

PIM-1

```
0050 0000      .TITLE      '#### PIM65 MONITOR  ####'
0051 0000      ;
0052 0000      ; *****
0053 0000      ; *****
0054 0000      ; *****
0055 0000      ; ***** PIM65  HEXADECIMAL  MONITOR *****
0056 0000      ; *****
0057 0000      ; ***** PROTON ELECTRONICS *****
0058 0000      ; ***** WETH. KONIGLAAN 2 *****
0059 0000      ; ***** 1412 GT  NAARDEN *****
0060 0000      ; *****
0061 0000      ; *****
0062 0000      ; *****
0063 0000      ;
0064 0000      ; DD: 01-06-81 SK <<< INVERTED SEGMENTS, SBC65/68 >>>
0065 0000      ;
```

COMMAND DESCRIPTION

```
0067 0000      ; COMMAND      FUNKTION
0068 0000      ;
0069 0000      ; +MEM, ENTR/-MEM  DISPLAY NEXT/PREVIOUS LOCATION, ENTER AD
0070 0000      ; GO              GO TO USER'S PROGRAM
0071 0000      ; TIME            SHOW THE TIME
0072 0000      ; ADDR            INPUT ADDRESS
0073 0000      ; INS             INSERT 1 BYTE
0074 0000      ; DEL             DELETE 1 BYTE
0075 0000      ; OFFSET          CALCULATE BRANCH OFFSET
0076 0000      ; SST             SINGLE STEP
0077 0000      ; LOAD            LAAD CODE VAN AUDIO-CASSETTE
0078 0000      ; DUMP            DUMP CODE NAAR CASSETTE
0079 0000      ;
0080 0000      ; THE MONITOR DRIVES A 8 DIGIT DISPLAY.
0081 0000      ; HANDLES INTERRUPTS AND BREAKPOINTS
0082 0000      ; AND UPDATES A 24 HRS CLOCK.
0083 0000      ;
0084 0000      ;
```

VIA ADDRESSING

```
0086 0000      ;
0087 0000      ; ROMBAS:  =#F800      ; START OF PROGRAM
0088 0000      ; IOSEL:   =#C000      ; I/O SELECT
0089 0000      ; MONRAM:  =#F0
0090 0000      ;
0091 0000      ;          *=IOSEL
```

```
0092 C000 ; VIA #0
0093 C000 DRB:   **++1
0094 C001 DRA:   **++1
0095 C002 DDRB:  **++1
0096 C003 DDRA:  **++1
0097 C004 T1CL:  **++1
0098 C005 T1CH:  **++1
0099 C006 T1LL:  **++1
0100 C007 T1LH:  **++1
0101 C008 T2LL:  **++1
0102 C009 T2CH:  **++1
0103 C00A SR:    **++1
0104 C00B ACR:   **++1
0105 C00C PCR:   **++1
0106 C00D IFR:   **++1
0107 C00E IER:   **++1
0108 C00F APORT: **++1
0109 C010 ;
0110 C010 ; VIA #1
0111 C010 DRB1:  **++1
0112 C011 DRA1:  **++1
0113 C012 DDRB1: **++1
0114 C013 DDRA1: **++1
0115 C014 T1CL1: **++1
0116 C015 T1CH1: **++1
0117 C016 T1LL1: **++1
0118 C017 T1LH1: **++1
0119 C018 T2LL1: **++1
0120 C019 T2CH1: **++1
0121 C01A SR1:   **++1
0122 C01B ACR1:  **++1
0123 C01C PCR1:  **++1
0124 C01D IFR1:  **++1
0125 C01E IER1:  **++1
0126 C01F APORT1: **++1
0127 C020 ;
0128 C020 ;
```

