```
; Disassembly of the file "C:\lab\if1-1.rom"
;
; CPU Type: Z80
;
; Created with dZ80 1.50
;
; on Sunday, 28 of April 2002 at 12:33 PM
;
; ---
;
; Last Updated: 14-JAN-2004
;
; Credits:      John Hutcheson          Documentation.
;               Geoff Wearmouth         Current Maintainer

#define DEFB .BYTE
#define DEFW .WORD
#define DEFM .TEXT
#define EQU  .EQU
#define ORG  .ORG

        ORG     $0000


; --------------------------------
; THE 'RETURN TO MAIN ROM' ROUTINE
; --------------------------------
;

;; MAIN-ROM
L0000:  POP     HL
        LD      (IY+$7C),$00    ; sv FLAGS_3
        JP      L0700           ; jump forward to UNPAGE

; -------------------
; THE 'START' ROUTINE
; -------------------
;

;; ST-SHADOW
L0008:  LD      HL,($5C5D)      ; sv CH_ADD
        POP     HL
        PUSH    HL
        JP      L009A           ; jump forward to START-2

; ----------------------------
; THE 'CALL A MAIN ROM' ROUTINE
; ----------------------------
;

;; CALBAS
L0010:  LD      ($5CBA),HL      ; sv SBRT
        POP     HL
        PUSH    DE
        JR      L0081           ; forward to CALBAS-2

        DEFB    $FF             ; unused

; ------------------------------------------
; THE 'TEST IF SYNTAX IS BEING CHECKED' ROUTINE
; ------------------------------------------
```

```
;

;; CHKSYNTAX
L0018:  BIT     7,(IY+$01)      ; sv FLAGS
        RET

        DEFB    $FF             ; unused
        DEFB    $FF             ; unused
        DEFB    $FF             ; unused

; -------------------------
; THE 'SHADOW-ERROR' ROUTINE
; -------------------------
;

;; SH-ERR
L0020:  RST     18H
        JR      Z,L0068         ; forward to ST-ERROR

        JR      L003A           ; forward to TEST-SP

        DEFB    $FF             ; unused
        DEFB    $FF             ; unused
        DEFB    $FF             ; unused

; ----------------------------------
; THE 'MAIN ROM ERROR RESTART' ROUTINE
; ----------------------------------
;

;; ROMERR
L0028:  RES     3,(IY+$02)      ; sv TV_FLAG
        JR      L0040           ; forward to RMERR-2

        DEFB    $FF             ; unused
        DEFB    $FF             ; unused

; ------------------------------------------------
; THE 'CREATE NEW SYSTEM VARIABLES RESTART' ROUTINE
; ------------------------------------------------
; This restart is used twice to create the new system variables.

;; NEWVARS
L0030:  JP      L01F7           ; jump forward to CRT-VARS

        DEFB    $FF             ; unused
        DEFB    $FF             ; unused
        DEFB    $FF             ; unused
        DEFB    $FF             ; unused
        DEFB    $FF             ; unused

; --------------------------------
; THE 'MASKABLE INTERRUPT' ROUTINE
; --------------------------------
;

;; INT-SERV
L0038:  EI
        RET


; ---------------------
; THE 'TEST-SP' ROUTINE
; ---------------------
```

```
;

;; TEST-SP
L003A:  CALL    L0077           ; routine CHECK-SP
        JP      L0258           ; jump forward to REP-MSG

; ---------------------------
; THE 'MAIN ROM ERROR' ROUTINE
; ---------------------------
;

;; RMERR-2
L0040:  RST     18H
        JR      Z,L0068         ; forward to ST-ERROR

        CALL    L0077           ; routine CHECK-SP
        CALL    L17B9           ; routine RCL-T-CH
        BIT     1,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0068         ; forward to ST-ERROR

        BIT     4,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0068         ; forward to ST-ERROR

        LD      A,(IY+$00)      ; sv ERR_NR
        CP      $14
        JR      NZ,L0068        ; forward to ST-ERROR

        LD      HL,L0000
        PUSH    HL
        RST     00H

        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF

; -----------------------------------
; THE 'NON-MASKABLE INTERRUPT' ROUTINE
; -----------------------------------
;

;; NMINT-SRV
L0066:  RETN


; ----------------------
; THE 'ST-ERROR' ROUTINE
; ----------------------
;

;; ST-ERROR
L0068:  LD      HL,($5C5D)      ; sv CH_ADD
        LD      ($5C5F),HL      ; sv X_PTR
        LD      SP,($5C3D)      ; sv ERR_SP
        LD      HL,$16C5
        PUSH    HL
        RST     00H

; ----------------------
; THE 'CHECK-SP' ROUTINE
; ----------------------
;
```

```
;; CHECK-SP
L0077:  BIT     2,(IY+$7C)      ; sv FLAGS_3
        RET     Z               ;

        LD      SP,($5C3D)      ; sv ERR_SP
        RST     00H             ;

; ----------------------
; THE 'CALBAS-2' ROUTINE
; ----------------------
;

;; CALBAS-2
L0081:  LD      E,(HL)          ;
        INC     HL              ;
        LD      D,(HL)          ;
        LD      ($5CBD),DE      ; sv SBRT
        INC     HL              ;
        EX      (SP),HL         ;
        EX      DE,HL           ;
        LD      HL,L0000        ;
        PUSH    HL              ;
        LD      HL,L0008        ;
        PUSH    HL              ;
        LD      HL,$5CB9        ; sv SBRT
        PUSH    HL              ;
        JP      L0700           ; jump forward to UNPAGE

; ----------------------
; THE 'CONTROL' ROUTINE
; ----------------------
;

;; START-2
L009A:  PUSH    AF
        LD      A,H
        OR      L
        JR      NZ,L00A5        ; forward to START-3

        POP     AF
        POP     HL
        LD      HL,($5CBA)      ; sv SBRT
        RET

; ---


;; START-3
L00A5:  PUSH    DE
        LD      DE,$15FE
        SBC     HL,DE
        POP     DE
        JR      NZ,L00BC        ; forward to START-4

        POP     AF
        LD      HL,L0700
        PUSH    HL
        LD      HL,$0004
        ADD     HL,DE
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        EX      DE,HL
        JP      (HL)
```

```
; ---

;; START-4
L00BC:  RST     30H             ; NEWVARS
        LD      A,$01
        OUT     ($F7),A
        LD      A,$EE
        OUT     ($EF),A
        POP     AF
        POP     HL
        PUSH    AF
        RST     10H             ; CALBAS
        DEFW    $007B           ; main TEMP-PTR3
        LD      ($5C3A),A       ; sv ERR_NR
        CP      $FF
        JR      NZ,L00E9        ; forward to TEST-CODE

        BIT     1,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L00E7         ; forward to NREPORT-0

        BIT     7,(IY+$0C)      ; sv PPC_hi
        JR      Z,L00E7         ; forward to NREPORT-0

        LD      HL,($5C59)      ; sv E_LINE
        LD      A,(HL)
        CP      $F7
        JP      Z,L0A95         ; jump forward to LOAD-RUN

;; NREPORT-0
L00E7:  RST     20H             ; sh_err
        DEFB    $FF             ; 'Program finished'

; ---

;; TEST-CODE
L00E9:  SUB     $1B
        JP      NC,L1981        ; jump forward to HOOK-CODE

        CP      $F0
        JR      Z,L00FB         ; forward to COPYCHADD

        CP      $F3
        JR      Z,L00FB         ; forward to COPYCHADD

        CP      $FC
        JP      NZ,L0028        ; jump to ROMERR

;; COPYCHADD
L00FB:  LD      HL,($5C5D)      ; sv CH_ADD
        LD      ($5CCB),HL      ; sv CHADD_
        POP     AF
        BIT     5,(IY+$37)      ; sv FLAGX
        JP      NZ,L0028        ; jump to ROMERR
        BIT     0,(IY+$7C)      ; sv FLAGS_3
        JP      NZ,L0028        ; jump to ROMERR
        SET     0,(IY+$7C)      ; sv FLAGS_3
        RST     18H
        JR      NZ,L011B        ; forward to RUNTIME

        LD      (IY+$0C),$FF    ; sv PPC_hi

;; RUNTIME
L011B:  LD      B,(IY+$0D)      ; sv SUBPPC
```

```
        LD      C,$00
        BIT     7,(IY+$0C)      ; sv PPC_hi
        JR      Z,L0130         ; forward to PROG-LINE

        PUSH    BC
        RST     10H             ; CALBAS
        DEFW    $19FB           ; main E-LINE-NO
        POP     BC
        RST     10H             ; CALBAS
        DEFW    $0018           ; main GET-CHAR

        JR      L016F           ; forward to S-STAT

; ---

;; PROG-LINE
L0130:  LD      HL,($5C53)      ; sv PROG

;; SC-L-LOOP
L0133:  LD      A,($5C46)       ; sv PPC_hi
        CP      (HL)
        JR      NC,L013B        ; forward to TEST-LOW


;; NREPORT-1
L0139:  RST     20H             ; sh_err
        DEFB    $00             ; 'Nonsense in BASIC'

; ---

;; TEST-LOW
L013B:  INC     HL
        JR      NZ,L0144        ; forward to LINE-LEN

        LD      A,($5C45)       ; sv PPC
        CP      (HL)
        JR      C,L0139         ; back to NREPORT-1


;; LINE-LEN
L0144:  INC     HL
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        INC     HL
        JR      Z,L016F         ; forward to S-STAT

        ADD     HL,DE
        JR      L0133           ; back to SC-L-LOOP

; ---


;; SKIP-NUM
L014E:  LD      DE,$0006
        ADD     HL,DE

;; EACH-ST
L0152:  LD      A,(HL)
        CP      $0E
        JR      Z,L014E         ; back to SKIP-NUM

        INC     HL
        CP      $22
```

```
        JR      NZ,L015D        ; forward to CHKEND

        DEC     C

;; CHKEND
L015D:  CP      $3A
        JR      Z,L0165         ; forward to CHKEVEN

        CP      $CB
        JR      NZ,L0169        ; forward to CHKEND-L


;; CHKEVEN
L0165:  BIT     0,C
        JR      Z,L016F         ; forward to S-STAT


;; CHKEND-L
L0169:  CP      $0D
        JR      NZ,L0152        ; back to EACH-ST

        JR      L0139           ; back to NREPORT-1

; ---


;; S-STAT
L016F:  DJNZ    L0152           ; back to EACH-ST

        DEC     HL
        LD      ($5C5D),HL      ; sv CH_ADD
        RST     18H
        JR      NZ,L01AA        ; forward to CL-WORK

        BIT     7,(IY+$0C)      ; sv PPC_hi
        JP      Z,L01F0         ; jump forward to ERR-6
        DEC     HL
        LD      C,$00

;; RCLM-NUM
L0182:  INC     HL
        LD      A,(HL)
        CP      $0E
        JR      NZ,L01A5        ; forward to NEXTNUM

        PUSH    BC
        LD      BC,$0006
        RST     10H             ; CALBAS
        DEFW    $19E8           ; main RECLAIM-2
        PUSH    HL
        LD      DE,($5CCB)      ; sv CHADD_
        AND     A
        SBC     HL,DE
        JR      NC,L01A3        ; forward to NXT-1

        EX      DE,HL
        LD      BC,$0006
        AND     A
        SBC     HL,BC
        LD      ($5CCB),HL      ; sv CHADD_

;; NXT-1
L01A3:  POP     HL
        POP     BC
```

```
;; NEXTNUM
L01A5:  LD      A,(HL)
        CP      $0D
        JR      NZ,L0182        ; back to RCLM-NUM


;; CL-WORK
L01AA:  RST     10H             ; CALBAS
        DEFW    $16BF           ; main SET-WORK
        CALL    $024D
        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        SUB     $CE
        CP      $01
        JP      Z,L0486         ; jump forward to CAT-SYN
        CP      $02
        JP      Z,L04B4         ; jump forward to FRTM-SYN
        CP      $03
        JP      Z,L053D         ; jump forward to MOVE-SYN
        CP      $04
        JP      Z,L0531         ; jump forward to ERASE-SYN
        CP      $05
        JP      Z,L04ED         ; jump forward to OPEN-SYN
        CP      $2A
        JP      Z,L082F         ; jump forward to SAVE-SYN
        CP      $21
        JP      Z,L0894         ; jump forward to LOAD-SYN
        CP      $08
        JP      Z,L089E         ; jump forward to VERIF-SYN
        CP      $07
        JP      Z,L08A8         ; jump forward to MRG-SYN
        CP      $2D
        JP      Z,L0559         ; jump forward to CLS#-SYN
        CP      $2F
        JP      Z,L057F         ; jump forward to CLR#-SYN

; finally if none of these,

        LD      HL,($5CB7)      ; sv VECTOR
        JP      (HL)

; ---

;; ERR-6
L01F0:  LD      HL,($5CCB)      ; sv CHADD_
        LD      ($5C5D),HL      ; sv CH_ADD
        RST     28H             ; main romerr

; ----------------------------------------
; THE 'CREATE NEW SYSTEM VARIABLES' ROUTINE
; ----------------------------------------
;

;; CRT-VARS
L01F7:  LD      HL,($5C4F)      ; sv CHANS
        LD      DE,$A349
        ADD     HL,DE
        JR      C,L0235         ; forward to VAR-EXIST

        LD      HL,L0224        ; Address DEFAULT below
        PUSH    HL
        LD      HL,($5C63)      ; sv STKBOT
        LD      ($5C65),HL      ; sv STKEND
```

```
        LD      HL,$5C92        ; sv MEM_0
        LD      ($5C68),HL      ; sv MEM
        LD      HL,$5CB5        ; sv P_RAMT_hi
        LD      BC,L003A
        LD      DE,L0000
        PUSH    DE
        LD      E,$08
        PUSH    DE
        LD      DE,$1655
        PUSH    DE
        JP      L0700           ; jump forward to UNPAGE

; and then back here

;; DEFAULT
L0224:  LD      HL,L023A
        LD      BC,$0013
        LD      DE,$5CB6        ; sv FLAGS_3
        LDIR

;   Note.  Accumulator may hold stream to close.

        LD      A,$01           ;
        LD      ($5CEF),A       ; sv COPIES
        RET                     ;


;; VAR-EXIST
L0235:  RES     1,(IY+$7C)      ; sv FLAGS_3
        RET


; -----------------------------------------
; THE 'SYSTEM VARIABLES DEFAULT VALUES' ROUTINE
; -----------------------------------------
;

;; SV_DEFVAL
L023A:  DEFB    $02
        DEFW    $01F0           ;
        LD      HL,$0000
        CALL    $0000
        LD      ($5CBA),HL      ; sv SBRT
        RET
        DEFW    $000C           ;
        DEFB    $01
        DEFB    $00
        DEFW    $0000           ;


; --------------------------------------
; THE 'RESET NEW SYSTEM VARIABLES' ROUTINE
; --------------------------------------
;

;; RES-VARS
L024D:  LD      HL,$5CCD        ; sv NTRESP
        LD      B,$22

;; EACH-VAR
L0252:  LD      (HL),$FF
        INC     HL
        DJNZ    L0252           ; back to EACH-VAR
```

```
        RET


; -------------------------------------
; THE 'SHADOW REPORT PRINTING' ROUTINE
; -------------------------------------
;

;; REP-MSG
L0258:  LD      (IY+$7C),$00    ; sv FLAGS_3
        EI
        HALT
        CALL    L17B9           ; routine RCL-T-CH
        RES     5,(IY+$01)      ; sv FLAGS
        BIT     1,(IY+$30)      ; sv FLAGS2
        JR      Z,L026E         ; forward to FETCH-ERR

        RST     10H             ; CALBAS
        DEFW    $0ECD           ; main COPY-BUFF



;; FETCH-ERR
L026E:  POP     HL
        LD      A,(HL)
        LD      (IY+$00),A      ; sv ERR_NR
        INC     A
        PUSH    AF

        LD      HL,$0000
        LD      (IY+$37),H      ; sv FLAGX
        LD      (IY+$26),H      ; sv X_PTR_hi
        LD      ($5C0B),HL      ; sv DEFADD
        INC     L
        LD      ($5C16),HL      ; sv STRMS_00
        RST     10H             ; CALBAS
        DEFW    $16B0           ; main SET-MIN
        RES     5,(IY+$37)      ; sv FLAGX
        RST     10H             ; CALBAS
        DEFW    $0D6E           ; main CLS-LOWER
        SET     5,(IY+$02)      ; sv TV_FLAG
        RES     3,(IY+$02)      ; sv TV_FLAG
        POP     AF
        LD      HL,$02B7
        LD      B,$04
        CPIR

;; PR-REP-LP
L029F:  LD      A,(HL)
        CP      $20
        JR      C,L02AC         ; forward to END-PR-MS

        PUSH    HL
        RST     10H             ; CALBAS
        DEFW    $0010           ; main PRINT-A
        POP     HL
        INC     HL
        JR      L029F           ; back to PR-REP-LP


;; END-PR-MS
L02AC:  LD      SP,($5C3D)      ; sv ERR_SP
        INC     SP
        INC     SP
```

```
        LD      HL,L1349
        PUSH    HL
        RST     00H


; -----------------------------------
; THE 'SHADOW REPORT MESSAGES' ROUTINE
; -----------------------------------
;

;;
L02B7:  DEFB    $00
        DEFM    "Program finished"
        DEFB    $01
        DEFM    "Nonsense in BASIC"
        DEFB    $02
        DEFM    "Invalid stream number"
        DEFB    $03
        DEFM    "Invalid device expression"
        DEFB    $04
        DEFM    "Invalid name"
        DEFB    $05
        DEFM    "Invalid drive number"
        DEFB    $06
        DEFM    "Invalid station number"
        DEFB    $07
        DEFM    "Missing name"
        DEFB    $08
        DEFM    "Missing station number"
        DEFB    $09
        DEFM    "Missing drive number"
        DEFB    $0A
        DEFM    "Missing baud rate"
        DEFB    $0B
        DEFM    "Header mismatch error" ; not used.
        DEFB    $0C
        DEFM    "Stream already open"
        DEFB    $0D
        DEFM    "Writing to a 'read' file"
        DEFB    $0E
        DEFM    "Reading a 'write' file"
        DEFB    $0F
        DEFM    "Drive 'write' protected"
        DEFB    $10
        DEFM    "Microdrive full"
        DEFB    $11
        DEFM    "Microdrive not present"
        DEFB    $12
        DEFM    "File not found"
        DEFB    $13
        DEFM    "Hook code error"
        DEFB    $14
        DEFM    "CODE error"
        DEFB    $15
        DEFM    "MERGE error"
        DEFB    $16
        DEFM    "Verification has failed"
        DEFB    $17
        DEFM    "Wrong file type"
        DEFB    $18


; --------------------------------
; THE 'CAT COMMAND SYNTAX' ROUTINE
```

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

;; CAT-SYN
L0486:  LD      HL,$5CD8        ; sv D_STR1
        LD      (HL),$02
        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CP      $0D
        JR      Z,L0494         ; forward to MISSING-D

        CP      $3A

;; MISSING-D
L0494:  JP      Z,L0683         ; jump forward to NREPORT-9
        CP      $23
        JR      NZ,L04A6        ; forward to CAT-SCRN

        CALL    L064E           ; routine EXPT-STRM
        CALL    L05B1           ; routine SEPARATOR
        JR      NZ,L04B2        ; forward to OREPORT-1

        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR

;; CAT-SCRN
L04A6:  CALL    L061E           ; routine EXPT-NUM
        CALL    L05B7           ; routine ST-END
        CALL    L066D           ; routine CHECK-M-2
        JP      L1E70           ; jump forward to CAT-RUN

;; OREPORT-1
L04B2:  RST     20H             ; sh_err
        DEFB    $00             ; 'Nonsense in BASIC'

; ----------------------------------
; THE 'FORMAT COMMAND SYNTAX' ROUTINE
; ----------------------------------
;

;; FRTM-SYN
L04B4:  CALL    L05F2           ; routine EXPT-SPEC
        CALL    L05B1           ; routine SEPARATOR
        JR      NZ,L04BF        ; forward to NO-FOR-M

        CALL    L062F           ; routine EXPT-NAME

;; NO-FOR-M
L04BF:  CALL    L05B7           ; routine ST-END
        LD      A,($5CD9)       ; sv D_STR1
        CP      $54
        JR      Z,L04CD         ; forward to FOR-B-T

        CP      $42
        JR      NZ,L04D3        ; forward to NOT-FOR-B


;; FOR-B-T
L04CD:  CALL    L06B0           ; routine TEST-BAUD
        JP      L0AC9           ; jump forward to SET-BAUD

;; NOT-FOR-B
L04D3:  CP      $4E
        JR      NZ,L04E7        ; forward to FOR-M
```

