                                ;
7D13    06A1'  D5               STARW:  PUSH DE
        06A2'  CD 0042'         ;       CALL SETDMA      ;Make sure DMA set up.
                                ;
                                ;
        06A5'  06 80            STARW0: LD B,128                ; Fill DMA with null charicters.
        06A7'  21 E680          ;       LD HL,DMA
```

```
06AA'  36 00          STARW2: LD (HL),0
06AC'  23                     INC HL
06AD'  10 FB                  DJNZ STARW2
                      ;
06AF'  0E 21                  LD C,RREAD
06B1'  CD 066B'               CALL DOS
06B4'  B7                     OR A
06B5'  28 09                  JR Z,STARW3
                      ;
                      ; Reading past EOF. This is not an error unless in read mode, in which case
                      ; read routines will pick-up on the EOF bit and call an error.
                      ;
06B7'  CD 08BC'               CALL TESTRND
06BA'  18 04                  JR STARW3
06BC'  DD CB 27 FE            SET 7,(IX+39)
06C0'  D1             STARW3: POP DE
06C1'  C9                     RET
                      ;
                      ;
                      ;
                      ; THIS ROUTINE MUST BE CALLED WHEN A PARTICULAR FILE HAS FINISHED
                      ; IO.
                      ; The routine writes out the current record to disk to clear the
                      ; DMA for an other file.
                      ;
7028   06C2'  DD 7E 24        STOPRW: LD A,(IX+36)
06C5'  B7                     OR A
06C6'  C8                     RET Z
06C7'  0E 22                  LD C,RWRITE
06C9'  CD 066B'               CALL DOS
06CC'  C9                     RET
                      ;
                      ;
                      ;
                      ;LINE$ Reset bit 6 of 1st. bytes of string variables. DE -> £ on entry
                      ;
06CD'  D5             LINE$:  PUSH DE
06CE'  E5                     PUSH HL
06CF'  13                     INC DE                 ;DE -> channel no.
06D0'  CD 06FE'               CALL SNEXT
06D3'  28 26                  JR Z,XLINE
06D5'  D5             LOOP1:  PUSH DE                ;DE -> seperator
06D6'  E1                     POP HL
06D7'  23                     INC HL                 ;HL -> 1st byte of variable
06D8'  CD 06FE'       LOOP2:  CALL SNEXT
06DB'  28 1E                  JR Z,XLINE
06DD'  FE 24                  CP '$'
06DF'  28 0B                  JR Z,LOOP3
06E1'  FE 22                  CP 22H                 ;Test for "
06E3'  28 0B                  JR Z,LOOP4
06E5'  CD 0703'               CALL ALPHNM            ;Z set if alphanumeric
06E8'  20 EB                  JR NZ,LOOP1
06EA'  18 EC                  JR LOOP2
06EC'  CB B6          LOOP3:  RES 6,(HL)
06EE'  18 E8                  JR LOOP2
06F0'  CD 06FE'       LOOP4:  CALL SNEXT
```

```
06F3'  28 06                    JR Z,XLINE
06F5'  FE 22                    CP 22H
06F7'  20 F7                    JR NZ,LOOP4
06F9'  18 DD                    JR LOOP2
06FB'  E1               XLINE:  POP HL
06FC'  D1                       POP DE
06FD'  C9                       RET
                        ;
06FE'  13               SNEXT:  INC DE
06FF'  1A                       LD A,(DE)
0700'  FE FF                    CP 0FFH
0702'  C9                       RET
                        ;
0703'  FE 1B            ALPHNM: CP 1BH
0705'  38 10                    JR C,ZSET
0707'  FE 30                    CP 30H
0709'  38 0E                    JR C,NZSET
070B'  FE 3A                    CP 3AH
070D'  38 08                    JR C,ZSET
070F'  FE 41                    CP 41H
0711'  38 06                    JR C,NZSET
0713'  FE 5B                    CP 5BH
0715'  30 02                    JR NC,NZSET
0717'  AF               ZSET:  XOR A
0718'  C9                       RET
0719'  3E 01             NZSET: LD A,1
071B'  B7                       OR A
071C'  C9                       RET
                        ;
                        ;
                        ;
                        ;               ****************************************
                        ;               *                                      *
                        ;               *       DISC BASIC UTILITY ROUTINES    *
                        ;               *                                      *
                        ;               ****************************************
                        ;
                        ;
                        ;
                        ;PRNTCHRS prints character in (HL), increments HL, decrements B and repaets
                        ;until B = 0. Preserves DE.
                        ;
7036   071D'            PRNTCHRS:
071D'  7E                       LD A,(HL)
071E'  CD 0CAB                  CALL PRINTX
0721'  23                       INC HL
0722'  10 F9                    DJNZ PRNTCHRS
0724'  C9                       RET
                        ;
                        ;
                        ;
                        ;GETCHAN: On entry, DE -> £<channel no.>. On exit. IX -> FCB for specified
                        ;channel if in range [1..4], else error. Destroys A,BC.
                        ;
```

```
703E  0725'                    GETCHAN:
      0725'  13                        INC DE                  ;Skip hash
      0726'  CD 07BC'                  CALL GETNXT             ;BC = channel
      0729'  78              GETCH0:   LD A,B
      072A'  A7                        AND A
      072B'  20 14           GETCH1:   JR NZ,CHANERR           ;'Out of range'
      072D'  B1                        OR C
      072E'  28 11                     JR Z,CHANERR            ;Should not be zero.
      0730'  FE 05                     CP 5                    ;A should be in [1..4]
      0732'  30 0D                     JR NC,CHANERR
                                ;
                                ;Here if valid channel. A = 1 to 4.
                                ;
      0734'                     CHNLIX:
      0734'  01 0028            GETCH2:   LD BC,40
      0737'  DD 21 D818                   LD IX,CHNL1-40
      073B'  DD 09              GETCH3:   ADD IX,BC
      073D'  3D                           DEC A
      073E'  20 FB                        JR NZ,GETCH3
      0740'  C9                           RET
                                ;
                                ;Here if channel no. out of range.
                                ;
      0741'                     CHANERR:
      0741'  C3 07C3'                     JP TOOBIG
                                ;
                                ;
                                ;
                                ;
                                ;Syntax: EOF£<chan.No.>,<line no>
                                ;
                                ;
                                ;
7062  0744'  CD 0B43'            EOF::     CALL USRCHK
      0747'  13                           INC DE
      0748'  CD 0725'                     CALL GETCHAN
      074B'  20 F4                        JR NZ,CHANERR   ; JMP if param.> 64K
                                ;
      074D'  DD CB 27 7E                  BIT 7,(IX+39)
      0751'  C8                           RET Z
      0752'  CD 0000*                     CALL PAGE0
      0755'  0B                           DB 11
                                ;
                                ;
                                ;
                                ;
                                ;GETFNAM evaluates string expression addressed by DE and, if valid,
                                ;initialises FCB addressed by HL. Error if bad filename. Preserves HL,IX.
                                ;
7073  0756'                     GETFNAM:
      0756'  DD E5                        PUSH IX
      0758'  E1                           POP HL
      0759'  CD 07FB'                     CALL INITFCB            ;Clear FCB (preserves everything)
      075C'  E5                 GETFNp:   PUSH HL
      075D'  E5                           PUSH HL
```

```
075E'  CD 0000*                    CALL PAGE0
0761'  02                          DB 2                    ;CALL EVALSE
0762'  E1                          POP HL
0763'  13                          INC DE
0764'  D5                          PUSH DE
0765'  E5                          PUSH HL
                          ;        CALL FIND1$             ;BC = string length, DE -> start of string.
0766'  CD 0000*                    CALL PAGE0
0769'  03                          DB 3
076A'  ED 53 FA81                  LD (CALCSTV),DE
076E'  78                          LD A,B
076F'  A7                          AND A
0770'  20 B9                       JR NZ,GETCH1            ;Jump if string too large
                                   ;
                                   ;Now see whether string is a valid filename. C = string length.
                                   ;
0772'  E1                          GETFN1: POP HL          ;HL -> FCB
0773'  CD 0810'                           CALL PARSE      ;NZ if bad filename (preserves HL)
0776'  20 1B                       JR NZ,FNAMERR
0778'  CD 08AB'                    CALL TESTSPA
                                   ;
                                   ;Here if valid filename. FCB contains drive, file name, file type.
                                   ;HL -> FCB. Now test for ambiguous file name.
                                   ;
077B'  CD 0893'                    GETFN2: CALL CHKAMB     ;Z if ambiguous
077E'  D1                                  POP DE
077F'  DD E1                               POP IX
0781'  C9                                  RET
                                   ;
                                   ;
                                   ;
0782'  DD E5                       GETNUL: PUSH IX
0784'  E1                                  POP HL
0785'  E5                                  PUSH HL
0786'  D5                                  PUSH DE
0787'  CD 07FB'                            CALL INITFCB
078A'  CD 08AB'                            CALL TESTSPA
078D'  18 EC                               JR GETFN2
                                   ;
                                   ;  Unambiguous file name
                                   ;
078F'  CD 0756'                    GETUFN: CALL GETFNAM
0792'  C0                                  RET NZ
                                   ;
                                   ;
                                   ;Here if bad filename.
                                   ;
7096   0793'                       FNAMERR:
0793'  EF                                  RST ERRRST
0794'  30                                  DB 48+0
                                   ;
                                   ;
                                   ;
                                   ;GETTYPE returns A = 1 if string expression at (DE) = "I".
                                   ;
                                   ;The type byte (IX+39) has the following meaning:
```

```
                        ;
                        ; BIT    Meanig
                        ; ---    ------
                        ;  0      1 - READ ENABLE
                        ;  1      1 - WRITE ENABLE
                        ;  2      1 - RAW DATA
                        ;  3
                        ;  4
                        ;  5
                        ;  6      1 - Reading unwriten data.
                        ;  7      1 - EOF condition meet.
                        ;
                        ;
                        ;Preserves IX.
                        ;
0795'                   GETTYPE:
0795'  DD E5                    PUSH IX
0797'  CD 0000*                 CALL PAGE0
079A'  02                       DB 2                    ;CALL EVALSE
079B'  D5                       PUSH DE
079C'  CD 0000*                 CALL PAGE0
079F'  03                       DB 3            ;CALL FIND1$,BC = string length, DE -> start of string
07A0'  ED 53 FA81               LD (CALCST),DE          ;Re-set stack pointer.
07A4'  78                       LD A,B
07A5'  B1                       OR C
07A6'  28 03                    JR Z,GETCH1
                        ;
                        ;DE -> character. Check for 'I' or 'O' or 'R'. C = 1.
                        ;
07A8'  1A               GETTY1: LD A,(DE)
07A9'  FE 49                    CP 'I'
07AB'  28 0A                    JR Z,GETTY2
07AD'  0C                       INC C
07AE'  FE 4F                    CP 'O'
07B0'  28 05                    JR Z,GETTY2
07B2'  0C                       INC C
07B3'  FE 52                    CP 'R'
07B5'  20 8A                    JR NZ,CHANERR           ;Jump if error
                        ;
07B7'  79               GETTY2: LD A,C
07B8'  D1                       POP DE
07B9'  DD E1                    POP IX
07BB'  C9                       RET
                        ;
                        ;
                        ;
                        ;GETNXT calls GETNEXT, and returns only if number positive and less than 64k.
                        ;
07BC'  CD 0000*         GETNXT: CALL PAGE0              ;RST GETRST
07BF'  0A                       DB 10
07C0'  20 01                    JR NZ,TOOBIG            ;Jump if > 64K  ??????????
07C2'  D0                       RET NC                  ;Return if not negative
                        ;
07C3'  EF               TOOBIG: RST ERRRST              ;"OUT OF RANGE".
07C4'  22                       DB 34
                        ;
```

```
                        ;
                        ;
                        ;GETREC evaluates numeric expression at (DE) as a record no.
                        ;On exit, ABC = record no. if in range, else error.
                        ;
07C5'  CD 0000*         GETREC::CALL PAGE0      ;CALL EVALAB
07C8'  01                       DB 1
07C9'  EF                       RST JT
07CA'  81                       DB 81H
07CB'  CD 0000*                 CALL PAGE0      ;CALL INT;(ACC1) now contains an integer
07CE'  06                       DB 6
07CF'  21 FDCF                  LD HL,ACC1+3            ;HL -> high byte of mantissa
07D2'  CB 7E                    BIT 7,(HL)
07D4'  CB FE                    SET 7,(HL)
07D6'  20 EB                    JR NZ,TOOBIG            ;Jump if negative no.
07D8'  AF                       XOR A
07D9'  47                       LD B,A
07DA'  4F                       LD C,A
07DB'  23                       INC HL                  ;HL -> exponent byte
07DC'  CB 7E                    BIT 7,(HL)
07DE'  CB BE                    RES 7,(HL)
07E0'  C8                       RET Z                   ;Return if no. = 0
07E1'  D5                       PUSH DE
07E2'  3E 18                    LD A,24
07E4'  96                       SUB (HL)                ;Z if no shifting necessary
07E5'  38 DC                    JR C,TOOBIG
07E7'  57                       LD D,A                  ;D = counter for shifting bits (23 >= D >= 0)
07E8'  2B                       DEC HL
07E9'  7E                       LD A,(HL)
07EA'  2B                       DEC HL
07EB'  46                       LD B,(HL)
07EC'  2B                       DEC HL
07ED'  4E                       LD C,(HL)
07EE'  28 09                    JR Z,GETRC2
                        ;
                        ;Now shift right D times A -> B -> C.
                        ;
07F0'  CB 3F            GETRC1: SRL A
07F2'  CB 18                    RR B
07F4'  CB 19                    RR C
07F6'  15                       DEC D
07F7'  20 F7                    JR NZ,GETRC1
07F9'  D1               GETRC2: POP DE
07FA'  C9                       RET
                        ;
                        ;
                        ;
                        ;INITFCB fills filename in FCB at (HL) with spaces and everything else
                        ;with zeroes. Preserves all registers.
                        ;
07FB'                   INITFCB:
07FB'  C5                       PUSH BC
07FC'  E5                       PUSH HL
07FD'  36 00                    LD (HL),0
07FF'  06 0B                    LD B,11
0801'  23               INITF1: INC HL
```