ASSEMBLER CONSTANTS

```
0130 C020 ENTER   =$10
0131 C020 MODMSK  =$80
0132 C020 ;
0133 C020 ; COMMAND-KEY VALUES
0134 C020 MEMCMD:  =$10
0135 C020 ADRCMD:  =$11
0136 C020 SSTCMD:  =$12
0137 C020 CMD2ND:  =$13
0138 C020 GDCMD:   =$14
0139 C020 INSCMD:  =$15
0140 C020 DELCMD:  =$16
```

```

0141 C020      LODCMD:  = $17
0142 C020      ;
0143 C020      ; COMMAND'S UNDER '2ND'
0144 C020      DMPCMD:  =LODCMD+MODMSK
0145 C020      TIMCMD:  =ADRCMD+MODMSK
0146 C020      OFFCMD:  =SSTCMD+MODMSK
0147 C020      TSTCMD:  =GOCMD+MODMSK
0148 C020      ;
0149 C020      ;==MONRAM      ; ZERO PAGE
0150 00F0      NMIVC      ;==+2      ; INDIRECT VECTORS
0151 00F2      INTVEC     ;==+2      ; IRQ BEFORE MONITOR
0152 00F4      DISVEC     ;==+2      ; 'SOFTWARE INTERRUPT'
0153 00F6      IRQVEC     ;==+2      ; IRQ AFTER MONITOR, NO 'BRK'
0154 00F8      ADL        ;==+1      ; MEMORY POINTER
0155 00F9      ADH        ;==+1
0156 00FA      AC         ;==+1      ; USER REG  A
0157 00FB      XR         ;==+1      ;           X
0158 00FC      YR         ;==+1      ;           Y
0159 00FD      PS         ;==+1      ; USER STATUS
0160 00FE      SPTR       ;==+1      ; USER STACKPOINTER
0161 00FF      PRVCMD     ;==+1      ; PREVIOUS COMMAND
0162 0100      ;
0163 0100      ; ***** FROM HERE IN PAGE 1 *****
0164 0100      ;
0165 0100      ;== $100
0166 0100      ID         ;==+1      ; ID NUMBER FOR CASSETTE
0167 0101      TAPID      ;==+1      ; ID. OF CURRENT FILE
0168 0102      EAL        ;==+1      ; END OF EDITING BUFFER
0169 0103      EAH        ;==+1
0170 0104      HOURS      ;==+1      ; CLOCK LOCATIONS
0171 0105      MINUTS     ;==+1
0172 0106      SECNDS     ;==+1
0173 0107      TIMER      ;==+2      ; 10 MSEC TIMER (DECR.)
0174 0109      KEY        ;==+1
0175 010A      CURCMD     ;==+1      ; CURRENT COMMAND
0176 010B      DSPBUF     ;==+8      ; DISPLAY BUFFER (LEFT TO RIGHT)
0177 0113      TGLKEY     ;==+1      ; KEYBOARD ROLL-OVER
0178 0114      DIV1       ;==+1
0179 0115      DIV2       ;==+1
0180 0116      DECNTR     ;==+1      ; KEYBOARD DEBOUNCE
0181 0117      TEMP       ;==+2
0182 0119      ASAV       ;==+1      ; SAVE:  A
0183 011A      XSAV       ;==+1      ;           X
0184 011B      YSAV       ;==+1      ;           Y
0185 011C      INDADR     ;==+1      ; JUMP FOR COMMAND'S
0186 011C      INDADL     ;==+1
0187 011D      INDADH     ;==+1
0188 011E      CHKSML     ;==+1      ; CHECKSUM
0189 011F      CHKSMH     ;==+1
0190 0120      GKX:       ;==+1      ; AUX SAVE .X
0191 0000      .EX1
0192 0000      ;

```

```

0193 0000
0194 0000
0195 F800 A2FF RESET: LDX #FF          ; INITIAL STACK VALUE
0196 F802 9A          TXS
0197 F803 86FE        STX SPTR          ; USER STACK
0198 F805 D8          CLD
0199 F806 78          SEI
0200 F807            ; TEST FOR WARM RESTART
0201 F807 A5F0        LDA NMIVC
0202 F809 CDA0F8      CMP VECTAB
0203 F80C D007        BNE RS3A
0204 F80E A5F1        LDA NMIVC+1
0205 F810 CDA1F8      CMP VECTAB+1
0206 F813 F025        BEQ WARM          ; WARM RESTART
0207 F815            ; COLD START
0208 F815 A00D RS3A   LDY #VECLEN
0209 F817 B9A0F8 RS1  LDA VECTAB, Y
0210 F81A 99F000      STA MONRAM, Y
0211 F81D 88          DEY
0212 F81E 10F7        BPL RS1
0213 F820 A900        LDA #0
0214 F822 A205        LDX #TIMER+1-EAH
0215 F824 900301 RS2  STA EAH, X
0216 F827 CA          DEX
0217 F828 10FA        BPL RS2
0218 F82A A970        LDA #70
0219 F82C 8D0201      STA EAL
0220 F82F A901        LDA #1
0221 F831 8D0001      STA ID          ; DEFAULT = 1
0222 F834 8D1401      STA DIV1
0223 F837 8D1501      STA DIV2
0224 F83A
0225 F83A 20B5FB WARM JSR INITRM      ; INITIALIZE TERMINAL-DRIVERS
0226 F83D

```

 ### MONITOR COMMAND LOOP ###

```

0228 F83D
0229 F83D A6FE MONITR LDX SPTR
0230 F83F 9A          TXS
0231 F840 D8          CLD
0232 F841 A200        LDX #0
0233 F843 8E1301      STX TGLKEY
0234 F846 2074FA      JSR XTIME          ; DISPLAY THE TIME
0235 F849 AD1601      LDA DBCNTR
0236 F84C D0EF        BNE MONITR
0237 F84E 20B3F9      JSR GETKEY
0238 F851
0239 F851 A6FE JUMPER LDX SPTR          ; SET SYSTEM STACK
0240 F853 9A          TXS
0241 F854 0D1301      ORA TGLKEY