```
        CALL    L068F           ; routine TEST-STAT
        LD      A,($5CD6)       ; sv D_STR1
        AND     A
        JP      Z,L069F         ; jump forward to NREPORT-6
        LD      ($5CC5),A       ; sv NTSTAT
        JP      L05C1           ; jump forward to END1

;; FOR-M
L04E7:  CALL    L0685           ; routine TEST-MNAM
        JP      L1E75           ; jump forward to IFOR-RUN

; ---------------------------------
; THE 'OPEN COMMAND SYNTAX' ROUTINE
; ---------------------------------
;

;; OPEN-SYN
L04ED:  CALL    L064E           ; routine EXPT-STRM
        CALL    L05B1           ; routine SEPARATOR
        JR      NZ,L04B2        ; back to OREPORT-1

        CALL    L05F2           ; routine EXPT-SPEC
        CALL    L05B1           ; routine SEPARATOR
        JR      NZ,L0500        ; forward to NOT-OP-M

        CALL    L062F           ; routine EXPT-NAME

;; NOT-OP-M
L0500:  CALL    L05B7           ; routine ST-END
        LD      A,($5CD8)       ; sv D_STR1
        RST     10H             ; CALBAS
        DEFW    $1727           ; main STR-DATA1
        LD      HL,$0011
        AND     A
        SBC     HL,BC
        JR      C,L052F         ; forward to NREPORT-C

        LD      A,($5CD9)       ; sv D_STR1
        CP      $54
        JR      Z,L051C         ; forward to OPEN-RS

        CP      $42
        JR      NZ,L051F        ; forward to NOT-OP-B


;; OPEN-RS
L051C:  JP      L0B47           ; jump forward to OP-RSCHAN

;; NOT-OP-B
L051F:  CP      $4E
        JR      NZ,L0529        ; forward to OP-M-C

        CALL    L068F           ; routine TEST-STAT
        JP      L0EA3           ; jump forward to OPEN-N-ST

;; OP-M-C
L0529:  CALL    L0685           ; routine TEST-MNAM
        JP      L1E7A           ; jump forward to OP-RUN

;; NREPORT-C
L052F:  RST     20H             ; sh_err
        DEFB    $0B             ; 'Stream already open'
```

```
; ---------------------------------
; THE 'ERASE COMMAND SYNTAX' ROUTINE
; ---------------------------------
;

;; ERASE-SYN
L0531:  CALL    L06A3           ; routine EXOT-EXPR
        CALL    L05B7           ; routine ST-END
        CALL    L0685           ; routine TEST-MNAM
        JP      L1E66           ; jump forward to ERASE-RUN

; ---------------------------------
; THE 'MOVE COMMAND SYNTAX' ROUTINE
; ---------------------------------
;

;; MOVE-SYN
L053D:  CALL    L06B9           ; routine EXPT-EXP1
        CALL    L059F           ; routine EX-D-STR
        RST     10H             ; CALBAS
        DEFW    $0018           ; main GET-CHAR
        CP      $CC
        JR      NZ,L0584        ; forward to NONSENSE

        CALL    L06B9           ; routine EXPT-EXP1
        CALL    L059F           ; routine EX-D-STR
        RST     10H             ; CALBAS
        DEFW    $0018           ; main GET-CHAR
        CALL    L05B7           ; routine ST-END
        JP      L1E6B           ; jump forward to MOVE-RUN

; --------------------------
; THE 'CLS# COMMAND' ROUTINE
; --------------------------
;

;; CLS#-SYN
L0559:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CP      $23
        JR      NZ,L0584        ; forward to NONSENSE

        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CALL    L05B7           ; routine ST-END
        LD      HL,L0038
        LD      ($5C8D),HL      ; sv ATTR_P
        LD      ($5C8F),HL      ; sv ATTR_T
        LD      (IY+$0E),L      ; sv BORDCR
        LD      (IY+$57),H      ; sv P_FLAG
        LD      A,$07
        OUT     ($FE),A
        RST     10H             ; CALBAS
        DEFW    $0D6B           ; main CLS
        JP      L05C1           ; jump forward to END1

; ---------------------------
; THE 'CLEAR# COMMAND' ROUTINE
; ---------------------------
;

;; CLR#-SYN
L057F:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
```

```
        CP      $23

;; NONSENSE
L0584:  JP      NZ,L04B2        ; jump to OREPORT-1
        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CALL    L05B7           ; routine ST-END
        XOR     A

;; ALL-STRMS
L058E:  PUSH    AF
        SET     1,(IY+$7C)      ; sv FLAGS_3
        CALL    L1718           ; routine CLOSE
        POP     AF
        INC     A
        CP      $10
        JR      C,L058E         ; back to ALL-STRMS

        JP      L05C1           ; jump forward to END1

; -------------------------------------
; THE 'EXCHANGE FILE SPECIFIERS' ROUTINE
; -------------------------------------
;

;; EX-D-STR
L059F:  LD      HL,$5CD6        ; sv D_STR1
        LD      DE,$5CDE        ; sv D_STR2
        LD      B,$08

;; ALL-BYTES
L05A7:  LD      A,(DE)
        LD      C,(HL)
        LD      (HL),A
        LD      A,C
        LD      (DE),A
        INC     HL
        INC     DE
        DJNZ    L05A7           ; back to ALL-BYTES

        RET


; ----------------------
; THE 'SEPARATOR' ROUTINE
; ----------------------
;

;; SEPARATOR
L05B1:  CP      $2C             ; the ',' character
        RET     Z

        CP      $3B             ; the ';' character
        RET


; ------------------------------
; THE 'END OF STATEMENT' ROUTINE
; ------------------------------
;

;; ST-END
L05B7:  CP      $0D
        JR      Z,L05BF         ; forward to TEST-RET
```

```
        CP      $3A
        JR      NZ,L0584        ; back to NONSENSE


;; TEST-RET
L05BF:  RST     18H
        RET     NZ


; -----------------------------------------
; THE 'RETURN TO THE MAIN INTERPRETER' ROUTINE
; -----------------------------------------
;

;; END1
L05C1:  LD      SP,($5C3D)      ; sv
        LD      (IY+$00),$FF    ; sv ERR_NR
        LD      HL,$1BF4
        RST     18H
        JR      Z,L05E0         ; forward to RETAD-SYN

        LD      A,$7F
        IN      A,($FE)
        RRA
        JR      C,L05DD         ; forward to RETAD-RUN

        LD      A,$FE
        IN      A,($FE)
        RRA
        JR      NC,L05E2        ; forward to BREAK-PGM


;; RETAD-RUN
L05DD:  LD      HL,$1B7D

;; RETAD-SYN
L05E0:  PUSH    HL
        RST     00H

;; BREAK-PGM
L05E2:  LD      (IY+$00),$14    ; sv ERR_NR
        RST     28H             ; romerr

; ---------------------------------------
; THE 'EVALUATE STRING EXPRESSION' ROUTINE
; ---------------------------------------
;

;; EXPT-STR
L05E7:  RST     10H             ; CALBAS
        DEFW    $1C8C           ; main EXPT-EXP
        RST     18H
        RET     Z

        PUSH    AF
        RST     10H             ; CALBAS
        DEFW    $2BF1           ; main STK-FETCH
        POP     AF
        RET


; ---------------------------------------
; THE 'EVALUATE CHANNEL EXPRESSION' ROUTINE
```

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

;; EXPT-SPEC
L05F2:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR

;; EXP-SPEC2
L05F5:  CALL    L05E7           ; routine EXPT-STR
        JR      Z,L060C         ; forward to TEST-NEXT

        PUSH    AF
        LD      A,C
        DEC     A
        OR      B
        JR      NZ,L062D        ; forward to NREPORT-3

        LD      A,(DE)
        RST     10H             ; CALBAS
        DEFW    $2C8D           ; main ALPHA
        JR      NC,L062D        ; forward to NREPORT-3

        AND     $DF
        LD      ($5CD9),A       ; sv D_STR1
        POP     AF

;; TEST-NEXT
L060C:  CP      $0D
        RET     Z

        CP      $3A
        RET     Z

        CP      $A5
        RET     NC

        CALL    L05B1           ; routine SEPARATOR
        JP      NZ,L04B2        ; jump to OREPORT-1
        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR

; ----------------------------------------
; THE 'EVALUATE NUMERIC EXPRESSION' ROUTINE
; ----------------------------------------
;

;; EXPT-NUM
L061E:  RST     10H             ; CALBAS
        DEFW    $1C82           ; main EXPT-1NUM
        RST     18H
        RET     Z

        PUSH    AF
        RST     10H             ; CALBAS
        DEFW    $1E99           ; main FIND-INT2
        LD      ($5CD6),BC      ; sv D_STR1
        POP     AF
        RET


;; NREPORT-3
L062D:  RST     20H             ; sh_err
        DEFB    $02             ; 'Invalid device expression'
```

```
; ------------------------------
; THE 'EVALUATE FILENAME' ROUTINE
; ------------------------------
;

;; EXPT-NAME
L062F:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CALL    L05E7           ; routine EXPT-STR
        RET     Z

        PUSH    AF
        LD      A,C
        OR      B
        JR      Z,L064C         ; forward to NREPORT-4

        LD      HL,$000A
        SBC     HL,BC
        JR      C,L064C         ; forward to NREPORT-4

        LD      ($5CDA),BC      ; sv D_STR1
        LD      ($5CDC),DE      ; sv D_STR1
        POP     AF
        RET


;; NREPORT-4
L064C:  RST     20H             ; sh_err
        DEFB    $03             ;

; -----------------------------------
; THE 'EVALUATE STREAM NUMBER' ROUTINE
; -----------------------------------
;

;; EXPT-STRM
L064E:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        RST     10H             ; CALBAS
        DEFW    $1C82           ; main EXPT-1NUM
        RST     18H
        RET     Z

        PUSH    AF
        RST     10H             ; CALBAS
        DEFW    $1E94           ; main FIND-INT1
        CP      $10
        JR      NC,$0663
        LD      ($5CD8),A       ; sv D_STR1
        POP     AF
        RET

        RST     20H             ; sh_err
        DEFB    $01             ;


; ---------------------------------
; THE 'CHECK "M" PARAMETERS' ROUTINE
; ---------------------------------
;

;; CHECK-M
L0665:  LD      A,($5CD9)       ; sv D_STR1
        CP      $4D
```

```
        JP      NZ,L062D        ; jump to NREPORT-3

;; CHECK-M-2
L066D:  LD      DE,($5CD6)      ; sv D_STR1
        LD      A,E
        OR      D
        JR      Z,L0681         ; forward to NREPORT-5

        INC     DE
        LD      A,E
        OR      D
        JR      Z,L0683         ; forward to NREPORT-9

        DEC     DE
        LD      HL,L0008
        SBC     HL,DE
        RET     NC


;; NREPORT-5
L0681:  RST     20H             ; sh_err
        DEFB    $04

;; NREPORT-9
L0683:  RST     20H             ; sh_err
        DEFB    $08

; ----------------------------------------------
; THE 'CHECK "M" PARAMETERS AND FILENAME' ROUTINE
; ----------------------------------------------
;

;; TEST-MNAM
L0685:  CALL    L0665           ; routine CHECK-M
        LD      A,($5CDB)       ; sv D_STR1
        AND     A
        RET     Z

        RST     20H             ; sh_err
        DEFB    $06


; ---------------------------------
; THE 'CHECK STATION NUMBER' ROUTINE
; ---------------------------------
;

;; TEST-STAT
L068F:  LD      DE,($5CD6)      ; sv D_STR1
        INC     DE
        LD      A,E
        OR      D
        JR      Z,L06A1         ; forward to NREPORT-8

        DEC     DE
        LD      HL,L0040
        SBC     HL,DE
        RET     NC


;; NREPORT-6
L069F:  RST     20H             ; sh_err
        DEFB    $05
```

```
;; NREPORT-8
L06A1:  RST     20H             ; sh_err
        DEFB    $07

; ----------------------------------
; THE 'EVALUATE "X";N;"NAME"' ROUTINE
; ----------------------------------
;

;; EXOT-EXPR
L06A3:  CALL    L05F2           ; routine EXPT-SPEC
        CALL    L05B1           ; routine SEPARATOR
        JP      NZ,L04B2        ; jump to OREPORT-1
        CALL    L062F           ; routine EXPT-NAME
        RET


; ----------------------------
; THE 'CHECK BAUD RATE' ROUTINE
; ----------------------------
;

;; TEST-BAUD
L06B0:  LD      HL,($5CD6)      ; sv D_STR1
        INC     HL
        LD      A,L
        OR      H
        RET     NZ

        RST     20H             ; sh_err
        DEFB    $09

; ----------------------------------------
; THE 'EVALUATE STREAM OR EXPRESSION' ROUTINE
; ----------------------------------------
;

;; EXPT-EXP1
L06B9:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CP      $23
        JP      Z,L064E         ; jump to EXPT-STRM
        CALL    L05F5           ; routine EXP-SPEC2
        CALL    L05B1           ; routine SEPARATOR
        JR      NZ,L06CC        ; forward to ENDHERE

        CALL    L062F           ; routine EXPT-NAME

;; ENDHERE
L06CC:  RST     18H
        RET     Z

        LD      A,($5CD9)       ; sv D_STR1
        CP      $54
        RET     Z

        CP      $42
        RET     Z

        CP      $4E
        JP      Z,L068F         ; jump to TEST-STAT
        JP      L0685           ; jump to TEST-MNAM
        DEFB    $FF
        DEFB    $FF
```

```
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF

; --------------------
; THE 'UNPAGE' ROUTINE
; --------------------
;

;; UNPAGE
L0700:  RET


; ---------------------------------
; THE 'EVALUATE PARAMETERS' ROUTINE
; ---------------------------------
;

;; EXPT-PRMS
L0701:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CP      $2A
        JR      NZ,L073C        ; forward to OREP-1-2

        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CALL    L05F5           ; routine EXP-SPEC2
        CALL    L05B1           ; routine SEPARATOR
        JR      NZ,L0716        ; forward to NO-NAME

        CALL    L062F           ; routine EXPT-NAME

;; NO-NAME
L0716:  PUSH    AF
        LD      A,($5CD9)       ; sv D_STR1
        CP      $4E
```

```
        JR      NZ,L0722        ; forward to NOT-NET

        SET     3,(IY+$7C)      ; sv FLAGS_3

;; NOT-NET
L0722:  POP     AF
        CP      $0D
        JR      Z,L0750         ; forward to END-EXPT

        CP      $3A
        JR      Z,L0750         ; forward to END-EXPT

        CP      $AA
        JR      Z,L0771         ; forward to SCREEN$

        CP      $AF
        JR      Z,L0789         ; forward to CODE

        CP      $CA
        JR      Z,L073E         ; forward to LINE

        CP      $E4
        JP      Z,L07D2         ; jump forward to DATA

;; OREP-1-2
L073C:  RST     20H             ; sh_err
        DEFB    $00

;; LINE
L073E:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        RST     10H             ; CALBAS
        DEFW    $1C82           ; main EXPT-1NUM
        CALL    L05B7           ; routine ST-END
        RST     10H             ; CALBAS
        DEFW    $1E99           ; main FIND-INT2
        LD      ($5CED),BC      ; sv HD_11
        JR      L0753           ; forward to PROG


;; END-EXPT
L0750:  CALL    L05B7           ; routine ST-END

;; PROG
L0753:  XOR     A
        LD      ($5CE6),A       ; sv HD_00
        LD      HL,($5C59)      ; sv E_LINE
        LD      DE,($5C53)      ; sv PROG
        LD      ($5CE9),DE      ; sv HD_0D
        SCF
        SBC     HL,DE
        LD      ($5CE7),HL      ; sv HD_0B
        LD      HL,($5C4B)      ; sv VARS
        SBC     HL,DE
        LD      ($5CEB),HL      ; sv HD_0F
        RET


;; SCREEN$
L0771:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CALL    L05B7           ; routine ST-END
        LD      HL,$1B00
        LD      ($5CE7),HL      ; sv HD_0B
```

```
        LD      HL,$4000
        LD      ($5CE9),HL      ; sv HD_0D
        LD      A,$03
        LD      ($5CE6),A       ; sv HD_00
        RET


;; CODE
L0789:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CP      $0D
        JR      Z,L079A         ; forward to DEFLT-0

        CP      $3A
        JR      NZ,L079F        ; forward to PAR-1

        BIT     5,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L073C        ; back to OREP-1-2


;; DEFLT-0
L079A:  RST     10H             ; CALBAS
        DEFW    $1CE6           ; main USE-ZERO
        JR      L07A7           ; forward to TEST-SAVE


;; PAR-1
L079F:  RST     10H             ; CALBAS
        DEFW    $1C82           ; main EXPT-1NUM
        CALL    L05B1           ; routine SEPARATOR
        JR      Z,L07B2         ; forward to PAR-2


;; TEST-SAVE
L07A7:  BIT     5,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L073C        ; back to OREP-1-2

        RST     10H             ; CALBAS
        DEFW    $1CE6           ; main USE-ZERO
        JR      L07B8           ; forward to END-CODE


;; PAR-2
L07B2:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        RST     10H             ; CALBAS
        DEFW    $1C82           ; main EXPT-1NUM

;; END-CODE
L07B8:  RST     10H             ; CALBAS
        DEFW    $0018           ; main GET-CHAR
        CALL    L05B7           ; routine ST-END
        RST     10H             ; CALBAS
        DEFW    $1E99           ; main FIND-INT2
        LD      ($5CE7),BC      ; sv HD_0B
        RST     10H             ; CALBAS
        DEFW    $1E99           ; main FIND-INT2
        LD      ($5CE9),BC      ; sv HD_0D
        LD      A,$03
        LD      ($5CE6),A       ; sv HD_00
        RET


;; DATA
```

```
L07D2:  BIT     6,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L07DA         ; forward to NO-M-ARR

        RST     20H             ; sh_err
        DEFB    $14

;; NO-M-ARR
L07DA:  RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        RST     10H             ; CALBAS
        DEFW    $28B2           ; main LOOK-VARS
        SET     7,C
        JR      NC,L07F2        ; forward to EXISTING

        LD      HL,L0000
        BIT     4,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L080E        ; forward to LD-DATA

        LD      (IY+$00),$01    ; sv ERR_NR
        RST     28H             ; romerr

;; EXISTING
L07F2:  JR      Z,L07F6         ; forward to G-TYPE


;; NONS-BSC
L07F4:  RST     20H             ; sh_err
        DEFB    $00

;; G-TYPE
L07F6:  RST     18H
        JR      Z,L081C         ; forward to END-DATA

        BIT     5,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0803         ; forward to VR-DATA

        BIT     7,(HL)
        JR      Z,L07F4         ; back to NONS-BSC


;; VR-DATA
L0803:  INC     HL
        LD      A,(HL)
        LD      ($5CE7),A       ; sv HD_0B
        INC     HL
        LD      A,(HL)
        LD      ($5CE8),A       ; sv HD_0B_hi
        INC     HL

;; LD-DATA
L080E:  LD      A,C
        LD      ($5CEB),A       ; sv HD_0F
        LD      A,$01
        BIT     6,C
        JR      Z,L0819         ; forward to NUM-ARR

        INC     A

;; NUM-ARR
L0819:  LD      ($5CE6),A       ; sv HD_00

;; END-DATA
L081C:  EX      DE,HL
        RST     10H             ; CALBAS
```

```
        DEFW    $0020           ; main NEXT-CHAR
        CP      $29
        JR      NZ,L07F4        ; back to NONS-BSC

        RST     10H             ; CALBAS
        DEFW    $0020           ; main NEXT-CHAR
        CALL    L05B7           ; routine ST-END
        LD      ($5CE9),DE      ; sv HD_0D
        RET


; ---------------------------------
; THE 'SAVE COMMAND SYNTAX' ROUTINE
; ---------------------------------
;

;; SAVE-SYN
L082F:  SET     5,(IY+$7C)      ; sv FLAGS_3
        CALL    L0701           ; routine EXPT-PRMS
        LD      A,($5CD9)       ; sv D_STR1
        CP      $42
        JR      Z,L084F         ; forward to SA-HEADER

        CP      $4E
        JR      NZ,L0849        ; forward to SAVE-M

        CALL    L068F           ; routine TEST-STAT
        CALL    L0EA9           ; routine OP-TEMP-N
        JR      L084F           ; forward to SA-HEADER


;; SAVE-M
L0849:  CALL    L0685           ; routine TEST-MNAM
        JP      L1E7F           ; jump forward to SAVE-RUN

;; SA-HEADER
L084F:  LD      B,$09
        LD      HL,$5CE6        ; sv HD_00

;; HD-LOOP
L0854:  CALL    L0880           ; routine SA-BYTE
        INC     HL
        DJNZ    L0854           ; back to HD-LOOP

        LD      HL,($5CE9)      ; sv HD_0D
        BIT     3,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L086E         ; forward to SA-BLOCK

        LD      A,($5CE6)       ; sv HD_00
        CP      $03
        JR      NC,L086E        ; forward to SA-BLOCK

        LD      DE,$0114
        ADD     HL,DE

;; SA-BLOCK
L086E:  LD      BC,($5CE7)      ; sv HD_0B

;; SA-BLK-LP
L0872:  LD      A,C
        OR      B
        JR      Z,L087D         ; forward to S-BLK-END

        CALL    L0880           ; routine SA-BYTE
```

```
        DEC     BC
        INC     HL
        JR      L0872           ; back to SA-BLK-LP


;; S-BLK-END
L087D:  JP      L0988           ; jump forward to TST-MR-M

; -------------------------------------------------
; THE 'SAVE A BYTE TO NETWORK OR RS232 LINK' ROUTINE
; -------------------------------------------------
;

;; SA-BYTE
L0880:  PUSH    HL
        PUSH    BC
        BIT     3,(IY+$7C)      ; sv FLAGS_3
        LD      A,(HL)
        JR      NZ,L088E        ; forward to SA-NET

        CALL    L0C5A           ; routine BCHAN-OUT
        JR      L0891           ; forward to SA-B-END


;; SA-NET
L088E:  CALL    L0D6C           ; routine NCHAN-OUT

;; SA-B-END
L0891:  POP     BC
        POP     HL
        RET


; ---------------------------------
; THE 'LOAD COMMAND SYNTAX' ROUTINE
; ---------------------------------
;

;; LOAD-SYN
L0894:  SET     4,(IY+$7C)      ; sv FLAGS_3
        CALL    L0701           ; routine EXPT-PRMS
        JP      L08AF           ; jump forward to LD-VF-MR

; ----------------------------------
; THE 'VERIFY COMMAND SYNTAX' ROUTINE
; ----------------------------------
;

;; VERIF-SYN
L089E:  SET     7,(IY+$7C)      ; sv FLAGS_3
        CALL    L0701           ; routine EXPT-PRMS
        JP      L08AF           ; jump forward to LD-VF-MR

; ----------------------------------
; THE 'MERGE COMMAND SYNTAX' ROUTINE
; ----------------------------------
;

;; MRG-SYN
L08A8:  SET     6,(IY+$7C)      ; sv FLAGS_3
        CALL    L0701           ; routine EXPT-PRMS

; ---------------------------------------
; THE 'LOAD-VERIFY-MERGE COMMANDS' ROUTINE
```