```
0802'  36 20                    LD (HL),' '
0804'  10 FB                    DJNZ INITF1
0806'  06 1D                    LD B,29
0808'  23               INITF2: INC HL
0809'  36 00                    LD (HL),0
080B'  10 FB                    DJNZ INITF2
080D'  E1                       POP HL
080E'  C1                       POP BC
080F'  C9                       RET
                        ;
                        ;
                        ;
                        ;PARSE checks whether string at (DE) of length C is a valid filename.
                        ;If so, copies name into FCB at (HL), and returns Z.
                        ;If name not valid, returns NZ. Preserves HL,IX.
                        ;
0810'  79               PARSE:  LD A,C
0811'  A7                       AND A
0812'  C8                       RET Z                  ;Return if string of zero length
0813'  E5                       PUSH HL                ;Save HL -> FCB
0814'  EB                       EX DE,HL
0815'  CD 081A'                 CALL PARS1
0818'  E1                       POP HL
0819'  C9                       RET
                        ;
                        ;Main parsing routine.
                        ;
081A'  E5               PARS1:  PUSH HL
081B'  C5                       PUSH BC
081C'  CD 088F'                 CALL GETCHR
081F'  28 13                    JR Z,PARS2             ;Jump if only one character in line
                        ;
                        ;Test for drive letter.
                        ;
0821'  CD 088F'                 CALL GETCHR            ;A = second char
0824'  FE 3A                    CP ':'
0826'  20 0C                    JR NZ,PARS2            ;Jump if no drive specified
                        ;
                        ;Test whether drive letter in range.
                        ;
0828'  F1                       POP AF                 ;Clear stack
0829'  E3                       EX (SP),HL             ;(SP) -> char after ':', HL -> drive char
082A'  7E                       LD A,(HL)
082B'  CD 0885'                 CALL TESTDRV
082E'  DA 0793'                 JP C,FNAMERR           ;Jump if bad drive name
0831'  12                       LD (DE),A              ;Store drive
0832'  18 01                    JR PARS3
                        ;
0834'  C1               PARS2:  POP BC                 ;C = no. of chars left
0835'  E1               PARS3:  POP HL                 ;HL -> first char of filename in string
                        ;
                        ;Now get file name.
                        ;
0836'  13                       INC DE                 ;DE -> start of filename in FCB
0837'  06 08                    LD B,8                 ;B = length of name field
                        ;
```

```
0839'  79              PARS4:  LD A,C
083A'  A7                      AND A
083B'  C8                      RET Z
083C'  CD 088F'                CALL GETCHR
083F'  FE 20                   CP ' '
0841'  C8                      RET Z                   ;Space terminates filename
0842'  FE 2A                   CP '*'
0844'  28 0A                   JR Z,PARS5              ;Jump if wildcard
0846'  FE 2E                   CP '.'
0848'  28 0E                   JR Z,PARS70             ;Jump if field terminator
084A'  12                      LD (DE),A               ;Store char
084B'  13                      INC DE
084C'  10 EB                   DJNZ PARS4
084E'  18 0D                   JR PARS8
                       ;
                       ;Here if char is wildcard.
                       ;
0850'  3E 3F           PARS5:  LD A,'?'
0852'  12              PARS6:  LD (DE),A
0853'  13                      INC DE
0854'  10 FC                   DJNZ PARS6
0856'  18 05                   JR PARS8
                       ;
                       ;
                       ;Here if at end of name field.
                       ;
0858'  0C              PARS70: INC C
0859'  2B              PARS7:  DEC HL
085A'  13              PARS7a: INC DE
085B'  10 FD                   DJNZ PARS7a
                       ;
                       ;Name field has been done, so now look for type field.
                       ;
085D'  79              PARS8:  LD A,C
085E'  A7                      AND A
085F'  C8                      RET Z                   ;Return if no more chars
                       ;
0860'  CD 088F'                CALL GETCHR             ;A should be '.'
0863'  FE 2E                   CP '.'
0865'  C0                      RET NZ                  ;Return - bad filename
                       ;
                       ;Now get file type.
                       ;
0866'  06 03           PARS9:  LD B,3
                       ;
0868'  79              PARS10: LD A,C
0869'  A7                      AND A
086A'  C8                      RET Z                   ;Return if nothing after '.'
086B'  CD 088F'                CALL GETCHR
086E'  FE 20                   CP ' '
0870'  C8                      RET Z                   ;Space terminates filename
0871'  FE 2A                   CP '*'
0873'  28 09                   JR Z,PARS11             ;Jump if wildcard
0875'  FE 2E                   CP '.'
0877'  C8                      RET Z
0878'  12                      LD (DE),A               ;Store char
```

```
0879'  13                       INC DE
087A'  10 EC                    DJNZ PARS10
087C'  AF                       XOR A                   ;Z
087D'  C9                       RET
                        ;
                        ;Here if wildcard. Fill remainder of field with '?'s
                        ;
087E'  3E 3F            PARS11: LD A,'?'
0880'  12               PARS12: LD (DE),A
0881'  13                       INC DE
0882'  10 FC                    DJNZ PARS12
0884'  C9                       RET
                        ;
                        ;
                        ;
                        ;TESTDRV tests for valid drive letter ('A' - 'P') in A, then subtracts 'A' - 1.
                        ;Returns C if drive not valid.
                        ;
0885'                   TESTDRV:
0885'  FE 41                    CP 'A'
0887'  D8                       RET C
0888'  FE 51                    CP 'P'+1
088A'  3F                       CCF
088B'  D8                       RET C
088C'  D6 40                    SUB 'A'-1               ;'A' -> 1, 'B' -> 2, etc
088E'  C9                       RET
                        ;
                        ;
                        ;
                        ;GETCHR returns in A character at (HL).
                        ;Increments HL, decrements C. Z if C = 0.
                        ;
088F'  7E               GETCHR: LD A,(HL)
0890'  23                       INC HL
0891'  0D                       DEC C
0892'  C9                       RET
                        ;
                        ;
                        ;
                        ;CHKAMB checks for ambiguous filename in FCB.
                        ;Returns Z if ambiguous. Destroys A.
                        ;
0893'  C5               CHKAMB: PUSH BC
0894'  E5                       PUSH HL
0895'  3E 3F                    LD A,'?'
0897'  01 000B                  LD BC,11
089A'  23                       INC HL
089B'  ED B1                    CPIR
089D'  E1                       POP HL
089E'  C1                       POP BC
089F'  C9                       RET
                        ;
                        ;
                        ;
                        ;
                        ; Routine to test for ocurence of A in 11 bytes starting at HL+1.
```

```
                        ;Returns NZ if found. Preserves HL,C, and DE.
                        ;
08A0'                   TESTCHR:
08A0'  E5                       PUSH HL
08A1'  06 0B                    LD B,11
08A3'  23               TESTQ1: INC HL
08A4'  BE                       CP (HL)
08A5'  20 02                    JR NZ,TESTQ2
08A7'  10 FA                    DJNZ TESTQ1
08A9'  E1               TESTQ2: POP HL
08AA'  C9                       RET
                        ;
                        ;
                        ;
                        ;TESTSPA tests filename in FCB at (HL) for all spaces.
                        ;If so, changes spaces to '?'s and returns Z, else returns NZ.
                        ;Preserves DE,HL.
                        ;
08AB'                   TESTSPA:
08AB'  3E 20                    LD A,' '
08AD'  CD 08A0'                 CALL TESTCHR
08B0'  C0                       RET NZ
08B1'  E5                       PUSH HL
08B2'  06 0B                    LD B,11
08B4'  23               TESTS2: INC HL
08B5'  36 3F                    LD (HL),'?'
08B7'  10 FB                    DJNZ TESTS2
08B9'  AF                       XOR A
08BA'  E1                       POP HL
08BB'  C9                       RET
                        ;
                        ;
                        ; Returns NZ if current file is random.
                        ;
08BC'                   TESTRND:
08BC'  DD 7E 27                 LD A,(IX+39)
08BF'  CB 47                    BIT 0,A
08C1'  C8                       RET Z
08C2'  CB 4F                    BIT 1,A
08C4'  C9                       RET
                        ;
                        ;Returns C set if new record biger than file size.
                        ;
08C5'  DD 7E 29         STEST:: LD A,(IX+41)
08C8'  DD BE 22                 CP (IX+34)
08CB'  C0                       RET NZ
08CC'  DD 7E 28                 LD A,(IX+40)
08CF'  DD BE 21                 CP (IX+33)
08D2'  C9                       RET
                        ;
                        ;
                        ;
08D3'  DD 7E 21         SWOP:   LD A,(IX+33)
08D6'  DD 77 28                 LD (IX+40),A
08D9'  DD 7E 22                 LD A,(IX+34)
08DC'  DD 77 29                 LD (IX+41),A
```

```
08DF'  C9                      RET

                               ;
                               ;
                               ;Syntax: USER COPY "newfile"="oldfile"
                               ;
                               ;
08E0'  01 D3 01                DB      1,EQ,1
                               ;
08E3'  D7                      COPY:   RST 10H
08E4'  4F                              DB 40H+8+7      ;Selecte VS,CLS,7
08E5'  CD 0B43'                        CALL USRCHK
08E8'  13                              INC DE
08E9'  AF                              XOR A
08EA'  32 D894                         LD (FLAG),A
08ED'  DD 21 D8E3                      LD IX,CHNL3
08F1'  CD 078F'                        CALL GETUFN             ;destination.
                               ;
08F4'  CD 0107'                        CALL SAVETYP            ;PRESERVE type.
08F7'  CD 00DA'                        CALL PUT$               ;Change type to $$$.
                               ;
08FA'  DD 21 D8BE                      LD IX,CHNL4
08FE'  CD 078F'                        CALL GETUFN             ;Source.
                               ;
0901'  CD 0968'                        CALL PORIG              ;request source.
0904'  CD 02E6'                        CALL SIOPEN             ;open source file, give error if not exist.
0907'  DD 36 27 05                     LD (IX+39),5
090B'  CD 06A1'                        CALL STARW
                               ;
090E'  CD 09DC'                        CALL GET                ;GET first (or only) , block.
0911'  C5                              PUSH BC
                               ;
0912'  CD 0983'                        CALL PCOPY              ;Request destination disc.
0915'  DD 21 D8E3                      LD IX,CHNL3
0919'  CD 02B3'                        CALL OPENc              ;Open file on disc as $$$ type.
091C'  DD 36 27 06                     LD (IX+39),6
0920'  C1                             POP BC
0921'  CD 0A0D'                       CALL PUT
                               ;
0924'  3A D894                 LOOP:   LD A,(FLAG)
0927'  B7                              OR A
0928'  20 2D                          JR NZ,DONE
                               ;
092A'  CD 06C2'                        CALL STOPRW
092D'  CD 0324'                        CALL IRCLOSE
                               ;
0930'  CD 0968'                        CALL PORIG
                               ;
0933'  CD 02E6'                        CALL SIOPEN
0936'  DD 36 27 05                     LD (IX+39),5
093A'  CD 06A1'                        CALL STARW
                               ;
093D'  CD 09DC'                        CALL GET
0940'  C5                              PUSH BC
0941'  CD 0983'                        CALL PCOPY
                               ;
```

```
0944'  DD 21 D8E3          LD IX,CHNL5
0948'  0E 0F               LD C,DOPEN
094A'  CD 066B'            CALL DOS
094D'  DD 36 27 06         LD (IX+39),6
                ;
0951'  C1                  POP BC
0952'  CD 0A0D'            CALL PUT
0955'  18 CD               JR LOOP
                ;
0957'  CD 06C2'     DONE:  CALL STOPRW
095A'  CD 0324'            CALL IRCLOSE
095D'  CD 0167'            CALL SAVEc      ;Rename $$$ file to be of original type.
0960'  0E 0D               LD C,13
0962'  CD 0000*            CALL BDOS
0965'  C3 0224             JP NEWINT3
                ;
                ;
0968'  DD 21 D8BE   PORIG: LD IX,CHNL4
096C'  CD 0994'            CALL DMESS1
096F'  CD 09B1'     DISCIN: CALL DMESS2
0972'  CD 0079      DISC1: CALL KBD
0975'  28 FB               JR Z,DISC1
0977'  FE 03               CP 3                    ;Test for BREAK
0979'  28 11               JR Z,COPYX
097B'  D7                  RST 10H
097C'  81 0C               DB 80H+1,FF
097E'  0E 0D               LD C,13
0980'  C3 0000*            JP BDOS
                ;
0983'  DD 21 D8E8   PCOPY: LD IX,CHNL5
0987'  CD 09BF'            CALL DMESS3
098A'  18 E3               JR DISCIN
                ;
                ;Exit point after BREAK
                ;
098C'  0E 0D        COPYX: LD C,13
098E'  CD 0000*            CALL BDOS
0991'  C3 0224             JP NEWINT
                ;
                ;
0994'  D7           DMESS1: RST SCRRST
0995'  9A 0C 49 6E         DB 80H+26,FF,'Insert Source Disc.......'
0999'  73 65 72 74
099D'  20 53 6F 75
09A1'  72 63 65 20
09A5'  44 69 73 63
09A9'  2E 2E 2E 2E
09AD'  2E 2E 2E
09B0'  C9                  RET
                ;
09B1'  D7           DMESS2: RST SCRRST
09B2'  8B 50 72 65         DB 80H+11,'Press a key'
09B6'  73 73 20 61
09BA'  20 6B 65 79
09BE'  C9                  RET
                ;
```

```
09BF'  D7                  DMESS3: RST SCRRST
09C0'  9A 0C 49 6E                 DB 80H+26,FF,'Insert Destination Disc..'
09C4'  73 65 72 74
09C8'  20 44 65 73
09CC'  74 69 6E 61
09D0'  74 69 6F 6E
09D4'  20 44 69 73
09D8'  63 2E 2E
09DB'  C9                          RET
                           ;
                           ;
                           ;GET block from disc. BC returns with number of bytes loaded.
                           ;(FLAG)  is non zero if EOF reached.
                           ;
09DC'  D5                  GET:    PUSH DE
09DD'  01 4000                     LD BC,16*1024
09E0'  DD 21 D8BE                  LD IX,CHNL4
09E4'  21 8000                     LD HL,BUFFER
09E7'  78                  GET1:   LD A,B
09E8'  B1                          OR C
09E9'  28 18                       JR Z,GET3
09EB'  CD 05F6'                    CALL BGET
09EE'  DD CB 27 7E                 BIT 7,(IX+39)
09F2'  20 05                       JR NZ,GET2
09F4'  77                          LD (HL),A
09F5'  23                          INC HL
09F6'  0B                          DEC BC
09F7'  18 EE                       JR GET1
                           ;
09F9'  3E FF               GET2:   LD A,0FFH
09FB'  32 D894                     LD (FLAG),A
09FE'  C5                          PUSH BC
09FF'  CD 0324'                    CALL IRCLOSE
0A02'  C1                          POP BC
                           ;
0A03'  21 4000             GET3:   LD HL,16*1024
0A06'  B7                          OR A
0A07'  ED 42                       SBC HL,BC
0A09'  44                          LD B,H
0A0A'  4D                          LD C,L
0A0B'  D1                          POP DE
0A0C'  C9                          RET
                           ;
                           ;
                           ;
0A0D'  DD 21 D8E8          PUT:    LD IX,CHNL5
0A11'  21 8000                     LD HL,BUFFER
0A14'  78                  DPUT:   LD A,B
0A15'  B1                          OR C
0A16'  C8                          RET Z
0A17'  7E                          LD A,(HL)
0A18'  23                          INC HL
0A19'  0B                          DEC BC
                           ;
0A1A'  C5                          PUSH BC
0A1B'  E5                          PUSH HL
```