```

```

0242 F857 8D0A01 STA CURCMD
0243 F85A A20A LDX #NCMND5
0244 F85C JP1
0245 F85C DD7FF8 CMP CMDTAB,X
0246 F85F F006 BEQ JP2
0247 F861 0A DEX
0248 F862 10F8 BPL JP1
0249 F864 4C3DF8 JMP MONITR
0250 F867
0251 F867 8A JP2 TXA
0252 F868 0A ASL A
0253 F869 AA TAX
0254 F86A BD8AF8 LDA JMPTAB,X
0255 F86D 8D1C01 STA INDADL
0256 F870 BD8BF8 LDA JMPTAB+1,X
0257 F873 8D1D01 STA INDADH
0258 F876 207CF8 JSR CMD
0259 F879 4C3DF8 JMP MONITR
0260 F87C
0261 F87C 6C1C01 CMD JMP (INDADR)
0262 F87F
0263 F87F CMDTAB
0264 F87F 10 .BYT MEMCMD,ADRCMD,SSTCMD,GOCMD
0265 F883 15 .BYT INSCMD,DELCMD,LODCMD,DMPCMD
0266 F887 91 .BYT TIMCMD,OFFCMD,TSTCMD
0267 F88A
0268 F88A NCMND5==*-CMDTAB-1
0269 F88A
0270 F88A JMPTAB ; FUNKTIONS MAY RETURN WITH RTS
0271 F88A C5F8 .WOR SEQADR,INPADR,SSUSER,GOUSER
0272 F892 10F9 .WOR INSDCL,INSDCL,LOAD,DUMP
0273 F89A 3DF8 .WOR MONITR,OFFSET,TTERMN
0274 F8A0
0275 F8A0
0276 F8A0 AEF8 VECTAB .WOR SSTINT,IRGHDCL,TERMNL,TERMNL
0277 F8A8 0000 .WOR 0 ; PROGRAM COUNTER
0278 F8AA 00 .BYT 0,0,0,0 ; REGISTERS
0279 F8AE VECLEN ==*-VECTAB-1
0280 F8AE

```

 ### ENTRY FROM SINGLE-STEP ###

```

0282 F8AE
0283 F8AE 85FA SSTINT: STA AC
0284 F8B0 86FB STX XR
0285 F8B2 84FC STY YR
0286 F8B4 68 PLA ; STATUS
0287 F8B5 85FD STA PS
0288 F8B7 68 PLA ; PROGRAM COUNTER
0289 F8B8 85F8 STA ADL
0290 F8BA 68 PLA

```

```
0291 F88B 85F9 STA ADH
0292 F8BD BA TSX ; RP
0293 F8BE 86FE STX SPTR
0294 F8C0 AD08C0 LDA T2LL ; CLEAR SST-INT.
0295 F8C3 58 CLI
0296 F8C4 D8 CLD
0297 F8C5 ; NOW SHOW NEXT LOCATION
0298 F8C5 ;
```

INPUT DATA ON SEQ. ADDRESSES

```
0300 F8C5 2069FA SEQADR JSR CLRDSP
0301 F8C8 A200 LDX #0
0302 F8CA A5F9 LDA ADH
0303 F8CC 2054FA JSR WBYTE
0304 F8CF A5F8 LDA ADL
0305 F8D1 2054FA JSR WBYTE
0306 F8D4 A200 SQA1 LDX #0
0307 F8D6 2041FA JSR RBYTE ; GET ADDRESS
0308 F8D9 85F9 STA ADH
0309 F8DB 2041FA JSR RBYTE
0310 F8DE 85F8 STA ADL
0311 F8E0 A200 LDX #0
0312 F8E2 A1F8 LDA (ADL,X) ; AND IT'S DATA
0313 F8E4 A206 LDX #6
0314 F8E6 2054FA JSR WBYTE
0315 F8E9 2024FA JSR GETBYT
0316 F8EC 2C1301 BIT TGLKEY
0317 F8EF 100D BPL SQA2
0318 F8F1 C6F8 DEC ADL
0319 F8F3 A5F8 LDA ADL
0320 F8F5 C9FF CMP #FF
0321 F8F7 D0CC BNE SEQADR
0322 F8F9 C6F9 DEC ADH
0323 F8FB 4CC5F8 JMP SEQADR
0324 F8FE E6F8 SQA2 INC ADL ; NEXT LOCATION
0325 F900 D0C3 BNE SEQADR
0326 F902 E6F9 INC ADH
0327 F904 4CC5F8 JMP SEQADR
0328 F907 ;
```

INPUT AN ADDRESS

```
0330 F907 2069FA INPADR JSR CLRDSP
0331 F90A 20FEF9 JSR GETADR
0332 F90D 4CD4F8 JMP SQA1
0333 F910 ;
```