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

;; LD-VF-MR
L08AF:  LD      HL,$5CE6        ; sv HD_00
        LD      DE,$5CDE        ; sv D_STR2
        LD      BC,$0007
        LDIR
        LD      A,($5CD9)       ; sv D_STR1
        CP      $4E
        JR      Z,L08CD         ; forward to TS-L-NET

        CP      $42
        JR      Z,L08D3         ; forward to TS-L-RS

        CALL    L0685           ; routine TEST-MNAM
        CALL    L1580           ; routine F-M-HM
        JR      L08F2           ; forward to TEST-TYPE


;; TS-L-NET
L08CD:  CALL    L068F           ; routine TEST-STAT
        CALL    L0EA9           ; routine OP-TEMP-N

;; TS-L-RS
L08D3:  LD      HL,$5CE6        ; sv HD_00
        LD      B,$09

;; LD-HEADER
L08D8:  PUSH    HL
        PUSH    BC
        BIT     3,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L08E7         ; forward to LD-HD-RS


;; LD-HD-NET
L08E0:  CALL    L0D12           ; routine NCHAN-IN
        JR      NC,L08E0        ; back to LD-HD-NET

        JR      L08EC           ; forward to LD-HDR-2


;; LD-HD-RS
L08E7:  CALL    L0B81           ; routine BCHAN-IN
        JR      NC,L08E7        ; back to LD-HD-RS


;; LD-HDR-2
L08EC:  POP     BC
        POP     HL
        LD      (HL),A
        INC     HL
        DJNZ    L08D8           ; back to LD-HEADER


;; TEST-TYPE
L08F2:  LD      A,($5CDE)       ; sv D_STR2
        LD      B,A
        LD      A,($5CE6)       ; sv HD_00
        CP      B
        JR      NZ,L0902        ; forward to NREPORT-N

        CP      $03
        JR      Z,L0911         ; forward to T-H-CODE
```

```
        JR      C,L0904         ; forward to TST-MERGE


;; NREPORT-N
L0902:  RST     20H             ; sh_err
        DEFB    $16

;; TST-MERGE
L0904:  BIT     6,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L0967        ; forward to MERGE-BLK

        BIT     7,(IY+$7C)      ; sv FLAGS_3
        JP      Z,L09A3         ; jump to LD-PR-AR


;; T-H-CODE
L0911:  BIT     6,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0919         ; forward to LD-BLOCK

        RST     20H             ; sh_err
        DEFB    $14

;; LD-BLOCK
L0919:  LD      HL,($5CDF)      ; sv D_STR2
        LD      DE,($5CE7)      ; sv HD_0B
        LD      A,H
        OR      L
        JR      Z,L0932         ; forward to LD-BLK-2

        SBC     HL,DE
        JR      NC,L0932        ; forward to LD-BLK-2

        BIT     4,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0930         ; forward to NREPORT-L

        RST     20H             ; sh_err
        DEFB    $13

;; NREPORT-L
L0930:  RST     20H             ; sh_err
        DEFB    $15

;; LD-BLK-2
L0932:  LD      HL,($5CE1)      ; sv D_STR2
        LD      A,(IX+$04)
        CP      $CD
        JR      NZ,L0941        ; forward to LD-BLK-3

        LD      HL,($5CE4)      ; sv D_STR2
        JR      L0952           ; forward to LD-BLK-4


;; LD-BLK-3
L0941:  BIT     3,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0952         ; forward to LD-BLK-4

        LD      A,($5CE6)       ; sv HD_00
        CP      $03
        JR      Z,L0952         ; forward to LD-BLK-4

        LD      BC,$0114
        ADD     HL,BC
```

```
;; LD-BLK-4
L0952:  LD      A,H
        OR      L
        JR      NZ,L0959        ; forward to LD-BLK-5

        LD      HL,($5CE9)      ; sv HD_0D

;; LD-BLK-5
L0959:  LD      A,($5CE6)       ; sv HD_00
        AND     A
        JR      NZ,L0962        ; forward to LD-NO-PGM

        LD      HL,($5C53)      ; sv PROG

;; LD-NO-PGM
L0962:  CALL    L0A5C           ; routine LV-ANY
        JR      L0988           ; forward to TST-MR-M


;; MERGE-BLK
L0967:  LD      A,($5CEE)       ; sv HD_11_hi
        AND     $C0
        JR      NZ,L0973        ; forward to NO-AUTOST

        CALL    L17B9           ; routine RCL-T-CH
        RST     20H             ; sh_err
        DEFB    $14

;; NO-AUTOST
L0973:  LD      BC,($5CE7)      ; sv HD_0B
        PUSH    BC
        INC     BC
        RST     10H             ; CALBAS
        DEFW    $0030           ; main BC-SPACES
        LD      (HL),$80
        EX      DE,HL
        POP     DE
        PUSH    HL
        CALL    L0A5C           ; routine LV-ANY
        POP     HL
        RST     10H             ; CALBAS
        DEFW    $08CE           ; main ME-CTRLX

;; TST-MR-M
L0988:  LD      A,(IX+$04)
        CP      $CD
        JR      NZ,L0994        ; forward to TST-MR-N

        CALL    L12A9           ; routine CLOSE-M2
        JR      L09A0           ; forward to MERGE-END


;; TST-MR-N
L0994:  BIT     3,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L09A0         ; forward to MERGE-END

        CALL    L0EF5           ; routine SEND-NEOF
        CALL    L17B9           ; routine RCL-T-CH

;; MERGE-END
L09A0:  JP      L05C1           ; jump to END1

;; LD-PR-AR
L09A3:  LD      DE,($5CE7)      ; sv HD_0B
```

```
        LD      HL,($5CE1)      ; sv D_STR2
        PUSH    HL
        LD      A,H
        OR      L
        JR      NZ,L09B5        ; forward to LD-PROG

        INC     DE
        INC     DE
        INC     DE
        EX      DE,HL
        JR      L09BE           ; forward to TST-SPACE


;; LD-PROG
L09B5:  LD      HL,($5CDF)      ; sv D_STR2
        EX      DE,HL
        SCF
        SBC     HL,DE
        JR      C,L09C7         ; forward to TST-TYPE


;; TST-SPACE
L09BE:  LD      DE,$0005
        ADD     HL,DE
        LD      B,H
        LD      C,L
        RST     10H             ; CALBAS
        DEFW    $1F05           ; main TEST-ROOM

;; TST-TYPE
L09C7:  POP     HL
        LD      A,($5CE6)       ; sv HD_00
        AND     A
        JR      Z,L0A15         ; forward to SET-PROG

        LD      A,H
        OR      L
        JR      Z,L09F3         ; forward to CRT-NEW

        LD      A,(IX+$04)
        CP      $CD
        JR      NZ,L09DE        ; forward to T-LD-NET

        LD      HL,($5CE4)      ; sv D_STR2
        JR      L09E8           ; forward to RCLM-OLD


;; T-LD-NET
L09DE:  BIT     3,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L09E8         ; forward to RCLM-OLD

        LD      DE,$0114
        ADD     HL,DE

;; RCLM-OLD
L09E8:  DEC     HL
        LD      B,(HL)
        DEC     HL
        LD      C,(HL)
        DEC     HL
        INC     BC
        INC     BC
        INC     BC
        RST     10H             ; CALBAS
```

```
        DEFW    $19E8           ; main RECLAIM-2

;; CRT-NEW
L09F3:  LD      HL,($5C59)      ; sv E_LINE
        DEC     HL
        LD      BC,($5CE7)      ; sv HD_0B
        PUSH    BC
        INC     BC
        INC     BC
        INC     BC
        LD      A,($5CE3)       ; sv D_STR2
        PUSH    AF
        RST     10H             ; CALBAS
        DEFW    $1655           ; main MAKE-ROOM
        INC     HL
        POP     AF
        LD      (HL),A
        POP     DE
        INC     HL
        LD      (HL),E
        INC     HL
        LD      (HL),D
        INC     HL

;; END-LD-PR
L0A0F:  CALL    L0A5C           ; routine LV-ANY
        JP      L0988           ; jump to TST-MR-M

;; SET-PROG
L0A15:  RES     1,(IY+$7C)      ; sv FLAGS_3
        LD      DE,($5C53)      ; sv PROG
        LD      HL,($5C59)      ; sv E_LINE
        DEC     HL
        RST     10H             ; CALBAS
        DEFW    $19E5           ; main RECLAIM-1
        LD      BC,($5CE7)      ; sv HD_0B
        LD      HL,($5C53)      ; sv PROG
        RST     10H             ; CALBAS
        DEFW    $1655           ; main MAKE-ROOM
        INC     HL
        LD      BC,($5CEB)      ; sv HD_0F
        ADD     HL,BC
        LD      ($5C4B),HL      ; sv VARS
        LD      A,($5CEE)       ; sv HD_11_hi
        LD      H,A
        AND     $C0
        JR      NZ,L0A4E        ; forward to NO-AUTO

        SET     1,(IY+$7C)      ; sv FLAGS_3
        LD      A,($5CED)       ; sv HD_11
        LD      L,A
        LD      ($5C42),HL      ; sv NEWPPC
        LD      (IY+$0A),$00    ; sv NSPPC

;; NO-AUTO
L0A4E:  LD      HL,($5C53)      ; sv PROG
        LD      DE,($5CE7)      ; sv HD_0B
        DEC     HL
        LD      ($5C57),HL      ; sv DATADD
        INC     HL
        JR      L0A0F           ; back to END-LD-PR


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

```
; THE 'LOAD OR VERIFY' ROUTINE
; ---------------------------
;

;; LV-ANY
L0A5C:  LD      A,D
        OR      E
        RET     Z

        LD      A,(IX+$04)
        CP      $CD
        JR      NZ,L0A6A        ; forward to LV-BN

        CALL    L15A9           ; routine LV-MCH
        RET


;; LV-BN
L0A6A:  PUSH    HL
        PUSH    DE
        BIT     3,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0A79         ; forward to LV-B


;; LV-N
L0A72:  CALL    L0D12           ; routine NCHAN-IN
        JR      NC,L0A72        ; back to LV-N

        JR      L0A7E           ; forward to LV-BN-E


;; LV-B
L0A79:  CALL    L0B81           ; routine BCHAN-IN
        JR      NC,L0A79        ; back to LV-B


;; LV-BN-E
L0A7E:  POP     DE
        DEC     DE
        POP     HL
        BIT     7,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L0A8A        ; forward to VR-BN

        LD      (HL),A
        JR      L0A8F           ; forward to LVBN-END


;; VR-BN
L0A8A:  CP      (HL)
        JR      Z,L0A8F         ; forward to LVBN-END

        RST     20H             ; sh_err
        DEFB    $15

;; LVBN-END
L0A8F:  INC     HL
        LD      A,E
        OR      D
        JR      NZ,L0A6A        ; back to LV-BN

        RET


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

```
; THE 'LOAD "RUN" PROGRAM' ROUTINE
; -------------------------------
;

;; LOAD-RUN
L0A95:  LD      BC,$0001
        LD      ($5CD6),BC      ; sv D_STR1
        LD      BC,$0003
        LD      ($5CDA),BC      ; sv D_STR1
        LD      BC,$0AC6
        LD      ($5CDC),BC      ; sv D_STR1
        SET     4,(IY+$7C)      ; sv FLAGS_3
        CALL    L0753           ; routine PROG
        LD      HL,$5CE6        ; sv HD_00
        LD      DE,$5CDE        ; sv D_STR2
        LD      BC,$0009
        LDIR
        SET     7,(IY+$0A)      ; sv NSPPC
        CALL    L1580           ; routine F-M-HM
        JP      L08F2           ; jump to TEST-TYPE
        LD      (HL),D
        LD      (HL),L
        LD      L,(HL)

; ---------------------------------------
; THE 'SET "BAUD"- SYSTEM VARIABLE' ROUTINE
; ---------------------------------------
;

;; SET-BAUD
L0AC9:  LD      BC,($5CD6)      ; sv D_STR1
        LD      HL,$0AEF

;; NXT-ENTRY
L0AD0:  LD      E,(HL)
        INC     HL
        LD      D,(HL)
        INC     HL
        EX      DE,HL
        LD      A,H
        CP      $4B
        JR      NC,L0AE4        ; forward to END-SET

        AND     A
        SBC     HL,BC
        JR      NC,L0AE4        ; forward to END-SET

        EX      DE,HL
        INC     HL
        INC     HL
        JR      L0AD0           ; back to NXT-ENTRY


;; END-SET
L0AE4:  EX      DE,HL
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        LD      ($5CC3),DE      ; sv BAUD
        JP      L05C1           ; jump to END1


