

Junior

paperware 2

- Source listing van de bootstrap-loader voor Ohio Scientific floppy's
- Hexdump van de EPROM ESS 515



Voor alle 6502-gebruikers die floppy-drives op hun computer willen aansluiten is deze source listing een belangrijk hulpmiddel. De floppy-disk-interface van Elek tuur (november en december 1982), die met goedkope standaard-onderdelen is opgebouwd, werkt met de hier beschreven software.

De object code van de gegeven programma-listing kan in een 2708 EPROM (ESS 515) worden gezet. Het disk operating system is van Ohio Scientific en heeft het typenummer OS-65D V3.1 of OS-65D V3.3. Op deze disketten staan verder een Microsoft BASIC, een assembler, een extended monitor en diverse andere programma's.

De EPROM ESS 515 bevat diverse utility-programma's die via het hex-keyboard of door middel van een JSR-kommando kunnen worden opgeroepen:

<AD> FF17 <GO> (HUBOUT) Laden van het disk operating system van Ohio Scientific en van de Microsoft-BASIC interpreter van de diskette naar de junior computer.

<AD> FF34 <GO> (HUBOUT) Startadres voor toekomstige uitbreidingen.

<AD> FFE2 <GO> (HUBOUT) Starten van een hulpprogramma voor het aanpassen van een OS-65D V3.1-diskette.

<AD> FFE8 <GO> (HUBOUT) Starten van een hulpprogramma voor het aanpassen van een OS-65D V3.3-diskette.

JSR FE43 Print character-subroutine.

JSR FE1B Receive character-subroutine.

-R

DOS

COMMANDS:

RE AS

PUT FN

LOAD FN

SOURCE FILE: 4000 - 4003

REQUIRED TRACK(S): 01

ENTER A FILE NAME: BOOTST

FOR *BOOTST* IS (ARE) 12 TRACK(S) ON FLOPPY

B*LO BOOTST

B*RE A

NEW?

-N1

-X

DOS ASSEMBLER

THE OBJECT MODULE

WILL BE STORED AT \$E000

DO YOU LIKE TO CHANGE IT?Y

ENTER OBJECT START ADDRESS: \$D000

PASS 1

EDITOR?

PASS 2

EDITOR?

LISTING?Y

```

0001:
0002:
0003:
0004: FC00          ORG    $FC00  BOOTSTRAP LOADER
0005:
0006:                SOURCE LISTING OF THE BOOT STRAP LOADER
0007:                FOR ELEKTOR'S DOS JUNIOR COMPUTER OR ANY
0008:                OTHER 6502 SYSTEM
0009:
0010:
0011:                WRITTEN BY GUIDO DE CUYPER AND A. NACHTMANN
0012:
0013:
0014:                DATE:  7 SEPTEMBER  1982
0015:
0016:
0017:
0018:                THIS IS THE SOURCE LISTING OF THE MODIFIED HEX
0019:                DISPLAY MONITOR OF "JUNIOR COMUTER BOOK 2" AND
0020:                A BOOT STRAP LOADER FOR OHIO SCIENTIFIC SOFTWARE.
0021:
0022:                SOME ADDRESSES IN THE HEX DISPLAY MONITOR HAVE
0023:                BEEN CHANGED. THE BOOTSTRAP LOADER HAS GOT
0024:                TWO ENTRIES: A COLD START AND A WARM START ENTRY.
0025:
0026:                THE PIA ON THE FLOPPY INTERFACE PC BOARD IS
0027:                INITIALIZED BY A POWER UP AND CAN'T BE REINITIA-
0028:                LIZED VIA THE RESET LINE!
0029:
0030:
0031:
0032:
0033:
0034:
0035:                ***DISPLAY BUFFERS***
0036:
0037: FC00          INL    *    $00F8
0038: FC00          INH    *    INL    +01
0039: FC00          POINTL *    INH    +01
0040: FC00          POINTH *    POINTL +01
0041:
0042:
0043:                ***TEMPS***
0044:
0045: FC00          TEMP   *    POINTH +01
0046: FC00          MODE   *    TEMP   +01
0047: FC00          NIBBLE *    MODE   +01
0048: FC00          KEY    *    NIBBLE +01
0049:
0050:
0051:                ***6532 I/O AND TIMER***
0052:
0053:
0054:                >PORT A & PORT B
0055:
0056: FC00          PAD    *    $FA80  DATA REGISTER

```

```

0057: FC00      PADD *      PAD   +01 DATA DIRECTION
0058: FC00      PBD  *      PAD   +02 DATA REGISTER
0059: FC00      PBDD *      PAD   +03 DATA DIRECTION
0060:
0061:
0062:          >WRITE EDGE DETECT CONTROL
0063:
    64: FC00      EDETA *      $FAE4 NEG EDET DISABLE PA7 IRQ
0065: FC00      EDSTB *     $FAE5 POS EDET DISABLE PA7 IRQ
0066: FC00      EDETC *     $FAE6 NEG EDET ENABLE PA7 IRQ
0067: FC00      EDETD *     $FAE7 POS EDET ENABLE PA7 IRQ
0068:
0069:          >READ FLAG REGISTER, CLEAR TIMER & IRQ FLAG
0070:
0071: FC00      RDFLAG *    $FAD5 BIT6=PA7 FLAG, BIT7=TIMER FLAG
0072:
0073:
0074:          >WRITE COUNT INTO TIMER, DISABLE TIMER IRQ
0075:
0076: FC00      CNTA *      $FAF4 CLK1T
0077: FC00      CNTB *      $FAF5 CLK8T
0078: FC00      CNTC *      $FAF6 CLK64T
0079: FC00      CNTD *      $FAF7 CLK1KT
0080:
0081:
0082:          >WRITE COUNT INTO TIMER, ENABLE TIMER IRQ
0083:
0084: FC00      CNTE *      $FAFC CLK1T
0085: FC00      CNTE *      $FAFD CLK8T
    86: FC00      CNTG *      $FAFE CLK64T
0087: FC00      CNTH *      $FAFF CLK1KT
0088:
0089:
0090:          ***INTERRUPT VECTORS***
0091:
0092: FC00      NMIL *      $FA7A
0093: FC00      NMIH *      $FA7B
0094: FC00      IRQL *      $FA7E
0095: FC00      IRQH *      $FA7F
0096:
0097:
0098:          ***RAM IN THE 6532: $FA00...$FA7F***
0099:
0100:
0101:
0102:
0103:          ***6522 VIA ON JUNIOR'S INTERFACE BOARD***
0104:
0105:
0106:          > 6522 REGISTER FILE
0107:
    08: FC00      ORB  *      $F800
0109: FC00      ORA  *      $F801 CONTROLS HANDSHAKE
0110: FC00      DDRB *      $F802
0111: FC00      DDRA *      $F803
0112:

```

```

0113: FC00      TALL      *      $F804  WRITE INTO LOW ORDER LATCH OF T1L-L
0114:
0115:
0116:
0117: FC00      TACH      *      $F805  WRITE INTO HIGH ORDER LATCH OF T1
0118:
0119:
    20:
0121:
0122:
0123:
0124: FC00      TAALL     *      $F806  WRITE INTO LOW ORDER LATCH OF T1
0125:
0126:
0127: FC00      TAALH    *      $F807  WRITE INTO HIGH ORDER LATCH OF T1 AND (1)
0128:
0129:
0130:
0131: FC00      TBLCL    *      $F808  WRITE INTO T2L-L
0132:
0133:
0134:
0135: FC00      TBCH     *      $F809  WRITE INTO T2C-H AND (1) TRANSFER T2L-L
0136:
0137:
0138:
0139:
0140: FC00      SR       *      $F80A  SERIAL PORT
0141: FC00      ACR      *      $F80B  AUXILIARY CONTROL REGISTER
    42: FC00      PCR      *      $F80C  PERIPHERAL CONTROL REGISTER
0143: FC00      IFR      *      $F80D  INTERRUPT FLAG REGISTER
0144: FC00      IER      *      $F80E  INTERRUPT ENABLE REGISTER
0145: FC00      ORAA     *      $F80F  OUTPUT REG. A, NO EFFECT ON HANDSHAKE
0146:
0147:
0148:
0149:
0150:
0151:
0152:
0153:
0154:
0155:
0156:
0157:
0158:
0159: FC00      DRA      *      $C000  DATA/DATA DIRECTION REGISTER A
0160: FC00      CRA      *      $C001  CONTROL REGISTER A
0161: FC00      DRB      *      $C002  DATA/DATA DIRECTION REGISTER B
0162: FC00      CRB      *      $C003  CONTROL REGISTER B
    63:
    64:
0165:
0166:
0167: FC00      CACIA    *      $C010  ACIA CONTROL REGISTER
0168: FC00      DACIA    *      $C011  ACIA DATA REGISTER

```

 ADDRESSES OF THE FLOPPY INTERFACE BOARD

PIA ADDRESSES (6821)

ACIA ADDRESSES (6850)

0169:
 0170:
 0171:
 0172:
 0173: FC00
 0174: FC00
 0175: FC00
 0176: FC00
 0177: FC00
 0178: FC00
 0179: FC00
 0180: FC00
 0181: FC00
 0182: FC00
 0183: FC00
 0184:
 0185:
 0186:
 0187:
 0188: FC00
 0189: FC00
 0190: FC00
 0191: FC00
 0192: FC00
 0193:
 0194: FC00
 0195: FC00
 0196: FC00
 0197: FC00
 0198: FC00
 0199:
 0200: FC00
 0201: FC00
 0202: FC00
 0203:
 0204:
 0205:
 0206:
 0207: FC00
 0208: FC00
 0209: FC00
 0210: FC00
 0211:
 0212:
 0213: FC00
 0214: FC00
 0215: FC00
 0216: FC00
 0217: FC00
 0218:
 0219:
 0220:
 0221:
 0222: FC00 A0 1E
 0223: FC07 80 83 FA
 0224: FC05 A9 01

OTHER FLOPPY I/O ADDRESSES

STPBIT * \$FA59 AMOUNT OF STOP BITS
 CNTLX * \$FA5A COUNT DOWN REGISTER
 CNTHX * \$FA5B
 CNTHL * \$FA5C COUNT DOWN REGISTER/2
 CNTHH * \$FA5D
 TIML * \$FA5E CURRENT COUNT REGISTER
 TIMH * \$FA5F
 TEMPA * \$FA60
 TEMPB * \$FA61
 CHA * \$FA62
 BRKT * \$FA7C

KERNEL ADDRESSES

KERNEL * \$2A51 KERNEL COMMAND INTERPRETER
 AHOLD * \$2363 DOS CHARACTER STORE
 MON * \$FC00 SAVE ENTRY TO HEX DISPLAY MONITOR
 LODVEC * \$00E0 LOAD VECTOR FOR THE BOOTSTRAP LOADER
 KPC00 * \$2325 KEY DEPRESSED DURING OUTPUT (CTL-C)
 SYSTEM IS BOOTED UP
 SETTK * \$26BC MOVE THE HEAD ON THE TRACK IN ACCU
 MEMHI * \$00FF FLOPPY LOAD POINTER
 MEMLO * MEMHI -01
 LDHEAD * \$2754 DO A HEAD LOAD
 READDK * \$2967 READ THIS TRACK INTO MEMORY POINTED
 BY THE LOAD POINTER MEMHI, MEMLO
 UNLDHD * \$2761 UNLOAD THE HEAD
 SETDRV * \$2906 SET FOR DRIVE IN ACCU
 STROUT * \$2D73 PRINT THE STRING FOLLOWING THE JSR INSTR.
 THE STRING HAS TO BE TERMINATED BY
 \$00. AFTER THE STRING IS PRINTED
 THE INSTRUCTION JUST BEHIND \$00
 IS EXECUTED!
 INDST * \$2321 INPUT DISTRIBUTOR
 OUTDST * INDST +01 OUTPUT DISTRIBUTOR
 DSTX * \$2ACC AN OTHER DISTRIBUTOR
 SECTNM * \$265E PRESENT SECTOR NUMBER
 INVEC * \$2301 VECTOR FOR THE PRINTER INPUT
 INVECA * INVEC +02 2ND INPUT DEVICE
 OUTVEC * \$2311 VECTOR FOR THE PRINTER OUTPUT
 OUTVCA * OUTVEC +02 2ND OUTPUT DEVICE
 STRATE * \$26A3 STEP RATE DELAY

HEX DISPLAY MONITOR

RESET LDAIM \$1E PB1...PB4 IS OUTPUT
 STA PBDD
 LDAIM \$01

```

0225: FC07 85 FD          STA  MODE  SET ADDRESS MODE
0226: FC09 A2 FF          LDXIM SEF
0227: FC0B 9A             TXS      INITIALIZE THE STACK POINTER
0228: FC0C 78             SET      DISABLE IRQ
0229:
0230: FC0D D8             START  CLD      DO YOUR JOB IN BINARY MODE
0231:
0232: FC0E 20 7C FC        STARTA JSR      SCAND  SCAN DISPLAY
0233: FC11 D0 FB          BNE     STARTA WAIT UNTIL KEY IS RELEASED
0234:
0235: FC13 20 7C FC        STARA  JSR      SCAND  SCAN DISPLAY AND DEBOUNCE
0236: FC16 F0 FB          BEQ     STARA  IS ANY KEY DEPRESSED?
0237: FC18 20 7C FC        JSR      SCAND  DEBOUNCE AGAIN
0238: FC1B F0 F6          BEQ     STARA  BRANCH ON ERROR
0239: FC1D 20 E1 FC        JSR      GETKEY RETURN WITH KEY VALUE IN ACCU
0240:
0241:
0242:
0243:
0244:
0245:
0246:
0247:
0248:
0249:
0250:
0251:
0252:
0253:
0254:
0255:
0256:
0257:
0258:
0259:
0260:
0261:
0262:
0263:
0264:
0265:
0266:
0267:
0268:
0269:
0270:
0271:
0272:
0273:
0274:
0275:
0276:
0277:
0278:
0279:
0280:

```

COMMAND FILTER

```

GOEXEC CMPIM $13  GO KEY?
BNE    ADMODE
LDXIM  SEF      SET THE STACK POINTER
TXS
CLD      JUST IN CASE
JMI    POINTL  GOTO ADDRESS POINTED BY THE DISPLAY
ADMODE CMPIM $10  AD KEY?
BNE    DAMODE
LDAIM  $01     SET ADDRESS MODE
STA    MODE
BNE    STEPA
DAMODE CMPIM $11  DA KEY?
BNE    STEP
LDAIM  $00     SET DATA MODE
STA    MODE
BEQ    STEPA
STEP   CMPIM $12  + KEY?
BNE    PCKEY
INC    POINTL  POINT=POINT+1
BNE    STEPA
INC    POINTH
STEPA  JMP      START
PCKEY  CMPIM $14  PC KEY?
BNE    ILLKEY
JMP    GOKERN  GOTO KERNEL COMMAND INTERPRETER
ILLKEY CMPIM $15  ILLEGAL KEY
BPL    STEPA  IF YES IGNORE IT
DATA   STA    KEY
LDY    MODE
BNE    ADDRES Y=0 IS DATA MODE, ELSE ADDRESS MODE
LDAIY POINTL  GET DATA SPECIFIED BY POINT