INSERT & DELETE

```

0335 F910 A000 INSDEL LDY #0
0336 F912 AD0201 ID0 LDA EAL
0337 F915 C5F8 CMP ADL ; TEST FOR END
0338 F917 AD0301 LDA EAH
0339 F91A E5F9 SBC ADH
0340 F91C B005 BCS ID1
0341 F91E F003 BEQ ID1
0342 F920 4CD4F8 JMP SQRA1 ; OUT
0343 F923 AD0A01 ID1 LDA CURCMD
0344 F926 C916 CMP #DELCMD
0345 F928 F00B BEQ ID0
0346 F92A B1F8 LDA (ADL),Y ; INSERT
0347 F92C 48 PHA
0348 F92D 8A TXA
0349 F92E 91F8 STA (ADL),Y
0350 F930 68 PLA
0351 F931 AA TAX
0352 F932 4C3BF9 JMP IDNEXT
0353 F935
0354 F935 C8 ID0 INY
0355 F936 B1F8 LDA (ADL),Y
0356 F938 88 DEY
0357 F939 91F8 STA (ADL),Y
0358 F93B IDNEXT
0359 F93B E6F8 INC ADL
0360 F93D D0D3 BNE ID0
0361 F93F E6F9 INC ADH
0362 F941 D0CF BNE ID0
0363 F943

```

 ### CALCULATE BRANCH OFFSET ###

```

0365 F943 A900 OFFSET LDA #0
0366 F945 8D1301 STA TGLKEY
0367 F948 2069FA JSR CLRDSP
0368 F94B 20FEF9 JSR GETADR
0369 F94E A202 LDX #2
0370 F950 2041FA JSR RBYTE
0371 F953 18 CLC
0372 F954 E5F8 SBC ADL
0373 F956 A000 LDY #0
0374 F958 91F8 STA (ADL),Y
0375 F95A 4CC5F8 JMP SEQADR
0376 F95D

```

 ### SINGLE STEP THE USER ###

```

0378 F95D ADA0F8 SSUSER LDA VECTAB ; SET NMI-VECTOR ON 'SSTINT'
0379 F960 85F0 STA NMIVC

```

```

0380 F962 AD01F8 LDA VECTAB+1
0381 F965 85F1 STA NMIVC+1
0382 F967 AD0EC0 LDA IER
0383 F96A 09A0 ORA #0A0
0384 F96C 8D0EC0 STA IER
0385 F96F AD0BC0 LDA ACR
0386 F972 29DF AND #0DF
0387 F974 8D0BC0 STA ACR
0388 F977 A924 LDA #024
0389 F979 8D08C0 STA T2LL
0390 F97C A900 LDA #0
0391 F97E 8D09C0 STA T2CH
0392 F981

```

; GENERATE 'NMI' ON OPCODE-FETCH

GO TO THE USER-PROGRAM

```

0394 F981 A6FE Gouser LDX SPTR
0395 F983 9A TXS
0396 F984 A5F9 LDA ADH
0397 F986 48 PHA
0398 F987 A5F8 LDA ADL
0399 F989 48 PHA
0400 F98A A5FD LDA PS
0401 F98C 48 PHA
0402 F98D A5FA LDA AC
0403 F98F A6FB LDX XR
0404 F991 A4FC LDY YR
0405 F993 40 RTI
0406 F994

```

; DEFINE STACK

ENTER MONITOR FROM "BRK"

```

0408 F994 85FA BKUSER: STA AC
0409 F996 86FB STX XR
0410 F998 84FC STY YR
0411 F99A 68 PLA
0412 F99B 85FD STA PS
0413 F99D D8 CLD
0414 F99E 68 PLA
0415 F99F 38 SEC
0416 F9A0 E901 SBC #1
0417 F9A2 85F8 STA ADL
0418 F9A4 68 PLA
0419 F9A5 E900 SBC #0
0420 F9A7 85F9 STA ADH
0421 F9A9 BA TSX
0422 F9AA 86FE STX SPTR
0423 F9AC AD08C0 LDA T2LL
0424 F9AF 58 CLI
0425 F9B0 4CC5F8 JMP SEQADR

```

; STATUS

; CORRECT 1 FROM 'BRK'
; PC

; CLEAR TIMER2

; SHOW NEXT LOCATION

0426 F9B3

 ### GET A KEY FROM KEYBOARD ###

```

0428 F9B3 8E2001 GETKEY STX GKX          ; SAVE .X
0429 F9B6 AD1601 GK2    LDA DBCNTR        ; DEBOUNCE
0430 F9B9 D0FB          BNE GK2
0431 F9BB A903          LDA #3
0432 F9BD 8D1601        STA DBCNTR
0433 F9C0 AE0901        LDX KEY
0434 F9C3 BDD9F9        LDA KEYVAL,X      ; GET KEY-CODE
0435 F9C6 C913          CMP #CMD2ND      ; '2ND'-KEY
0436 F9C8 D00B          BNE GK1
0437 F9CA A980          LDA #MODMSK
0438 F9CC 4D1301        EOR TGLKEY
0439 F9CF 8D1301        STA TGLKEY
0440 F9D2 4CB6F9        JMP GK2
0441 F9D5 AE2001 GK1    LDX GKX          ; RESTORE .X
0442 F9D8 60            RTS
0443 F9D9              ;
0444 F9D9              KEYVAL              ;
0445 F9D9 11            . BYT ADRCMD, 0, MEMCMD, SSTCMD ; KEYBOARD FROM BOTTOMLINE UI
0446 F9DD 01            . BYT 1, 2, 3, GO CMD
0447 F9E1 04            . BYT 4, 5, 6, $F
0448 F9E5 07            . BYT 7, 8, 9, $E
0449 F9E9 0A            . BYT $A, $B, $C, $D
0450 F9ED 13            . BYT CMD2ND, INSCMD, DELCMD, LODCMD
0451 F9F1              ;
0452 F9F1              ;