; ----------------------------------
; THE 'RS232 TIMING CONSTANTS' ROUTINE
```

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

;;
L0AEF:  DEFW    $0032           ;
        DEFW    $0A82           ;
        DEFW    $006E           ;
        DEFW    $04C5           ;
        DEFW    $012C           ;
        DEFW    $01BE           ;
        DEFW    $0258           ;
        DEFW    $00DE           ;
        DEFW    $04B0           ;
        DEFW    $006E           ;
        DEFW    $0960           ;
        DEFW    $0036           ;
        DEFW    $12C0           ;
        DEFW    $001A           ;
        DEFW    $2580           ;
        DEFW    $000C           ;
        DEFW    $4B00           ;
        DEFW    $0005           ;


; ----------------------------------------------
; THE 'OPEN RS232 CHANNEL IN CHANS AREA' ROUTINE
; ----------------------------------------------
;

;; OP-RS-CH
L0B13:  LD      HL,($5C53)      ; sv PROG
        DEC     HL
        LD      BC,$000B
        PUSH    BC
        RST     10H             ; CALBAS
        DEFW    $1655           ; main MAKE-ROOM
        POP     BC
        PUSH    DE
        CALL    L1691           ; routine REST-N-AD
        POP     DE
        LD      HL,$0B6E
        LD      BC,$000B
        LDDR
        INC     DE
        LD      A,($5CD9)       ; sv D_STR1
        CP      $42
        RET     NZ

        PUSH    DE
        LD      HL,$0005
        ADD     HL,DE
        LD      DE,L0C5A
        LD      (HL),E
        INC     HL
        LD      (HL),D
        INC     HL
        LD      DE,$0B75
        LD      (HL),E
        INC     HL
        LD      (HL),D
        POP     DE
        RET
```

```
; --------------------------------------
; THE 'ATTACH CHANNEL TO A STREAM' ROUTINE
; --------------------------------------
;

;; OP-RSCHAN
L0B47:  CALL    L0B13           ; routine OP-RS-CH

;; OP-STREAM
L0B4A:  LD      HL,($5C4F)      ; sv CHANS
        DEC     HL
        EX      DE,HL
        AND     A
        SBC     HL,DE
        EX      DE,HL
        LD      HL,$5C16        ; sv STRMS_00
        LD      A,($5CD8)       ; sv D_STR1
        RLCA
        LD      C,A
        LD      B,$00
        ADD     HL,BC
        LD      (HL),E
        INC     HL
        LD      (HL),D
        JP      L05C1           ; jump to END1


; ------------------------------
; THE '"T" CHANNEL DATA' ROUTINE
; ------------------------------
;

;;
L0B64:  DEFW    $0008             ; main ERROR-1
        DEFW    $0008             ; main ERROR-1
        DEFB    $54
        DEFW    $0C3C             ;
        DEFW    $0B6F             ;
        DEFW    $000B             ;


; ------------------------------
; THE '"T" CHANNEL INPUT' ROUTINE
; ------------------------------
;

;; T-INPUT
L0B6F:  LD      HL,$0B7B
        JP      L0CBD           ; jump to CALL-INP


; ------------------------------
; THE '"B" CHANNEL INPUT' ROUTINE
; ------------------------------
;

;; B-INPUT
L0B75:  LD      HL,L0B81
        JP      L0CBD           ; jump to CALL-INP


; --------------------------------------
; THE '"T" CHANNEL INPUT SERVICE' ROUTINE
; --------------------------------------
```

```
;

;; TCHAN-IN
L0B7B:  CALL    L0B81           ; routine BCHAN-IN
        RES     7,A
        RET


; --------------------------------------
; THE '"B" CHANNEL INPUT SERVICE' ROUTINE
; --------------------------------------
;

;; BCHAN-IN
L0B81:  LD      HL,$5CC7        ; sv SER_FL
        LD      A,(HL)
        AND     A
        JR      Z,L0B8E         ; forward to REC-BYTE

        LD      (HL),$00
        INC     HL
        LD      A,(HL)
        SCF
        RET


;; REC-BYTE
L0B8E:  LD      A,$7F
        IN      A,($FE)
        RRCA
        JR      C,L0B9A         ; forward to REC-PROC

        LD      (IY+$00),$14    ; sv ERR_NR
        RST     28H             ; romerr

;; REC-PROC
L0B9A:  DI
        LD      A,($5CC6)       ; sv IOBORD
        OUT     ($FE),A
        LD      DE,($5CC3)      ; sv BAUD
        LD      HL,$0320
        LD      B,D
        LD      C,E
        SRL     B
        RR      C
        LD      A,$FE
        OUT     ($EF),A

;; READ-RS
L0BB1:  IN      A,($F7)
        RLCA
        JR      NC,L0BC5        ; forward to TST-AGAIN

        IN      A,($F7)
        RLCA
        JR      NC,L0BC5        ; forward to TST-AGAIN

        IN      A,($F7)
        RLCA
        JR      NC,L0BC5        ; forward to TST-AGAIN

        IN      A,($F7)
        RLCA
        JR      C,L0BD1         ; forward to START-BIT
```

```
;; TST-AGAIN
L0BC5:  DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L0BB1        ; back to READ-RS

        PUSH    AF
        LD      A,$EE
        OUT     ($EF),A
        JR      L0BF0           ; forward to WAIT-1


;; START-BIT
L0BD1:  LD      H,B
        LD      L,C
        LD      B,$80
        DEC     HL
        DEC     HL
        DEC     HL

;; SERIAL-IN
L0BD8:  ADD     HL,DE
        NOP

;; BD-DELAY
L0BDA:  DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L0BDA        ; back to BD-DELAY

        ADD     A,$00
        IN      A,($F7)
        RLCA
        RR      B
        JR      NC,L0BD8        ; back to SERIAL-IN

        LD      A,$EE
        OUT     ($EF),A
        LD      A,B
        CPL
        SCF
        PUSH    AF

;; WAIT-1
L0BF0:  ADD     HL,DE

;; WAIT-2
L0BF1:  DEC     HL
        LD      A,L
        OR      H
        JR      NZ,L0BF1        ; back to WAIT-2

        ADD     HL,DE
        ADD     HL,DE
        ADD     HL,DE

;; T-FURTHER
L0BF9:  DEC     HL
        LD      A,L
        OR      H
        JR      Z,L0C36         ; forward to END-RS-IN
```

```
        IN      A,($F7)
        RLCA
        JR      NC,L0BF9        ; back to T-FURTHER

        IN      A,($F7)
        RLCA
        JR      NC,L0BF9        ; back to T-FURTHER

        IN      A,($F7)
        RLCA
        JR      NC,L0BF9        ; back to T-FURTHER

        IN      A,($F7)
        RLCA
        JR      NC,L0BF9        ; back to T-FURTHER

        LD      H,D
        LD      L,E
        SRL     H
        RR      L
        LD      B,$80
        DEC     HL
        DEC     HL
        DEC     HL

;; SER-IN-2
L0C1D:  ADD     HL,DE
        NOP

;; BD-DELAY2
L0C1F:  DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L0C1F        ; back to BD-DELAY2

        ADD     A,$00
        IN      A,($F7)
        RLCA
        RR      B
        JR      NC,L0C1D        ; back to SER-IN-2

        LD      HL,$5CC7        ; sv SER_FL
        LD      (HL),$01
        INC     HL
        LD      A,B
        CPL
        LD      (HL),A

;; END-RS-IN
L0C36:  CALL    L0CA9           ; routine BORD-REST
        POP     AF
        EI
        RET


; --------------------------------
; THE '"T" CHANNEL OUTPUT' ROUTINE
; --------------------------------
;

;; TCHAN-OUT
L0C3C:  CP      $A5
        JR      C,L0C46         ; forward to NOT-TOKEN
```

```
        SUB     $A5
        RST     10H             ; CALBAS
        DEFW    $0C10           ; main PO-TOKENS
        RET


;; NOT-TOKEN
L0C46:  CP      $7F
        JR      C,L0C4C         ; forward to NOT-GRAPH

        LD      A,$3F

;; NOT-GRAPH
L0C4C:  CP      $0D
        JR      NZ,L0C57        ; forward to NOT-CR

        CALL    L0C5A           ; routine BCHAN-OUT
        LD      A,$0A
        JR      L0C5A           ; forward to BCHAN-OUT


;; NOT-CR
L0C57:  CP      $20
        RET     C


; -------------------------------
; THE '"B" CHANNEL OUTPUT' ROUTINE
; -------------------------------
;

;; BCHAN-OUT
L0C5A:  LD      B,$0B
        CPL
        LD      C,A
        LD      A,($5CC6)       ; sv IOBORD
        OUT     ($FE),A
        LD      A,$EF
        OUT     ($EF),A
        CPL
        OUT     ($F7),A
        LD      HL,($5CC3)      ; sv BAUD
        LD      D,H
        LD      E,L

;; BD-DEL-1
L0C6F:  DEC     DE
        LD      A,D
        OR      E
        JR      NZ,L0C6F        ; back to BD-DEL-1


;; TEST-DTR
L0C74:  LD      A,$7F
        IN      A,($FE)
        OR      $FE
        IN      A,($FE)
        RRA
        JP      NC,L0CB4        ; jump to BRK-INOUT
        IN      A,($EF)
        AND     $08
        JR      Z,L0C74         ; back to TEST-DTR

        SCF
```

```
        DI

;; SER-OUT-L
L0C88:  ADC     A,$00
        OUT     ($F7),A
        LD      D,H
        LD      E,L

;; BD-DEL-2
L0C8E:  DEC     DE
        LD      A,D
        OR      E
        JR      NZ,L0C8E        ; back to BD-DEL-2

        DEC     DE
        XOR     A
        SRL     C
        DJNZ    L0C88           ; back to SER-OUT-L

        EI
        LD      A,$01
        LD      C,$EF
        LD      B,$EE
        OUT     ($F7),A
        OUT     (C),B

;; BD-DEL-3
L0CA4:  DEC     HL
        LD      A,L
        OR      H
        JR      NZ,L0CA4        ; back to BD-DEL-3


; -----------------------------------
; THE 'BORDER COLOUR RESTORE' ROUTINE
; -----------------------------------
;

;; BORD-REST
L0CA9:  LD      A,($5C48)       ; sv BORDCR
        AND     $38
        RRCA
        RRCA
        RRCA
        OUT     ($FE),A
        RET


; --------------------------------------
; THE 'BREAK INTO I/O OPERATION' ROUTINE
; --------------------------------------
;

;; BRK-INOUT
L0CB4:  EI
        CALL    L0CA9           ; routine BORD-REST
        LD      (IY+$00),$14    ; sv ERR_NR
        RST     28H             ; romerr

; ----------------------
; THE 'CALL-INP' ROUTINE
; ----------------------
;
```

```
;; CALL-INP
L0CBD:  RES     3,(IY+$02)      ; sv TV_FLAG
        PUSH    HL
        LD      HL,($5C3D)      ; sv ERR_SP
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        AND     A
        LD      HL,$107F
        SBC     HL,DE
        JR      NZ,L0CFB        ; forward to INKEY$

        POP     HL
        LD      SP,($5C3D)      ; sv ERR_SP
        POP     DE
        POP     DE
        LD      ($5C3D),DE      ; sv ERR_SP

;; IN-AGAIN
L0CDB:  PUSH    HL
        LD      DE,$0CE1
        PUSH    DE
        JP      (HL)
        JR      C,L0CED         ; forward to ACC-CODE

        JR      Z,L0CEA         ; forward to NO-READ


;; OREPORT-8
L0CE5:  LD      (IY+$00),$07    ; sv ERR_NR
        RST     28H             ; romerr

;; NO-READ
L0CEA:  POP     HL
        JR      L0CDB           ; back to IN-AGAIN


;; ACC-CODE
L0CED:  CP      $0D
        JR      Z,L0CF7         ; forward to END-INPUT

        RST     10H             ; CALBAS
        DEFW    $0F85           ; main ADD-CHRX
        POP     HL
        JR      L0CDB           ; back to IN-AGAIN


;; END-INPUT
L0CF7:  POP     HL
        JP      L0700           ; jump to UNPAGE

;; INKEY$
L0CFB:  POP     HL
        LD      DE,$0D01
        PUSH    DE
        JP      (HL)
        RET     C

        RET     Z

        BIT     4,(IY+$7C)      ; sv FLAGS_3
        JR      Z,L0CE5         ; back to OREPORT-8

        OR      $01
```

```
        RET


; ------------------------------
; THE '"N" CHANNEL INPUT' ROUTINE
; ------------------------------
;

;; N-INPUT
L0D0C:  LD      HL,L0D12
        JP      L0CBD           ; jump to CALL-INP

; --------------------------------------
; THE '"N" CHANNEL INPUT SERVICE' ROUTINE
; --------------------------------------
;

;; NCHAN-IN
L0D12:  LD      IX,($5C51)      ; sv CURCHL
        LD      A,(IX+$10)
        AND     A
        JR      Z,L0D1E         ; forward to TEST-BUFF

        RST     20H             ; sh_err
        DEFB    $0D

;; TEST-BUFF
L0D1E:  LD      A,(IX+$14)
        AND     A
        JR      Z,L0D38         ; forward to TST-N-EOF

        LD      E,(IX+$13)
        DEC     A
        SUB     E
        JR      C,L0D38         ; forward to TST-N-EOF

        LD      D,$00
        INC     E
        LD      (IX+$13),E
        ADD     IX,DE
        LD      A,(IX+$14)
        SCF
        RET


;; TST-N-EOF
L0D38:  LD      A,(IX+$0F)
        AND     A
        JR      Z,L0D3F         ; forward to GET-N-BUF

        RET


;; GET-N-BUF
L0D3F:  LD      A,($5CC6)       ; sv IOBORD
        OUT     ($FE),A
        DI

;; TRY-AGAIN
L0D45:  CALL    L0F1E           ; routine WT-SCOUT
        JR      NC,L0D5F        ; forward to TIME-OUT

        CALL    L0E18           ; routine GET-NBLK
        JR      NZ,L0D5F        ; forward to TIME-OUT
```

```
        EI
        CALL    L0CA9           ; routine BORD-REST
        LD      (IX+$13),$00
        LD      A,($5CD2)       ; sv NTTYPE
        LD      (IX+$0F),A
        JR      L0D1E           ; back to TEST-BUFF


;; TIME-OUT
L0D5F:  LD      A,(IX+$0B)
        AND     A
        JR      Z,L0D45         ; back to TRY-AGAIN

        EI
        CALL    L0CA9           ; routine BORD-REST
        AND     $00
        RET


; -------------------------------
; THE '"N" CHANNEL OUTPUT' ROUTINE
; -------------------------------
;

;; NCHAN-OUT
L0D6C:  LD      IX,($5C51)      ; sv CURCHL
        LD      B,A
        LD      A,(IX+$14)
        AND     A
        LD      A,B
        JR      Z,L0D7A         ; forward to TEST-OUT

        RST     20H             ; sh_err
        DEFB    $0C

;; TEST-OUT
L0D7A:  LD      E,(IX+$10)
        INC     E
        JR      NZ,L0D88        ; forward to ST-BF-LEN

        PUSH    AF
        XOR     A
        CALL    L0DAB           ; routine S-PACK-1
        POP     AF
        LD      E,$01

;; ST-BF-LEN
L0D88:  LD      (IX+$10),E
        LD      D,$00
        ADD     IX,DE
        LD      (IX+$14),A
        RET


; ----------------------
; THE 'OUT-BLK-N' ROUTINE
; ----------------------
;

;; OUT-BLK-N
L0D93:  CALL    L0FC5           ; routine OUTPAK
        LD      A,(IX+$0B)
        AND     A
```

```
        RET     Z

        LD      HL,$5CCD        ; sv NTRESP
        LD      (HL),$00
        LD      E,$01
        CALL    L0F92           ; routine INPAK
        RET     NZ

        LD      A,($5CCD)       ; sv NTRESP
        DEC     A
        RET


; ----------------------
; THE 'S-PACK-1' ROUTINE
; ----------------------
;

;; S-PACK-1
L0DAB:  CALL    L0DB2           ; routine SEND-PACK
        RET     NZ

        JP      L0E0F           ; jump to BR-DELAY

; ----------------------
; THE 'SEND-PACK' ROUTINE
; ----------------------
;

;; SEND-PACK
L0DB2:  LD      (IX+$0F),A
        LD      B,(IX+$10)
        LD      A,($5CC6)       ; sv IOBORD
        OUT     ($FE),A
        PUSH    IX
        POP     DE
        LD      HL,$0015
        ADD     HL,DE
        XOR     A

;; CHKS1
L0DC5:  ADD     A,(HL)
        INC     HL
        DJNZ    L0DC5           ; back to CHKS1

        LD      (IX+$11),A
        LD      HL,$000B
        ADD     HL,DE
        PUSH    HL
        LD      B,$07
        XOR     A

;; CHKS2
L0DD4:  ADD     A,(HL)
        INC     HL
        DJNZ    L0DD4           ; back to CHKS2

        LD      (HL),A
        DI

;; SENDSCOUT
L0DDA:  CALL    L0F61           ; routine SEND-SC
        POP     HL
        PUSH    HL
```

```
        LD      E,$08
        CALL    L0D93           ; routine OUT-BLK-N
        JR      NZ,L0DDA        ; back to SENDSCOUT

        PUSH    IX
        POP     HL
        LD      DE,$0015
        ADD     HL,DE
        LD      E,(IX+$10)
        LD      A,E
        AND     A
        JR      Z,L0DFD         ; forward to INC-BLKN

        LD      B,$20

;; SP-DL-1
L0DF6:  DJNZ    L0DF6           ; back to SP-DL-1

        CALL    L0D93           ; routine OUT-BLK-N
        JR      NZ,L0DDA        ; back to SENDSCOUT


;; INC-BLKN
L0DFD:  INC     (IX+$0D)
        JR      NZ,L0E05        ; forward to SP-N-END

        INC     (IX+$0E)

;; SP-N-END
L0E05:  POP     HL
        CALL    L0CA9           ; routine BORD-REST
        EI
        LD      A,(IX+$0B)
        AND     A
        RET


; ----------------------
; THE 'BR-DELAY' ROUTINE
; ----------------------
;

;; BR-DELAY
L0E0F:  LD      DE,$1500

;; DL-LOOP
L0E12:  DEC     DE
        LD      A,E
        OR      D
        JR      NZ,L0E12        ; back to DL-LOOP

        RET


; ---------------------------------------------
; THE 'HEADER AND DATA BLOCK RECEIVING' ROUTINE
; ---------------------------------------------
;

;; GET-NBLK
L0E18:  LD      HL,$5CCE        ; sv NTDEST
        LD      E,$08
        CALL    L0F92           ; routine INPAK
        RET     NZ
```

```
        LD      HL,$5CCE        ; sv NTDEST
        XOR     A
        LD      B,$07

;; CHKS3
L0E27:  ADD     A,(HL)
        INC     HL
        DJNZ    L0E27           ; back to CHKS3

        CP      (HL)
        RET     NZ

        LD      A,($5CCE)       ; sv NTDEST
        AND     A
        JR      Z,L0E40         ; forward to BRCAST

        CP      (IX+$0C)
        RET     NZ

        LD      A,($5CCF)       ; sv NTSRCE
        CP      (IX+$0B)
        RET     NZ

        JR      L0E45           ; forward to TEST-BLKN


;; BRCAST
L0E40:  LD      A,(IX+$0B)
        OR      A
        RET     NZ


;; TEST-BLKN
L0E45:  LD      HL,($5CD0)      ; sv NTNUMB
        LD      E,(IX+$0D)
        LD      D,(IX+$0E)
        AND     A
        SBC     HL,DE
        JR      Z,L0E65         ; forward to GET-NBUFF

        DEC     HL
        LD      A,H
        OR      L
        RET     NZ

        CALL    L0E65           ; routine GET-NBUFF

;   Note. The DEC instruction does not affect the carry flag.

        DEC     (IX+$0D)
        JR      NC,L0E62        ; forward, with no carry, to GETNB-END !!

        DEC     (IX+$0E)

;; GETNB-END
L0E62:  OR      $01
        RET


;; GET-NBUFF
L0E65:  LD      A,($5CCE)       ; sv NTDEST
        OR      A
        CALL    NZ,L0FBE        ; routine SEND-RESP
```

```
        LD      A,($5CD3)       ; sv NTLEN
        AND     A
        JR      Z,L0E93         ; forward to STORE-LEN

        PUSH    IX
        POP     HL
        LD      DE,$0015
        ADD     HL,DE
        PUSH    HL
        LD      E,A
        CALL    L0F92           ; routine INPAK
        POP     HL
        RET     NZ

        LD      A,($5CD3)       ; sv NTLEN
        LD      B,A
        LD      A,($5CD4)       ; sv NTDCS

;; CHKS4
L0E87:  SUB     (HL)
        INC     HL
        DJNZ    L0E87           ; back to CHKS4

        RET     NZ

        LD      A,($5CCE)       ; sv NTDEST
        AND     A
        CALL    NZ,L0FBE        ; routine SEND-RESP

;; STORE-LEN
L0E93:  LD      A,($5CD3)       ; sv NTLEN
        LD      (IX+$14),A
        INC     (IX+$0D)
        JR      NZ,L0EA1        ; forward to GETNBF-END

        INC     (IX+$0E)

;; GETNBF-END
L0EA1:  CP      A
        RET


; -------------------------------------
; THE 'OPEN "N" CHANNEL COMMAND' ROUTINE
; -------------------------------------
;

;; OPEN-N-ST
L0EA3:  CALL    L0EB5           ; routine OP-PERM-N
        JP      L0B4A           ; jump to OP-STREAM

; ----------------------------------------
; THE 'OPEN TEMPORARY "N" CHANNEL' ROUTINE
; ----------------------------------------
;

;; OP-TEMP-N
L0EA9:  CALL    L0EB5           ; routine OP-PERM-N
        LD      IX,($5C51)      ; sv CURCHL
        SET     7,(IX+$04)
        RET


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

```
; THE 'OPEN PERMANENT "N" CHANNEL' ROUTINE
; ---------------------------------------
;

;; OP-PERM-N
L0EB5:  LD      HL,($5C53)      ; sv PROG
        DEC     HL
        LD      BC,$0114
        PUSH    BC
        RST     10H             ; CALBAS
        DEFW    $1655           ; main MAKE-ROOM
        INC     HL
        POP     BC
        CALL    L1691           ; routine REST-N-AD
        LD      ($5C51),HL      ; sv CURCHL
        EX      DE,HL
        LD      HL,$0EEA
        LD      BC,$000B
        LDIR
        LD      A,($5CD6)       ; sv D_STR1
        LD      (DE),A
        INC     DE
        LD      A,($5CC5)       ; sv NTSTAT
        LD      (DE),A
        INC     DE
        XOR     A
        LD      (DE),A
        LD      H,D
        LD      L,E
        INC     DE
        LD      BC,$0106
        LDIR
        LD      DE,($5C51)      ; sv CURCHL
        RET



; ------------------------------
; THE '"N" CHANNEL DATA' ROUTINE
; ------------------------------
;

;;
L0EEA:  DEFW    $0008           ; main ERROR-1
        DEFW    $0008           ; main ERROR-1
        DEFB    $4E
        DEFW    $0D6C           ;
        DEFW    $0D0C           ;
        DEFW    $0114           ;


; --------------------------------------
; THE 'SEND EOF BLOCK TO NETWORK' ROUTINE
; --------------------------------------
;

;; SEND-NEOF
L0EF5   LD      IX,($5C51)      ; sv CURCHL
        LD      A,(IX+$10)
        AND     A
        RET     Z

        LD      A,$01
        JP      L0DAB           ; jump to S-PACK-1
```

```
; --------------------------
; THE 'NETWORK STATE' ROUTINE
; --------------------------
;

;; NET-STATE
L0F03:  LD      A,R
        OR      $C0
        LD      B,A
        CALL    L0F0E           ; routine CHK-REST
        JR      C,L0F03         ; back to NET-STATE

        RET


; --------------------------
; THE 'CHECK-RESTING' ROUTINE
; --------------------------
;

;; CHK-REST
L0F0E:  LD      A,$7F
        IN      A,($FE)
        RRCA
        JR      NC,L0F4D        ; forward to E-READ-N


;; MAKESURE
L0F15:  PUSH    BC
        POP     BC
        IN      A,($F7)
        RRCA
        RET     C

        DJNZ    L0F15           ; back to MAKESURE

        RET


; -------------------------
; THE 'WAIT-SCOUT' ROUTINE
; -------------------------
;

;; WT-SCOUT
L0F1E:  LD      HL,$01C2

;; CLAIMED
L0F21:  LD      B,$80
        CALL    L0F0E           ; routine CHK-REST
        JR      NC,L0F35        ; forward to WT-SYNC

        DEC     HL
        DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L0F21        ; back to CLAIMED

        LD      A,(IX+$0B)
        AND     A
        JR      Z,L0F21         ; back to CLAIMED

        RET
```

```
;; WT-SYNC
L0F35:  IN      A,($F7)
        RRCA
        JR      C,L0F56         ; forward to SCOUT-END

        LD      A,$7F
        IN      A,($FE)
        RRCA
        JR      NC,L0F4D        ; forward to E-READ-N

        DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L0F35        ; back to WT-SYNC

        LD      A,(IX+$0B)
        AND     A
        JR      Z,L0F35         ; back to WT-SYNC

        RET


;; E-READ-N
L0F4D:  EI
        CALL    L0CA9           ; routine BORD-REST
        LD      (IY+$00),$14    ; sv ERR_NR
        RST     28H             ; romerr

;; SCOUT-END
L0F56:  LD      L,$09

;; LP-SCOUT
L0F58:  DEC     L
        SCF
        RET     Z

        LD      B,$0E

;; DELAY-SC
L0F5D:  DJNZ    L0F5D           ; back to DELAY-SC

        JR      L0F58           ; back to LP-SCOUT


; ------------------------
; THE 'SEND-SCOUT' ROUTINE
; ------------------------
;

;; SEND-SC
L0F61:  CALL    L0F03           ; routine NET-STATE
        LD      C,$F7
        LD      HL,$0009
        LD      A,($5CC5)       ; sv NTSTAT
        LD      E,A
        IN      A,($F7)
        RRCA
        JR      C,L0F61         ; back to SEND-SC