```
0A1C'  CD 0679'                CALL CALPOS
0A1F'  77                      LD (HL),A
0A20'  DD 34 24                INC (IX+36)
0A23'  F2 0A3F'                JP P,DPUT1          ;Jump DMA not full
                       ;
0A26'  0E 22                   LD C,RWRITE
0A28'  CD 066B'                CALL DOS
                       ;
0A2B'  3C                      INC A
0A2C'  20 02                   JR NZ,DPUTQ
0A2E'  EF                      RST ERRRST
0A2F'  23                      DB 35
                       ;
0A30'  CD 0685'        DPUTQ:  CALL INCRR
0A33'  DD 36 24 00             LD (IX+36),0
0A37'  CD 08C5'                CALL STEST
0A3A'  30 03                   JR NC,DPUT1
0A3C'  CD 08D3'                CALL SWOP
                       ;
0A3F'  E1              DPUT1:  POP HL
0A40'  C1                      POP BC
0A41'  C3 0A14'                JP DPUT
                       ;
                       ;
                       ; Routine to write out BC bytes  starting at (HL) to a file whos
                       ;FCB is pointed to by IX.
                       ;
                       ;
                       ;SYSCOPY Copies 1st 52 sectors from source to destination discs
                       ;
0A44'  C9                      RET                 ;Syntax byte
0A45'                  SYSCOPY:
0A45'  D7                      RST 10H
0A46'  4F                      DB 40H+8+7          ;Select VS 7, CLS
0A47'  CD 0A79'                CALL SRCMESS
0A4A'  CD 0AB9'                CALL SET0
0A4D'  06 34                   LD B,52
0A4F'  CD 0000*                CALL BLKRD
0A52'  28 02                   JR Z,SYS1
0A54'  EF                      RST ERRRST
0A55'  3B                      DB 48+11
0A56'  CD 0A88'        SYS1:   CALL DSTMESS
0A59'  CD 0AB9'                CALL SET0
0A5C'  06 34                   LD B,52
0A5E'  CD 0AD5'                CALL BLKWR
0A61'  28 02                   JR Z,SYS2
0A63'  EF                      RST ERRRST
0A64'  3B                      DB 48+11
0A65'  CD 0A8D'        SYS2:   CALL EXITMESS
0A68'  CD 0079         SYS3:   CALL KBD
0A6B'  28 FB                   JR Z,SYS3
0A6D'  FE 0D                   CP CR
0A6F'  28 E5                   JR Z,SYS1
0A71'  0E 0D                   LD C,0DH
0A73'  CD 0000*                CALL BDOS           ;Reset disc system
0A76'  C3 0224                 JP NEWINT
```

```
0A79'                           ;
0A79'                           SRCMESS:
0A79'  CD 0994'                         CALL DMESS1
0A7C'  CD 09B1'                 SRCM1:  CALL DMESS2
0A7F'  CD 0079                  SRCM2:  CALL KBD
0A82'  28 FB                            JR Z,SRCM2
0A84'  D7                               RST 10H
0A85'  81 0C                            DB 80H+1,FF
0A87'  C9                               RET
0A88'                           ;
0A88'  CD 09BF'                 DSTMESS:
0A8B'  18 EF                            CALL DMESS3
                                        JR SRCM1
0A8D'                           ;
0A8D'  D7                       EXITMESS:
0A8E'  99 0C 52 45                      RST SCRRST
0A92'  54 20 74 6F                      DB 80H+25,FF,'RET to continue, any oth'
0A96'  20 63 6F 6E
0A9A'  74 69 6E 75
0A9E'  65 2C 20 61
0AA2'  6E 79 20 6F
0AA6'  74 68
0AA8'  D7                               RST SCRRST
0AA9'  8E 65 72 20                      DB 80H+14,'er key to quit'
0AAD'  6B 65 79 20
0AB1'  74 6F 20 71
0AB5'  75 69 74
0AB8'  C9                               RET
0AB9'  0E 0D                    ;
0ABB'  CD 0000*                 SET0:   LD C,0DH
0ABE'  AF                               CALL BDOS
0ABF'  32 0000*                         XOR A
0AC2'  21 0000                          LD (DRVRQ),A
0AC5'  22 0000*                         LD HL,0
0AC8'  21 0001                          LD (TRKRQ),HL
0ACB'  22 0000*                         LD HL,1
0ACE'  21 8000                          LD (SECRQ),HL
0AD1'  22 0000*                         LD HL,BUFFER
0AD4'  C9                               LD (DMARQ),HL
                                        RET
0AD5'  C5                       ;
0AD6'  06 0A                    BLKWR:  PUSH BC
0AD8'  05                               LD B,10
0AD9'  28 22                    BLKWP:  DEC B
0ADB'  CD 0000*                         JR Z,BLKWE
0ADE'  B7                               CALL EXWR
0ADF'  20 F7                            OR A
0AE1'  2A 0000*                         JR NZ,BLKWP
0AE4'  11 0080                          LD HL,(DMARQ)
0AE7'  19                               LD DE,128
0AE8'  22 0000*                         ADD HL,DE
0AEB'  3A 0000*                         LD (DMARQ),HL
0AEE'  3C                               LD A,(SECRQ)
0AEF'  FE 1B                            INC A
                                        CP 27
```

```
0AF1'  CC 0B02'              CALL Z,NTRK
0AF4'  32 0000*              LD (SECRQ),A
0AF7'  C1                    POP BC
0AF8'  05                    DEC B
0AF9'  20 DA                 JR NZ,BLKWR
0AFB'  AF                    XOR A
0AFC'  C9                    RET
0AFD'  3E 01          BLKWE: LD A,1
0AFF'  C1                    POP BC
0B00'  B7                    OR A
0B01'  C9                    RET
0B02'  3A 0000*       NTRK:  LD A,(TRKRQ)
0B05'  3C                    INC A
0B06'  32 0000*              LD (TRKRQ),A
0B09'  3E 01                 LD A,1
0B0B'  C9                    RET
                      ;
                      ;
                      ;
                      ;REBOOT::CALL WBOOT
                      ;       LD C,13
                      ;       CALL BDOS
                      ;       LD A,(DDRIVE)
                      ;       LD E,A
                      ;       LD C,14
                      ;       CALL BDOS
                      ;       JP BASIC2
                      ;
                      ;DDRIVE::DS 1
                      ;
                      ;
                      ;
                      ;       ========================
                      ;       =     Linking Module   =
                      ;       ========================
                      ;
7EF8  0B0C'  CD 0B43' DUSER:: CALL USRCHK
0B0F'  28 02                 JR Z,DUSER1
                      ;
0B11'  EF                    RST ERRRST             ; Statement not recognised.
0B12'  31                    DB 48+1
                      ;
0B13'  CD 0000*       DUSER1: CALL JPLINK
0B16'  E9                    JP (HL)
                      ;
7F04  0B17'           JMPTAB::
0B17'  0187'                 DW      DLOAD     7858
0B19'  0110'                 DW      DSAVE     7708
0B1B'  04C2'                 DW      DPRINT    7B38
0B1D'  0458'                 DW      DINPUT    7AC8
0B1F'  044E'                 DW      DLINPUT   7A8E
0B21'  0269'                 DW      OPEN      78E7
0B23'  02F5'                 DW      CLOSE     7933
0B25'  0342'                 DW      KILL      7987
0B27'  057F'                 DW      READ      78E8
0B29'  059F'                 DW      WRITE     7C08
```

```
0B2B' 004E'                     DW      DIR     7748
0B2D' 00A4'                     DW      ERASE   7797
0B2F' 035F'                     DW      ETYPE   79C8
0B31' 03F4'                     DW      RECORD  7A64
0B33' 0386'                     DW      REN     79F0
0B35' 03C9'                     DW      RUN     7A38
0B37' 000D'                     DW      QUIT    770E
0B39' 0744'                     DW      EOF     7D62
0B3B' 08E3'                     DW      COPY
                        ;
0B3D' 0000*                     DW      SFORMAT
0B3F' 0A45'                     DW      SYSCOPY
0B41' 0000*                     DW      STAT
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        .COMMENT %
                        SYNPAT: CP 87H                  ;Test for DISC token
                                JR Z,SYNPT1             ;Jump if DISC token
                        ;       LD A,0
                        ;       LD (PAGE),A
                        ;       OUT (0),A
                                LD HL,RWTAB
                                JP FINDJP
                        ;
                        ;Here if DISC entered. HL -> DISC token.
                        ;
                        SYNPT1: POP IX                  ;Pop return address
                                POP HL                  ;HL -> DISC token, (SP) -> return address
                                INC HL
                                RST OSPRST              ;HL -> first non-space char
                                EX DE,HL                ;DE -> first non-space char after DISC token
                                CALL USRCHK             ;Check for valid DISC command
                                JP NZ,SYNER1            ;Jump if not valid.
                        ;
                        ;Here, HL -> DISC routine, DE -> token or last letter of command word.
                        ;
                                PUSH DE
                                JP (IX)                 ;Return
                        %
                        ;
                        ;
                        ;
                        ;USRCHK checks line pointed to by DE for valid DISC command.
                        ;Returns Z if valid, else NZ.
                        ;
5768 0B43'              USRCHK::
                                LD A,(DE)
0B43' 1A
0B44' CB 7F                     BIT 7,A
```

```
0B46'  20 27                    JR NZ,USRCH1
                                ;
0B48'  06 FF                    LD B,-1
0B4A'  21 0B99'                 LD HL,DWORDLST
0B4D'  D5               USRCHa: PUSH DE
0B4E'  04                       INC B
0B4F'  7E               USRCH0: LD A,(HL)
0B50'  E6 7F                    AND 7FH
0B52'  EB                       EX DE,HL
0B53'  BE                       CP (HL)
0B54'  EB                       EX DE,HL
0B55'  20 0C                    JR NZ,MISMATCH
0B57'  CB 7E                    BIT 7,(HL)
0B59'  20 04                    JR NZ,MATCH
0B5B'  13                       INC DE
0B5C'  23                       INC HL
0B5D'  18 F0                    JR USRCH0
                                ;
0B5F'  F1               MATCH:  POP AF
0B60'  78                       LD A,B
0B61'  18 0C                    JR USRCH1
                                ;
0B63'                   MISMATCH:
0B63'  CB 7E                    BIT 7,(HL)
0B65'  23                       INC HL
0B66'  28 FB                    JR Z,MISMATCH
0B68'  7E                       LD A,(HL)
0B69'  A7                       AND A
0B6A'  D1                       POP DE
0B6B'  20 E0                    JR NZ,USRCHa
0B6D'  3C                       INC A
0B6E'  C9                       RET
                                ;
0B6F'  21 0B98'         USRCH1: LD HL,TOKTAB
0B72'  01 0016                  LD BC,NUBTOK
0B75'  ED B9                    CPDR
0B77'  C0                       RET NZ
0B78'  21 0B17'                 LD HL,JMPTAB
0B7B'  09                       ADD HL,BC
0B7C'  09                       ADD HL,BC
0B7D'  D5                       PUSH DE
0B7E'  CF                       RST DEHL
0B7F'  EB                       EX DE,HL
0B80'  D1                       POP DE
0B81'  BF                       CP A
0B82'  C9                       RET
                                ;
                                ;
0016                    NUBTOK  EQU     22
                                ;
0B83'  9E                       DB 9EH          ;LOAD
0B84'  B5                       DB 0B5H         ;SAVE
0B85'  90                       DB 90H          ;PRINT
0B86'  98                       DB 98H          ;INPUT
0B87'  C1                       DB 0C1H         ;LINE
0B88'  00                       DB 0            ;OPEN
```

```
0B89'  01                       DB 1                    ;CLOSE
0B8A'  02                       DB 2                    ;KILL
0B8B'  B0                       DB 0B0H                 ;READ
0B8C'  03                       DB 3                    ;WRITE
0B8D'  04                       DB 4                    ;DIR
0B8E'  05                       DB 5                    ;ERA
0B8F'  06                       DB 6                    ;TYPE
0B90'  07                       DB 7                    ;REC
0B91'  08                       DB 8                    ;REN
0B92'  B4                       DB 0B4H                 ;RUN
0B93'  09                       DB 9                    ;QUIT
0B94'  0A                       DB 10                   ;EOF
0B95'  0B                       DB 11                   ;COPY
                        ;
0B96'  0C                       DB 12                   ;FORMAT
0B97'  0D                       DB 13                   ;SYSCOPY
0B98'  0E               TOKTAB: DB 14                   ;STAT
                        ;
0B99'                   DWORDLST:
0B99'  4F 50 45 CE              DC 'OPEN'
0B9D'  43 4C 4F 53              DC 'CLOSE'
0BA1'  C5
0BA2'  4B 49 4C CC              DC 'KILL'
0BA6'  57 52 49 54              DC 'WRITE'
0BAA'  C5
0BAB'  44 49 D2                 DC 'DIR'
0BAE'  45 52 C1                 DC 'ERA'
0BB1'  54 59 50 C5              DC 'TYPE'
0BB5'  52 45 C3                 DC 'REC'
0BB8'  52 45 CE                 DC 'REN'
0BBB'  51 55 49 D4              DC 'QUIT'
0BBF'  45 4F C6                 DC 'EOF'
0BC2'  43 4F 50 D9              DC 'COPY'
0BC6'  46 4F 52 4D              DC 'FORMAT'
0BCA'  41 D4
0BCC'  53 59 53 43              DC 'SYSCOPY'
0BD0'  4F 50 D9
0BD3'  53 54 41 D4              DC 'STAT'
0BD7'  00                       DB 0
                        ;
                        ;
                        ;SYNERR is called when first char in BASIC statement is not a token.
                        ;Inserts LET or GOTO token into line and tries syntax check again.
                        ;
0BD8'  FE 30            SYNERR: CP '0'
0BDA'  38 08                    JR C,SYNER1             ;Error if char < '0'
0BDC'  FE 3A                    CP '9'+1
0BDE'  30 06                    JR NC,SYNER2            ;Jump if not a number
0BE0'  3E 96                    LD A,96H                ;GOTO token
0BE2'  18 0C                    JR SYNER3
                        ;
                        ;Here if syntax error.
                        ;
0BE4'  EF               SYNER1: RST ERRRST
0BE5'  01                       DB 1                    ;'Mistake'
```