```

```

0281: FC5F 0A          ASLA
0282: FC60 0A          ASLA
0283: FC61 0A          ASLA
0284: FC62 0A          ASLA      SHIFT LOW ORDER NIBBLE TO THE LEFT
0285: FC63 05 FF      ORA   KEY      KEY VALUE IS NOW LOW ORDER NIBBLE
0286: FC65 91 FA      STAY  POINTL  MOVE DATA TO MEMORY
0287: FC67 4C 49 FC      JMP   STEPA
0288:
0289: FC6A A2 04      ADDRESS LDXIM $04      DO 4 SHIFTS
0290:
0291: FC6C 06 FA      ADLOOP ASL   POINTL  SHIFT POINTH, POINTL TO THE LEFT
0292: FC6E 26 FB      ROL   POINTH
0293: FC70 CA          DEX
0294: FC71 00 F9      BNE   ADLOOP
0295: FC73 A5 FA      LDA   POINTL
0296: FC75 05 FF      ORA   KEY      THE KEY VALUE IS THE LOW ORDER ADDR. NIBBLE
0297: FC77 85 FA      STA   POINTL
0298: FC79 4C 49 FC      JMP   STEPA
0299:
0300:
0301:
0302:
0303:
0304:
0305:
0306:
0307:
0308:
0309:
0310:
0311:
0312:
0313:
0314:
0315: FC7C A0 00      SCAND  LDYIM $00
0316: FC7E B1 FA      LDAIY POINTL  GET DATA FROM MEMORY
0317: FC80 85 F9      STA   INH     STORE DATA IN THE DATA BUFFER
0318:
0319: FC82 A9 7F      SCANDS LDAIM $7F   PA0...PA6 IS OUTPUT
0320: FC84 8D 81 FA      STA   PADD
0321: FC87 A2 0B      LDXIM $0B     ENABLE DISPLAY
0322:
0323: FC89 A5 FB      SCDSA  LDA   POINTH  OUTPUT POINTH
0324: FC8B 20 B8 FC      JSR   SHOW
0325: FC8E A5 FA      LDA   POINTL  OUTPUT POINTL
0326: FC90 20 B8 FC      JSR   SHOW
0327: FC93 A5 F9      LDA   INH     OUTPUT INH
0328: FC95 20 B8 FC      JSR   SHOW
0329:
0330: FC98 A9 00      SCDSB  LDAIM $00   PA0...PA7 IS INPUT
0331: FC9A 8D 81 FA      STA   PADD
0332:
0333: FC9D A0 03      AK     LDYIM $03   SCAN 3 KEYBOARD ROWS
0334: FC9F A2 00      LDXIM $00   RESET ROW COUNTER
0335:
0336: FCA1 A9 FF      ONEKEY LDAIM $FF

```

SUBROUTINES OF THE HEX DISPLAY MONITOR

SCAND IS A SUBROUTINE SHOWING THE DATA SPECIFIED BY POINT.
 SCANDS SHOWS THE CONTENTS OF THE DISPL, BUFFERS POINTH, POINTL AND INH ON THE 7-SEGMENT DISPLAYS.
 THE SUBROUTINE AK SCANS THE KEYBOARD. IT IS AUTOMATICALLY CALLED VIA SCAND OR SCANDS. AK RETURNS WITH A=0 IF NO KEY IS DEPRESSED AND RETURNS WITH A<>0 IF A KEY IS DEPRESSED.
 WHEN SCAND OR SCANDS ARE LEFT, PORT A IS AN INPUT PORT.


```

0337:
0338: FCA3 8E 82 FA AKA STX PBD OUTPUT ROW NUMBER
0339: FCA6 E8 INX ENABLE FOLLOWING ROW
0340: FCA7 E8 INX
0341: FCA8 2D 80 FA AND PAD INPUT ROW PATTERN
0342: FCAB 88 DEY ARE ALL ROWS SCANNED?
0343: FCAC D0 F5 BNE AKA
0344: FCAE A0 06 LDYIM $06 TURN DISPLAY OFF
0345: FCB0 8C 82 FA STY PBD
0346: FCB3 09 80 ORAIM $80
0347: FCB5 49 FF EORIM $FF INVERT KEY PATTERN
0348: FCB7 60 RTS
0349:
0350:

```

```

0351: THE SUBROUTINE SHOW DISPLAYS THE CONTENTS
0352: OF ANY DISPLAY BUFFER ON THE 7-SEGMENT DISPLAY.
0353: THE X REGISTER IS USED AS A DISPLAY POINTER
0354:
0355:

```

```

0356: FCB8 48 SHOW PHA SAVE CURRENT BYTE
0357: FCB9 84 FC STY TEMP SAVE Y REGISTER
0358: FCBB 4A LSRA GET HIGH ORDER NIBBLE
0359: FCBC 4A LSRA
0360: FCBD 4A LSRA
0361: FCBE 4A LSRA
0362: FCBF 20 CB FC JSR CONVD OUTPUT HIGH ORDER NIBBLE
0363: FCC2 68 PLA GET BYTE AGAIN
0364: FCC3 29 0F ANDIM $0F MASK OFF HIGH ORDER NIBBLE
0365: FCC5 20 CB FC JSR CONVD OUTPUT LOW ORDER NIBBLE
0366: FCC8 A4 FC LDY TEMP RESTORE Y REGISTER
0367: FCCA 60 RTS
0368:
0369:

```

```

0370: THE SUBROUTINE CONVD DOES A BYTE TO 7-SEGMENT
0371: CONVERSION.
0372:
0373:

```

```

0374: FCCB A8 CONVD TAY USE NIBBLE AS INDEX
0375: FCCC B9 08 FD LDAY LOOK FETCH SEGMENT PATTERN
0376: FCCF 8D 80 FA STA PAD OUTPUT SEGMENT PATTERN
0377: FCD2 8E 82 FA STX PBD OUTPUT DISPLAY ENABLE
0378: FCD5 A0 FF LDYIM $FF DELAY A BIT
0379:
0380: FCD7 88 DELAY DEY
0381: FCD8 D0 FD BNE DELAY
0382: FCDA 88 DEY TURN THE SEGMENTS OFF
0383: FCDB 8C 82 FA STY PBD
0384: FCDE E8 INX ENABLE NEXT DISPLAY
0385: FCDF E8 INX
0386: FCE0 60 RTS
0387:
0388:

```

```

0389: GETKEY RETURNS WITH THE VALUE OF A DEPRESSED
0390: KEY IN THE ACCU REGISTER, IF AN INVALID KEY
0391: WAS DEPRESSED, IT RETURNS WITH $15 IN ACCU.
0392:

```

```

0393:
0394: FCE1 A2 21      GETKEY LDXIM $21      START AT ROW ZERO
0395:
0396: FCE3 A0 01      GETKEA LDYIM $01      GET ONE ROW
0397: FCE5 20 A1 EC      JSR      ONEKEY
0398: FCE8 D0 07      BNE     KEYIN  A=0 THEN NO KEY DEPRESSED
0399: FCEA E0 27      CFXIM  $27      ARE ALL ROWS SCANNED?
0400: FCEC D0 F5      BNE     GETKEA
0401: FCEE A9 15      LDAIM  $15      RETURN HERE IF ILLEGAL KEY
0402: FCF0 60
0403:
0404: FCF1 A0 FF      KEYIN  LDYIM $FF
0405:
0406: FCF3 0A          KEYINA ASLA          SHIFT LEFT UNTIL Y=KEY NUMBER
0407: FCF4 B0 03      BCS    KEYINB
0408: FCF6 08          INY
0409: FCF7 10 FA      BPL    KEYINA
0410:
0411: FCF9 8A          KEYINB TXA
0412: FCFB 29 0F      ANDIM  $0F      MASK MSD
0413: FCFD 4A          LSRA          DIVIDE BY 2
0414: FCFF AA          TAX
0415: FCFE 98          TYA
0416: FCFE 10 03      BPL    KEYIND
0417:
0418: FDD1 10          KEYINC CLC
0419: FDD2 69 07      ADCIM  $07      ADD ROW DISPLACEMENT
0420:
0421: FDD4 CA          KEYIND DEX
0422: FDD5 D0 FA      BNE    KEYINC
0423: FDD7 60          RTS
0424:
0425:
0426: FDD8 40          LOOK    =          $40    0
0427: FDD9 79          =          $79    1
0428: FDDA 24          =          $24    2
0429: FDDB 30          =          $30    3
0430: FDDC 19          =          $19    4
0431: FDDD 12          =          $12    5
0432: FDDF 02          =          $02    6
0433: FDE0 78          =          $78    7
0434: FDE1 00          =          $00    8
0435: FDE2 10          =          $10    9
0436: FDE3 08          =          $08    A
0437: FDE4 03          =          $03    B
0438: FDE5 46          =          $46    C
0439: FDE6 21          =          $21    D
0440: FDE7 06          =          $06    E
0441: FDE8 0E          =          $0E    F
0442:
0443:
0444:
0445:
0446:
0447:
0448:

```

*** BOOTSTRAP STARTS HERE***

```

0449: FD18 20 1E FD   DKBOOT JSR   SIGMA  INITIALIZE THE ENTIRE I/O
0450: FD1B 6C FD 00           JMI   LODVEC  CONTINUE EXECUTION VIA THE LOAD VECTOR
0451:
0452:
0453:
0454:           ***INITIALIZE THE DOS COMPUTER I/O***
0455:
0456: FD1E 20 28 FD   SIGMA  JSR   INIDSK  INITIALIZE THE 6821 PIA
0457: FD21 20 CF FD           JSR   INITPR  INITIALIZE THE 6532 PIA (TERMINAL)
0458: FD24 20 4F FD           JSR   BOOT   LOAD TRACK 0 FROM THE DISKETTE TO $2200
0459: FD27 60           RTS
0460:
0461:
0462:           ***SUBROUTINES OF THE BOOTSTRAP***
0463:
0464: FD28 A0 00           INIDSK LDYIM $00
0465: FD2A 8C 01 C0           STY   CRA    SELECT DATA DIRECTION REGISTER A
0466: FD2D A9 40           LDAIM $00
0467: FD2F 8D 00 C0           STA   DRA
0468: FD32 A9 04           LDAIM $04
0469: FD34 8D 01 C0           STA   CRA
0470: FD37 A9 40           LDAIM $40
0471: FD39 8D 00 C0           STA   DRA
0472: FD3C A2 04           LDXIM $04
0473: FD3E 8E 01 C0           STX   CRA    SELECT DATA REGISTER A
0474: FD41 8C 03 C0           STY   CRB    SELECT DATA DIRECTION REGISTER B
0475: FD44 88           DEY         Y=FF
0476: FD45 8C 02 C0           STY   DRB    PORT B IS TOTAL OUTPUT
0477: FD48 8E 03 C0           STX   CRB    SELECT DATA REGISTER B AGAIN
0478: FD4B 8C 02 C0           STY   DRB    SET ALL OUTPUTS HIGH
0479: FD4E 60           RTS
0480:
0481:
0482: FD4F A9 FB           BOOT   LDAIM $FB  FOR STEP IN DIRECTION
0483: FD51 D0 09           BNE   TRKO   BRANCH ALWAYS
0484:
0485: FD53 A9 02           STEPLO LDAIM $02  LOAD MASK PATTERN
0486: FD55 2C 00 C0           BIT   DRA    ARE WE ON TRACK ZERO?
0487: FD58 F0 1C           BEQ   INDEXP IF YES, DO INDEX TEST
0488: FD5A A9 FF           LDAIM $FF  TOGGLE BITS FOR STEP OUT
0489:
0490: FD5C 8D 02 C0           TRKO  STA   DRB  STORE IN PIA
0491: FD5F 20 CE FD           JSR   DUMMY  DELAY SOME CYCLES
0492: FD62 29 F7           ANDIM $F7  CLEAR STEP BIT
0493: FD64 8D 02 C0           STA   DRB
0494: FD67 20 CE FD           JSR   DUMMY  DELAY SOME CYCLES
0495: FD6A 09 08           ORAIM $08  SET THE STEP BIT
0496: FD6C 8D 02 C0           STA   DRB  AND OUTPUT IT
0497: FD6F A2 18           LDXIM $18  DELAY VALUE
0498: FD71 20 BA FD           JSR   FDELAY WAIT A MOMENT
0499: FD74 F0 DD           BEQ   STEPLO TRY IT AGAIN
0500:
0501: FD76 A2 7F           INDEXP LDXIM $7F  RESET HEAD LOAD BIT
0502: FD78 8E 02 C0           STX   DRB  HEAD LOAD
0503: FD7B 20 D7 FC           JSR   DELAY  WAIT A MOMENT
0504:

```

```

0505: FD7E AD 00 C0 INDBEG LDA DRA GET DISK STATUS
0506: FD81 30 FB BMI INDBEG LOOP TILL INDEX START
0507:
0508: FD83 AD 00 C0 INDBEG LDA DRA GET DISK STATUS
0509: FD86 10 FB BPL INDBEG LOOP TILL INDEX END
0510: FD88 A9 03 LDAIM $03
0511: FD8A 8D 10 C0 STA CACIA RESET ACIA
0512: FD8D A9 58 LDAIM $58 INITIALIZE ACIA FOR
0513: FD8F 8D 10 C0 STA CACIA 8 BIT EVEN PARITY, DIVIDE BY 1 RATE
0514: FD92 20 C5 FD JSR RBYTE GET FIRST BYTE FROM DISKETTE
0515: FD95 85 FE STA LODVEC +01 SAVE HIGH START ADDRESS
0516: FD97 AA TAX SAVE IT FOR LATER USE
0517: FD98 20 C5 FD JSR RBYTE GET SECOND BYTE
0518: FD9B 85 FE STA LODVEC SAVE LOW START ADDRESS
0519: FD9D 20 C5 FD JSR RBYTE READ THIRD BYTE FROM DISKETTE
0520: FDA0 85 FE STA $00FF SAVE THE AMOUNT OF PAGES WE'RE GOING TO READ
0521: FROM DISKETTE (PAGES/TRACK)
0522: FDA2 A0 00 LDYIM $00 INITIALIZE LOAD POINTER
0523:
0524: FDA4 20 C5 FD RPAGE JSR RBYTE READ A BYTE FROM DISKETTE
0525: FDA7 91 FD STAYI LODVEC AND STORE IT IN MEMORY
0526: FDA9 CB INY SET UP FOR NEXT MEMORY LOCATION
0527: FDAA D0 F8 BNE RPAGE LOOP TILL A PAGE IS READ
0528: FDAC E6 FE INC LODVEC +01 ADJUST LOAD VECTOR
0529: FDAE C6 FE DEC $00FF DECREMENT PAGE COUNTER
0530: FDB0 D0 F2 BNE RPAGE LOOP TILL ALL PAGES ARE READ
0531: FDB2 86 FE STX LODVEC +01 RESTORE LOAD VECTOR
0532: FDB4 A9 FF LDAIM $FF
0533: FDB6 8D 02 C0 STA DRB UNLOAD HEAD
0534: FDB9 60 RTS
0535:
0536:
0537: FDBA A0 F8 FDELAY LDYIM $F8
0538:
0539: FDBC 88 FDELA DEY
0540: FDBD D0 FD BNE FDELA LOOP A BIT
0541: FDBF 55 FF EORX $00FF DUMMY INSTRUCTION
0542: FDC1 CA DEX ADJUST TIMER COUNT
0543: FDC2 D0 F6 BNE FDELAY LOOP TILL TIME OUT
0544: FDC4 60 RTS
0545:
0546:
0547: FDC5 AD 10 C0 RBYTE LDA CACIA GET ACIA STATUS
0548: FDC8 4A LSRA RECEIVER BIT IN CARRY
0549: FDC9 90 FA BCC RBYTE LOOP TILL RECEIVER BUFFER IS FULL
0550: FDCB AD 11 C0 LDA DACIA GET DATA BYTE FROM ACIA
0551:
0552: FDCE 60 DUMMY RTS
0553:
0554:
0555: ***INITIALIZE THE TERMINAL***
0556:
0557: FDCF D8 INITPR CLD
0558: FDD0 78 SEI DISABLE INTERRUPTS
0559: FDD1 A9 67 LDAIM $67
0560: FDD3 8D 82 FA STA PBD

```

```

0561: FDD6 A9 00          LDAIM $00
0562: FDD8 8D 80 FA      STA PAD
0563: FDDB A2 FC          LDXIM $FC      YOUR EPSON LIKES THIS SPEED
0564: FDDD 8E 5A FA      STX CNTLX
0565: FDE0 A2 FF          LDXIM $FF
0566: FDE2 8E 5B FA      STX CNTHX
0567: FDE5 EA          NOP
0568: FDE6 A9 7F          LDAIM $7F
0569: FDE8 8D 81 FA      STA PADD      SET DATA DIRECTION
0570: FDEB 4A          LERA
0571: FDEC 8D 83 FA      STA PBDD      SET PORT B FOR EPSON BUSY ON PB5
0572: FDEF A2 03          LDXIM $03
0573: FDP1 8E 59 FA      STX STPBIT    TRANSMIT NO PARITY BUT 2 STOP BITS
0574:
0575: FDF4 2C 80 FA      STRTBT BIT    PAD      WAIT FOR A START BIT
0576: FDF7 30 FB          BMI STRTBT
0577: FDF9 20 4F FE      JSR COMTIM    COMPUTE THE START BIT TIME
0578: FDEC 4E 5F FA      LSR TIMH     DIVIDE BY 2
0579: FDEF 6E 5E FA      ROR TIML
0580: FE02 AD 5E FA      LDA TIML
0581: FE05 8D 5C FA      STA CNTHL    SAVE HALF START BIT TIME
0582: FE08 AD 5F FA      LDA TIMH
0583: FE0B 8D 5D FA      STA CNTHH
0584: FE0E A2 08          LDXIM $08
0585: FE10 20 72 FE      JSR DELHBT
0586: FE13 20 2B FE      JSR RECD     GET THE REST OF THE CHARACTER
0587: FE16 C9 7F          CMPIM $7F    WAS IT A RUBOUT CHARACTER?
0588: FE18 D0 B5          BNE INITPR   IF NOT, THEN TRY IT AGAIN
0589: FE1A 60          RTS
0590:
0591:
0592:
0593:
0594:
0595: FE1B 2C 80 FA      RECCHA BIT    PAD      WAIT FOR A START BIT
0596: FE1E 30 FB          BMI RECCHA
0597: FE20 8E 61 FA      STX TEMPB    SAVE X
0598: FE23 A2 08          LDXIM $08    WE RECEIVE 8 BITS
0599: FE25 20 72 FE      JSR DELHBT   DELAY FOR HALF BIT TIME
0600:
0601: FE28 70 81 FE      RECA JSR      DELBIT   DELAY FOR ONE BIT TIME
0602:
0603: FE2B 2C 80 FA      RECD BIT    PAD      ONE/ZERO CHECK
0604: FE2E 10 09          BPL RECB     BRANCH ON ZERO
0605: FE30 38          SEC          IT IS A '1'
0606: FE31 6E 62 FA      ROR CHA     ROTATE CARRY INT CHARACTER
0607: FE34 CA          DEX         SET UP FOR NEXT BIT
0608: FE35 D0 F1          BNE RECA    ARE ALL BITS READ?
0609: FE37 F0 07          BEQ RECC
0610:
0611: FE39 18          RECB CLC     IT IS A '0'
0612: FE3A 6E 62 FA      ROR CHA
0613: FE3D CA          DEX
0614: FE3E D0 E8          BNE RECA
0615:
0616: FE40 20 81 FE      RECC JSR      DELBIT   WAIT FOR THE STOP BIT

```

```

0617: FE43 AD 62 FA      LDA   CHA   LOAD CHARACTER IN ACCU
0618: FE46 29 7F      ANDIM S7F   MASK OFF BIT 7
0619: FE48 8D 63 23     STA   AHOLD SAVE CHARACTER FOR DOS
0620: FE4B AE 61 FA      LDX   TEMPB RESTORE X
0621: FE4E 60      RTS
0622:
0623:
0624:      ***COMPUTE BIT TIME***
0625:
0626:
0627: FE4F 18      COMTIM CLC
0628: FE50 AD 5A FA      LDA   CNTLX 16 BIT ADDITION
0629: FE53 69 01      ADCIM S01
0630: FE55 8D 5A FA      STA   CNTLX
0631: FE58 AD 5B FA      LDA   CNTHX
0632: FE5B 69 00      ADCIM S00
0633: FE5D 8D 5B FA      STA   CNTHX
0634: FE60 2C 80 FA      BIT   PAD   IS START BIT FINISHED?
0635: FE63 10 EA      BPL   COMTIM
0636: FE65 AD 5A FA      LDA   CNTLX SET UP FOR HALF BIT TIME COMPUTATION
0637: FE68 8D 5E FA      STA   TIML
0638: FE6B AD 5B FA      LDA   CNTHX
0639: FE6E 8D 5F FA      STA   TIMH
0640: FE71 60      RTS
0641:
0642:
0643:
0644:      ***DELAY A FULL/HALF BIT TIME***
0645:
0646:
0647: FE72 AD 5C FA      DELHBT LDA   CNTHL  FETCH HALF BIT TIME
0648: FE75 8D 5E FA      STA   TIML
0649: FE78 AD 5D FA      LDA   CNTHH
0650: FE7B 8D 5F FA      STA   TIMH
0651: FE7E 38      SEC
0652: FE7F B0 0C      BCS   CNTDN  BRANCH ALWAYS
0653:
0654: FE81 AD 5A FA      DELBIT LDA   CNTLX  FETCH FULL BIT TIME
0655: FE84 8D 5E FA      STA   TIML
0656: FE87 AD 5B FA      LDA   CNTHX
0657: FE8A 8D 5F FA      STA   TIMH
0658:
0659: FE8D 38      CNTDN SEC
0660: FE8E AD 5E FA      LDA   TIML  16 BIT SUBTRACTION
0661: FE91 E9 01      SBCIM S01
0662: FE93 8D 5E FA      STA   TIML
0663: FE96 AD 5F FA      LDA   TIMH
0664: FE99 E9 00      SBCIM S00
0665: FE9B 8D 5F FA      STA   TIMH
0666: FE9E EA      NOP
0667: FE9F EA      NOP   EQUALIZE 4 MICRO SECONDS
0668: FEA0 B0 0B      BCS   CNTDN  TIME OUT?
0669: FEA2 10      RTS
0670:
0671:
0672:

```

```

0673:          ***PRINT THE CHARACTER IN ACCU***
0674:
0675:
0676: FEA3 8E 60 FA PRCHA STX  TEMPA  SAVE X
0677: FEA6 8D 62 FA          STA  CHA   SAVE CHARACTER
0678:
0679: FEA9 AD 82 FA CTE    LDA  PBD   GET PIA DATA OF 6532
680: FEAC 29 40          ANDIM $40  MASK EPSON BUSY BIT
681: FEAE D0 F9          BNE  CTS   LOOP IF PB5 IS HIGH
0682: FEB0 AD 82 FA          LDA  PBD
0683: FEB3 29 FE          ANDIM $FE  TRANSMIT START BIT
0684: FEB5 8D 82 FA          STA  PBD
0685: FEB8 20 81 FE          JSR  DELBIT DELAY ONE BIT TIME
0686: FEBB A2 07          LDXIM $07  TRANSMIT 7 DATA BITS
0687:
0688: FEBD 4E 62 FA PRA    LSR  CHA   SHIFT OUT CHARACTER
0689: FEC0 90 30          BCC  PRC   BRANCH ON ZERO
0690: FEC2 AD 82 FA          LDA  PBD
0691: FEC5 09 01          ORAIM $01  OUTPUT A ONE
0692: FEC7 8D 82 FA          STA  PBD
0693:
0694: FECA 20 81 FE PRB    JSR  DELBIT DELAY ONE BIT TIME
0695: FECD CA            DEX
0696: FECE D0 ED          BNE  PRA   ARE ALL BITS TRANSMITTED?
0697: FED0 AE 59 FA          LDX  STPBIT GET AMOUNT OF STOPBITS+1
0698:
0699: FED3 AD 82 FA PRD    LDA  PBD
0700: FED6 09 01          ORAIM $01  FIRST NONE PARITY
0701: FED8 8D 82 FA          STA  PBD   AND THE STOP BITS
702: FEDB 20 81 FE          JSR  DELBIT
0703: FEDE CA            DEX
0704: FEDF D0 F2          BNE  PRD
0705: FEE1 2C 80 FA          BIT  PAD   CHECK IF BREAK KEY IS DEPRESSED!
0706: FEE4 10 04          BPL  BRKTST
0707: FEE6 AE 60 FA          LDX  TEMPA RESTORE X
0708: FEE9 60            RTS
0709:
0710: FEEA 2C 80 FA BRKTST BIT  PAD   IS THE BREAK KEY ACTIVE?
0711: FEED 10 FB          BPL  BRKTST
0712: FEED 6C 7C FA          JMI  BRKT  JUMP VIA AN USER SELECTABLE VECTOR
0713:
0714:
0715: FEF2 AD 82 FA PRC    LDA  PBD
0716: FEF5 29 FE          ANDIM $FE  OUTPUT A ZERO
0717: FEF7 8D 82 FA          STA  PBD
0718: FEFA 18            CLC
0719: FEFB 90 CD          BCC  PRB   BRANCH ALWAYS
0720:
0721:
0722:
0723:
724:
0725:
0726:
0727:
0728:          *****
          ***WARM START ENTRY FOR DOS***

```

```

0729:
0730:
0731:
0732: FEED 20 03 FF GOKERN JSR RESTTY RESET PRINTER I/O.
0733: FF00 4C 51 2A JMP KERNEL AND GOTO DOS KOMMAND INTERPRETER
0734:
0735:
0736:
0737: FF03 A9 27 RESTTY LDAIM $27
0738: FF05 8D 82 FA STA RBD
0739: FF08 A9 00 LDAIM $00
0740: FF0A 8D 80 FA STA PAD
0741: FF0D A9 7F LDAIM $7F
0742: FF0F 8D 81 FA STA PADD
0743: FF12 4A LSRA
0744: FF13 8D 83 FA STA PBDD
0745: FF16 60 RTS
0746:
0747:
0748:
0749:
0750:
0751:
0752:
0753:
0754:
0755: FF17 A9 2E BASBRK LDAIM BASTST SET BASIC'S BREAK VECTOR
0756: FF19 8D 7D FA STA BRKT
0757: FF1C A9 FF LDAIM BASTST /256
0758: FF1E 8D 7E FA STA BRKT +01
0759: FF21 A9 00 11 LDAIM MON IF NMI BREAK THEN GOTO HEX DISPLAY MONITOR
0760: FF23 8D 7A FA STA NMIL
0761: FF26 A9 FC LDAIM MON /256
0762: FF28 8D 7B FA STA NMIH
0763: FF2B AC 1B FD JMP DXBOOT NOW BOOT UP THE SYSTEM
0764:
0765:
0766: FF2E A9 03 BASTST LDAIM $03 SET BREAK FLAG (CTL-C)
0767: FF30 8D 25 23 STA KPDD AND RETURN TO CALLER
0768: FF33 60 RTS
0769:
0770:
0771:
0772:
0773:
0774:
0775:
0776:
0777: FF34 A9 51 DOBRK LDAIM KERNEL IF BREAK THEN GOTO KERNEL
0778: FF36 8D 7C FA STA BRKT
0779: FF39 A9 2A LDAIM KERNEL /256
0780: FF3B 8D 7D FA STA BRKT +01
0781: FF3E A9 00 LDAIM MON IF NMI BREAK THEN GOTO HEX DISPLAY MONITOR
0782: FF40 8D 7A FA STA NMIL
0783: FF43 A9 FC LDAIM MON /256
0784: FF45 8D 7B FA STA NMIH