;; ALL-BITS
L0F72:  OUT     (C),H
```

```
        LD      D,H
        LD      H,$00
        RLC     E
        RL      H
        LD      B,$08

;; S-SC-DEL
L0F7D:  DJNZ    L0F7D           ; back to S-SC-DEL

        IN      A,($F7)
        AND     $01
        CP      D
        JR      Z,L0F61         ; back to SEND-SC

        DEC     L
        JR      NZ,L0F72        ; back to ALL-BITS

        LD      A,$01
        OUT     ($F7),A
        LD      B,$0E

;; END-S-DEL
L0F8F:  DJNZ    L0F8F           ; back to END-S-DEL

        RET


; -------------------
; THE 'INPAK' ROUTINE
; -------------------
;

;; INPAK
L0F92:  LD      B,$FF

;; N-ACTIVE
L0F94:  IN      A,($F7)
        RRA
        JR      C,L0F9D         ; forward to INPAK-2

        DJNZ    L0F94           ; back to N-ACTIVE

        INC     B
        RET


;; INPAK-2
L0F9D:  LD      B,E

;; INPAK-L
L0F9E:  LD      E,$80
        LD      A,$CE
        OUT     ($EF),A
        NOP
        NOP
        INC     IX
        DEC     IX
        INC     IX
        DEC     IX

;; UNTIL-MK
L0FAE:  LD      A,$00
        IN      A,($F7)
        RRA
```

```
        RR      E
        JP      NC,L0FAE        ; jump to UNTIL-MK
        LD      (HL),E
        INC     HL
        DJNZ    L0F9E           ; back to INPAK-L

        CP      A
        RET


; --------------------------------
; THE 'SEND RESPONSE BYTE' ROUTINE
; --------------------------------
;

;; SEND-RESP
L0FBE:  LD      A,$01
        LD      HL,$5CCD        ; sv NTRESP
        LD      (HL),A
        LD      E,A

; --------------------
; THE 'OUTPAK' ROUTINE
; --------------------
;

;; OUTPAK
L0FC5:  XOR     A
        OUT     ($F7),A
        LD      B,$04

;; DEL-D-1
L0FCA:  DJNZ    L0FCA           ; back to DEL-D-1


;; OUTPAK-L
L0FCC:  LD      A,(HL)
        CPL
        SCF
        RLA
        LD      B,$0A

;; UNT-MARK
L0FD2:  OUT     ($F7),A
        RRA
        AND     A
        DEC     B
        LD      D,$00
        JP      NZ,L0FD2        ; jump to UNT-MARK
        INC     HL
        DEC     E
        PUSH    HL
        POP     HL
        JP      NZ,L0FCC        ; jump to OUTPAK-L
        LD      A,$01
        OUT     ($F7),A
        RET


; ----------------------------------------
; THE 'SET A TEMPORARY "M" CHANNEL' ROUTINE
; ----------------------------------------
;
```

```
;; SET-T-MCH
L0FE8:  EXX
        LD      HL,L0000
        EXX
        LD      IX,($5C4F)      ; sv CHANS
        LD      DE,$0014
        ADD     IX,DE

;; CHK-LOOP
L0FF6:  LD      A,(IX+$00)
        CP      $80
        JR      Z,L1034         ; forward to CHAN-SPC

        LD      A,(IX+$04)
        AND     $7F
        CP      $4D
        JR      NZ,L102A        ; forward to NEXT-CHAN

        LD      A,($5CD6)       ; sv D_STR1
        CP      (IX+$19)
        JR      NZ,L102A        ; forward to NEXT-CHAN

        EXX
        LD      L,(IX+$1A)
        LD      H,(IX+$1B)
        EXX
        LD      BC,($5CDA)      ; sv D_STR1
        LD      HL,($5CDC)      ; sv D_STR1
        CALL    L131E           ; routine CHK-NAME
        JR      NZ,L102A        ; forward to NEXT-CHAN

        BIT     0,(IX+$18)
        JR      Z,L102A         ; forward to NEXT-CHAN

        RST     20H             ; sh_err
        DEFB    $0D

;; NEXT-CHAN
L102A:  LD      E,(IX+$09)
        LD      D,(IX+$0A)
        ADD     IX,DE
        JR      L0FF6           ; back to CHK-LOOP


;; CHAN-SPC
L1034:  LD      HL,($5C53)      ; sv PROG
        DEC     HL
        PUSH    HL
        LD      BC,$0253
        RST     10H             ; CALBAS
        DEFW    $1655           ; main MAKE-ROOM
        POP     DE
        PUSH    DE
        LD      HL,$13CC
        LD      BC,$0019
        LDIR
        LD      A,($5CD6)       ; sv D_STR1
        LD      (IX+$19),A
        LD      BC,$0253
        PUSH    IX
        POP     HL
        CALL    L1691           ; routine REST-N-AD
        EX      DE,HL
        LD      BC,($5CDA)      ; sv D_STR1
```

```
        BIT     7,B
        JR      NZ,L106F        ; forward to TEST-MAP


;; T-CH-NAME
L1061:  LD      A,B
        OR      C
        JR      Z,L106F         ; forward to TEST-MAP

        LD      A,(HL)
        LD      (IX+$0E),A
        INC     HL
        INC     IX
        DEC     BC
        JR      L1061           ; back to T-CH-NAME


;; TEST-MAP
L106F:  POP     IX
        EXX
        LD      A,H
        OR      L
        JR      NZ,L108A        ; forward to ST-MAP-AD

        LD      HL,($5C4F)      ; sv CHANS
        PUSH    HL
        DEC     HL
        LD      BC,L0020
        RST     10H             ; CALBAS
        DEFW    $1655           ; main MAKE-ROOM
        POP     HL
        LD      BC,L0020
        ADD     IX,BC
        CALL    L1691           ; routine REST-N-AD

;; ST-MAP-AD
L108A:  LD      (IX+$1A),L
        LD      (IX+$1B),H
        LD      A,$FF
        LD      B,$20

;; FILL-14AP
L1094:  LD      (HL),A
        INC     HL
        DJNZ    L1094           ; back to FILL-14AP

        PUSH    IX
        POP     HL
        LD      DE,$001C
        ADD     HL,DE
        EX      DE,HL
        LD      HL,$13E5
        LD      BC,$000C
        LDIR
        PUSH    IX
        POP     HL
        LD      DE,$0037
        LD      BC,$000C
        ADD     HL,DE
        EX      DE,HL
        LD      HL,$13E5
        LDIR
        PUSH    IX
        POP     HL
```

```
        LD      DE,($5C4F)      ; sv CHANS
        OR      A
        SBC     HL,DE
        INC     HL
        RET


; ---------------------------------
; THE 'RECLAIM "M" CHANNEL' ROUTINE
; ---------------------------------
;

;; DEL-M-BUF
L10C4:  LD      L,(IX+$1A)
        LD      H,(IX+$1B)
        PUSH    HL
        LD      A,(IX+$19)
        PUSH    AF
        PUSH    IX
        POP     HL
        LD      BC,$0253
        RST     10H             ; CALBAS
        DEFW    $19E8           ; main RECLAIM-2
        PUSH    IX
        POP     HL
        LD      DE,($5C4F)      ; sv CHANS
        OR      A
        SBC     HL,DE
        INC     HL
        LD      BC,$0253
        CALL    L135F           ; routine RE-ST-STRM
        POP     AF
        POP     HL
        LD      B,A
        LD      IX,($5C4F)      ; sv CHANS
        LD      DE,$0014
        ADD     IX,DE

;; TEST-MCHL
L10F5:  LD      A,(IX+$00)
        CP      $80
        JR      Z,L1114         ; forward to RCLM-MAP

        LD      A,(IX+$04)
        AND     $7F
        CP      $4D
        JR      NZ,L110A        ; forward to NXTCHAN

        LD      A,(IX+$19)
        CP      B
        RET     Z


;; NXTCHAN
L110A:  LD      E,(IX+$09)
        LD      D,(IX+$0A)
        ADD     IX,DE
        JR      L10F5           ; back to TEST-MCHL


;; RCLM-MAP
L1114:  LD      BC,L0020
        PUSH    HL
        PUSH    BC
```

```
        RST     10H             ; CALBAS
        DEFW    $19E8           ; main RECLAIM-2
        POP     BC
        POP     HL
        CALL    L1391           ; routine REST-MAP
        RET


; -------------------------------
; THE '"M" CHANNEL INPUT' ROUTINE
; -------------------------------
;

;; M-INPUT
L1122:  LD      IX,($5C51)      ; sv CURCHL
        LD      HL,$112C
        JP      L0CBD           ; jump to CALL-INP


; --------------------------------------
; THE '"M" CHANNEL INPUT SERVICE' ROUTINE
; --------------------------------------
;

;; MCHAN-IN
L112C:  BIT     0,(IX+$18)
        JR      Z,L1134         ; forward to TEST-M-BF

        RST     20H             ; sh_err
        DEFB    $0D

;; TEST-M-BF
L1134:  LD      E,(IX+$0B)
        LD      D,(IX+$0C)
        LD      L,(IX+$45)
        LD      H,(IX+$46)
        SCF
        SBC     HL,DE
        JR      C,L1158         ; forward to CHK-M-EOF

        INC     DE
        LD      (IX+$0B),E
        LD      (IX+$0C),D
        DEC     DE
        PUSH    IX
        ADD     IX,DE
        LD      A,(IX+$52)
        POP     IX
        SCF
        RET


;; CHK-M-EOF
L1158:  BIT     1,(IX+$43)
        JR      Z,L1162         ; forward to NEW-BUFF

        XOR     A
        ADD     A,$0D
        RET


;; NEW-BUFF
L1162:  LD      DE,L0000
        LD      (IX+$0B),E
```

```
        LD      (IX+$0C),D
        INC     (IX+$0D)
        CALL    L1177           ; routine GET-RECD
        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        JR      L1134           ; back to TEST-M-BF


; --------------------------
; THE 'GET A RECORD' ROUTINE
; --------------------------
;

;; GET-RECD
L1177:  LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE

;; GET-R-2
L117D:  LD      BC,$04FB
        LD      ($5CC9),BC      ; sv SECTOR

;; GET-R-LP
L1184:  CALL    L11A5           ; routine G-RD-RC
        JR      C,L119E         ; forward to NXT-SCT

        JR      Z,L119E         ; forward to NXT-SCT

        LD      A,(IX+$44)
        CP      (IX+$0D)
        JR      NZ,L119E        ; forward to NXT-SCT

        PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF
        RET     Z


;; NXT-SCT
L119E:  CALL    L1312           ; routine DEC-SECT
        JR      NZ,L1184        ; back to GET-R-LP

        RST     20H             ; sh_err
        DEFB    $11

; --------------------------------------
; THE 'GET HEADER AND DATA BLOCK' ROUTINE
; --------------------------------------
;

;; G-RD-RC
L11A5:  CALL    $12C4
        LD      DE,$001B
        ADD     HL,DE
        CALL    L18A9           ; routine GET-M-BUF
        CALL    L1341           ; routine CHKS-HD-R
        JR      NZ,L11D6        ; forward to G-REC-ERR

        BIT     0,(IX+$43)
        JR      NZ,L11D6        ; forward to G-REC-ERR

        LD      A,(IX+$43)
        OR      (IX+$46)
```

```
        AND     $02
        RET     Z

        PUSH    IX
        POP     HL
        LD      DE,$0047
        ADD     HL,DE
        LD      BC,$000A
        CALL    L131E           ; routine CHK-NAME
        JR      NZ,L11D6        ; forward to G-REC-ERR

        LD      A,$FF
        OR      A
        RET


;; G-REC-ERR
L11D6:  SCF
        RET


; --------------------------------
; THE '"M" CHANNEL OUTPUT' ROUTINE
; --------------------------------
;

;; MCHAN-OUT
L11D8:  LD      IX,$FFFA
        ADD     IX,DE
        BIT     0,(IX+$18)
        JR      NZ,L11E6        ; forward to NOREAD

        RST     20H             ; sh_err
        DEFB    $0C

;; NOREAD
L11E6:  LD      E,(IX+$0B)
        LD      D,(IX+$0C)
        PUSH    IX
        ADD     IX,DE
        LD      (IX+$52),A
        POP     IX
        INC     DE
        LD      (IX+$0B),E
        LD      (IX+$0C),D
        BIT     1,D
        RET     Z


; ----------------------------------------
; THE 'WRITE RECORD ONTO MICRODRIVE' ROUTINE
; ----------------------------------------
;

;; WR-RECD
L11FF:  LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE
        CALL    L120D           ; routine WRITE-PRC
        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        RET


;; WRITE-PRC
```

```
L120D:  CALL    L1264           ; routine CHK-FULL
        JR      NZ,L121B        ; forward to NOFULL

        CALL    L10C4           ; routine DEL-M-BUF
        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        RST     20H             ; sh_err
        DEFB    $0F

;; NOFULL
L121B:  PUSH    IX
        LD      B,$0A

;; CP-NAME
L121F:  LD      A,(IX+$0E)
        LD      (IX+$47),A
        INC     IX
        DJNZ    L121F           ; back to CP-NAME

        POP     IX
        LD      C,(IX+$0B)
        LD      (IX+$45),C
        LD      A,(IX+$0C)
        LD      (IX+$46),A
        LD      A,(IX+$0D)
        LD      (IX+$44),A
        PUSH    IX
        POP     HL
        LD      DE,$0043
        ADD     HL,DE
        CALL    L1341           ; routine CHKS-HD-R
        LD      DE,$000F
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF
        PUSH    IX
        POP     HL
        LD      DE,$0047
        CALL    L1275           ; routine SEND-BLK
        LD      DE,L0000
        LD      (IX+$0B),E
        LD      (IX+$0C),D
        INC     (IX+$0D)
        RET


; ----------------------
; THE 'CHK-FULL' ROUTINE
; ----------------------
;

;; CHK-FULL
L1264:  LD      L,(IX+$1A)
        LD      H,(IX+$1B)
        LD      B,$20

;; NXT-B-MAP
L126C:  LD      A,(HL)
        CP      $FF
        RET     NZ

        INC     HL
        DJNZ    L126C           ; back to NXT-B-MAP

        XOR     A
```

```
        RET


; ----------------------
; THE 'SEND-BLK' ROUTINE
; ----------------------
;

;; SEND-BLK
L1275:  PUSH    IX
        POP     HL
        LD      DE,$0037
        ADD     HL,DE
        PUSH    HL

;; FAILED
L127D:  CALL    L12C4           ; routine GET-M-RD2
        CALL    L12DF           ; routine CHECK-MAP
        JR      NZ,L127D        ; back to FAILED

        EX      (SP),HL
        PUSH    BC
        IN      A,($EF)
        AND     $01
        JR      NZ,L128F        ; forward to NO-PRT

        RST     20H             ; sh_err
        DEFB    $0E


;; NO-PRT
L128F:  LD      A,$E6
        OUT     ($EF),A
        LD      BC,$0168
        CALL    L18FA           ; routine DELAY-BC
        CALL    L1878           ; routine OUT-H-BUF
        LD      A,$EE
        OUT     ($EF),A
        POP     BC
        POP     HL
        LD      A,B
        OR      (HL)
        LD      (HL),A
        RET


; ------------------------
; THE 'CLOSE FILE' ROUTINE
; ------------------------
;

;; CLOSE-M
L12A6:  PUSH    HL
        POP     IX

;; CLOSE-M2
L12A9:  BIT     0,(IX+$18)
        JR      Z,L12B6         ; forward to NOEMP

        SET     1,(IX+$43)
        CALL    L11FF           ; routine WR-RECD

;; NOEMP
L12B6:  XOR     A
```

```
        CALL    L17F7           ; routine SEL-DRIVE
        CALL    L10C4           ; routine DEL-M-BUF
        RET


; --------------------
; THE 'ERR-RS' ROUTINE
; --------------------
;

;; ERR-RS
L12BE:  POP     HL
        LD      A,(HL)
        LD      ($5C3A),A       ; sv ERR_NR
        RST     28H             ; romerr

; ----------------------------------------
; THE 'FETCH HEADER FROM MICRODRIVE' ROUTINE
; ----------------------------------------
;

;; GET-M-RD2
L12C4:  PUSH    IX
        POP     HL
        LD      DE,L0028
        ADD     HL,DE
        CALL    L18A3           ; routine GET-M-HD
        CALL    L1341           ; routine CHKS-HD-R
        JR      NZ,L12C4        ; back to GET-M-RD2

        BIT     0,(IX+$28)
        JR      Z,L12C4         ; back to GET-M-RD2

        RET


; ---------------------------------
; THE 'CHECK MAP BIT STATE' ROUTINE
; ---------------------------------
;

;; CHK-MAP-2
L12DA:  LD      E,(IX+$44)
        JR      L12E2           ; forward to ENTRY


;; CHECK-MAP
L12DF:  LD      E,(IX+$29)

;; ENTRY
L12E2:  LD      L,(IX+$1A)
        LD      H,(IX+$1B)

;; ENTRY-2
L12E8:  XOR     A
        LD      D,A
        LD      A,E
        AND     $07
        SRL     E
        SRL     E
        SRL     E
        ADD     HL,DE
        LD      B,A
        INC     B
```

```
        XOR     A
        SCF

;; ROTATE
L12F8:  RLA
        DJNZ    L12F8           ; back to ROTATE

        LD      B,A
        AND     (HL)
        RET


; -----------------------------------
; THE 'RESET BIT IN MAP AREA' ROUTINE
; -----------------------------------
;

;; RES-B-HAP
L12FE:  CALL    L12DF           ; routine CHECK-MAP
        LD      A,B
        CPL
        AND     (HL)
        LD      (HL),A
        RET


; ----------------------------------------
; THE 'CHECK 'PSEUDO-MAP' BIT STATE' ROUTINE
; ----------------------------------------
;

;; TEST-PHAP
L1306:  PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      E,(IX+$29)
        JR      L12E8           ; back to ENTRY-2


; -------------------------------------
; THE 'DECREASE SECTOR COUNTER' ROUTINE
; -------------------------------------
;

;; DEC-SECT
L1312:  LD      BC,($5CC9)      ; sv SECTOR
        DEC     BC
        LD      ($5CC9),BC      ; sv SECTOR
        LD      A,B
        OR      C
        RET


; ------------------------
; THE 'CHECK-NAME' ROUTINE
; ------------------------
;

;; CHK-NAME
L131E:  PUSH    IX
        LD      B,$0A

;; ALL-CHARS
```

```
L1322:  LD      A,(HL)
        CP      (IX+$0E)
        JR      NZ,L133E        ; forward to CHKNAM-END

        INC     HL
        INC     IX
        DEC     B
        DEC     C
        JR      NZ,L1322        ; back to ALL-CHARS

        LD      A,B
        OR      A
        JR      Z,L133E         ; forward to CHKNAM-END


;; ALLCHR-2
L1333:  LD      A,(IX+$0E)
        CP      $20
        JR      NZ,L133E        ; forward to CHKNAM-END

        INC     IX
        DJNZ    L1333           ; back to ALLCHR-2


;; CHKNAM-END
L133E:  POP     IX
        RET


; -----------------------------------------
; THE 'CALCULATE/COMPARE CHECKSUMS' ROUTINE
; -----------------------------------------
;

;; CHKS-HD-R
L1341:  LD      BC,$000E
        JR      L1349           ; forward to CNKS-ALL


;; CHKS-BUFF
L1346:  LD      BC,$0200

;; CNKS-ALL
L1349:  PUSH    HL
        LD      E,$00

;; NXT-BYTE
L134C:  LD      A,E
        ADD     A,(HL)
        INC     HL
        ADC     A,$01
        JR      Z,L1354         ; forward to STCHK

        DEC     A

;; STCHK
L1354:  LD      E,A
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,L134C        ; back to NXT-BYTE

        LD      A,E
        CP      (HL)
```

```
        LD      (HL),A
        POP     HL
        RET


; ---------------------------------
; THE 'RESTORE STREAM DATA' ROUTINE
; ---------------------------------
;

;; RE-ST-STRM
L135F:  PUSH    HL
        LD      A,$10
        LD      HL,$5C16        ; sv STRMS_00

;; NXT-STRM
L1365:  LD      ($5C5F),HL      ; sv X_PTR
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        POP     HL
        PUSH    HL
        OR      A
        SBC     HL,DE
        JR      NZ,L1377        ; forward to NOTRIGHT

        LD      DE,L0000
        JR      L137E           ; forward to STO-DISP


;; NOTRIGHT
L1377:  JR      NC,L1384        ; forward to UPD-POINT

        EX      DE,HL
        OR      A
        SBC     HL,BC
        EX      DE,HL

;; STO-DISP
L137E:  LD      HL,($5C5F)      ; sv X_PTR
        LD      (HL),E
        INC     HL
        LD      (HL),D

;; UPD-POINT
L1384:  LD      HL,($5C5F)      ; sv X_PTR
        INC     HL
        INC     HL
        DEC     A
        JR      NZ,L1365        ; back to NXT-STRM