```
                        ;
                        ;Here if first char in line is not a number.
                        ;Should be a capital letter.
                        ;
0BE6'  FE 41            SYNER2: CP 'A'
0BE8'  38 FA                    JR C,SYNER1
0BEA'  FE 5B                    CP 'Z'+1
0BEC'  30 F6                    JR NC,SYNER1            ;Error if not a letter
0BEE'  3E 9C                    LD A,9CH                ;LET token
                        ;
                        ;Now insert GOTO or LET token. HL = address for insertion.
                        ;
0BF0'  F5               SYNER3: PUSH AF                 ;Save token
0BF1'  CD 1B43                  CALL CALLEN             ;BC = length of line upto 0FFH. Preserves HL
0BF4'  09                       ADD HL,BC               ;HL -> first byte after 0FFH
0BF5'  54                       LD D,H
0BF6'  5D                       LD E,L
0BF7'  2B                       DEC HL
0BF8'  ED B8                    LDDR
0BFA'  EB                       EX DE,HL                ;HL -> address for token
0BFB'  F1                       POP AF
0BFC'  77                       LD (HL),A               ;Store token
0BFD'  C9                       RET
                        ;
                        ;
                        ;
                        ;
                        ;
                        END
```

Macros:

Symbols:

ACC1	FDCC	ADJVAL	C4F4	ADLOAD	0190I'	AE	3C45
ALPHNM	0703'	ARRTOP	FACC	AUTORU	0190I'	BASIC2	0250
BDOS	0ABC*	BGET	05F6'	BGET1	0600'	BGET2	061C'
BGET3	0628'	BGET4	062C'	BLKRD	0A50*	BLKWE	0AFD'
BLKWP	0AD8'	BLKWR	0AD5'	BPUT	05BE'	BPUT1	05E8'
BPUTQ	05D4'	BREAKM	09F2	BUFFER	8000	CALCBO	FA7F
CALCST	FA81	CALLEN	1B43	CALPOS	0679'	CALSIZ	017E'
CHANER	0741'	CHKAMB	0893'	CHNL1	D840	CHNL2	D86A
CHNL3	D894	CHNL4	D8BE	CHNL5	D8E8	CHNLIX	0734'
CLOSAL	032DI'	CLOSE	02F5'	CLOSEA	0321'	CLOSEB	0305'
CLSA1	0336'	COMPUT	0023	COPY	08E3'	COPYX	098C'
CR	000D	DATA	0037'	DBSTKL	012AI'	DCLOSE	0010
DECRR	0693'	DECVA	0239'	DEHL	0008	DELETE	0013
DESAVE	FB49	DINPT0	0459'	DINPT1	0471'	DINPT2	047A'
DINPT3	0480'	DINPT4	0487'	DINPUT	0458I'	DIR	004E'
DIR$	0060'	DIR0	0063'	DIR1	0072'	DIR2	0092'
DIRPAD	0084I'	DISC1	0972'	DISCIN	096F'	DLINPU	044EI'
DLOAD	0187'	DMA	E680	DMARQ	0AE9*	DMASET	001A
DMESS1	0994'	DMESS2	09B1'	DMESS3	09BF'	DONE	0957'
DOPEN	000F	DOS	066B'	DOSREN	0017	DPRIN0	04E4'
DPRIN1	04E5'	DPRIN2	04F8'	DPRIN3	0517'	DPRIN4	0526'
DPRIN5	0530'	DPRIN6	053E'	DPRINT	04C2'	DPRINZ	04D8'
DPUT	0A14'	DPUT1	0A3F'	DPUTQ	0A30'	DRVRQ	0AC0*
DSAVE	0110'	DSTMES	0A88'	DUSER	0B0CI'	DUSER1	0B13'
DWORDL	0B99'	EOF	0744I'	EOL	065F'	EOL1	0638'
EQ	00D3	ERASE	00A4'	ERASE1	00C4'	ERASE2	00D3'
ERROR	049A'	ERRRST	0028	ETYPE	035FI'	EXITME	0A8D'
EXWR	0ADC*	FCLOS1	031A'	FCLOS2	0324'	FCLOSE	030E'
FF	000C	FILEA2	0009'	FILEAD	0000'	FINDJP	0689
FLAG	D894	FNAMER	0793'	FOPEN	02EC'	GET	09DC'
GET1	09E7'	GET2	09F9'	GET3	0A03'	GETBC	0215'
GETCH0	0729'	GETCH1	072B'	GETCH2	0734'	GETCH3	073B'
GETCHA	0725'	GETCHR	088F'	GETFN1	0772'	GETFN2	077B'
GETFNA	0756I'	GETFNP	075C'	GETNUL	0782'	GETNXT	07BC'
GETRC1	07F0'	GETRC2	07F9'	GETREC	07C5I'	GETRST	0030
GETTY1	07A8'	GETTY2	07B7'	GETTYP	0795'	GETUFN	078FI'
GOTZER	10A6	INCL1	0257'	INCL2	025F'	INCLRA	0242'
INCRR	0685'	INITF1	0801'	INITF2	0808'	INITFC	07FB'
IRCLOS	0324'	IXTEMP	D8F6	JMPTAB	0B17I'	JPHL	07B8
JPLINK	0B14*	JT	0028	KBD	0079	KBDBUF	FA83
KILL	0342'	KILL1	034C'	LDNX1	01D5'	LDNXT	01C7'
LF	000A	LINE$	06CD'	LINPUT	D8F4	LOADTY	00FE'
LOOP	0924'	LOOP1	06D5'	LOOP2	06D8'	LOOP3	06EC'
LOOP4	06F0'	LSTPG	FA7A	MAKE	0016	MATCH	0B5F'
MISMAT	0B63'	MOVNAM	00E6'	MULT	0434'	MULT1	0437'
NEWINT	0224	NEXT	0012	NFERR	02EA'	NOFILE	0097I'
NTRK	0B02'	NUBTOK	0016	NZSET	0719'	OK3	02CC'
OLD	02BC'	OPEN	0269'	OPEN2	02A9'	OPENC	02B3'
OSPRST	0018	PAGE	FAD2	PAGE0	07CC*	PARS1	081A'
PARS10	0868'	PARS11	087E'	PARS12	0880'	PARS2	0834'
PARS3	0835'	PARS4	0839'	PARS5	0850'	PARS6	0852'
PARS7	0859'	PARS70	0858'	PARS7A	085A'	PARS8	085D'
PARS9	0866'	PARSE	0810'	PCOPY	098C'	PGPORT	0000
PORIG	0968'	PRINTX	0CAB	PRNTCH	071D'	PUT	0A0D'

PUT$	00DA'	PUT$1	00DF'	PUT0	0567'	PUT01	0572'
PUTHL	0176'	PUTNXT	0145'	QUIT	000DI'	QUIT1	0027'
RANDIN	049C'	RAWREA	0351'	RAWSKI	0644'	READ	057F'
READI1	0211'	READIN	01FE'	RECORD	03F4I'	REN	0386I'
RENO	039C'	REN1	03AD'	REN2	03BD'	RNDOUT	0544'
RREAD	0021	RUN	03C9I'	RWRITE	0022	RWTAB	26F7
SAVEC	0167'	SAVETY	0107'	SCRRST	0010	SEARCH	0011
SECRQ	0AF5*	SET0	0AB9'	SETDMA	0042I'	SETREC	0407'
SETREG	00F4'	SETVA	021E'	SETZER	10A9	SFORMA	0B3D*
SIOPEN	02E6I'	SKIP	043E'	SNEXT	06FE'	SOPEN	02AE'
SRAMPG	022A'	SRCM1	0A7C'	SRCM2	0A7F'	SRCMES	0A79'
SREAD	0014	STACKS	0744	STARW	06A1'	STARW0	06A5'
STARW2	06AA'	STARW3	06C0'	STAT	0B41*	STEST	08C5I'
STKLIM	FA92	STOPRW	06C2'	STRGET	0632'	STRGT1	0635'
STRGT2	0651'	STRGT3	0655'	STRPUT	05EB'	SWOP	08D3'
SWRITE	0015	SYNER1	0BE4'	SYNER2	0BE6'	SYNER3	0BF0'
SYNERR	0BD8'	SYS1	0A56'	SYS2	0A65'	SYS3	0A68'
SYSCOP	0A45'	SYSTOP	FA94	TESTCH	08A0'	TESTDR	0885'
TESTQ1	08A3'	TESTQ2	08A9'	TESTRN	08BC'	TESTS2	08B4'
TESTSP	08AB'	TOKTAB	0B98'	TOOBIG	07C3'	TRKRQ	0B07*
TYPE1	0371'	USER	FA89	USERSA	D912	USRCH0	0B4F'
USRCH1	0B6F'	USRCHA	0B4D'	USRCHK	0B43I'	VARNAM	FA7B
VAZERO	FD65	WRCRLF	0509'	WRITE	059F'	XLINE	06FB'
ZSET	0717'						

No Fatal error(s)

```
                        ;       ****************************
                        ;       *                          *
                        ;       * SM1 UNIVERSAL FORMATTER *
                        ;       *    HEADER PROGRAM        *
                        ;       *                          *
                        ;       ****************************

                        ;EXTERNAL       F140J           ; DTC140JD CONTROLLER
                        ;EXTERNAL       F5000           ; SMS5000  CONTROLLER
                        ;EXTERNAL       F1791           ; SM21791  CONTROLLER
                        ;EXTERNAL       FSIDI           ; SILICON DISC HANDLERS
                        ;EXTERNAL       F520A           ; DTC520A CONTROLLER
                        ;
0003                    DTYPE   EQU     3       ;DISK TYPE
0000                    F140J   EQU     0
0000                    F5000   EQU     0
0000                    FSIDI   EQU     0
0000                    F520A   EQU     0
                        ;
                        ext drvrq,cfgbyt,cnfg,BDOS
                        ;
0224                    NEWINT  EQU     224H
                        ;
                        .z80
0000'                   CSEG
                        ;
0000'   C9                      RET                     ;Syntax byte
                        ;
0001'                   SFORMAT::
0001'   21 000F'                ld hl,formld
0004'   11 8000                 ld de,8000h
0007'   01 2000                 ld bc,2000h
000A'   ED B0                    ldir
000C'   C3 8000                 jp 8000h

                        .8080

000F'                   formld:
                        .phase 8000h

                        ;DRVRQ  EQU     0FFE8H
                        ;CFGBYT EQU     0FFE9H
                        ;CNFG   EQU     0FFF0H
                        ;VERS   EQU     0FFFFH
0079                    KBD EQU 79H
00BC                    GETSTR EQU 0BCH
                        ;TBUFF  EQU     80H
                        ;BDOS   EQU     5


0007                    BELL    EQU     7
000A                    LF      EQU     10
000D                    CR      EQU     13
```

```
                        ; ------------------------------------------------
                        ;SFORMAT::
                        ;       LDA     TBUFF
                        ;       CPI     3
                        ;       JNZ     ER0             ; COMMAND ERROR
                        ;       LDA     TBUFF+3
                        ;       CPI     ':'
                        ;       JNZ     ER0             ; COMMAND ERROR
                        ;       LDA     TBUFF+2
                        ;       CPI     'B'
                        ;       JC      ER0             ; COMMAND ERROR
                        ;       CPI     'I'+1
                        ;       JNC     ER0             ; COMMAND ERROR
                        ;       DCR     A
                        ;       DCR     A
                        ;       ANI     111B
8000    3E 00                   mvi a,0
8002    32 0000*                STA     DRVRQ

8005    3E 03                   MVI     A,DTYPE         ; INTERROGATE CONFIG
8007    32 0000*                STA     CFGBYT
                        ;       CALL    CNFG
                        ;       LDA     CFGBYT          ; FIND CONFIG BYTE
                        ;       CPI     255
                        ;       JZ      ER0             ; DRIVE NOT CONFIGURED

                        ;       LXI     D,MSGHW


                        ;       LDA     CFGBYT
                        ;       ANI     11110000B
                        ;       CPI     20H
                        ;       JZ      HCFG
                        ;       CPI     30H
                        ;       JZ      HCFG
800A    11 80CC                 LXI     D,MSGRDY

800D    3E 42                   mvi a,66
                        ;HCFG:  LDA     TBUFF+2
800F    32 8123                 STA     FRIG1

8012    3A 0000*                LDA     DRVRQ
                        ;       ANI     100B

8015    3E 03                   mvi a,3         ;version 3 in 0ffff

                        ;       LDA     VERS
                        ;       JZ      SKPVS
                        ;       RRC
                        ;       RRC
                        ;       RRC
                        ;       RRC
                        ;SKPVS: ANI     1111B
                        ;       JZ      ER0             ; FIRMWARE ERROR
                                                        ; DRIVE CONFIG'D BUT NO CNTRLR!
8017    6F                      MOV     L,A
```

```
8018  26 00      MVI     H,0
801A  29         DAD     H
801B  01 8058    LXI     B,SERTAB
801E  09         DAD     B
801F  7E         MOV     A,M
8020  23         INX     H
8021  66         MOV     H,M
8022  6F         MOV     L,A             ; SERVICE ROUTINE ADDR IN HL
8023  B4         ORA     H
8024  CA 8042    JZ      ERO             ; CONTROLLER TYPE NOT SUPPORTED
8027  01 804D    LXI     B,ERROR
802A  C5         PUSH    B               ; SERVICE RTN ERROR HNDLR
802B  E5         PUSH    H               ; SERVICE RTN
              ;  MVI     C,9
802C  CD 8487    CALL    CRTOUT;         BDOS
              ;  MVI     C,1             ; AWAIT ACKN.
802F  CD 8497    CALL    KEYBD           ;BDOS
8032  F5         PUSH    PSW
8033  0E 09      www:    mvi c,9
8035  11 8127           lxi d,mgwait
8038  CD 8487           call CRTOUT       ;bdos
              ;  MVI     E,LF
              ;  MVI     C,2
              ;  CALL    BDOS
803B  F1         POP     PSW             ; LINE FEED
803C  FE 0D      CPI     CR
803E  C8         RZ                      ; GO TO IT
803F  C3 813D    JMP     finish          ; ABORT
              ;
              ;
              ;
              ;
              ;
8042  0E 09      ERO:    MVI     C,9
8044  11 8078            LXI     D,MSGEO
8047  CD 8487            CALL    CRTOUT          ;BDOS
804A  C3 813D            JMP     finish