```

 ### GET HEX KEY OR DO COMMAND ###

```

0454 F9F1 20B3F9 GETHEX JSR GETKEY
0455 F9F4 C910          CMP #ENTER
0456 F9F6 3005          BMI GH1
0457 F9F8 F003          BEQ GH1
0458 F9FA 4C51F8        JMP JUMPER
0459 F9FD 60            GH1              RTS
0460 F9FE              ;

```

 ### GET 4 DIGIT HEX ADDRESS ###

```

0462 F9FE A900          GETADR LDA #0
0463 FA00 AA            TAX
0464 FA01 2095FA GA0    JSR WCHAR        ; CLEAR ADR-FIELD
0465 FA04 E8            INX
0466 FA05 E004          CPX #4
0467 FA07 30F8          BMI GA0
0468 FA09 20F1F9 GA1    JSR GETHEX

```

```

0459 FA0C F015      BEQ GAEND
0470 FA0E A200      LDX #0
0471 FA10 48        PHA
0472 FA11 B00C01 GA2    ; SAVE DATA
0473 FA14 900B01      LDA DSPBUF+1,X ; LEFT-SHIFT 4 DIGITS
0474 FA17 E8        STA DSPBUF,X
0475 FA18 E003      INX
0476 FA1A 30F5      CPX #3
0477 FA1C 68        BMI GA2
0478 FA1D 2095FA      PLA
0479 FA20 18        JSR WCHAR
0480 FA21 90E6      CLC
0481 FA23 60        BCC GA1
0482 FA24          RTS

```

GET BYTE IN DATA-FIELD

```

0484 FA24 20F1F9 GETBYT JSR GETHEX
0485 FA27 F00E      BEQ GBEND
0486 FA29 AE1201      LDX DSPBUF+7 ; LEFT-SHIFT ON DATA-FIELD
0487 FA2C 8E1101      STX DSPBUF+6
0488 FA2F A207      LDX #7
0489 FA31 2095FA      JSR WCHAR
0490 FA34 18        CLC
0491 FA35 90ED      BCC GETBYT
0492 FA37 A206      LDX #6
0493 FA39 2041FA      JSR RBYTE
0494 FA3C A200      LDX #0
0495 FA3E 81F8      STA (ADL,X)
0496 FA40 60        RTS
0497 FA41

```

READ A BYTE FROM DISPL(X)

```

0499 FA41 20A4FA RBYTE JSR RCHAR
0500 FA44 0A        ASL A
0501 FA45 0A        ASL A
0502 FA46 0A        ASL A
0503 FA47 0A        ASL A
0504 FA48 8D1801      STA TEMP+1
0505 FA4B E8        INX
0506 FA4C 20A4FA      JSR RCHAR
0507 FA4F 0D1801      ORA TEMP+1
0508 FA52 E8        INX
0509 FA53 60        RTS
0510 FA54

```

WRITE BYTE TO DISPL(X)

```
0512 FA54 801801 WBYTE STA TEMP+1
0513 FA57 4A LSR A
0514 FA58 4A LSR A
0515 FA59 4A LSR A
0516 FA5A 4A LSR A
0517 FA5B 2095FA JSR WCHAR
0518 FA5E E8 INX
0519 FA5F AD1801 LDA TEMP+1
0520 FA62 290F AND #0F
0521 FA64 2095FA JSR WCHAR
0522 FA67 E8 INX
0523 FA68 60 RTS
0524 FA69
```

CLEAR DISPLAY TO BLANKS

```
0526 FA69 A900 CLRDSP LDA #0
0527 FA6B A207 FILDSP LDX #7
0528 FA6D 9D0B01 CD1 STA DSPBUF,X
0529 FA70 CA DEX
0530 FA71 10FA BPL CD1
0531 FA73 60 RTS
0532 FA74
```

PUT THE TIME ON DISPLAY

```
0534 FA74 A200 XTIME LDX #0
0535 FA76 AD0401 LDA HOURS
0536 FA79 2054FA JSR WBYTE
0537 FA7C A900 LDA #0
0538 FA7E 9D0B01 STA DSPBUF,X
0539 FA81 E8 INX
0540 FA82 AD0501 LDA MINUTS
0541 FA85 2054FA JSR WBYTE
0542 FA88 A900 LDA #0
0543 FA8A 9D0B01 STA DSPBUF,X
0544 FA8D E8 INX
0545 FA8E AD0601 LDA SECONDS
0546 FA91 2054FA JSR WBYTE
0547 FA94 60 RTS
0548 FA95
```