        LD      ($5C5F),A       ; sv X_PTR
        POP     HL
        RET


; -----------------------------------
; THE 'RESTORE MAP ADDRESSES' ROUTINE
; -----------------------------------
;

;; REST-MAP
L1391:  LD      BC,L0020
        LD      IX,($5C4F)      ; sv CHANS
```

```
        LD      DE,$0014
        ADD     IX,DE

;; LCHAN
L139D:  LD      A,(IX+$00)
        CP      $80
        RET     Z

        PUSH    HL
        LD      A,(IX+$04)
        AND     $7F
        CP      $4D
        JR      NZ,L13C1        ; forward to LPEND

        LD      E,(IX+$1A)
        LD      D,(IX+$1B)
        SBC     HL,DE
        JR      NC,L13C1        ; forward to LPEND

        EX      DE,HL
        OR      A
        SBC     HL,BC
        LD      (IX+$1A),L
        LD      (IX+$1B),H

;; LPEND
L13C1:  POP     HL
        LD      E,(IX+$09)
        LD      D,(IX+$0A)
        ADD     IX,DE
        JR      L139D           ; back to LCHAN



; ------------------------------
; THE '"M" CHANNEL DATA' ROUTINE
; ------------------------------
;

;;
L13CC:  DEFW    $0008           ; main ERROR-1
        DEFW    $0008           ; main ERROR-1
        DEFB    $CD
        DEFW    $11D8           ;
        DEFW    $1122           ;
        DEFW    $0253           ;
        DEFW    $0000           ;
        DEFB    $00
        DEFM    "          "    ; 10 spaces
        DEFB    $FF



; ---------------------------
; THE 'PREAMBLE DATA' ROUTINE
; ---------------------------
;

;;
L13E5:  DEFB    $00, $00, $00
        DEFB    $00, $00, $00
        DEFB    $00, $00, $00
        DEFB    $00, $FF, $FF
```

```
; -------------------------
; THE 'MOVE COMMAND' ROUTINE
; -------------------------
;

;; MOVE
L13F1:  SET     4,(IY+$7C)      ; sv FLAGS_3
        CALL    L1455           ; routine OP-STRM
        LD      HL,($5C4F)      ; sv CHANS
        PUSH    HL
        CALL    L14C7           ; routine EX-DSTR2
        CALL    L1455           ; routine OP-STRM
        CALL    L14C7           ; routine EX-DSTR2
        POP     DE
        LD      HL,($5C4F)      ; sv CHANS
        OR      A
        SBC     HL,DE
        LD      DE,($5CDA)      ; sv D_STR1
        ADD     HL,DE
        LD      ($5CDA),HL      ; sv D_STR1

;; M-AGAIN
L1414:  LD      HL,($5CDA)      ; sv D_STR1
        LD      ($5C51),HL      ; sv CURCHL

;; I-AGAIN
L141A:  RST     10H             ; CALBAS
        DEFW    $15E6           ; main INPUT-AD
        JR      C,L1423         ; forward to MOVE-OUT

        JR      Z,L141A         ; back to I-AGAIN

        JR      L142E           ; forward to MOVE-EOF


;; MOVE-OUT
L1423:  LD      HL,($5CE2)      ; sv D_STR2
        LD      ($5C51),HL      ; sv CURCHL
        RST     10H             ; CALBAS
        DEFW    $0010           ; main PRINT-A
        JR      L1414           ; back to M-AGAIN


;; MOVE-EOF
L142E:  RES     4,(IY+$7C)      ; sv FLAGS_3
        LD      HL,($5C4F)      ; sv CHANS
        PUSH    HL
        CALL    L14C7           ; routine EX-DSTR2
        CALL    L14A4           ; routine CL-CHAN
        CALL    L14C7           ; routine EX-DSTR2
        POP     DE
        LD      HL,($5C4F)      ; sv CHANS
        OR      A
        SBC     HL,DE
        LD      DE,($5CDA)      ; sv D_STR1
        ADD     HL,DE
        LD      ($5CDA),HL      ; sv D_STR1
        CALL    L14A4           ; routine CL-CHAN
        CALL    L17B9           ; routine RCL-T-CH
        RET
```

```
; ------------------------------------------
; THE 'USE STREAM OR TEMPORARY CHANNEL' ROUTINE
; ------------------------------------------
;

;; OP-STRM
L1455:  LD      A,($5CD8)       ; sv D_STR1
        INC     A
        JR      Z,L1466         ; forward to OP-CHAN

        DEC     A
        RST     10H             ; CALBAS
        DEFW    $1601           ; main CHAN-OPEN
        LD      HL,($5C51)      ; sv CURCHL
        LD      ($5CDA),HL      ; sv D_STR1
        RET


;; OP-CHAN
L1466:  LD      A,($5CD9)       ; sv D_STR1
        CP      $4D
        JR      NZ,L147F        ; forward to CHECK-N

        CALL    L1B29           ; routine OP-TEMP-M
        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        LD      ($5CDA),IX      ; sv D_STR1
        BIT     2,(IX+$43)
        RET     Z

        RST     20H             ; sh_err
        DEFB    $16


;; CHECK-N
L147F:  CP      $4E
        JR      NZ,L148B        ; forward to CHECK-R

        CALL    L0EA9           ; routine OP-TEMP-N
        LD      ($5CDA),IX      ; sv D_STR1
        RET


;; CHECK-R
L148B:  CP      $54
        JR      Z,L1495         ; forward to USE-R

        CP      $42
        JR      Z,L1495         ; forward to USE-R

        RST     20H             ; sh_err
        DEFB    $00


;; USE-R
L1495:  CALL    L0B13           ; routine OP-RS-CH
        LD      ($5CDA),DE      ; sv D_STR1
        PUSH    DE
        POP     IX
        SET     7,(IX+$04)
        RET