804D  0E 09      ERROR:  MVI     C,9             ; SERVICE RTN ERROR HNDLR
804F  11 80A8            LXI     D,MSGER         ; JMP 0 --> NO ERROR
8052  CD 8487            CALL    CRTOUT          ;BDOS           ; RET   --> AN ERROR
8055  C3 813D            JMP     finish

                ; ---------------------------------------------------

8058  0000      SERTAB: DW      0               ; FIRMWARE ERROR
805A  0000              DW      F140J           ; DTC140JD CONTROLLER
805C  0000              DW      F5000           ; SMS5000  CONTROLLER
805E  8168              DW      F1791           ; SM21791  CONTROLLER
8060  0000              DW      FSIDI           ; SIDISC HANDLER (4)
8062  0000              DW      F520A           ; DTC 520A CONTROLLER
8064  0000              DW      0
8066  0000              DW      0
8068  0000              DW      0
806A  0000              DW      0
806C  0000              DW      0
```

```
806E  0000                    DW      0
8070  0000                    DW      0
8072  0000                    DW      0
8074  0000                    DW      0
8076  0000                    DW      0

8078  0D 0A 07          MSGE0:  DB      CR,LF,BELL
807B  43 6F 6D 6D             DB      'Command or System error'
807F  61 6E 64 20
8083  6F 72 20 53
8087  79 73 74 65
808B  6D 20 65 72
808F  72 6F 72
8092  0D 0A                   DB      cr,lf
8094  2D 20 6E 6F             DB      '- no action taken'
8098  20 61 63 74
809C  69 6F 6E 20
80A0  74 61 6B 65
80A4  6E
80A5  0D 0A 24                DB      CR,LF,'$'

80A8  0D 0A 07          MSGER:  DB      CR,LF,BELL
80AB  41 4E 20 45             DB      'AN ERROR CONDITION HAS OCCURED'; IN THE SERVICE ROUTINE'
80AF  52 52 4F 52
80B3  20 43 4F 4E
80B7  44 49 54 49
80BB  4F 4E 20 48
80BF  41 53 20 4F
80C3  43 43 55 52
80C7  45 44
80C9  0D 0A 24                DB      CR,LF,'$'

                        ;MSGHW: DB      CR,LF,BELL
                        ;       DB      'N' AND 11111B  ; BLINK
                        ;       DB      'WARNING --- HARD DISC DRIVE --- WARNING'
                        ;       DB      'O' AND 11111B  ; BLINK OFF
                        ;       DB      LF
80CC  0D 0A 07          MSGRDY: DB      CR,LF,BELL
80CF  52 65 61 64             DB      'Ready to format'
80D3  79 20 74 6F
80D7  20 66 6F 72
80DB  6D 61 74

80DE  0D 0A             ;       DB       the disc in drive '
80E0  49 6E 73 65               DB      CR,LF
80E4  72 74 20 64               db      'Insert disc and',cr,lf
80E8  69 73 63 20
80EC  61 6E 64 0D
80F0  0A
80F1  74 79 70 65               DB      'type RET to go ahead',cr,lf
80F5  20 52 45 54
80F9  20 74 6F 20
80FD  67 6F 20 61
8101  68 65 61 64
8105  0D 0A
8107  6F 72 20 61               DB      'or any other key to abandon';^C to abort.'
```

```
810B  6E 79 20 6F
810F  74 68 65 72
8113  20 68 65 79
8117  20 74 6F 20
811B  61 62 61 6E
811F  64 6F 6E
8122  24
8123  58 3A 20 3F
8127  0D 0A 0A 57
812B  41 49 54 2E
812F  2E 46 4F 52
8133  4D 41 54 54
8137  49 4E 47 0D
813B  0A 24

813D  0E 09
813F  11 8152
8142  CD 8487

8145  0E 0D
8147  CD 0000*

814A  3E FF
814C  32 C000

814F  C3 0224

8152  0D 0A 45 6E
8156  64 20 6F 66
815A  20 66 6F 72
815E  6D 61 74 74
8162  69 6E 67 0D
8166  0A
8167  24

                        DB      '$'
FRIG1:                  DB      'X: ?'
mqwait:                 db      CR,LF,LF,"WAIT..FORMATTING",cr,lf,'$'



finish:                 MVI     C,9             ; SERVICE RTN ERROR HNDLR
                        LXI     D,mgfin         ; JMP 0 --> NO ERROR
                        CALL    CRTOUT  ;BDOS           ; RET   --> AN ERROR
ext retbasic
;
                        MVI C,0DH
                        CALL BDOS               ;Reset disc system
;
                        MVI A,0FFH
                        STA 0C000H              ;Ensure FF at bases of CALCSTK
;
                        JMP NEWINT              ;Clear program
;
;
;                       JMP     RETBASIC
;
mgfin:                  db cr,lf,"End of formatting",cr,lf



                        db '$'


                        ;       ****************************
                        ;       *                          *
                        ;       * DISC FORMATING SOFTWARE  *
                        ;       * FOR SJM'S FLOPPY CNTRLR  *
                        ;       *                          *
                        ;       ****************************

                PUBLIC  F1791

80ED            LDIR    EQU 00EDH+0B000H
0000            FALSE   EQU     0
FFFF            TRUE    EQU     NOT(FALSE)
                ;
                ;
                ; ---------------------------------------------------
```

```
                        ;
                        ;Hardware Interface, Intermediate Code and Executives
                        ;
                        ;
                        ;
                        .Z80

                        ;
0010                    FDCPORT EQU     010H
                        ;
0010                    FDCCOM  EQU     FDCPORT         ;FDC command register port (OUT)
0010                    FDCSTA  EQU     FDCPORT         ;FDC status register port (IN)
0011                    FDCTRK  EQU     FDCPORT+1       ;FDC track register port (IN & OUT)
0012                    FDCSEC  EQU     FDCPORT+2       ;FDC sector register port (IN & OUT)
0013                    FDCDAT  EQU     FDCPORT+3       ;FDC data register port (IN & OUT)
                        ;
0014                    FDCTLI  EQU     FDCPORT+4       ;Controller board input port
0014                    FDCTLO  EQU     FDCPORT+4       ;Controller board output port
                        ;
0001                    DSLBIT  EQU     00000001B       ;Drive select: 0 - drive A, 1 - drive B
0002                    SSLBIT  EQU     00000010B       ;Side select: 0 - side 0, 1 - side 1
0004                    MONBIT  EQU     00000100B       ;Motor on: 1 - turns drive motor on
0008                    MRYBIT  EQU     00001000B       ;Motor ready: 1 - drive motor ready
0010                    DENBIT  EQU     00010000B       ;Density: 0 - FM, 1 - MFM
                        ;
0001                    HLDBIT  EQU     00000001B       ;Head load: 1 - head load on drive
0002                    DSDBIT  EQU     00000010B       ;Double-sided: 1 if drive double-sided
0004                    TPIBIT  EQU     00000100B       ;TPI: 0 - 48 TPI drive, 1 - 96 TPI drive
0008                    STPBIT  EQU     00001000B       ;Track stepping rate: 0 - 12 ms, 1 - 6 ms
0010                    NODBIT  EQU     00010000B       ;No. of drives: 0 - 1 drive, 1 - 2 drives
0020                    RDYBIT  EQU     00100000B       ;Ready: 1 - drive ready
0040                    INTBIT  EQU     01000000B       ;Interrupt: 1 - FDC interrupt request
0080                    DRQBIT  EQU     10000000B       ;Data request: 1 - FDC data request
                        ;
0001                    BUSYBIT EQU     00000001B

                        .8080

0004                    SPEED   EQU     4       ; 4=4MHZ, 6=6MHZ
                        ;DRVRQ  EQU     0FFE8H
                        ;CFGBYT EQU     0FFE9H
                        ;INITLZ EQU     0FFFCH
                        EXT INITLIZ
B000                    IMMG    EQU     0B000H          ;4000H: TRACK IMMAGE BUFFER
                        ; ----------------------------------------

8168    CD 8382         F1791:  CALL    DRVSET
816B    C2 81CE                 JNZ     ERRORF
816E    CD 81D9                 CALL    C1213           ; TRK 0 D/D 8" IS S/D INTERLD
8171    CD 8382                 CALL    DRVSET
8174    C2 81CE                 JNZ     ERRORF
8177    CD 821F                 CALL    IMMAGE
817A    C2 81CE                 JNZ     ERRORF          ; NO SKEW TABLE
817D    0E 00                   MVI     C,0             ; START @ CYL 00
817F    06 00                   MVI     B,0             ; SIDE 0
8181    CD 8267                 CALL    UPDATE          ; UPDATE THE IMMAGE
```

```
84    CD 83CD            CALL    FMT             ; FORMAT THE TRACK
87    C2 81CE            JNZ     ERRORF
8A    CD 81ED            CALL    C213            ; RESORE CONFIG BYTE
8D    CD 821F            CALL    IMMAGE
90    C2 81CE            JNZ     ERRORF
93    CD 8382            CALL    DRVSET
96    C2 81CE            JNZ     ERRORF
99    0E 00              MVI     C,0
9B    C3 81AB            JMP     SKIPS2
9E    06 00        LOOP1: MVI     B,0             ; SIDE 0
A0    CD 8267            CALL    UPDATE          ; UPDATE THE IMMAGE
A3    C5                 PUSH    B
A4    CD 83CD            CALL    FMT             ; FORMAT THE TRACK
A7    C1                 POP     B
A8    C2 81CE            JNZ     ERRORF
AB    3A 0000*   SKIPS2: LDA     CFGBYT
AE    E6 01              ANI     1               ; 2 SIDED ?
B0    CA 81C0            JZ      SKIP1
B3    06 01              MVI     B,1
B5    CD 8267            CALL    UPDATE
B8    C5                 PUSH    B
B9    CD 83CD            CALL    FMT
BC    C1                 POP     B
BD    C2 81CE            JNZ     ERRORF
C0    0C           SKIP1: INR     C
C1    CD 81F3            CALL    MAXCYL          ; AT MAX CYL YET?
C4    B9                 CMP     C
C5    CA 81D2            JZ      EXIT
C8    CD 8206            CALL    CSEEK           ; SEEK DEPENDS ON TPI
CB    CA 819E            JZ      LOOP1
CE    CD 0000*   ERRORF: CALL    INITLIZ
D1    C9                 RET
D2    CD 0000*   EXIT:   CALL    INITLIZ
D5    C1                 pop bc
D6    C3 0001'           jmp SFORMAT
                 ;JMP    0

F9    3A 0000*   C213:   LDA     CFGBYT
FC    32 81EE            STA     R213+1
FF    FE 12              CPI     12H
101   CA 81E7            JZ      C2131
104   FE 13              CPI     13H
106   C0                 RNZ
107   3E 90        C2131: MVI     A,10H+80H
109   32 0000*           STA     CFGBYT
10C   C9                 RET
10D   3E 00        R213:  MVI     A,0             ; SETUP BY C213
10F   32 0000*           STA     CFGBYT
112   C9                 RET

113   3A 0000*   MAXCYL: LDA     CFGBYT
116   FE 10              CPI     10H
118   3E 4D              MVI     A,77
11A   D0                 RNC
11B   3A 0000*           LDA     CFGBYT
11E   FE 04              CPI     04H
```

```
0    3E 50                      MVI     A,80
2    D0                         RNC
3    3E 28                      MVI     A,40
5    C9                         RET

6    3A 0000*          CSEEK:   LDA     CFGBYT
9    E6 04                      ANI     100B            ; 96 TPI ?
B    79                         MOV     A,C
C    C2 821A                    JNZ     CSKIP2
F    DB 14                      IN      FDCTLI
1    EE 0F                      XRI     0FH
3    E6 04                      ANI     TPIBIT
5    79                         MOV     A,C
6    CA 821A                    JZ      CSKIP2
8    81                         ADD     C
9    D3 13             CSKIP2:  OUT     FDCDAT
B    C3 8403                    JMP     SEEK

E    3A 0000*          IMMAGE:  LDA     CFGBYT
1    21 8332                    LXI     H,SNGTAB
3    E6 02                      ANI     010B            ; CONFIG D/D ?
5    CA 822D                    JZ      IMMAG0
8    21 8356                    LXI     H,DBLTAB
B    11 B000           IMMAG0:  LXI     D,IMMG
E    CD 8257                    CALL    IMLP0
1    22 8255                    SHLD    TEMP
4    CD 8283                    CALL    GETTAB
7    C0                         RNZ
8    CD 82B4                    CALL    COUNT           ; C=#SECTORS
B    2A 8255           IMLP2:   LHLD    TEMP
E    CD 8257                    CALL    IMLP0
1    0D                         DCR     C
2    C2 823D                    JNZ     IMLP2
5    01 03E8                    LXI     B,1000
8    1B                         DCX     D
9    1A                IMLP3:   LDAX    D
A    13                         INX     D
B    12                         STAX    D
C    0B                         DCX     B
D    78                         MOV     A,B
E    B1                         ORA     C
F    C2 824B                    JNZ     IMLP3
2    C9                         RET
3                      TEMP:    DS      2
5    7E                IMLP0:   MOV     A,M
6    23                         INX     H
7    B7                         ORA     A
8    C8                         RZ
9    47                         MOV     B,A
A    7E                         MOV     A,M
B    23                         INX     H
C    12                IMLP1:   STAX    D
D    13                         INX     D
E    05                         DCR     B
```

```
                                UPDATE: CALL    GETTAB          ; HL=SECTOR SKEW TABL
                                        LXI     D,IMMG
                                UPLP2:  MOV     A,M
                                        ORA     A
                                        RZ                      ; NO MORE SECTORS
                                UPLP1:  LDAX    D
                                        INX     D
                                        CPI     0FEH            ; ID MARK
                                        JNZ     UPLP1
                                        MOV     A,C             ; CYL
                                        STAX    D
                                        INX     D
                                        MOV     A,B
                                        STAX    D               ; SIDE
                                        INX     D
                                        MOV     A,M             ; SECTOR
                                        INX     H
                                        STAX    D
                                        JMP     UPLP2

                                GETTAB: LXI     H,TABTOP
                                        LDA     CFGBYT
                                        PUSH    B
                                        MOV     C,A
                                GTBL0:  MOV     A,M
                                        CPI     255
                                        JZ      NOTAB
                                GTBL1:  MOV     A,M
                                        INX     H
                                        CPI     255
                                        JZ      TSKP0
                                        CMP     C
                                        JNZ     GTBL1
                                        POP     B
                                GTBL2:  MOV     A,M
                                        INX     H
                                        CPI     255
                                        JNZ     GTBL2
                                        RET
                                TSKP0:  MOV     A,M
                                        INX     H
                                        ORA     A
                                        JNZ     TSKP0
                                        JMP     GTBL0
                                NOTAB:  POP     B
                                        LXI     H,0
                                        ORA     A
                                        RET