WRITE CHR TO DISPL(X)

```
0550 FA95 8C1B01 WCHAR STY YSAV
0551 FA98 A3 TAY
0552 FA99 B911FD LDA CHASET,Y
0553 FA9C 9D0B01 STA DSPBUF,X
0554 FA9F 98 TYA
```

```
0555 FAA0 AC1B01 LDY YSAV
0556 FAA3 60 RTS
0557 FAA4
```

READ CHR FROM DISPL(X)

```
0559 FAA4 8E1A01 RCHAR STX XSAV
0560 FAA7 BD0B01 LDA DSPBUF,X
0561 FAAA A213 LDX #NCHARS
0562 FAAC DD11FD RC1 CMP CHASET,X
0563 FAAF F004 BEQ RC2
0564 FAB1 CA DEX
0565 FAB2 10F8 BPL RC1
0566 FAB4 E8 INX
0567 FAB5 8A RC2 TXA
0568 FAB6 AE1A01 LDX XSAV
0569 FAB9 60 RTS
0570 FAB8
0571 FAB8
0572 FAB8 ; #### TEST HEX-TERMINAL ####
0573 FAB8
0574 FAB8 ; ..TEST THE DIGIT-STROBE D0-D7..
0575 FAB8
0576 FAB8 A207 TDIGIT LDX #7 ; SHIFT "8." FROM RIGHT TO LEFT
0577 FABC 8E1801 TDIG1 STX TEMP+1
0578 FADF 2069FA JSR CLRDSP
0579 FAC2 AE1801 LDX TEMP+1
0580 FAC5 A9FF LDA #FF
0581 FAC7 9D0B01 STA DSPBUF,X
0582 FACA 2063FB JSR DLY500 ; DELAY 500 MSEC
0583 FACD CA DEX
0584 FACE 10EC BPL TDIG1
0585 FAD0
0586 FAD0 ; ..TEST SEGMENTS A...G & DP
0587 FAD0
0588 FAD0 A901 TSEGM LDA #1
0589 FAD2 8D1801 TSEGM1 STA TEMP+1 ; ROTATE SEGM.
0590 FAD5 206BFA JSR FILDSP
0591 FAD8 2063FB JSR DLY500
0592 FADB 0A ASL A
0593 FADC B000 BCS TSEGM9
0594 FADE A8 TAY
0595 FADF 0D1801 ORA TEMP+1
0596 FAE2 206BFA JSR FILDSP
0597 FAE5 2063FB JSR DLY500
0598 FAE8 98 TYA
0599 FAE9 D0E7 BNE TSEGM1
0600 FAEB 60 TSEGM9 RTS
0601 FAEC
0602 FAEC ; ..TEST LED-I/O BOARD..
0603 FAEC
```

```

0604 FAEC A9FF TLED LDA #FF
0605 FAEE 8D12C0 STA DDRB1
0606 FAF1 8D10C0 STA DRB1
0607 FAF4 A9EE LDA #EE ; CA2 & CB2 OUTPUT
0608 FAF6 8D10C0 STA PCR1
0609 FAF9 2063FB LDB1 JSR DLY500
0610 FAFC 0E10C0 ASL DRB1
0611 FAFF B0F8 BCS LDB1
0612 FB01 A901 LDA #1
0613 FB03 8D10C0 STA DRB1
0614 FB06 2063FB LDB2 JSR DLY500
0615 FB09 AD10C0 LDA PCR1
0616 FB0C 4922 EOR #22
0617 FB0E 8D10C0 STA PCR1
0618 FB11 0E10C0 ASL DRB1
0619 FB14 90F0 BCC LDB2
0620 FB16 ; ... NOW READ SWITCHES TO LEDS ...
0621 FB16 AD11C0 LDB3 LDA DRA1
0622 FB19 8D10C0 STA DRB1
0623 FB1C AD1601 LDA DBCNTR
0624 FB1F D0F5 BNE LDB3 ; UNTIL A KEY
0625 FB21 A905 LDA #5
0626 FB23 8D1601 STA DBCNTR
0627 FB26 60 RTS
0628 FB27 ;

```

ROM TEST

```

0630 FB27 ;
0631 FB27 ;
0632 FB27 ;
0633 FB27 ROMT: ; START ROM TEST
0634 FB27 D8 CLD
0635 FB28 A900 LDA #0
0636 FB2A 8D1E01 STA CHKSM1 ; CLEAR CHECKSUM
0637 FB2D 8D1F01 STA CHKSM1+1
0638 FB30 85F8 STA ADL
0639 FB32 A9F8 LDA #>ROMBAS
0640 FB34 85F9 STA ADL+1
0641 FB36 ;
0642 FB36 ADD ; THE ADD ROUTINE
0643 FB36 A000 LDY #0
0644 FB38 10 CLC
0645 FB39 AD1E01 LDA CHKSM1
0646 FB3C 71F8 ADC (ADL),Y ; ADD LOCATION TO CHECKSM1
0647 FB3E 8D1E01 STA CHKSM1
0648 FB41 AD1F01 LDA CHKSM1+1
0649 FB44 6900 ADC #0
0650 FB46 8D1F01 STA CHKSM1+1
0651 FB49 ; INC ADL
0652 FB49 E6F8 INC ADL