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

```
; THE 'CLOSE 'MOVE' CHANNEL' ROUTINE
; ---------------------------------
;

;; CL-CHAN
L14A4:  LD      A,($5CD8)       ; sv D_STR1
        INC     A
        RET     NZ

        LD      A,($5CD9)       ; sv D_STR1
        CP      $4D
        JR      NZ,L14B8        ; forward to CL-CHK-N

        LD      IX,($5CDA)      ; sv D_STR1
        CALL    L12A9           ; routine CLOSE-M2
        RET


;; CL-CHK-N
L14B8:  CP      $4E
        RET     NZ

        LD      IX,($5CDA)      ; sv D_STR1
        LD      ($5C51),IX      ; sv CURCHL
        CALL    L0EF5           ; routine SEND-NEOF
        RET


; -----------------------------------------------
; THE 'EXCHANGE DSTRI AND STR2 CONTENTS' ROUTINE
; -----------------------------------------------
;

;; EX-DSTR2
L14C7:  LD      DE,$5CD6        ; sv D_STR1
        LD      HL,$5CDE        ; sv D_STR2
        LD      B,$08

;; ALL-BYT-2
L14CF:  LD      A,(DE)
        LD      C,(HL)
        EX      DE,HL
        LD      (HL),C
        LD      (DE),A
        EX      DE,HL
        INC     HL
        INC     DE
        DJNZ    L14CF           ; back to ALL-BYT-2

        RET


; -----------------------------------------------
; THE 'SAVE DATA BLOCK INTO MICRODRIVE' ROUTINE
; -----------------------------------------------
;

;; SA-DRIVE
L14DA:  LD      A,($5CD6)       ; sv D_STR1
        CALL    L17F7           ; routine SEL-DRIVE
        IN      A,($EF)
        AND     $01
        JR      NZ,L14E8        ; forward to START-SA
```

```
        RST     20H             ; sh_err
        DEFB    $0E


;; START-SA
L14E8:  LD      HL,($5CE9)      ; sv HD_0D
        LD      ($5CE4),HL      ; sv D_STR2
        CALL    L1B29           ; routine OP-TEMP-M
        BIT     0,(IX+$18)
        JR      NZ,L14FC        ; forward to NEW-NAME

        CALL    L12A9           ; routine CLOSE-M2
        RST     20H             ; sh_err
        DEFB    $0C

;; NEW-NAME
L14FC:  SET     2,(IX+$43)
        LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE
        PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        EX      DE,HL
        LD      HL,$5CE6        ; sv HD_00
        LD      BC,$0009
        LD      (IX+$0B),C
        LDIR
        PUSH    DE
        LD      HL,$0009
        LD      BC,($5CE7)      ; sv HD_0B
        ADD     HL,BC
        SRL     H
        INC     H
        PUSH    HL
        CALL    L1D38           ; routine FREESECT
        POP     HL
        LD      A,E
        CP      H
        JR      NC,L1530        ; forward to SA-DRI-2

        RST     20H             ; sh_err
        DEFB    $0F

;; SA-DRI-2
L1530:  POP     DE
        LD      HL,($5CE4)      ; sv D_STR2
        LD      BC,($5CE7)      ; sv HD_0B

;; SA-DRI-3
L1538:  LD      A,B
        OR      C
        JR      Z,L155E         ; forward to SA-DRI-4

        LD      A,(IX+$0C)
        CP      $02
        JR      NZ,L1552        ; forward to SA-DRI-WR

        PUSH    HL
        PUSH    BC
        CALL    L120D           ; routine WRITE-PRC
        POP     BC
        PUSH    IX
        POP     HL
```

```
        LD      DE,$0052
        ADD     HL,DE
        EX      DE,HL
        POP     HL

;; SA-DRI-WR
L1552:  LDI
        INC     (IX+$0B)
        JR      NZ,L1538        ; back to SA-DRI-3

        INC     (IX+$0C)
        JR      L1538           ; back to SA-DRI-3


;; SA-DRI-4
L155E:  SET     1,(IX+$43)
        CALL    L120D           ; routine WRITE-PRC
        LD      A,($5CEF)       ; sv COPIES
        DEC     A
        JR      Z,L1579         ; forward to END-SA-DR

        LD      ($5CEF),A       ; sv COPIES
        RES     1,(IX+$43)
        LD      A,$00
        LD      (IX+$0D),A
        JR      L14FC           ; back to NEW-NAME


;; END-SA-DR
L1579:  XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        JP      L10C4           ; jump to DEL-M-BUF

; -------------------------------------------------
; THE 'GET HEADER INFORMATION FROM MICRODRIVE' ROUTINE
; -------------------------------------------------
;

;; F-M-HM
L1580:  LD      HL,($5CE1)      ; sv D_STR2
        LD      ($5CE4),HL      ; sv D_STR2
        CALL    L1B29           ; routine OP-TEMP-M
        BIT     0,(IX+$18)
        JR      Z,L1591         ; forward to F-HD-2

        RST     20H             ; sh_err
        DEFB    $11


;; F-HD-2
L1591:  BIT     2,(IX+$43)      ;
        JR      NZ,L1599        ; forward to F-HD-3


        RST     20H             ; sh_err
        DEFB    $16

;; F-HD-3
L1599:  PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      DE,$5CE6        ; sv HD_00
        LD      BC,$0009
```

```
        LDIR
        RET


; ------------------------------------------------
; THE 'LOAD OR VERIFY BLOCK FROM MICRODRIVE' ROUTINE
; ------------------------------------------------
;

;; LV-MCH
L15A9:  LD      ($5CE9),HL      ; sv HD_0D
        LD      E,(IX+$53)
        LD      D,(IX+$54)
        LD      HL,L0008
        ADD     HL,DE
        SRL     H
        INC     H
        LD      A,H
        LD      ($5CE7),A       ; sv HD_0B
        CALL    L1613           ; routine SA-MAP
        LD      DE,$0009
        LD      L,(IX+$45)
        LD      H,(IX+$46)
        OR      A
        SBC     HL,DE
        LD      (IX+$45),L
        LD      (IX+$46),H
        PUSH    IX
        POP     HL
        LD      DE,$005B
        ADD     HL,DE
        LD      DE,($5CE9)      ; sv HD_0D
        JR      L15F9           ; forward to LOOK-MAP


;; USE-REC
L15DF:  CALL    L166C           ; routine F-REC2
        LD      A,(IX+$44)
        OR      A
        JR      Z,L15DF         ; back to USE-REC

        RLA
        DEC     A
        LD      D,A
        LD      E,$F7
        LD      HL,($5CE9)      ; sv HD_0D
        ADD     HL,DE
        EX      DE,HL
        PUSH    IX
        POP     HL
        LD      BC,$0052
        ADD     HL,BC

;; LOOK-MAP
L15F9:  EXX
        CALL    L12DA           ; routine CHK-MAP-2
        JR      NZ,L15DF        ; back to USE-REC

        LD      A,(HL)
        OR      B
        LD      (HL),A
        EXX
        CALL    L1648           ; routine LD-VE-M
        LD      A,($5CE7)       ; sv HD_0B
```

```
        DEC     A
        LD      ($5CE7),A       ; sv HD_0B
        JR      NZ,L15DF        ; back to USE-REC

        CALL    L162D           ; routine RE-MAP
        RET


; ----------------------------------------
; THE 'SAVE MICRODRIVE MAP CONTENTS' ROUTINE
; ----------------------------------------
;

;; SA-MAP
L1613:  POP     HL
        LD      ($5CC9),HL      ; sv SECTOR
        LD      L,(IX+$1A)
        LD      H,(IX+$1B)
        LD      BC,$1000

;; SA-HAP-LP
L1620:  LD      E,(HL)
        LD      (HL),C
        INC     HL
        LD      D,(HL)
        LD      (HL),C
        INC     HL
        PUSH    DE
        DJNZ    L1620           ; back to SA-HAP-LP

        LD      HL,($5CC9)      ; sv SECTOR
        JP      (HL)

; ----------------------------------------
; THE 'RESTORE MICRODRIVE MAP CONTENTS' ROUTINE
; ----------------------------------------
;

;; RE-MAP
L162D:  POP     HL
        LD      ($5CC9),HL      ; sv SECTOR
        LD      L,(IX+$1A)
        LD      H,(IX+$1B)
        LD      DE,$001F
        ADD     HL,DE
        LD      B,$10

;; RE-MAP-LP
L163D:  POP     DE
        LD      (HL),D
        DEC     HL
        LD      (HL),E
        DEC     HL
        DJNZ    L163D           ; back to RE-MAP-LP

        LD      HL,($5CC9)      ; sv SECTOR
        JP      (HL)

; ---------------------
; THE 'LD-VE-M' ROUTINE
; ---------------------
;

;; LD-VE-M
```

```
L1648:  LD      C,(IX+$45)
        LD      B,(IX+$46)
        LD      A,($5CB6)       ; sv FLAGS_3
        BIT     7,A
        JR      NZ,L1658        ; forward to VE-M-E

        LDIR
        RET


;; VE-M-E
L1658:  LD      A,(DE)
        CP      (HL)
        JR      NZ,L1664        ; forward to VE-FAIL

        INC     HL
        INC     DE
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,L1658        ; back to VE-M-E

        RET


;; VE-FAIL
L1664:  RST     20H             ; sh_err
        DEFB    $15


; -----------------------------------------
; THE 'FETCH RECORD FROM MICRODRIVE.' ROUTINE
; -----------------------------------------
;

;; F-REC1
L1666:  LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE

;; F-REC2
L166C:  LD      BC,$04FB
        LD      ($5CC9),BC      ; sv SECTOR

;; UNTILFIVE
L1673:  CALL    L11A5           ; routine G-RD-RC
        JR      C,L168A         ; forward to F-ERROR

        JR      Z,L168A         ; forward to F-ERROR

        CALL    L12DA           ; routine CHK-MAP-2
        JR      NZ,L168A        ; forward to F-ERROR

        PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF
        RET     Z


;; F-ERROR
L168A:  CALL    L1312           ; routine DEC-SECT
        JR      NZ,L1673        ; back to UNTILFIVE
```

```
        RST     20H             ; sh_err
        DEFB    $11


; ----------------------------------------
; THE 'RESTORE ADDRESS OF FILENAME' ROUTINE
; ----------------------------------------
;

;; REST-N-AD
L1691:  PUSH    HL
        PUSH    HL
        LD      DE,($5CE4)      ; sv D_STR2
        CALL    L16AC           ; routine TST-PLACE
        LD      ($5CE4),DE      ; sv D_STR2
        POP     HL
        LD      DE,($5CDC)      ; sv D_STR1
        CALL    L16AC           ; routine TST-PLACE
        LD      ($5CDC),DE      ; sv D_STR1
        POP     HL
        RET


;; TST-PLACE
L16AC:  SCF
        SBC     HL,DE
        RET     NC

        LD      HL,($5C65)      ; sv STKEND
        SBC     HL,DE
        RET     C

        EX      DE,HL
        ADD     HL,BC
        EX      DE,HL
        RET

        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
```

```
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF


; --------------------------
; THE 'CLOSE STREAM' ROUTINE
; --------------------------
;

;; CLOSE-CH
L1708:  INC     HL
        RST     30H             ; NEWVARS
        SRL     A
        SUB     $03
        RES     1,(IY+$7C)      ; sv FLAGS_3
```

```
        CALL    L1718           ; routine CLOSE
        JP      L05C1           ; jump to END1

; ---------------------------
; THE 'CLOSE COMMAND' ROUTINE
; ---------------------------
;

;; CLOSE
L1718:  RST     10H             ; CALBAS
        DEFW    $1727           ; main STR-DATA1
        LD      A,C
        OR      B
        RET     Z

        PUSH    BC
        PUSH    HL
        LD      HL,($5C4F)      ; sv CHANS
        DEC     HL
        ADD     HL,BC
        EX      (SP),HL
        RST     10H             ; CALBAS
        DEFW    $16EB           ; main CLOSEX
        LD      HL,($5C4F)      ; sv CHANS
        LD      DE,$0014
        ADD     HL,DE
        POP     DE
        SCF
        SBC     HL,DE
        POP     BC
        RET     NC

        PUSH    BC
        PUSH    DE
        EX      DE,HL
        LD      ($5C51),HL      ; sv CURCHL
        INC     HL
        INC     HL
        INC     HL
        INC     HL
        LD      A,(HL)
        LD      DE,$0005
        ADD     HL,DE
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        PUSH    DE
        CP      $42
        JR      Z,L1751         ; forward to CL-RS-CH

        CP      $54
        JR      NZ,L175E        ; forward to CL-N-CH


;; CL-RS-CH
L1751:  BIT     1,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L177F        ; forward to RCLM-CH

        LD      A,$0D
        CALL    L0C5A           ; routine BCHAN-OUT
        JR      L177F           ; forward to RCLM-CH


;; CL-N-CH
```

```
L175E:  CP      $4E
        JR      NZ,L176D        ; forward to CL-M-CN

        BIT     1,(IY+$7C)      ; sv FLAGS_3
        JR      NZ,L177F        ; forward to RCLM-CH

        CALL    L0EF5           ; routine SEND-NEOF
        JR      L177F           ; forward to RCLM-CH


;; CL-M-CN
L176D:  CP      $4D
        JR      NZ,L177F        ; forward to RCLM-CH

        POP     DE
        POP     IX
        POP     DE
        BIT     1,(IY+$7C)      ; sv FLAGS_3
        JP      Z,L12A9         ; jump to CLOSE-M2
        JP      L10C4           ; jump to DEL-M-BUF

;; RCLM-CH
L177F:  POP     BC
        POP     HL
        PUSH    BC
        RST     10H             ; CALBAS
        DEFW    $19E8           ; main RECLAIM-2
        XOR     A
        LD      HL,$5C16        ; sv STRMS_00

;; UPD-STRM
L1789:  LD      E,(HL)
        INC     HL
        LD      D,(HL)
        DEC     HL
        LD      ($5C5F),HL      ; sv X_PTR
        POP     BC
        POP     HL
        PUSH    HL
        PUSH    BC
        AND     A
        SBC     HL,DE
        JR      NC,L17A4        ; forward to UPD-NXT-S

        EX      DE,HL
        AND     A
        SBC     HL,BC
        EX      DE,HL
        LD      HL,($5C5F)      ; sv X_PTR
        LD      (HL),E
        INC     HL
        LD      (HL),D

;; UPD-NXT-S
L17A4:  LD      HL,($5C5F)      ; sv X_PTR
        INC     HL
        INC     HL
        INC     A
        CP      $10
        JR      C,L1789         ; back to UPD-STRM

        LD      (IY+$26),$00    ; sv X_PTR_hi
        POP     HL
        POP     HL
```

```
        RES     1,(IY+$7C)      ; sv FLAGS_3
        RET


; ---------------------------------------
; THE 'RECLAIM TEMPORARY CHANNELS' ROUTINE
; ---------------------------------------
;

;; RCL-T-CH
L17B9:  LD      IX,($5C4F)      ; sv CHANS
        LD      DE,$0014
        ADD     IX,DE

;; EX-CHANS
L17C2:  LD      A,(IX+$00)
        CP      $80
        JR      NZ,L17D2        ; forward to CHK-TEMPM

        LD      A,$EE
        OUT     ($EF),A
        XOR     A
        JP      L17F7           ; jump to SEL-DRIVE

; ---

        RET


;; CHK-TEMPM
L17D2:  LD      A,(IX+$04)
        CP      $CD
        JR      NZ,L17DE        ; forward to CHK-TEMPN

        CALL    L10C4           ; routine DEL-M-BUF
        JR      L17B9           ; back to RCL-T-CH


;; CHK-TEMPN
L17DE:  CP      $CE
        JR      NZ,L17ED        ; forward to PT-N-CHAN

        LD      BC,$0114
        PUSH    IX
        POP     HL
        RST     10H             ; CALBAS
        DEFW    $19E8           ; main RECLAIM-2
        JR      L17B9           ; back to RCL-T-CH


;; PT-N-CHAN
L17ED:  LD      E,(IX+$09)
        LD      D,(IX+$0A)
        ADD     IX,DE
        JR      L17C2           ; back to EX-CHANS


; -------------------------------
; THE 'SELECT DRIVE MOTOR' ROUTINE
; -------------------------------
;

;; SEL-DRIVE
L17F7:  PUSH    HL
```

```
        CP      $00
        JR      NZ,L1802        ; forward to TURN-ON

        CALL    L182A           ; routine SW-MOTOR
        EI
        POP     HL
        RET


;; TURN-ON
L1802:  DI
        CALL    L182A           ; routine SW-MOTOR
        LD      HL,$1388

;; TON-DELAY
L1809:  DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L1809        ; back to TON-DELAY

        LD      HL,$1388

;; REPTEST
L1811:  LD      B,$06

;; CHK-PRES
L1813:  CALL    L18E9           ; routine TEST-BRK
        IN      A,($EF)
        AND     $04
        JR      NZ,L1820        ; forward to NOPRES

        DJNZ    L1813           ; back to CHK-PRES

        POP     HL
        RET


;; NOPRES
L1820:  DEC     HL
        LD      A,H
        OR      L
        JR      NZ,L1811        ; back to REPTEST

        CALL    L17F7           ; routine SEL-DRIVE
        RST     20H             ; sh_err
        DEFB    $10


;; SW-MOTOR
L182A:  PUSH    DE
        LD      DE,$0100
        NEG
        ADD     A,$09
        LD      C,A
        LD      B,$08

;; ALL-MOTRS
L1835:  DEC     C
        JR      NZ,L184B        ; forward to OFF-MOTOR

        LD      A,D
        OUT     ($F7),A
        LD      A,$EE
        OUT     ($EF),A
```

```
        CALL    L1867           ; routine DEL-S-1
        LD      A,$EC
        OUT     ($EF),A
        CALL    L1867           ; routine DEL-S-1
        JR      L185C           ; forward to NXT-MOTOR


;; OFF-MOTOR
L184B:  LD      A,$EF
        OUT     ($EF),A
        LD      A,E
        OUT     ($F7),A
        CALL    L1867           ; routine DEL-S-1
        LD      A,$ED
        OUT     ($EF),A
        CALL    L1867           ; routine DEL-S-1

;; NXT-MOTOR
L185C:  DJNZ    L1835           ; back to ALL-MOTRS

        LD      A,D
        OUT     ($F7),A
        LD      A,$EE
        OUT     ($EF),A
        POP     DE
        RET


; ---------------------------------
; THE '1 MILLISECOND DELAY' ROUTINE
; ---------------------------------
;

;; DEL-S-1
L1867:  PUSH    BC
        PUSH    AF
        LD      BC,$0087
        CALL    L18FA           ; routine DELAY-BC
        POP     AF
        POP     BC
        RET


; -----------------------------------------------
; THE 'SEND DATA BLOCK TO MICRODRIVE HEAD' ROUTINE
; -----------------------------------------------
;

;; OUT-M-HD
L1872:  PUSH    HL
        LD      DE,$001E
        JR      L187C           ; forward to OUT-M-BLK


;; OUT-H-BUF
L1878:  PUSH    HL
        LD      DE,$021F

;; OUT-M-BLK
L187C:  IN      A,($EF)
        AND     $01
        JR      NZ,L1884        ; forward to NOT-PROT

        RST     20H             ; sh_err
```

```
        DEFB    $0E


;; NOT-PROT
L1884   LD      A,($5CC6)       ; sv IOBORD
        OUT     ($FE),A
        LD      A,$E2
        OUT     ($EF),A
        INC     D
        LD      A,D
        LD      B,E
        LD      C,$E7
        NOP
        NOP
        NOP

;; OUT-M-BYT
L1895:  OTIR
        DEC     A
        JR      NZ,L1895        ; back to OUT-M-BYT

        LD      A,$E6
        OUT     ($EF),A
        CALL    L0CA9           ; routine BORD-REST
        POP     HL
        RET


; -----------------------------------------------
; THE 'RECEIVE BLOCK FROM MICRODRIVE HEAD' ROUTINE
; -----------------------------------------------
;

;; GET-M-HD
L18A3:  PUSH    HL
        LD      DE,$000F
        JR      L18AD           ; forward to GET-M-BLK


;; GET-M-BUF
L18A9:  PUSH    HL
        LD      DE,$0210

;; GET-M-BLK
L18AD:  LD      B,E
        LD      C,D
        INC     C
        PUSH    BC

;; CHK-AGAIN
L18B1:  LD      B,$08

;; CHKLOOP
L18B3:  CALL    L18E9           ; routine TEST-BRK
        IN      A,($EF)
        AND     $04
        JR      Z,L18B1         ; back to CHK-AGAIN

        DJNZ    L18B3           ; back to CHKLOOP


;; CHK-AC-2
L18BE:  LD      B,$06
```

```
;; CHK-LP-2
L18C0:  CALL    L18E9           ; routine TEST-BRK
        IN      A,($EF)
        AND     $04
        JR      NZ,L18BE        ; back to CHK-AC-2

        DJNZ    L18C0           ; back to CHK-LP-2

        POP     BC
        LD      A,$EE
        OUT     ($EF),A
        POP     HL
        PUSH    HL

;; DR-READY
L18D2:  IN      A,($EF)
        AND     $02
        JR      NZ,L18D2        ; back to DR-READY

        CALL    L18E9           ; routine TEST-BRK
        LD      A,C
        LD      C,$E7

;; IN-M-BLK
L18DE:  INIR
        DEC     A
        JR      NZ,L18DE        ; back to IN-M-BLK

        LD      A,$EE
        OUT     ($EF),A
        POP     HL
        RET


; ----------------------
; THE 'TEST-BRK' ROUTINE
; ----------------------
;

;; TEST-BRK
L18E9:  LD      A,$7F
        IN      A,($FE)
        RRA
        RET     C

        LD      A,$FE
        IN      A,($FE)
        RRA
        RET     C

        LD      (IY+$00),$14    ; sv ERR_NR
        RST     28H             ; romerr

; ----------------------
; THE 'DELAY-BC' ROUTINE
; ----------------------
;

;; DELAY-BC
L18FA:  PUSH    AF

;; DELAY-BC1
L18FB:  DEC     BC
        LD      A,B
```

```
        OR      C
        JR      NZ,L18FB        ; back to DELAY-BC1

        POP     AF
        RET


; -----------------------------------------------
; THE '32-BIT CYCLICAL REDUNDANCY CHECKSUM' ROUTINE
; -----------------------------------------------
;   This routine calculates and then checks and inserts a CRC-32 checksum
;   in the four bytes following the 512 bytes of data.  There is only one
;   byte allocated for the checksum in production models and this routine
;   was removed from the second Interface 1 ROM.

;; CRC-32
L1902:  PUSH    HL
        PUSH    IX

        POP     HL
        LD      BC,$0052
        ADD     HL,BC
        LD      B,H             ; BC=&CHDATA
        LD      C,L
        LD      HL,L0000        ; HL=0
        LD      DE,L0000        ; DE=0
        EXX
        LD      BC,$0200        ; BC'=512
        LD      HL,L0000        ; HL'=0
        LD      DE,L0000        ; DE'=0

;; CRC-32a
L191C:  EXX
        LD      A,(BC)          ; Get CHDATA byte
        INC     BC              ; point to next byte
        ADD     A,E             ; Accumulate in E
        LD      E,A
        JR      NC,L1929        ; forward to CRC-32b

        INC     D               ; overflow into D
        JR      NZ,L1929        ; forward to CRC-32b

        EXX
        INC     DE              ; overflow into DE'
        EXX

;; CRC-32b
L1929:  ADD     HL,DE           ; accumulate DED'E' in HLH'L'
        EXX
        ADC     HL,DE
        DEC     BC              ; count down
        LD      A,B
        OR      C
        JR      NZ,L191C        ; back to CRC-32a

        LD      D,E             ; bits 0-7 move to 8-15
        EXX
        LD      A,D             ; copy to A
        LD      E,$00           ; clear bits 0-7
        SLA     D               ; move 8-14 to 9-15, 15 to cy
        EXX
        LD      E,A             ; 8-15 to 0-7
        RL      E               ; cy to 0 0-6 to 1-7,7 to cy
        RL      D               ; cy to 8 8-14 to 9-15,15 to cy
```

```
        EXX
        ADD     HL,DE           ; accumulate 0-15 in HL
        EXX
        ADC     HL,DE           ; accumulate 16-31 in H'L'
        PUSH    HL              ; save CRC 16-31
        EXX

        PUSH    HL              ; swap CRC 0-15 w/ CHDATA
        PUSH    BC
        POP     HL
        POP     BC

        LD      E,$00           ; say data 'Ok'
        LD      A,C
        CP      (HL)            ; test CRC-ll on data
        JR      Z,L1952         ; forward to CRC-32c

        INC     E               ; say data 'corrupted'
        LD      (HL),A          ; set correct CRC-ll

;; CRC-32c
L1952:  INC     HL              ; point to high byte
        LD      A,B
        CP      (HL)            ; test CRC-lh on data
        JR      Z,L1959         ; forward to CRC-32d

        INC     E               ; say data 'corrupted'
        LD      (HL),A          ; set CRC-lh

;; CRC-32d
;; UNKN-5
L1959:  INC     HL
        POP     BC              ; pop CRC 15-31
        LD      A,C
        CP      (HL)            ; test CRC-hl on data
        JR      Z,L1961         ; forward to CRC-32e

        INC     E               ; say data 'corrupted'
        LD      (HL),A          ; set CRC-hl

;; CRC-32e
L1961:  INC     HL
        LD      A,B
        CP      (HL)            ; test CRC-hh on data
        JR      Z,L1968         ; forward to CRC-32f

        INC     E               ; say data 'corrupted'
        LD      (HL),A          ; set CRC-hh

;; CRC-32f
L1968:  LD      A,E             ;set Z Flag when data OK
        OR      A
        POP     HL
        RET                     ; return.


; ----------------------------------------
; THE 'ENCRYPT/DECRYPT CHANNEL DATA' ROUTINE
; ----------------------------------------
;   This subroutine encrypts the 512 bytes of the microdrive buffer on the
;   first call and decrypyts the contents if they are already encrytped.

;; ENCR-CHDAT
L196C:  PUSH    IX
```

```
        POP     HL

        LD      DE,$0052        ; CHDATA
        ADD     HL,DE           ; set hl to ix+CHDATA
        LD      BC,$0200        ; 512 bytes

;; ENCR-CHD1
L1976:  LD      A,(HL)          ; get a byte

        XOR     $55             ; smash some bits

        LD      (HL),A          ; set the byte
        INC     HL              ; next byte
        DEC     BC              ; count down
        LD      A,B             ; test for BC=0
        OR      C
        JR      NZ,L1976        ; back to ENCR-CHD1

        RET                     ; return.


; -----------------------
; THE 'HOOK-CODE' ROUTINE
; -----------------------
;

;; HOOK-CODE
L1981:  CP      $18
        JR      C,L1987         ; forward to CLR-ERR

        RST     20H             ; sh_err
        DEFB    $12

;; CLR-ERR
L1987:  LD      (IY+$00),$FF    ; sv ERR_NR
        SET     2,(IY+$01)      ; sv FLAGS
        INC     HL
        EX      (SP),HL
        PUSH    HL
        ADD     A,A
        LD      D,$00
        LD      E,A
        LD      HL,$19A9
        ADD     HL,DE
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        POP     AF
        LD      HL,L0700
        PUSH    HL
        EX      DE,HL
        JP      (HL)


; ---------------------------
; THE 'HOOK CODE +32' ROUTINE
; ---------------------------
;

;; HOOK-32
L19A4:  LD      HL,($5CED)      ; sv HD_11
        JP      (HL)
```

```
; ---------------------------
; THE 'HOOK CODE +31' ROUTINE
; ---------------------------
;

;; HOOK-31
L19A8:  RET


; ---------------------------------
; THE 'HOOK CODE ADDRESSES' ROUTINE
; ---------------------------------