                                COUNT:  MVI     C,0
                                CNTLP:  MOV     A,M
                                        INX     H
                                        ORA     A
```

```
C3 82B6                 JMP     CNTLP

                ; ---------------------------------------

                        ; SECTOR SKEW TABLES

00 01 04 05     TABTOP: DB      00H,01H,04H,05H ; SINGLE DENSITY 5" DISCS
02 03 06 07             DB      02H,03H,06H,07H ; D/D 5" DISCS
FF                      DB      255
01 0C 07 02             DB      1,12,7,2,13,8,3,14,9,4,15,10,5,16,11,6
0D 08 03 0E
09 04 0F 0A
05 10 0B 06
00                      DB      0

10                      DB      10H             ; S/S S/D 8" DISCS.
FF                      DB      255
01 16 11 0C             DB      1,22,17,12,7,2,23,18,13,8,3,24,19
07 02 17 12
0D 08 03 18
13
0E 09 04 19             DB      14,9,4,25,20,15,10,5,26,21,16,11,6
14 0F 0A 05
1A 15 10 0B
06
00                      DB      0

11 12 13                DB      11H,12H,13H
FF                      DB      255
01 0A 13 02             DB      1,10,19,2,11,20,3,12,21,4,13,22,5
0B 14 03 0C
15 04 0D 16
05
0E 17 06 0F             DB      14,23,6,15,24,7,16,25,8,17,26,9,18
18 07 10 19
08 11 1A 09
12
00                      DB      0

90                      DB      10H+80H         ; D/D 8" TRK 00
FF                      DB      255
01 0A 13 02             DB      1,10,19,2,11,20,3,12,21,4,13,22,5
0B 14 03 0C
15 04 0D 16
05
0E 17 06 0F             DB      14,23,6,15,24,7,16,25,8,17,26,9,18
18 07 10 19
08 11 1A 09
12
00                      DB      0

FF                      DB      255             ; TERMINATE

                ; ---------------------------------------------------
```

```
4    06 00                      DB      6,      0
6    01 FC                      DB      1,      0FCH    ; INDEX MARK
B    1A FF                      DB      2       0FFH
A    00                         DB      0
B    06 00                      DB      6,      0
D    01 FE                      DB      1,      0FEH
F    01 00                      DB      1,      0       ; TRACK £
1    01 00                      DB      1,      0       ; SIDE  £
3    01 00                      DB      1,      0       ; SEC   £
5    01 00                      DB      1,      0       ; SEC LEN
7    01 F7                      DB      1,      0F7H    ; ID CRC
9    0B FF                      DB      11,     0FFH
1    06 00                      DB      6,      0
3    01 FB                      DB      1,      0FBH    ; DATA ADDR MARK
     80 E5                      DB      128,    0E5H
     01 F7                      DB      1,      0F7H    ; DATA CRC
     1B FF                      DB      27,     0FFH
     00                         DB      0

     50 4E              DBLTAB: DB      80,     04EH
     0C 00                      DB      12,     0
     03 F6                      DB      3,      0F6H
     01 FC                      DB      1,      0FCH    ; INDEX
     32 4E                      DB      50,     04EH
     00                         DB      0
     0C 00                      DB      12,     0
     03 F5                      DB      3,      0F5H
     01 FE                      DB      1,      0FEH    ; ID ADDR
     01 00                      DB      1,      0       ; TRACK
     01 00                      DB      1,      0       ; SIDE
     01 00                      DB      1,      0       ; SECTOR
     01 01                      DB      1,      1       ; SEC LEN
     01 F7                      DB      1,      0F7H    ; CRC
     16 4E                      DB      22,     04EH
     0C 00                      DB      12,     0
     03 F5                      DB      3,      0F5H
     01 FB                      DB      1,      0FBH    ; DATA ADDR
     80 E5                      DB      128,    0E5H    ; FIRST HALF OF DATA
     80 E5                      DB      128,    0E5H    ; SECND HALF OF DATA
     01 F7                      DB      1,      0F7H    ; CRC
     32 4E                      DB      50,     04EH
     00                         DB      0

                    ; ---------------------------------------------------
                    .Z80
                    ;
                    ;DRVSET selects drive given by (DRVRQ).
                    ;Returns Z if select successful, else NZ.
                    ;
     CD 843E        DRVSET: CALL WAIT                  ;Wait until FDC not busy
                    ;
     3A 0000*               LD A,(DRVRQ)               ;A = drive number to select
     F6 0C                  OR MONBIT+MRYBIT
     06 0D                  LD B,DSLBIT+MONBIT+MRYBIT       ;Drive select, drive enable
     CD 8427                CALL REPLACE               ;Update status
                    ;
```

```
                        ;
                        ;Here if drive change required.
                        ;
3F   DB 11                      IN A,(FDCTRK)           ;A = current track number
41   D3 13                      OUT (FDCDAT),A          ;Load track number into DR
43   3E 10                      LD A,00010000B          ;Seek current track with head unloaded
45   D3 10                      OUT (FDCCOM),A          ;Issue command ('Unload head')
                        ;
47   CD 8448                    CALL WAIT1              ;Wait until FDC has finished command
                        ;
                        ;
                        ;Test whether drive is double-sided.
                        ;
4A   3A 0000*                   LD A,(CFGBYT)           ;A = configure byte
4D   E6 01                      AND 01B                 ;NZ if drive configured as D/S
4F   28 09                      JR Z,SKIP2              ;Jump if drive configured S/S
                        ;
51   DB 14                      IN A,(FDCTLI)           ;A = input control byte
53   EE 0F                      XOR 0FH                 ;INVERT SWITCHES
55   E6 02                      AND DSDBIT              ;NZ if drive D/S
57   CA 8452                    JP Z,DRVSE5             ;Jump if drive select error
                        ;
                        ;Test whether drive is 96 TPI.
                        ;
5A   3A 0000*           SKIP2:  LD A,(CFGBYT)           ;A = configure byte
5D   E6 04                      AND 0100B               ;NZ if drive configured 96 TPI
5F   28 09                      JR Z,SKIP3              ;Jump if drive configured 48 TPI
                        ;
61   DB 14                      IN A,(FDCTLI)           ;A = input control byte
63   EE 0F                      XOR 0FH                 ;INVERT SWITCHES
65   E6 04                      AND TPIBIT              ;NZ if drive 96 TPI
67   CA 8452                    JP Z,DRVSE5             ;Jump if drive select error
                        ;
6A   DB 14              SKIP3:  IN A,(FDCTLI)
6C   EE 0F                      XOR 0FH
6E   E6 02                      AND 10B                 ;NZ if drive configured D/D
70   3E 00                      LD A,0
72   28 01                      JR Z,SKIP4              ;Jump if drive configured S/D
74   3D                         DEC A                   ;A = 255
                        ;
75   06 10              SKIP4:  LD B,DENBIT             ;Select single or double density
77   CD 8427                    CALL REPLACE            ;Update status
                        ;
                        ;
7A   C3 8407                    JP RECALB               ;Move disc head to track 00
                        .8080

                        ;
                        .Z80
7D   78                 FMT:    LD A,B
7E   06 02                      LD B,SSLBIT             ;Select side
80   07                         RLCA                    ;SET/RES SIDE BIT
81   CD 8427                    CALL REPLACE            ;Update status
                        ;
```

```
                ;       LD A,L
                ;       OUT (DMA[illegible]),A          ;Set low (DMA address)
                ;       LD A,H
                ;       OUT (DMAHI),A                   ;Set high (DMA address)
                ;
04  CD 8456            CALL WAIT2
07  C0                  RET     NZ

08  DB 14              IN A,(FDCTLI)
0A  E6 80              AND DRQBIT
0C  C0                 RET NZ                          ;STILL REQUESTING DATA SO ERR
0D  21 B000            LD HL,IMMG
10  3E F4              LD A,11110100B
12  D3 10              OUT (FDCCOM),A                  ;Issue command


14  F3                 DI                              ;Ensure no interruptions
15  0E 13              LD C,FDCDAT                     ;C = FDC data register port
                ;
                ;Main loop for writing bytes from disc.
                ;Time taken to write each byte = 73 T-states.
                ;
17  DB 14       DISCW1: IN A,(FDCTLI)                  ;11. A = control input byte
19  E6 C0               AND INTBIT+DRQBIT              ;7. NZ if interrupt or data request
1B  28 FA               JR Z,DISCW1                    ;7/12. Jump if no request
                ;
                ;Here if data byte needed or command finished.
                ;
1D  CB 77               BIT 6,A                        ;8. NZ if command finished
1F  20 05               JR NZ,DISCW2                   ;7/12. Jump if command finished
                ;
                ;Here if data byte needed for FDC data register.
                ;
21  ED A3              OUTI                           ;16. Output byte and increment pointer
23  C3 83E7             JP DISCW1                      ;10. Get next byte
                ;
                ;Here if write command finished.
                ;
26  FB          DISCW2: EI
27  DB 10               IN A,(FDCSTA)
29  E6 E4               AND 11100100B
2B  C8                  RET Z

2C  32 0000*    RWEF:   LD (CFGBYT),A
2F  3E 06               LD A,6
31  B7                  OR A
32  C9                  RET



                ;
                ;
                ;
                ;SEEK [illegible]
```

```
                        ;
3   3E 18               SEEK:   LD A,00010000B          ;Seek command, head loaded
5   18 02                       JR SKTRK
                        ;
                        ;
                        ;
                        ;RECALB moves disc head to track 00.
                        ;Returns A = 0, Z if seek track 00 successful.
                        ;
7   3E 08               RECALB: LD A,00001000B          ;Restore command, head loaded
                        ;
9   47                  SKTRK:  LD B,A                  ;B = command byte
A   DB 14                       IN A,(FDCTLI)           ;A = input status byte
C   EE 0F                       XOR 0FH                 ;INVERT SWITCHES
E   E6 08                       AND STPBIT              ;A = track stepping rate
0   3E 00                       LD A,00000000B          ;00=6mS
2   20 02                       JR NZ,SKTR1             ;Z If Step '20 ms'
4   3E 02                       LD A,00000010B          ;01=20mS
6   B0                  SKTR1:  OR B                    ;A = seek command byte
7   D3 10                       OUT (FDCSTA),A          ;Issue command
                        ;
9   CD 8448                     CALL WAIT1              ;Wait until command has finished
                        ;
;   CD 8437                     CALL DELAY1             ;Must wait 54 microseconds
    DB 10                       IN A,(FDCSTA)           ;A = FDC status byte
    E6 10                       AND 00010000B           ;NZ if seek error
    C8                          RET Z
    3E 04                       LD A,4                  ;KQ1
    C9                          RET
                        ;
                        ;
                        ;
                        ;
                        ;REPLACE updates hardware status byte.
                        ;On entry, A = new value of status byte,
                        ;B = mask for old status byte.
                        ;
                        ;N.B.   Those bits which are zero in mask
                        ;       will remain unchanged in status byte.
                        ;
                        RPLCE:
                        REPLACE:
    A0                          AND B
    4F                          LD C,A                  ;C = masked new value
    78                          LD A,B
    2F                          CPL
    47                          LD B,A                  ;B = complemented mask
    3A 8486                     LD A,(LSTOUT)           ;Get old value of status byte
    A0                          AND B
    B1                          OR C
    32 8486                     LD (LSTOUT),A           ;Store new value of status byte
    D3 14                       OUT (FDCTLO),A          ;Update status byte
    C9                          RET
```

```
                        ;
                        ;
37    3E 32             DELAY1: LD A,50
                        ;
39    3D                DELY11: DEC A
3A    C2 8439                   JP NZ,DELY11
3D    C9                        RET
                        ;
                        ;
                        ;
                        ;WAIT calls DELAY1, then waits until FDC is not busy before returning.
                        ;
E     CD 8437           WAIT:   CALL DELAY1
1     DB 10                     IN A,(FDCSTA)          ;A = FDC status register
3     E6 01                     AND BUSYBIT            ;NZ if FDC busy (bit 0)
5     20 F7                     JR NZ,WAIT
7     C9                        RET
                        ;
                        ;
                        ;
                        ;WAIT1 calls DELAY1, then waits until FDC has finished command.
                        ;
8     CD 8437           WAIT1:  CALL DELAY1
8     DB 14                     IN A,(FDCTLI)          ;A = hardware status byte
)     E6 40                     AND INTBIT             ;NZ if INTRQ from FDC (bit 4)
F     28 F7                     JR Z,WAIT1
      C9                        RET
                        ;
                        ;
                        ;
                        ;WAIT3: CALL WAIT2
                        ;       RET NZ
                        ;       LD A,(CFGBYT)
                        ;       AND 00010000B          ;Jump if 8" drive
                        ;       IN A,(FDCTLI)          ;A = hardware status byte
                        ;       JR Z,DRVT5
                        ;
                        ;       AND RY5BIT             ;NZ if 5" drive ready (bit 7)
                        ;       JR NZ,SKIP5            ;Jump if 5" drive ready
                        ;
                        ;Here if drive (5" or 8") not ready.
                        ;
   3E 05                DRVSE5: LD A,5
   A7                           AND A                  ;NZ
   C9                           RET
                        ;
                        ;DRVT5: AND RY8BIT             ;NZ if 8" drive ready
                        ;       JR Z,DRVSE5            ;Jump if 8" drive not ready
                        ;
                        ;Here if drive (5" or 8") ready.
                        ;
                        ;SKIP5: XOR A
                        ;       RET
                        ;
```

```
456                     ;
456  CD 847C            WAIT2:
459  C8                         CALL TEST
                                RET Z               ;DRIVE IS READY
                        ;
                        ;HERE IF DRIVE IS NOT RAEDY
                        ;
45A  06 08                      LD B,MRYBIT
45C  AF                         XOR A
45D  CD 8427                    CALL REPLACE    ;TURN OFF MOTOR READY

460  3E 0C                      LD A,MRYBIT+MONBIT
462  47                         LD B,A
463  CD 8427                    CALL REPLACE    ;ENSURE MOTOR ON & MOTOR READY
                        ;
466  CD 8470                    CALL DELAY2
                        ;
469  CD 847C                    CALL TEST
46C  C8                         RET Z
46D  3E 09                      LD A,9
46F  C9                         RET

470  01 0320            DELAY2: LD BC,800
473  CD 8437            DEL22:  CALL DELAY1
476  0B                         DEC BC
477  79                         LD A,C
478  B0                         OR B
479  C8                         RET Z
47A  18 F7                      JR DEL22