```



```

0729:
0730:
0731:
0732: FF0D 20 03 FF GOKERN JSR RESTTY RESET PRINTER I/O
0733: FF00 4C 51 25 JMP KERNEL AND GOTO DOS KOMMAND INTERPRETER
0734:
0735:
0736:
0737: FF03 A9 27 RESTTY LDAIM $27
0738: FF05 8D 82 FA STA PBD
0739: FF08 A9 00 LDAIM $00
0740: FF0A 8D 80 FA STA PAD
0741: FF0D A9 7F LDAIM $7F
0742: FF0F 8D 81 FA STA PADD
0743: FF12 4A LSRA
0744: FF13 8D 83 FA STA PBDD
0745: FF16 60 RTS
0746:
0747:
0748:
0749:
0750:
0751:
0752:
0753:
0754:
0755: FF17 A9 2E BASBRK LDAIM BASTST SET BASIC'S BREAK VECTOR
0756: FF19 8D 7C FA STA BRKT
0757: FF1C A9 FF LDAIM BASTST /256
0758: FF1E 8D 7D FA STA BRKT +01
0759: FF21 A9 00 LDAIM MON IF NMI BREAK THEN GOTO HEX DISPLAY MONITOR
0760: FF23 8D 7A FA STA NMIL
0761: FF26 A9 FC LDAIM MON /256
0762: FF28 8D 7B FA STA NMIH
0763: FF2B 4C 18 FD JMP DKBOOT NOW BOOT UP THE SYSTEM
0764:
0765:
0766: FF2E A9 03 BASTST LDAIM $03 SET BREAK FLAG (CTL-C)
0767: FF30 8D 25 23 STA KPDO AND RETURN TO CALLER
0768: FF33 60 RTS
0769:
0770:
0771:
0772:
0773:
0774:
0775:
0776:
0777: FF34 A9 51 DOSBRK LDAIM KERNEL IF BREAK THEN GOTO KERNEL
0778: FF36 8D 7C FA STA BRKT
0779: FF39 A9 2A LDAIM KERNEL /256
0780: FF3B 8D 7D FA STA BRKT +01
0781: FF3E A9 00 LDAIM MON IF NMI BREAK THEN GOTO HEX DISPLAY MONITOR
0782: FF40 8D 7A FA STA NMIL
0783: FF43 A9 FC LDAIM MON /256
0784: FF45 8D 7B FA STA NMIH

```

```

0785: FF48 4C 18 FD          JMP    DKBOOT NOW BOOT UP THE SYSTEM
0786:
0787:
0788:
0789:          ***INDIRECT INTERRUPT VECTORS***
0790:
0791: FF4B 6C 7A FA    JNDNMI JMI    NMIL
0792: FF4E 6C 7E FA    INDIRQ JMI    IRQ
0793:
0794:
0795:
0796:
0797:
0798:
0799:
0800:          *****
0801:          *** INITIALIZE BOOTSTRAP FOR OSI MODIFICATION ***
0802:          *****
0803:
0804:
0805:
0806:
0807:
0808:
0809:
0810: FF51 20 1E FD    FIRSTX JSR    SIGMA  INIT THE DOS COMPUTER I/O
0811: FF54 A9 28          LDAIM $28    SETP RATE DELAY
0812: FF56 8D A3 26    STA    STRATE
0813: FF59 A9 01          LDAIM $01    TRACK=1/SECTOR=1
0814: FF5B 8D 5E 26    STA    SECTNM SAVE SECTOR NUMBER
0815: FF5E 20 BC 26    JSR    SETTK  MOVE HEAD ON TRACK 1
0816: FF61 A9 2A          LDAIM $2A    THE LOAD VECTOR IS $2A00
0817: FF63 85 FF          STA    MEMHI  SAVE IT
0818: FF65 20 54 27    JSR    LDHEAD LOAD THE HEAD
0819: FF68 86 FE          STX    MEMLO
0820: FF6A 20 67 29    JSR    READDK READ TRACK 1/SECTOR 2
0821: FF6D A9 01          LDAIM $01    SET FOR DRIVE A AND THE DISTRIBUTOR
0822: FF6F 8D 21 23    STA    INDST  INPUT DISTRIBUTOR
0823: FF72 8D 22 23    STA    OUTDST OUTPUT DISTRIBUTOR
0824: FF75 8D C6 2A    STA    DSTX   COMMON DISTRIBUTOR
0825: FF78 20 C6 29    JSR    SETDRV
0826: FF7B A9 1A          LDAIM RECCHA -01
0827: FF7D 8D 01 23    STA    INVEC  SET PRINTER INPUT VECTOR
0828: FF80 8D 03 23    STA    INVECA
0829: FF83 A9 A2          LDAIM PRCHA  -01
0830: FF85 8D 11 23    STA    OUTVEC SET PRINTER OUTPUT VECTOR
0831: FF88 8D 13 23    STA    OUTVCA
0832: FF8B A9 FE          LDAIM RECCHA /256 HIGH ORDER ADDRESS OF RECCHA&PRCHA
0833: FF8D 8D 02 23    STA    INVEC  +01
0834: FF90 8D 04 23    STA    INVECA +01
0835: FF93 8D 12 23    STA    OUTVEC +01
0836: FF96 8D 14 23    STA    OUTVCA +01
0837: FF99 60          RTS
0838:
0839:
0840: FF9A EE 5E 26    FIRSTY INC   SECTNM SECTOR NUMBER IS 2

```

```

0841: FF9D A9 06          LDAIM $06      TRACK=06
0842: FF9F 20 BC 26      JSR  SETTK    MOVE HEAD ON TRACK 06
0843: FFA2 20 67 29      JSR  READDK   READ TRACK 06/SECTOR 2
0844: FFA5 EE 5E 26      INC  SECTNM   SECTNM=3
0845: FFA8 A9 00          LDAIM $00      LOAD VECTOR = $0000
0846: FFAA 85 FE          STA  MEMLO
0847: FFAC 85 FF          STA  MEMHI
0848: FFAE 20 67 29      JSR  READDK   READ TRACK 06/SECTOR 3
0849: FFB1 A9 01          LDAIM $01      SECTNM=1
0850: FFB3 8D 5E 26      STA  SECTNM
0851: FFB6 A9 13          LDAIM $13      TRACK=13
0852: FFB8 20 BC 26      JSR  SETTK    MOVE HEAD ON TRACK 13
0853: FFB8 A9 32          LDAIM $32      SET LOAD POINTER TO $3274
0854: FFB8 85 FF          STA  MEMHI
0855: FFBF A9 74          LDAIM $74
0856: FFC1 85 FE          STA  MEMLO
0857: FFC3 20 64 27      JSR  LDHEAD
0858: FFC6 20 67 29      JSR  READDK   READ TRACK 13/SECTOR 1
0859:
0860: FFC9 20 61 27  FIRSTZ JSR  UNLDHD  UNLOAD THE HEAD
0861:
0862: FFCC 20 73 2D      JSR  STROUT
0863: FFCE 0D
0864: FFDE 0A           =    $0A    CRLF
0865: FFD1 2A           =    '*'
0866: FFD2 54           =    'T'
0867: FFD3 52           =    'R'
0868: FFD4 41           =    'A'
0869: FFD5 43           =    'C'
0870: FFD6 4B           =    'K'
0871: FFD7 20           =    ' '
0872: FFD8 30           =    '0'
0873: FFD9 26           =    '&'
0874: FFDA 31           =    '1'
0875: FFDB 2A           =    '*'
0876: FFDC 0D           =    $0D
0877: FFDD 0A           =    $0A
0878: FFDE 00           =    $00
0879:
0880: FFDF 4C 00 FC      JMP  RESET   GOTO HEX DISPLAY MONITOR
0881:
0882:
0883: *****
0884: ***BOOT OS 65D V3.1 FROM HERE***
0885: *****
0886:
0887:
0888: FFE2 20 51 FF  VONE JSR  FIRSTX
0889: FFE5 4C C9 FF  JMP  FIRSTZ
0890:
0891:
0892:
0893:
0894:
0895:
0896:

```

```

0897:
0898:
0899:
0900:
0901:
1902: FFEB 20 51 FF VTHREE JSR FIRSTX
0903: FFEB 4C 9A FF JMP FIRSTY
0904:
0905:
0906:
-T