;

;;
L19A9:  DEFW    L19D9           ; CONS-IN
        DEFW    L19EC           ;
        DEFW    L0B81           ;
        DEFW    L0C5A           ;
        DEFW    L19FC           ;
        DEFW    L1A01           ;
        DEFW    L17F7           ;
        DEFW    L1B29           ;
        DEFW    L12A9           ;
        DEFW    L1D6E           ;
        DEFW    L1A09           ;
        DEFW    L11FF           ;
        DEFW    L1A17           ;
        DEFW    L1A4B           ;
        DEFW    L1A86           ;
        DEFW    L1A91           ;
        DEFW    L1B29           ;
        DEFW    L10C4           ;
        DEFW    L0EA9           ;
        DEFW    L1A24           ;
        DEFW    L1A31           ;
        DEFW    L0DB2           ;
        DEFW    L19A8           ;
        DEFW    L19A4           ;


; ---------------------------
; THE 'CONSOLE INPUT' ROUTINE
; ---------------------------
;

;; CONS-IN
L19D9:  EI
        RES     5,(IY+$01)      ; sv FLAGS

;; WTKEY
L19DE:  HALT
        RST     10H             ; CALBAS
        DEFW    $02BF           ; main KEYBOARD
        BIT     5,(IY+$01)      ; sv FLAGS
        JR      Z,L19DE         ; back to WTKEY

        LD      A,($5C08)       ; sv LASTK
        RET


; ---------------------------
; THE 'CONSOLE OUTPUT' ROUTINE
; ---------------------------
```

```
;

;; CONS-OUT
L19EC:  PUSH    AF
        LD      A,$FE

;; OUT-CODE
L19EF:  LD      HL,$5C8C        ; sv SCR_CT
        LD      (HL),$FF
        RST     10H             ; CALBAS
        DEFW    $1601           ; main CHAN-OPEN
        POP     AF
        RST     10H             ; CALBAS
        DEFW    $0010           ; main PRINT-A
        RET


; ----------------------------
; THE 'PRINTER OUTPUT' ROUTINE
; ----------------------------
;

;; PRT-OUT
L19FC:  PUSH    AF
        LD      A,$03
        JR      L19EF           ; back to OUT-CODE


; ----------------------------
; THE 'KEYBOARD TEST' ROUTINE
; ----------------------------
;

;; KBD-TEST
L1A01:  XOR     A
        IN      A,($FE)
        AND     $1F
        SUB     $1F
        RET


; -----------------------------
; THE 'READ SEQUENTIAL' ROUTINE
; -----------------------------
;

;; READ-SEQ
L1A09:  BIT     1,(IX+$43)
        JR      Z,L1A14         ; forward to INCREC

        LD      (IY+$00),$07    ; sv ERR_NR
        RST     28H             ; romerr

;; INCREC
L1A14:  INC     (IX+$0D)

; -------------------------
; THE 'READ RANDOM' ROUTINE
; -------------------------
;

;; RD-RANDOM
L1A17:  CALL    L1177           ; routine GET-RECD
        BIT     2,(IX+$43)
```

```
        RET     Z

        CALL    L10C4           ; routine DEL-M-BUF
        RST     20H             ; sh_err
        DEFB    $16


; -----------------------------------
; THE 'CLOSE NETWORK CHANNEL' ROUTINE
; -----------------------------------
;

;; CLOSE-NET
L1A24:  CALL    $0EF5
        PUSH    IX
        POP     HL
        LD      BC,$0114
        RST     10H             ; CALBAS
        DEFW    $19E8           ; main RECLAIM-2
        RET


; -------------------------------------
; THE 'GET PACKET FROM NETWORK' ROUTINE
; -------------------------------------
;

;; GET-PACK
L1A31:  LD      A,($5CC6)       ; sv IOBORD
        OUT     ($FE),A
        DI
        CALL    L0F1E           ; routine WT-SCOUT
        JR      NC,L1A46        ; forward to GP-ERROR

        CALL    L0E18           ; routine GET-NBLK
        JR      NZ,L1A46        ; forward to GP-ERROR

        EI
        AND     A
        JP      L0CA9           ; jump to BORD-REST

;; GP-ERROR
L1A46:  SCF
        EI
        JP      L0CA9           ; jump to BORD-REST

; -------------------------
; THE 'READ SECTOR' ROUTINE
; -------------------------
;

;; RD-SECTOR
L1A4B:  LD      HL,$00F0        ; counts 240 sectors.
        LD      ($5CC9),HL      ; sv SECTOR

;; NO-GOOD
L1A51:  CALL    L12C4           ; routine GET-M-RD2
        LD      A,(IX+$29)
        CP      (IX+$0D)
        JR      Z,L1A63         ; forward to USE-C-RC

        CALL    L1312           ; routine DEC-SECT
        JR      NZ,L1A51        ; back to NO-GOOD
```

```
        RST     20H             ; sh_err
        DEFB    $11

;; USE-C-RC
L1A63:  PUSH    IX
        POP     HL
        LD      DE,$0043
        ADD     HL,DE
        CALL    L18A9           ; routine GET-M-BUF
        CALL    L1341           ; routine CHKS-HD-R
        JR      NZ,L1A81        ; forward to DEL-B-CT

        LD      DE,$000F
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF
        JR      NZ,L1A81        ; forward to DEL-B-CT

        OR      A
        BIT     2,(IX+$43)
        RET     Z


;; DEL-B-CT
L1A81:  CALL    L1AE0           ; routine CLR-BUFF
        SCF
        RET


; ------------------------------
; THE 'READ NEXT SECTOR' ROUTINE
; ------------------------------
;

;; RD-NEXT
L1A86:  LD      HL,$00F0        ; counts 240 sectors.
        LD      ($5CC9),HL      ; sv SECTOR
        CALL    L12C4           ; routine GET-M-RD2
        JR      L1A63           ; back to USE-C-RC


; --------------------------
; THE 'WRITE SECTOR' ROUTINE
; --------------------------
;

;; WR-SECTOR
L1A91:  LD      HL,$00F0        ; counts 240 sectors.
        LD      ($5CC9),HL      ; sv SECTOR
        PUSH    IX
        POP     HL
        LD      DE,$0037
        ADD     HL,DE
        PUSH    HL
        LD      DE,$000C
        ADD     HL,DE
        CALL    L1341           ; routine CHKS-HD-R
        LD      DE,$000F
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF

;; WR-S-1
L1AAD:  CALL    L12C4           ; routine GET-M-RD2
        LD      A,(IX+$29)
        CP      (IX+$0D)
```

```
        JR      Z,L1ABF         ; forward to WR-S-2

        CALL    L1312           ; routine DEC-SECT
        JR      NZ,L1AAD        ; back to WR-S-1

        RST     20H             ; sh_err
        DEFB    $11


;; WR-S-2
L1ABF:  IN      A,($EF)
        AND     $01
        JR      NZ,L1AC7        ; forward to WR-S-3

        RST     20H             ; sh_err
        DEFB    $0E


;; WR-S-3
L1AC7:  LD      A,$E6
        OUT     ($EF),A
        LD      BC,$0168
        CALL    L18FA           ; routine DELAY-BC
        POP     HL
        CALL    L1878           ; routine OUT-H-BUF
        LD      A,$EE
        OUT     ($EF),A
        CALL    L12DF           ; routine CHECK-MAP
        LD      A,B
        OR      (HL)
        LD      (HL),A
        RET


; -----------------------------------
; THE 'CLEAR BUFFER CONTENTS' ROUTINE
; -----------------------------------
;

;; CLR-BUFF
L1AE0:  PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      D,H
        LD      E,L
        INC     DE
        LD      BC,$01FF
        LDIR
        RET


; ----------------------------------------
; THE 'OPEN A PERMANENT "M" CHANNEL' ROUTINE
; ----------------------------------------
;

;; OP-M-STRM
L1AF0:  LD      A,($5CD8)       ; sv D_STR1
        ADD     A,A
        LD      HL,$5C16        ; sv STRMS_00
        LD      E,A
        LD      D,$00
        ADD     HL,DE
```

```
         PUSH    HL
         CALL    L1B29           ; routine OP-TEMP-M
         BIT     0,(IX+$18)
         JR      Z,L1B0D         ; forward to MAKE-PERM

         IN      A,($EF)
         AND     $01
         JR      NZ,L1B0D        ; forward to MAKE-PERM

         RST     20H             ; sh_err
         DEFB    $0E


;; MAKE-PERM
L1B0D:   RES     7,(IX+$04)
         XOR     A
         CALL    L17F7           ; routine SEL-DRIVE
         BIT     0,(IX+$18)
         JR      NZ,L1B23        ; forward to STORE-DSP

         BIT     2,(IX+$43)
         JR      Z,L1B23         ; forward to STORE-DSP

         RST     20H             ; sh_err
         DEFB    $16


;; STORE-DSP
L1B23:   EX      DE,HL
         POP     HL
         LD      (HL),E
         INC     HL
         LD      (HL),D
         RET


; -----------------------------------------
; THE 'OPEN A TEMPORARY "M" CHANNEL' ROUTINE
; -----------------------------------------
;

;; OP-TEMP-M
L1B29:   CALL    L0FE8           ; routine SET-T-MCH
         PUSH    HL
         LD      A,(IX+$19)
         CALL    L17F7           ; routine SEL-DRIVE
         LD      BC,$00FF
         LD      ($5CC9),BC      ; sv SECTOR

;; OP-F-1
L1B3A:   CALL    L11A5           ; routine G-RD-RC
         JR      C,L1B5F         ; forward to OP-P-4

         JR      Z,L1B5C         ; forward to OP-F-3

         RES     0,(IX+$18)
         LD      A,(IX+$44)
         OR      A
         JR      NZ,L1B57        ; forward to OP-F-2

         PUSH    IX
         POP     HL
         LD      DE,$0052
         ADD     HL,DE
```

```
        CALL    L1346           ; routine CHKS-BUFF
        JR      Z,L1B6C         ; forward to OP-F-5


;; OP-F-2
L1B57:  CALL    L117D           ; routine GET-R-2
        JR      L1B6C           ; forward to OP-F-5


;; OP-F-3
L1B5C:  CALL    L12FE           ; routine RES-B-HAP

;; OP-P-4
L1B5F:  CALL    L1312           ; routine DEC-SECT
        JR      NZ,L1B3A        ; back to OP-F-1

        RES     1,(IX+$43)
        RES     2,(IX+$43)

;; OP-F-5
L1B6C:  POP     HL
        RET


; --------------------------------
; THE 'FORMAT "M" COMMAND' ROUTINE
; --------------------------------
;

;; FORMAT
L1B6E:  CALL    L0FE8           ; routine SET-T-MCH
        LD      A,(IX+$19)
        CALL    L182A           ; routine SW-MOTOR
        LD      BC,$32C8
        CALL    L18FA           ; routine DELAY-BC
        DI
        IN      A,($EF)
        AND     $01
        JR      NZ,L1B86        ; forward to FORMAT-1

        RST     20H             ; sh_err
        DEFB    $0E


;; FORMAT-1
L1B86   LD      A,$E6
        OUT     ($EF),A
        LD      BC,$00FF
        LD      ($5CC9),BC      ; sv SECTOR
        PUSH    IX
        POP     HL
        LD      DE,$002C
        ADD     HL,DE
        EX      DE,HL
        LD      HL,$FFE2
        ADD     HL,DE
        LD      BC,$000A
        LDIR
        XOR     A
        LD      (IX+$47),A
        SET     0,(IX+$28)
        RES     0,(IX+$43)
        SET     1,(IX+$43)
        PUSH    IX
```

```
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      B,$00
        LD      A,$FC

;; FILL-B-F
L1BBD:  LD      (HL),A
        INC     HL
        DJNZ    L1BBD           ; back to FILL-B-F


;; FILL-B-F2
L1BC1:  LD      (HL),A
        INC     HL
        DJNZ    L1BC1           ; back to FILL-B-F2

        PUSH    IX
        POP     DE
        LD      HL,$0043
        ADD     HL,DE
        CALL    L1341           ; routine CHKS-HD-R
        LD      DE,$000F
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF

;; WR-F-TEST
L1BD6:  CALL    L1312           ; routine DEC-SECT
        JR      Z,L1C0A         ; forward to TEST-SCT

        LD      (IX+$29),C
        PUSH    IX
        POP     HL
        LD      DE,L0028
        ADD     HL,DE
        CALL    L1341           ; routine CHKS-HD-R
        LD      DE,$FFF4
        ADD     HL,DE
        CALL    L1872           ; routine OUT-M-HD
        LD      BC,$01B2
        CALL    L18FA           ; routine DELAY-BC
        PUSH    IX
        POP     HL
        LD      DE,$0037
        ADD     HL,DE
        CALL    L1878           ; routine OUT-H-BUF
        LD      BC,$033F
        CALL    L18FA           ; routine DELAY-BC
        CALL    L18E9           ; routine TEST-BRK
        JR      L1BD6           ; back to WR-F-TEST


;; TEST-SCT
L1C0A:  LD      A,$EE
        OUT     ($EF),A
        LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE
        LD      BC,$00FF
        LD      ($5CC9),BC      ; sv SECTOR

;; CHK-SCT
L1C1B:  CALL    L12C4           ; routine GET-M-RD2
        CALL    L12DF           ; routine CHECK-MAP
        JR      Z,L1C3E         ; forward to CHK-NSECT
```

```
        PUSH    IX
        POP     HL
        LD      DE,$0043
        ADD     HL,DE
        CALL    L18A9           ; routine GET-M-BUF
        CALL    L1341           ; routine CHKS-HD-R
        JR      NZ,L1C3E        ; forward to CHK-NSECT

        LD      DE,$000F
        ADD     HL,DE
        CALL    L1346           ; routine CHKS-BUFF
        JR      NZ,L1C3E        ; forward to CHK-NSECT

        CALL    L12FE           ; routine RES-B-HAP

;; CHK-NSECT
L1C3E:  CALL    L1312           ; routine DEC-SECT
        JR      NZ,L1C1B        ; back to CHK-SCT

        CALL    L1E3E           ; routine IN-CHK

;; MARK-FREE
L1C46:  CALL    L1264           ; routine CHK-FULL
        JR      NZ,L1C53        ; forward to MK-BLK

        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        CALL    L10C4           ; routine DEL-M-BUF
        RET


;; MK-BLK
L1C53:  CALL    L1275           ; routine SEND-BLK
        JR      L1C46           ; back to MARK-FREE


; -------------------------
; THE 'CAT COMMAND' ROUTINE
; -------------------------
;

;; CAT
L1C58:  LD      A,($5CD8)       ; sv D_STR1
        RST     10H             ; CALBAS
        DEFW    $1601           ; main CHAN-OPEN
        CALL    $0FE8
        LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE
        LD      BC,$00FF
        LD      ($5CC9),BC      ; sv SECTOR

;; CAT-LP
L1C6E:  CALL    L12C4           ; routine GET-M-RD2
        CALL    L1E53           ; routine G-RDES
        JR      NZ,L1C6E        ; back to CAT-LP

        LD      A,(IX+$43)
        OR      (IX+$46)
        AND     $02
        JR      NZ,L1C85        ; forward to IN-NAME

        CALL    L12FE           ; routine RES-B-HAP
        JR      L1CEE           ; forward to F-N-SCT
```

```
;; IN-NAME
L1C85:  LD      A,(IX+$47)
        OR      A
        JR      Z,L1CEE         ; forward to F-N-SCT

        PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      DE,$000A
        LD      B,$00
        LD      C,(IX+$0D)

;; SE-NAME
L1C9A:  LD      A,C
        OR      A
        JR      Z,L1CD4         ; forward to INS-NAME

        PUSH    HL
        PUSH    IX
        PUSH    BC
        LD      B,$0A

;; T-MA-1
L1CA4:  LD      A,(HL)
        CP      (IX+$47)
        JR      NZ,L1CAF        ; forward to T-NA-2

        INC     HL
        INC     IX
        DJNZ    L1CA4           ; back to T-MA-1


;; T-NA-2
L1CAF:  POP     BC
        POP     IX
        POP     HL
        JR      Z,L1CEE         ; forward to F-N-SCT

        JR      NC,L1CBB        ; forward to ORD-NAM

        ADD     HL,DE
        DEC     C
        JR      L1C9A           ; back to SE-NAME


;; ORD-NAM
L1CBB:  PUSH    HL
        PUSH    DE
        PUSH    BC
        PUSH    HL
        SLA     C
        LD      H,B
        LD      L,C
        ADD     HL,BC
        ADD     HL,BC
        ADD     HL,BC
        ADD     HL,BC
        LD      B,H
        LD      C,L
        POP     HL
        DEC     HL
```

```
        ADD     HL,BC
        EX      DE,HL
        ADD     HL,DE
        EX      DE,HL
        LDDR
        POP     BC
        POP     DE
        POP     HL

;; INS-NAME
L1CD4:  PUSH    IX
        LD      B,$0A

;; MOVE-NA
L1CD8:  LD      A,(IX+$47)
        LD      (HL),A
        INC     IX
        INC     HL
        DJNZ    L1CD8           ; back to MOVE-NA

        POP     IX
        LD      A,(IX+$0D)
        INC     A
        LD      (IX+$0D),A
        CP      $32
        JR      Z,L1CF4         ; forward to BF-FILLED


;; F-N-SCT
L1CEE:  CALL    L1312           ; routine DEC-SECT
        JP      NZ,L1C6E        ; jump to CAT-LP

;; BF-FILLED
L1CF4:  PUSH    IX
        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        PUSH    IX
        POP     HL
        LD      DE,$002C
        ADD     HL,DE
        CALL    L1D50           ; routine PRNAME
        LD      A,$0D
        CALL    L1D66           ; routine PRCHAR
        PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      B,(IX+$0D)
        LD      A,B
        OR      A
        JR      Z,L1D1C         ; forward to NONAMES


;; OT-NAMS
L1D17:  CALL    L1D50           ; routine PRNAME
        DJNZ    L1D17           ; back to OT-NAMS


;; NONAMES
L1D1C:  CALL    L1D38           ; routine FREESECT
        LD      A,E
        SRL     A
        RST     10H             ; CALBAS
        DEFW    $2D28           ; main STACK-A
```

```
        LD      A,$0D
        CALL    L1D66           ; routine PRCHAR
        RST     10H             ; CALBAS
        DEFW    $2DE3           ; main PRINT-FP
        LD      A,$0D
        CALL    L1D66           ; routine PRCHAR
        POP     IX
        CALL    L10C4           ; routine DEL-M-BUF
        RET


; ----------------------
; THE 'FREESECT' ROUTINE
; ----------------------
;

;; FREESECT
L1D38:  LD      L,(IX+$1A)
        LD      H,(IX+$1B)
        LD      E,$00
        LD      C,$20

;; FR-SC-LP
L1D42:  LD      A,(HL)
        INC     HL
        LD      B,$08

;; FR-S-LPB
L1D46:  RRA
        JR      C,L1D4A         ; forward to FR-S-RES

        INC     E

;; FR-S-RES
L1D4A:  DJNZ    L1D46           ; back to FR-S-LPB

        DEC     C
        JR      NZ,L1D42        ; back to FR-SC-LP

        RET


; --------------------
; THE 'PRNAME' ROUTINE
; --------------------
;

;; PRNAME
L1D50:  PUSH    BC
        LD      B,$0A

;; PRNM-LP
L1D53:  LD      A,(HL)
        CALL    L1D66           ; routine PRCHAR
        INC     HL
        DJNZ    L1D53           ; back to PRNM-LP

        LD      A,$0D
        CALL    L1D66           ; routine PRCHAR
        PUSH    HL
        RST     10H             ; CALBAS
        DEFW    $0D4D           ; main TEMPS
        POP     HL
        POP     BC
```

```
        RET


; --------------------
; THE 'PRCHAR' ROUTINE
; --------------------
;

;; PRCHAR
L1D66:  PUSH    IX
        RST     10H             ; CALBAS
        DEFW    $0010           ; main PRINT-A
        POP     IX
        RET


; ---------------------------
; THE 'ERASE COMMAND' ROUTINE
; ---------------------------
;

;; ERASE
L1D6E:  CALL    L0FE8           ; routine SET-T-MCH
        LD      A,(IX+$19)
        CALL    L17F7           ; routine SEL-DRIVE
        IN      A,($EF)
        AND     $01
        JR      NZ,L1D7F        ; forward to ERASE-1

        RST     20H             ; sh_err
        DEFB    $0E


;; ERASE-1
L1D7F   PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        PUSH    HL
        POP     DE
        INC     DE
        LD      BC,$001F
        XOR     A
        LD      (HL),A
        LDIR
        LD      A,$FF
        LD      (IX+$0D),A
        LD      BC,$04FB
        LD      ($5CC9),BC      ; sv SECTOR

;; ERASE-LP
L1D9C:  CALL    L1312           ; routine DEC-SECT
        JR      Z,L1DF8         ; forward to ERASE-MK

        CALL    L12C4           ; routine GET-M-RD2
        CALL    L1E53           ; routine G-RDES
        JR      NZ,L1DDA        ; forward to TST-NUM

        LD      A,(IX+$43)
        OR      (IX+$46)
        AND     $02
        JR      NZ,L1DB8        ; forward to ERASE-2

        CALL    L12FE           ; routine RES-B-HAP
```

```
        JR      L1DDA           ; forward to TST-NUM


;; ERASE-2
L1DB8:  PUSH    IX
        POP     HL
        LD      DE,$0047
        ADD     HL,DE
        LD      BC,$000A
        CALL    L131E           ; routine CHK-NAME
        JR      NZ,L1DDA        ; forward to TST-NUM

        CALL    L1306           ; routine TEST-PHAP
        LD      A,B
        OR      (HL)
        LD      (HL),A
        BIT     1,(IX+$43)
        JR      Z,L1DDA         ; forward to TST-NUM

        LD      A,(IX+$44)
        INC     A
        LD      (IX+$0D),A

;; TST-NUM
L1DDA:  PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      E,$00
        LD      C,$20

;; LP-P-HAP
L1DE5:  LD      A,(HL)
        INC     HL
        LD      B,$08

;; LP-B-HAP
L1DE9:  RRA
        JR      NC,L1DED        ; forward to NOINC-C

        INC     E

;; NOINC-C
L1DED:  DJNZ    L1DE9           ; back to LP-B-HAP

        DEC     C
        JR      NZ,L1DE5        ; back to LP-P-HAP

        LD      A,(IX+$0D)
        CP      E
        JR      NZ,L1D9C        ; back to ERASE-LP


;; ERASE-MK
L1DF8:  CALL    L1E3E           ; routine IN-CHK

;; ERASE-MK2
L1DFB:  CALL    L12C4           ; routine GET-M-RD2
        CALL    L1306           ; routine TEST-PHAP
        JR      Z,L1E26         ; forward to T-OTHER

        PUSH    HL
        PUSH    BC
        LD      A,$E6
```

```
        OUT     ($EF),A
        LD      BC,$0168
        CALL    L18FA           ; routine DELAY-BC
        PUSH    IX
        POP     HL
        LD      DE,$0037
        ADD     HL,DE
        CALL    L1878           ; routine OUT-H-BUF
        LD      A,$EE
        OUT     ($EF),A
        CALL    L12FE           ; routine RES-B-HAP
        POP     BC
        POP     HL
        LD      A,B
        CPL
        AND     (HL)
        LD      (HL),A

;; T-OTHER
L1E26:  PUSH    IX
        POP     HL
        LD      DE,$0052
        ADD     HL,DE
        LD      B,$20

;; CHK-W-MAP
L1E2F:  LD      A,(HL)
        OR      A
        JR      NZ,L1DFB        ; back to ERASE-MK2

        INC     HL
        DJNZ    L1E2F           ; back to CHK-W-MAP

        XOR     A
        CALL    L17F7           ; routine SEL-DRIVE
        CALL    L10C4           ; routine DEL-M-BUF
        RET


; ---------------------------------
; THE 'SIGNAL 'FREE SECTOR'' ROUTINE
; ---------------------------------
;

;; IN-CHK
L1E3E:  XOR     A
        LD      (IX+$43),A
        LD      (IX+$45),A
        LD      (IX+$46),A
        PUSH    IX
        POP     HL
        LD      DE,$0043
        ADD     HL,DE
        CALL    L1341           ; routine CHKS-HD-R
        RET


; -------------------------------------
; THE 'OBTAIN RECORD DESCRIPTOR' ROUTINE
; -------------------------------------
;

;; G-RDES
L1E53:  PUSH    IX
```

```
        POP     HL
        LD      DE,$0043
        ADD     HL,DE
        CALL    L18A3           ; routine GET-M-HD
        CALL    L1341           ; routine CHKS-HD-R
        RET     NZ

        BIT     0,(IX+$43)
        RET


; -------------------------------------
; THE 'CALLS TO THE COMMAND S' ROUTINE
; -------------------------------------
;

;; ERASE-RUN
L1E66:  CALL    L1D6E           ; routine ERASE
        JR      L1E84           ; forward to ENDC


;; MOVE-RUN
L1E6B:  CALL    L13F1           ; routine MOVE
        JR      L1E84           ; forward to ENDC


;; CAT-RUN
L1E70:  CALL    L1C58           ; routine CAT
        JR      L1E84           ; forward to ENDC


;; IFOR-RUN
L1E75:  CALL    L1B6E           ; routine FORMAT
        JR      L1E84           ; forward to ENDC


;; OP-RUN
L1E7A:  CALL    L1AF0           ; routine OP-M-STRM
        JR      L1E84           ; forward to ENDC


;; SAVE-RUN
L1E7F:  CALL    L14DA           ; routine SA-DRIVE
        JR      L1E84           ; forward to ENDC


;; ENDC
L1E84:  JP      L05C1           ; jump to END1

; ----------------------
; THE 'DISP-HEX' ROUTINE
; ----------------------
;

;; DISP-HEX
L1E87:  PUSH    AF
        RRA
        RRA
        RRA
        RRA
        CALL    L1E90           ; routine DISP-NIB
        POP     AF

;; DISP-NIB
```

```
L1E90:  AND     $0F
        CP      $0A
        JR      C,L1E98         ; forward to CDNV-L

        ADD     A,$07

;; CDNV-L
L1E98:  ADD     A,$30
        CALL    L1EA9           ; routine DISP-CH
        RET


; ----------------------
; THE 'DISP-HEX2' ROUTINE
; ----------------------
;

;; DISP-HEX2
L1E9E:  PUSH    AF
        CALL    L1E87           ; routine DISP-HEX
        LD      A,$20
        CALL    L1EA9           ; routine DISP-CH
        POP     AF
        RET


; ---------------------
; THE 'DISP-CH' ROUTINE
; ---------------------
;

;; DISP-CH
L1EA9:  PUSH    HL
        PUSH    DE
        PUSH    BC
        PUSH    AF
        EXX
        PUSH    HL
        PUSH    DE
        PUSH    BC
        PUSH    AF
        LD      HL,($5C51)      ; sv CURCHL
        PUSH    HL
        PUSH    AF
        LD      A,$02
        RST     10H             ; CALBAS
        DEFW    $1601           ; main CHAN-OPEN
        POP     AF
        RST     10H             ; CALBAS
        DEFW    $0010           ; main PRINT-A
        POP     HL
        LD      ($5C51),HL      ; sv CURCHL
        POP     AF
        POP     BC
        POP     DE
        POP     HL
        EXX
        POP     AF
        POP     BC
        POP     DE
        POP     HL
        RET
```

```
; ----------------------
; THE 'HEX-LINE' ROUTINE
; ----------------------
;

;; HEX-LINE
L1ECE:  PUSH    HL
        PUSH    BC
        PUSH    AF
        LD      B,$0A

;; HEX-LINE2
L1ED3:  LD      A,(HL)
        CALL    L1E9E           ; routine DISP-HEX2
        INC     HL
        DJNZ    L1ED3           ; back to HEX-LINE2

        LD      A,$0D
        CALL    L1EA9           ; routine DISP-CH
        POP     AF
        POP     BC
        POP     HL
        RET

        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
```

```
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
```

```
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
```

```
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
DEFB    $FF
```

```
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF
        DEFB    $FF

;;
LEND    DEFB    $FF

.END

;
```