```

```

0653 FB4B D0E9      BNE ADD
0654 FB4D E6F9      INC ADL+1
0655 FB4F A5F9      LDA ADL+1
0656 FB51 C900      CMP #DREND      ; HIGH BYTE TEST
0657 FB53 D0E1      BNE ADD
0658 FB55 A204      LDX #4
0659 FB57 AD1F01    LDA CHKSMH      ; PUT CHECKSUM ON DISPL
0660 FB5A 2054FA    JSR WBYTE
0661 FB5D AD1E01    LDA CHKSM1
0662 FB60 4C54FA    JMP WBYTE
0663 FB63           ;
0664 FB63           ;
0665 FB63           ; DELAY 500 MSEC
0666 FB63           ;
0667 FB63 48      DLY500  PHA      ; PRESERVE ALL REGS.
0668 FB64 A932      LDA #50
0669 FB66 8D0701    STA TIMER
0670 FB69 AD0701 DLY501  LDA TIMER
0671 FB6C D0FB      BNE DLY501
0672 FB6E 68        PLA
0673 FB6F 60        RTS
0674 FB70           ;
0675 FB70 2069FA TKEYB  JSR CLRDSP      ; CLEAR DISPLAY
0676 FB73 A940      LDA #$40
0677 FB75 8D0B01    STA DSPBUF
0678 FB78 8D0C01    STA DSPBUF+1
0679 FB7B AD1601 TSTKB1 LDA DBCNTR
0680 FB7E D0FB      BNE TSTKB1
0681 FB80 AD0901    LDA KEY      ; ECHO KEY-NUMBER TO DISPLAY
0682 FB83 4A        LSR A
0683 FB84 4A        LSR A
0684 FB85 4A        LSR A
0685 FB86 4A        LSR A
0686 FB87 AA        TAX
0687 FB88 BD11FD    LDA CHASET, X
0688 FB8B 8D0B01    STA DSPBUF
0689 FB8E AD0901    LDA KEY
0690 FB91 290F      AND #$0F
0691 FB93 AA        TAX
0692 FB94 BD11FD    LDA CHASET, X
0693 FB97 8D0C01    STA DSPBUF+1
0694 FB9A A903      LDA #$03
0695 FB9C 8D1601    STA DBCNTR
0696 FB9F 10DA      BPL TSTKB1
0697 FBA1           ;
0698 FBA1 2069FA TTERMN JSR CLRDSP
0699 FBA4 A900      LDA #0
0700 FBA6 8D1301    STA TGLKEY
0701 FBA9 2027FB    JSR ROMT
0702 FBAC 20ECFA    JSR TLED
0703 FBAF 20BAFA    JSR TDIGIT
0704 FBB2 4C70FB    JMP TKEYB

```

05 FBB5 ;
0706 FBB5 ;
0707 FBB5 ;

 ### KEYBOARD-DISPLAY DRIVER ###

```

0709 FBB5      ;
0710 FBB5      ;
0711 FBB5      ;
0712 FBB5      ; *****
0713 FBB5      ; *****
0714 FBB5      ; *****
0715 FBB5      ; ***** TERMINAL-ROUTINE. *****
0716 FBB5      ; *****
0717 FBB5      ; ***** DRIVE LED-DISPLAYS *****
0718 FBB5      ; ***** SCAN KEYBOARD *****
0719 FBB5      ; ***** UPDATE CLOCK *****
0720 FBB5      ; *****
0721 FBB5      ; *****
0722 FBB5      ; *****
0723 FBB5      ;
0724 FBB5      ;
0725 FBB5      ;
0726 FBB5      DSPON      =#07
0727 FBB5      DSPOFF     =#08
0728 FBB5      T1IMSK     =%11000000
0729 FBB5      IFREQ      =2000-2
0730 FBB5      ;
0731 FBB5      A90F      INITRM:  LDA #$0F
0732 FBB7      8D03C0     STA DDRA
0733 FBB8      A9FF      LDA #$FF
0734 FBB8      8D02C0     STA DDRB
0735 FBBF      A908      LDA #DSPOFF
0736 FBC1      8D01C0     STA DRA
0737 FBC4      A9C0      LDA #$C0      ; CB2 LOW
0738 FBC6      8D00C0     STA PCR      ; TO FREE PB7
0739 FBC9      A940      LDA #$40
0740 FBCB      8D08C0     STA ACR
0741 FBCE      A9CE      LDA #<IFREQ
0742 FBD0      8D06C0     STA T1LL
0743 FBD3      A907      LDA #>IFREQ
0744 FBD5      8D05C0     STA T1CH
0745 FBD8      A9F0      LDA #$F0
0746 FBDA      8D1601     STA DBCNTR
0747 FBDD      A97F      LDA #$7F
0748 FBDF      8D0EC0     STA IER
0749 FBE2      A9C0      LDA #T1IMSK
0750 FBE4      8D0EC0     STA IER
0751 FBE7      8D00C0     STA IFR
0752 FBEB      58         CLI
0753 FBEB      60         RTS
0754 FBEC      ;
0755 FBEC      ; ### HANDLER FOR IRQ ###
0756 FBEC      8D1701     IRQHDL: STA TEMP
  