                        ;;
                        ;
                        TEST:   IN A,(FDCTLI)
                                BIT 5,A
                                JR Z,TEST1
                                XOR A
                                RET
                        ;
                        TEST1:  INC A
                                RET



                        .8080



                        .Z80
                        LSTOUT: DB      0
                        CRTOUT: PUSH DE
                                PUSH AF
                        CRT1:   LD A,(DE)
```

```
A   FE 24                       CP "$"
C   28 06                       JR Z,CRTEXIT
E   CD 00BC                     CALL GETS
1   13                          INC DE
2   18 F5                       JR CRT1
4   F1                  CRTEXIT:POP AF
5   D1                          POP DE
6   C9                          RET
7   CD 0079             KEYBD:  CALL KBD
A   28 FB                       JR Z,KEYBD
C   C9                          RET
                        .8080
                                END     SFORMAT
```

```
:

s:
  0157*   BELL    0007    BUSYBI  0001    C1213   81D9
  81E7    CFGBYT  040C*   CNFG    0000*   CNTLP   82B6
  82B4    CR      000D    CRT1    8489    CRTEXI  8494
  8487    CSEEK   8206    CSKIP2  821A    DBLTAB  8356
  8473    DELAY1  8437    DELAY2  8470    DELY11  8439
  0010    DISCW1  83E7    DISCW2  83F6    DRQBIT  0080
  0395*   DRVSE5  8452    DRVSET  8382    DSDBIT  0002
  0001    DTYPE   0003    ER0     8042    ERROR   804D
  81CE    EXIT    81D2    F1403   0000    F1791   8168I
  0000    F520A   0000    FALSE   0000    FDCCOM  0010
  0013    FDCPOR  0010    FDCSEC  0012    FDCSTA  0010
  0014    FDCTL0  0014    FDCTRK  0011    FINISH  813D
  83CD    FORMLD  000F'   FRIG1   8123    FSIDI   0000
  00BC    GETTAB  8283    GTBL0   828B    GTBL1   8291
  829D    HLDBIT  0001    IMLP0   8257    IMLP1   825E
  823D    IMLP3   824B    IMMAG0  822D    IMMAGE  821F
  B000    INITLI  01E2*   INTBIT  0040    KBD     0079
  8497    LDIR    B0ED    LF      000A    LOOP1   819E
  8486    MAXCYL  81F3    MGFIN   8152    MGWAIT  8127
  0004    MRYBIT  0008    MSGE0   8078    MSGER   80A8
  80CC    NEWINT  0224    NODBIT  0010    NOTAB   82AE
  81ED    RDYBIT  0020    RECALB  8407    REPLAC  8427
  0000*   RPLCE   8427    RWEF    83FC    SEEK    8403
  8058    SFORMA  0001I'  SKIP1   81C0    SKIP2   83AA
  83BA    SKIP4   83C5    SKIPS2  81AB    SKTR1   8416
  8409    SNGTAB  8332    SPEED   0004    SSLBIT  0002
  0008    TABTOP  82BE    TEMP    8255    TEST    847C
  8484    TPIBIT  0004    TRUE    FFFF    TSKP0   82A5
  8267    UPLP1   8270    UPLP2   826D    WAIT    843E
  8448    WAIT2   8456    WWW     8033

al error(s)
```

```
                        .Z80
                        CSEG
                        ORG 100H
                        ;*********************************************
                        EXT     NOFILE,SETDMA,GETFNAM,BDOS
                        PUBLIC  STAT
                        ;*********************************************
                        SCRRST  EQU 10H
                        LF      EQU 10
                        CR      EQU 13
                        DIS     EQU 33
                        PRINT   EQU 0BCH                ;(Getstr in ROM.)
                        DMA     EQU 0E680H
                        CHNL5   EQU 0D840H+168
                        ;*************************************************************
                        ;Temp variables (all in DMA except STORE,which is in FCB).
                        ;*************************************************************
                        BLKSZ   EQU     DMA+0
                        FLAG    EQU     DMA+1
                        BLKNO   EQU     DMA+2
                        READTY  EQU     DMA+4
                        STORE   EQU     CHNL5+16
                        ;*****************************************************************************
                        ;PROGRAM
                        ;
                        ;D.STAT
                        ;D.STAT <filename.>
                        ;D.STAT <filename.>,R/<attr.>
                        ;*****************************************************************************
1300                    STAT:   INC DE
1A                              LD A,(DE)
FE FF                           CP 0FFH
CA 01F2'                        JP Z,STAT1              ;Jump to 'STAT'
FE 22                           CP '"'
20 31                           JR NZ,ERROR
D5                              PUSH DE
13                      LOPS:   INC DE
1A                              LD A,(DE)
FE FF                           CP 0FFH
28 2A                           JR Z,ERROR
FE 22                           CP '"'
20 F6                           JR NZ,LOPS
D1                              POP DE
DD 21 D8E8                      LD IX,CHNL5
CD 0000*                        CALL GETFNAM
CA 01CE'                        JP Z,STCHK
1B                              DEC DE                  ;DE returns pointing two after the ".
1A                              LD A,(DE)
FE FF                           CP 0FFH
CA 016D'                        JP Z,FSIZE
FE 2C                           CP ','
20 10                           JR NZ,ERROR
13                              INC DE
1A                              LD A,(DE)
FE 52                           CP 'R'
```

```
2'  13                      INC DE
3'  1A                      LD A,(DE)
4'  FE 57                   CP 'W'
6'  28 0B                   JR Z,RW
8'  FE 4F                   CP 'O'
A'  28 02                   JR Z,RO
C'  EF                ERROR:RST 28H
D'  01                      DB 1
                    ;*****************************************************
                    ;*****************************************************
E'  21 D8F1           RO:   LD HL,CHNL5+9
F'  CB FE                   SET 7,(HL)
1'  32 E684           RW:   LD (READTY),A
3'  0E 1E                   LD C,30
5'  11 D8E8                 LD DE,CHNL5
7'  CD 0000*                CALL BDOS
9'  FE FF                   CP 0FFH
B'  CC 0000*                CALL Z,NOFILE           ;(RET address is popped by NOFILE.)
D'  CD 02B5'                CALL NAMPR              ;Print requested name
F'  D7                      RST SCRRST
    89 20 73 65             DB 80H+9,' set to R'
    74 20 74 6F
    20 52
    3A E684                 LD A,(READTY)
    CD 00BC                 CALL PRINT
    D7                      RST SCRRST
    83 0D 0A 0A             DB 80H+3,CR,LF,LF
    C9                      RET
                    ;*****************************************************
                    ;Routine to print unambiguous file size.(Refer to CHNL5)
                    ;*****************************************************
                    FSIZE:
                    ;************************************
    11 D8E8         SFR:    LD DE,CHNL5
    0E 11                   LD C,17
    CD 0000*                CALL BDOS               ;Search for first.
    FE FF                   CP 0FFH
    CC 0000*                CALL Z,NOFILE           ;RET address is popped by NOFILE.
    21 E689                 LD HL,DMA+9
    0F                      RRCA
    0F                      RRCA
    0F                      RRCA
    4F                      LD C,A
    06 00                   LD B,0
    09                      ADD HL,BC
    11 D8F1                 LD DE,CHNL5+9
    7E                      LD A,(HL)
    12                      LD (DE),A
    11 D8E8                 LD DE,CHNL5
    0E 23                   LD C,35
    CD 0000*                CALL BDOS               ;Get file size (used later on.)
    CD 02B5'                CALL NAMPR              ;Print file name.
                    ;******************
                    ;Compute file size.
                    ;******************
  2A D909                   LD HL,(CHNL5+DIG)       ;Point to file size (In FCB.)
```

```
7'   E5                         PUSH HL
8'   01 0007                    LD BC,7
B'   09                         ADD HL,BC
C'   CB 3C                      SRL H                    ;Now divide HL by 8.
E'   CB 1D                      RR L                     ;(=file size in K.)
0'   CB 3C                      SRL H
2'   CB 1D                      RR L
4'   CB 3C                      SRL H
6'   CB 1D                      RR L
B'   CD 025D'                   CALL HEXDEC              ;Display file in K
B'   D7                         RST SCRRST
2'   83 6B 20 20                DB 80H+3,'k  '
0'   E1                         POP HL
1'   CD 025D'                   CALL HEXDEC              ;Display number of records.
4'   D7                         RST SCRRST
5'   88 20 52 65                DB 80H+8,' Recs  R'
9'   63 73 20 20
0'   52

                        ;*************
                        ;Is file R/O?
                        ;*************
3'   3A D8F1            STCHK:  LD A,(CHNL5+9)
.'   CB 7F                      BIT 7,A
5'   28 04                      JR Z,RDWRTE
7'   3E 4F                      LD A,'O'
'    18 02                      JR PRN
'    3E 57              RDWRTE: LD A,'W'
'    CD 00BC            PRN:    CALL PRINT
                        ;*********************************************
                        ;Find and print free  space for current drive.
                        ;*********************************************
'    3A D8E8            STCHK:  LD A,(CHNL5)
'    A7                         AND A
'    28 1E                      JR Z,STAT1               ;Continue if no drive specified.
'    3D                 STAT0:  DEC A
'    F5                         PUSH AF                  ;Requested drive not default.(0).
'    0E 19                      LD C,25
'    CD 0000*                   CALL BDOS                ;
'    32 D8F8                    LD (STORE),A             ;Store current drive.
'    F1                         POP AF
'    5F                         LD E,A                   ;A contains drive number.
'    0E 0E                      LD C,14
'    CD 0000*                   CALL BDOS                ;Select requested drive.
'    CD 01F6'                   CALL STAT2               ;Non default entry point.
'    3A D8F8                    LD A,(STORE)
'    5F                         LD E,A
'    0E 0E                      LD C,14
'    CD 0000*                   CALL BDOS                ;Reselect drive.
'    C9                         RET
                        ;***************
'    AF                 STAT1:  XOR A
'    32 D8E8                    LD (CHNL5),A             ;Set FCB to show defualt drive.
'    0E 1F              STAT2:  LD C,31
'    CD 0000*                   CALL BDOS                ;Get Disc param vector.
```

```
    23                      INC HL                          ;Point to BLM.
    7E                      LD A,(HL)                       ;Get BLM.
    3C                      INC A                           ;No of sectors per block.
    CB 3F                   SRL A
    CB 3F                   SRL A
    CB 3F                   SRL A                           ;Block size in K
    32 E680                 LD (BLKSZ),A
    23                      INC HL
    23                      INC HL                          ;Point to DSM.
    56                      LD D,(HL)
    23                      INC HL
    66                      LD H,(HL)
    6A                      LD L,D                          ;HL now contains DSM
    23                      INC HL
    22 E682                 LD (BLKNO),HL
                    ;***********************************************
                    ;Determine iff disc R/O.Assume is CURRENT drive.
                    ;***********************************************
                    ;       LD E,29
                    ;       CALL BDOS
                    ;       BIT 0,L                         ;Test CURRENT drive.
                    ;       JR NZ,RDONLY
                    ;       LD A,'W'
                    ;       JR JMP5
                    ;RDONLY:        LD A,'O'
                    ;JMP5:  LD (READTY),A
                    ;***************************************
    0E 1B                   LD C,27
    CD 0000*                CALL BDOS                       ;Get allocation vector.
    EB                      EX DE,HL
    2A E682                 LD HL,(BLKNO)
    7D                      LD A,L
    CB 3C                   SRL H
    CB 1D                   RR L
    CB 3C                   SRL H
    CB 1D                   RR L
    CB 3C                   SRL H
    CB 1D                   RR L                            ;Number of blocks/8
    E6 07                   AND 00000111B
    4F                      LD C,A
    D9                      EXX
    3A E680                 LD A,(BLKSZ)
    5F                      LD E,A
    16 00                   LD D,0
    21 0000                 LD HL,0
    D9                      EXX
    06 08           LOOP2:  LD B,8
    CD 02AA'                CALL ALLOC
    2B                      DEC HL
    7C                      LD A,H
    B5                      OR L
    20 F6                   JR NZ,LOOP2
    41                      LD B,C
    78                      LD A,B
    A7                      AND A
```

```
47'   CD 02DB'                  CALL DRVLET;******          ;Print drive
4A'   D9                        EXX                         ;HL Contains space.
                        ;       RST SCRRST
                        ;       DB 80H+1,'R'
                        ;       LD A,(READTY)
                        ;       CALL PRINT
4B'   D7                        RST SCRRST
4C'   86 53 70 61               DB 80H+6,'Space '
50'   63 65 20
53'   CD 025D'                  CALL HEXDEC                 ;Convert to string.(Five digit.)
56'   D7                        RST SCRRST
57'   84 6B 0D 0A               DB 80H+4,'k',CR,LF,LF
5B'   0A
5C'   C9                        RET
5D'   D5                HEXDEC: PUSH DE
5E'   F5                        PUSH AF
5F'   C5                        PUSH BC
60'   AF                        XOR A
61'   32 E681                   LD (FLAG),A
64'   01 2710                   LD BC,10000
67'   CD 0286'                  CALL DIGIT
6A'   01 03E8                   LD BC,1000
6D'   CD 0286'                  CALL DIGIT
70'   01 0064                   LD BC,100
73'   CD 0286'                  CALL DIGIT
76'   01 000A                   LD BC,10
79'   CD 0286'                  CALL DIGIT
7C'   7D                        LD A,L
7D'   C6 30                     ADD A,'0'
7F'   CD 00BC                   CALL PRINT
82'   C1                        POP BC
83'   F1                        POP AF
84'   D1                        POP DE
85'   C9                        RET
                        ;**************************************
86'   A7                DIGIT:  AND A
87'   16 00                     LD D,0
89'   ED 42             LOOP3:  SBC HL,BC
8B'   14                        INC D
8C'   30 FB                     JR NC,LOOP3
8E'   7A                        LD A,D
8F'   3D                        DEC A
90'   C6 30                     ADD A,'0'
92'   FE 30                     CP '0'
94'   20 0A                     JR NZ,JMP3
96'   3A E681                   LD A,(FLAG)
99'   A7                        AND A
9A'   3E 30                     LD A,'0'
9C'   20 02                     JR NZ,JMP3
9E'   09                        ADD HL,BC
9F'   C9                        RET
                        ;*******
A0'   09                JMP3:   ADD HL,BC
A1'   CD 00BC                   CALL PRINT
A4'   3E 01                     LD A,1
```