```

SYMBOL	TABLE	3400	3700				
ACR	F80B	ADDRS	FC6A	ADLOOP	FC6C	ADMODE	FC2B
AHOLD	2363	AK	FC9D	AKA	FCA3	BASBRK	FF17
BASTST	FF2E	BOOT	FD4F	BRKT	FA7C	BRKTST	FEEA
CACIA	C010	CHA	FA62	CNTA	FAF4	CNTB	FAF5
CNTC	FAF6	CNTD	FAF7	CNTDN	FE8D	CNTE	FAFC
CNTF	FAFD	CNTG	FAFE	CNTH	FAFF	CNTHH	FASD
CNTHL	FA5C	CNTHX	FA5B	CNTLX	FA5A	COMTIM	FE4F
CONVD	FCCB	CRA	C001	CRB	C003	CTS	FEA9
DACIA	C011	DAMODE	FC35	DATA	FC57	DDRA	F803
DDRB	F802	DELAY	FCD7	DELBIT	FE81	DELHBT	FE72
DKBOOT	FD18	DOSBRK	FF34	DRA	C000	DRB	C002
DSTX	2AC6	DUMMY	FDCE	EDETA	FAE4	EDETB	FAE5
EDETC	FAE6	EDETD	FAE7	FDELA	F0BC	FDELAY	F0BA
FIRSTX	FF51	FIRSTY	FF9A	FIRSTZ	FFC9	GETKEA	FCB3
GETKEY	FCB1	GOEXEC	FC20	GOKERN	FEFD	IER	F80E
IER	F80D	ILLKEY	FC53	INDBEG	FD7E	INDEND	FD83
INDEXP	FD76	INDIRQ	FF4E	INDNMI	FF4B	INDST	2321
INH	00F9	INIDSK	FD28	INITPR	FD0F	INL	00F8
INVEC	2301	INVECA	2303	IRQH	FA7F	IRQL	FA7E
KERNEL	2A51	KEYIN	FCF1	KEYINA	FCF3	KEYINB	FCF9
KEYINC	FD01	KEYIND	FD04	KEY	00FF	KPDO	2325
LDHEAD	2754	LODVEC	00FD	LOOK	FD08	MEMHI	00FE
MEMLO	00FE	MODE	00FD	MON	FC00	NIBBLE	00FE
NMIL	FA7B	NMIL	FA7A	ONEKEY	FCA1	ORAA	F80F
ORA	F801	ORB	F800	OUTDST	2322	OUTVCA	2313
OUTVEC	2311	PADD	FA81	PAD	FA80	PBDD	FA83
PBD	FA82	PCKEY	FC4C	PCR	F80C	POINTH	00FB
POINTL	00FA	PRA	FEBD	PRB	FECA	PRCHA	FEA3
PRC	FEF2	PRD	FEF3	RBYTE	FDC5	RDFLAG	FAD5
READDK	2967	RECA	FE2B	RECB	FE39	RECC	FE40
RECCHA	FE1B	RECD	FE2B	RESET	FC00	RESTTY	FF03
RPAGE	FDA4	SCAND	FC7C	SCANDS	FC82	SCDSA	FC89
SCDSB	FC98	SECTNM	265E	SETDRV	29C6	SETTK	26BC
SHOW	FCB8	SIGMA	FD1E	SR	F80A	STARA	FC13
START	FC0D	STARTA	FC0E	STEP	FC3F	STEPA	FC49
STEPLO	FD53	STPBIT	FA59	STRATE	26A3	STROUT	2D73
STRTBT	FD44	TAALH	F807	TAALL	F806	TACH	F805
TALL	F804	TBCH	F809	TBLCL	F808	TEMP	00FC
TEMPA	FA60	TEMPB	FA61	TIMH	FA5F	TIML	FA5E
TRKO	FD5C	UNLDHD	2761	VONE	FFE2	VTHREE	FFEB

SYMBOL TABLE 3400 37C0

INL	00F8	INH	00F9	POINTL	00FA	POINTH	00FB
TEMP	00FC	LODVEC	00FD	MODE	00FD	MEMLO	00FE
NIBBLE	00FE	KEY	00FF	MEMHI	00FF	INVEC	2301
INVECA	2303	OUTVEC	2311	OUTVCA	2313	INDST	2321
OUTDST	2322	KPDO	2325	AHOLD	2363	SECTNM	265E
STRATE	26A3	SETTK	26BC	LDHEAD	2754	UNLDHD	2761
READDK	2967	SETDRV	29C6	KERNEL	2A51	DSTX	2AC6
STROUT	2D73	DRA	C000	CRA	C001	DRB	C002
CRB	C003	CACIA	C010	DACIA	C011	ORB	F800
ORA	F801	DDR8	F802	DDRA	F803	TALL	F804
TACH	F805	TAALL	F806	TAALH	F807	TBLCL	F808
TBCH	F809	SR	F80A	ACR	F80B	PCR	F80C
IFR	F80D	IER	F80E	ORAA	F80F	STPBIT	FA59
CNTLX	FA5A	CNTHX	FA5B	CNTHL	FA5C	CNTHH	FA5D
TIML	FA5E	TIMH	FA5F	TEMPA	FA60	TEMPB	FA61
CHA	FA62	NMIL	FA7A	NMIH	FA7B	BRKT	FA7C
IRQL	FA7E	IRQH	FA7F	PAD	FA80	PADD	FA81
PBD	FA82	PBDD	FA83	RDFLAG	FAD5	EDETA	FAE4
EDETB	FAE5	EDETC	FAE6	EDETD	FAE7	CNTA	FAF4
CNTB	FAF5	CNTC	FAF6	CNTD	FAF7	CNTE	FAFC
CNTF	FAFD	CNTG	FAFE	CNTH	FAFF	MON	FC00
RESET	FC00	START	FC0D	STARTA	FC0E	STARA	FC13
GOEXEC	FC20	ADM0DE	FC2B	DAM0DE	FC35	STEP	FC3F
STAPA	FC49	PCKEY	FC4C	ILLKEY	FC53	DATA	FC57
ADDRES	FC6A	ADLOOP	FC6C	SCAND	FC7C	SCANDS	FC82
SCDSA	FC89	SCDSB	FC98	AK	FC9D	ONEKEY	FCAL
AKA	FCA3	SHOW	FCB8	CONVD	FCCB	DELAY	FCD7
GETKEY	FCE1	GETKEA	FCE3	KEYIN	PCF1	KEYINA	FCF3
KEYINB	FCF9	KEYINC	FD01	KEYIND	FD04	LOOK	FD08
DKBOOT	FD18	SIGMA	FD1E	INIDSK	FD28	BOOT	FD4F
STEPLO	FD53	TRKO	FD5C	INDEXP	FD76	INDBEG	FD7E
INDEND	FD83	RPAGE	FDA4	FDELAY	FDBA	FDELA	FDBC
RBYTE	FDC5	DUMMY	FDCE	INITPR	FDCE	STRTBT	EDF4
RECCHA	FE1B	RECA	FE28	RECD	FE2B	RECB	FE39
RECC	FE40	COMTIM	FE4F	DELHBT	FE72	DELBIT	FE81
CNTDN	FE8D	PRCHA	FEA3	CTS	FEA9	PRA	FE8D
PRB	FECA	PRD	FED3	BRKTST	FEEA	PRC	FEF2
GOKERN	FEFD	RESTTY	FF03	BASBRK	FF17	BASTST	FF2E
DOSBRK	FF34	INDNMI	FF4B	INDIRQ	FF4E	FIRSTX	FF51
FIRSTY	FF9A	FIRSTZ	FFC9	VONE	FFE2	VTHREE	FFEB