```
2A9'   C9                        RET
                                 ;*************************************
2AA'   1A                        ALLOC:  LD A,(DE)                  ;Get allocation byte.
2AB'   13                                INC DE
2AC'   87                        LOOP1:  ADD A,A
2AD'   38 03                             JR C,ISUSE
2AF'   D9                                EXX
2B0'   19                                ADD HL,DE
2B1'   D9                                EXX
2B2'   10 F8                     ISUSE:  DJNZ LOOP1
2B4'   C9                                RET
                                 ;*************************************
                                 ;Print name of requested file.
                                 ;(Uses FCB rather than input buffer.)
                                 ;*************************************
2B5'   CD 02DB'                  NAMPR:  CALL DRVLET                ;Print drive (Unless A:)
2B8'   23                                INC HL
2B9'   06 08                             LD B,8
2BB'   7E                        LOOPS:  LD A,(HL)
2BC'   23                                INC HL
2BD'   FE 20                             CP ' '
2BF'   28 03                             JR Z,MISS
2C1'   CD 00BC                           CALL PRINT
2C4'   10 F5                     MISS:   DJNZ LOOPS
2C6'   3E 2E                             LD A,'.'
2C8'   CD 00BC                           CALL PRINT
2CB'   06 03                             LD B,3
2CD'   7E                        LOOPT:  LD A,(HL)
2CE'   CB BF                             RES 7,A
2D0'   23                                INC HL
2D1'   CD 00BC                           CALL PRINT
2D4'   10 F7                             DJNZ LOOPT
2D6'   D7                                RST SCRRST
2D7'   82 20 20                          DB 80H+2,'  '
2DA'   C9                                RET
                                 ;*******************
                                 ;Print drive number.
                                 ;*******************
2DB'   D7                        DRVLET: RST SCRRST
2DC'   82 0D 0A                          DB 80H+2,CR,LF
2DF'   21 D8E3                           LD HL,CHNLS
2E2'   7E                                LD A,(HL)
2E3'   A7                                AND A
2E4'   C8                                RET Z
2E5'   C6 40                             ADD A,'A'-1
2E7'   CD 00BC                           CALL PRINT
2EA'   3E 3A                             LD A,':'
2EC'   CD 00BC                           CALL PRINT
2EF'   C9                                RET
                                 ;*************************************
                                 END
```

:

s:

02AA'	BDOS	0216*	BLKNO	E682	BLKSZ	E680
D8E8	CR	000D	DIGIT	0286'	DIS	0021
E680	DRVLET	02DB'	ERROR	013C'	FLAG	E681
016D'	GETFNA	011C*	HEXDEC	025D'	ISUSE	02B2'
02A0'	LF	000A	LOOP1	02AC'	LOOP2	0237'
0289'	LOOPS	02BB'	LOOPT	02CD'	LOPS	010C'
02C4'	NAMPR	02B5'	NOFILE	0178*	PRINT	00BC
01CB'	RDWRTE	01C9'	READTY	E684	RO	013E'
014J'	SCRRST	0010	SETDMA	0000*	SFR	016D'
0100I'	STAT0	01D4'	STAT1	01F2'	STAT2	01F6'
01CE'	STORE	D8F8				

tal error(s)

```
                .Z80
000'            DSEG
                ;
                ;********** DISC SYSTEM VARIABLES **********
                ;
000             PGPORT          EQU     0
008             DEHL            EQU     8
0F4             SWITCH0         EQU     0F4H
250             BASIC2          EQU     250H
AD2             PAGE            EQU     0FAD2H
                ;
                ;Externals for Page 0 calls
                ;
C45             AE              EQU     3C45H
D84             EVALAB          EQU     3D84H
E7E             EVALSE          EQU     3E7EH
FE9             FIND1$          EQU     3FE9H
927             GETINP          EQU     2927H
0B7             GOTMIN1         EQU     20B7H
00A             INT             EQU     200AH
AF5             SLOAD1          EQU     2AF5H
FC6             STR$            EQU     3FC6H
C4F             ADJVAL          EQU     0C4FH
                ;
                ;
                ;
                EXT     EXCNFG,EXRD,EXWR,BLKRD,INITLZ,CCPSTART
                ;
                ;CP/M Memory Map
                ;
                ;D700H          BDOS (28 records)
                ;D840H          FCBs for DB
                ;E500H          BIOS (7 records)
                ;E680H          DMA default address
                ;E880H          Disc assigment table
                ;E8C0H          Directory buffer
                ;E940H          Allocation vectors
                ;E9C0H          Start of free space
                ;
                .PHASE 0E9C0H
                ;
E9C0            DBUF::          DS 256
EAC0            TDBUF::         DS 192
EB80            NSTK::
                ;
                ;Scratch Pad variables
                ;
EB80            PTRKP::         DS 2
EB82            LCA::           DS 1
EB83            EFLAG::         DS 1
EB84            LSTOUT::        DS 1
EB85            SWUF::          DS 1
EB86            CURDRV::        DS 1
EB87            TRACKS::        DS 4
EB8B            BFID::          DS 8
                ;
```

```
                        CFGTAB::        DS 8
                        ;
                        TRUST::         DS 1
                        DRVRQ::         DS 1
                        CFGBYT::        DS 1
                        TRKRQ::         DS 2
                        SECRQ::         DS 2
                        DMARQ::         DS 2
                        ;
                        CDDRV::         DS 1            ;4
                        BPNT::          DS 1            ;44H
                        RETRY::         DS 1            ;45H
                        TOAM::          DS 2            ;46H
                        ;
                        JPLINK::                        ;Link for jump table
C9                              RET
00                              NOP
00                              NOP
                        ULINK1::                        ;Spare link
C9                              RET
00                              NOP
00                              NOP
                        ;
                        ;
                        SKEW6::         DS 1            ;Dummy address - not needed with type 3
                        ;
                        PCODE::
                        ;
                C       INCLUDE PCODE.RAM
                C       ;Include file for DC and DV
                C       ;
CD EBE1         C       CNFG::  CALL DISCROM
0000*           C               DW EXCNFG
C9              C               RET
                C       ;
E5              C       READ::  PUSH HL
CD EBE1         C               CALL DISCROM
0000*           C               DW EXRD
E1              C               POP HL
C9              C               RET
                C       ;
                C       WRITE::
E5              C               PUSH HL
CD EBE1         C               CALL DISCROM
0000*           C               DW EXWR
E1              C               POP HL
C9              C               RET
                C       ;
                C       BBLKRD::
CD EBE1         C               CALL DISCROM
0000*           C               DW BLKRD
C9              C               RET
                C       ;
                C       INITLIZ::
CD EBE1         C               CALL DISCROM
0000*           C               DW INITLZ
```

```
01    C9            C                 RET
                    C       ;
02    CD EBE1       C                 CALL DISCROM
05    0000*         C                 DW CCPSTART
07    C9            C                 RET
08    00            C                 NOP
                    C       ;
                    C       ;
                    C       ;Switch in DISC ROM.
                    C       ;
9                   C       SWROM::
9     3E 70         C                 LD A,70H                ;Dummy value
                    C       ;
                    C       ;Switch page.
                    C       ;
B                   C       SWPAGE::
B     32 FAD2       C                 LD (PAGE),A
E     D3 00         C                 OUT (PGPORT),A
0     C9            C                 RET
                    C       ;
                    C       ;
                    C       ;DISCROM entry point, return address on stack points to CALL address
                    C       ;Return address is adjusted and subroutine in ROM is called. DE and BC
                    C       ;preserved. Entry value of A preserved.
                    C       ;
                    C       DISCROM::
      E5            C                 PUSH HL
      FD E1         C                 POP IY                  ;Save HL
      E1            C                 POP HL
      D5            C                 PUSH DE
      5E            C                 LD E,(HL)
      23            C                 INC HL
      56            C                 LD D,(HL)
      23            C                 INC HL
      E3            C                 EX (SP),HL              ;New return address to stack
      EB            C                 EX DE,HL                ;DE restored
      E5            C                 PUSH HL                 ;Call address to stack
      6F            C                 LD L,A
      3A FAD2       C                 LD A,(PAGE)
      67            C                 LD H,A
      CD E3D9       C                 CALL SWROM              ;Select ROM
      7D            C                 LD A,L                  ;Restore A
      E3            C                 EX (SP),HL              ;Page to stack
      E5            C                 PUSH HL                 ;Call address to stack
      21 EC01       C                 LD HL,CALSUB
      E3            C                 EX (SP),HL              ;CALLSUB to stack
      E5            C                 PUSH HL                 ;Call address to stack
      FD E5         C                 PUSH IY
      E1            C                 POP HL                  ;Restore HL
      C9            C                 RET
                    C       ;
                    C       CALSUB::
      E1            C                 POP HL
      F5            C                 PUSH AF
      7C            C                 LD A,H
      C2 EBD9       C                 CALL SUBSET
```

```
07    F1            C              POP AF
08    C9            C              RET
                    C    ;
                    C    ;
                    C    ;
                    C    ;PAGE0 calls a specified routine in ROM page zero. Byte on top of machine
                    C    ;stack gives offset into jump table. Switches to page zero before jumping to
                    C    ;routine, and switches back to NODE ROM page on return.
                    C    ;Affects no registers, except AF'.
                    C    ;
09                  C    PAGE0::
09    08            C              EX AF,AF'                ;Save true AF
0A    E3            C              EX (SP),HL               ;HL -> data byte
0B    7E            C              LD A,(HL)                ;A = offset
0C    23            C              INC HL
0D    E3            C              EX (SP),HL
                    C    ;
                    C    ;Registers can now be pushed onto stack.
                    C    ;
0E    E5            C              PUSH HL                  ;Save HL
0F    21 EC28       C              LD HL,PAGEX              ;Return address for page 0 routine
12    E3            C              EX (SP),HL               ;Restore HL
13    E5            C              PUSH HL
14    D5            C              PUSH DE
15    C5            C              PUSH BC                  ;Save HL,DE,BC
                    C    ;
16    4F            C              LD C,A
17    06 00         C              LD B,0                   ;BC = offset
19    08            C              EX AF,AF'                ;Restore true AF
                    C    ;
1A    21 EC2E       C              LD HL,JPTABLE
1D    09            C              ADD HL,BC
1E    09            C              ADD HL,BC                ;HL -> address of required routine
1F    CF            C              RST DEHL
20    EB            C              EX DE,HL                 ;HL = address of routine
                    C    ;
21    C1            C              POP BC
22    D1            C              POP DE                   ;Restore BC,DE
                    C    ;
                    C    ;Here HL -> address of routine, (SP) = true value of HL.
                    C    ;Switch in page 0 and call routine.
                    C    ;
23    CD 00F4       C              CALL SWITCH0             ;Switch in ROM page 0
26    E3            C              EX (SP),HL               ;HL = true HL, (SP) -> routine.
27    C9            C              RET                      ;'Call' routine
                    C    ;
                    C    ;The called routine returns to here.
                    C    ;
                    C    PAGEX::
29    F5            C              PUSH AF
2A    CD EBD9       C              CALL SWROM               ;Switch in RING ROM page
2D    F1            C              POP AF
                    C    ;
                    C    SPARE::
2E    C9            C              RET
```

```
                        C       ;
                        C       ;
2E                      C       JPTABLE::
                        C       ;
2E  3C45                C       ADD0:   DW      AE
30  3D84                C       ADD1:   DW      EVALAB
32  3E7E                C       ADD2:   DW      EVALSE
34  3FE9                C       ADD3:   DW      FIND1$
36  2927                C       ADD4:   DW      GETINP
38  20B7                C       ADD5:   DW      GOTMIN1
3A  200A                C       ADD6:   DW      INT
3C  2AF5                C       ADD7:   DW      SLOAD1
3E  3FC6                C       ADD8:   DW      STR$
40  0C4F                C       ADD9:   DW      ADJVAL
42  0030                C       ADD10:  DW      30H             ;RESET GETRST
44  288F                C       ADD11:  DW      288FH           ;SGOTO
                        C       ;
                        C       ;
46                      C       RETBASIC::
46  3E 00               C               LD A,0
48  CD EBDB             C               CALL SWPAGE
4B  C3 0250             C               JP BASIC2
                        C       ;
                        C       ;       DISC PARAMETER BLOCK SET
                        C       ;
5E                      C       PBASE::
5E  03                  C               DB      3       ;       5IN S/T D/D D/S
5F  001A                C               DW      26
61  04                  C               DB      4
62  0F                  C               DB      15
63  01                  C               DB      1
64  0098                C               DW      155
66  003F                C               DW      63
68  80                  C               DB      10000000B
69  00                  C               DB      00000000B
6A  0010                C               DW      16
6C  0002                C               DW      2
                        C       ;
6E  FF                  C       PEND::  DB      0FFH            ;TERMINATOR
                        C       ;
                        C               END
```

os:

ols:

EC2E	ADD1	EC30	ADD10	EC42	ADD11	EC44
EC32	ADD3	EC34	ADD4	EC36	ADD5	EC38
EC3A	ADD7	EC3C	ADD8	EC3E	ADD9	EC40
AL 0C4F	AE	3C45	BASIC2	0250	BBLKRD	EBC6I
EB8BI	BLKRD	0209*	BPNT	EBA5I	CALSUB	EC01I
TA 0215*	CDDRV	EBA4I	CFGBYT	EB9DI	CFGTAB	EB93I
EBB0I	CURDRV	EB86I	DBUF	E9C0I	DEHL	0008
RO EBE1I	DMARQ	EBA2I	DRVRQ	EB9CI	EFLAG	EB83I
AB 3D84	EVALSE	3E7E	EXCNFG	01F3*	EXRD	01FA*
0202*	FIND1$	3FE9	GETINP	2927	GOTMIN	20B7
LI EBCCI	INITLZ	020F*	INT	200A	JPLINK	EBA9I
BL EC2EI	LCA	EB82I	LSTOUT	EB84I	NSTK	EB80I
FAD2	PAGE0	EC09I	PAGEX	EC28I	PBASE	EC4EI
E EBB0I	PEND	EC5EI	PGPORT	0000	PTRKP	EB80I
EBB6I	RETBAS	EC46I	RETRY	EBA6I	SECRQ	EBA0I
6 EBAFI	SLOAD1	2AF5	SPARE	EC2DI	STR$	3FC6
CH 00F4	SWPAGE	EBDBI	SWROM	EBD9I	SWUF	EB85I
F EAC0I	TOAM	EBA7I	TRACKS	EB87I	TRKRQ	EB9EI
- EB9BI	ULINK1	EBACI	WRITE	EBBEI		

atal error(s)

```
                        ;
                        .Z80
000'                    DSEG
                        .PHASE 0F5B0H
                        ;
                        ;
                        ;The position of these variables is independent of the CP/M system size.
                        ;
5B0                     BDOS::          DS 3
5B3                     USRJMP::        DS 32
5D3                     DSCFLG::        DS 1
5D4                     KEYJP::         DS 44
                        ;
                        .DEPHASE
                                END
```