```

```

0757 FBEF 68          PLA
0758 FBF0 48          PHA
0759 FBF1 2910        AND ##10
0760 FBF3 D01A        BNE BRKINT
0761 FBF5 AD0DC0      LDA IFR
0762 FBF8 2920        AND ##20
0763 FBFA D019        BNE STEPIN          ; SST-INTERRUPT
0764 FBFC AD0DC0      LDA IFR
0765 FBFF 29C0        AND #T1IMSK        ; T1 INT ??
0766 FC01 D006        BNE DSINT
0767 FC03 AD1701      LDA TEMP
0768 FC06 6CF600      JMP (IROVEC)
0769 FC09             ;
0770 FC09 AD1701 DSINT LDA TEMP
0771 FC0C 6CF400      JMP (DISVEC)
0772 FC0F             ;
0773 FC0F AD1701 BRKINT LDA TEMP
0774 FC12 4C94F9      JMP BKUSER
0775 FC15             ;
0776 FC15             ; #### HANDLER FOR NMI ####
0777 FC15             ;
0778 FC15 AD1701 STEPIN LDA TEMP
0779 FC18 6CF000 NMI: JMP (NMIVEC)
0780 FC1B             ;
0781 FC1B             ;
0782 FC1B             ; #### DISPLAY MULTIPLEXER ####
0783 FC1B 48          TERMNL: PHA
0784 FC1C 8A          TXA
0785 FC1D 48          PHA
0786 FC1E 98          TYA
0787 FC1F 48          PHA
0788 FC20 A908        LDA #DSPOFF
0789 FC22 8D01C0      ORA DRA
0790 FC25 8D01C0      STA DRA
0791 FC28 205AFC      JSR SCNKYB
0792 FC2B 2045FC      JSR DSPLAY
0793 FC2E 20B9FC      JSR XCLOCK
0794 FC31 A907        LDA #DSPON
0795 FC33 2D01C0      AND DRA
0796 FC36 8D01C0      STA DRA
0797 FC39 AD0DC0      LDA IFR
0798 FC3C 8D0DC0      STA IFR
0799 FC3F 68          PLA
0800 FC40 A8          TAY
0801 FC41 68          PLA
0802 FC42 AA          TAX
0803 FC43 68          PLA
0804 FC44 40          RTI
0805 FC45             ;
0806 FC45 EE01C0 DSPLAY INC DRA
0807 FC48 AD01C0      LDA DRA
0808 FC4B 2907        AND ##07

```

```

0809 FC4D RA TAX
0810 FC4E BD0B01 LDA DSPBUF,X
0811 FC51 0D1301 ORA TGLKEY
0812 FC54 49FF EOR #$FF ; ***** FOR INVERTED SEGMENTS
0813 FC56 8D00C0 STA DRB
0814 FC59 60 RTS
0815 FC5A ;
0816 FC5A ; DEBOUNCE CONSTANTS
0817 FC5A ;
0818 FC5A DDBNC =10 ; * IFREQ (MS.)
0819 FC5A UDBNC =$FF-DDBNC
0820 FC5A ;
0821 FC5A ; ### SCAN KEYBOARD ###
0822 FC5A AD1601 SCNKYB LDA DBCNTR
0823 FC5D F053 BEQ SCEND
0824 FC5F ;
0825 FC5F A205 LDX #5
0826 FC61 B0B3FC SCROW LDA ROW,X
0827 FC64 8D00C0 STA DRB
0828 FC67 AD01C0 LDA DRA
0829 FC6A 8D1701 STA TEMP
0830 FC6D A003 LDY #3
0831 FC6F 2C1701 SCCOL BIT TEMP
0832 FC72 1019 BPL DOWN
0833 FC74 0E1701 ASL TEMP
0834 FC77 88 DEY
0835 FC78 10F5 BPL SCCOL
0836 FC7A CA DEX
0837 FC7B 10E4 BPL SCROW
0838 FC7D ;
0839 FC7D 2C1601 UP BIT DBCNTR
0840 FC80 3027 BMI SC1
0841 FC82 CE1601 DEC DBCNTR
0842 FC85 D02B BNE SCEND
0843 FC87 A9F5 LDA #UDBNC ; DEBOUNCE KEY-RELEASE
0844 FC89 8D1601 STA DBCNTR
0845 FC8C 60 RTS
0846 FC8D ;
0847 FC8D 2C1601 DOWN BIT DBCNTR
0848 FC90 3006 BMI DECODE
0849 FC92 A90A LDA #DDBNC
0850 FC94 8D1601 STA DBCNTR
0851 FC97 60 RTS
0852 FC98 8A DECODE TXA ; ROW-COUNT
0853 FC99 0A ASL A
0854 FC9A 0A ASL A ; * 4
0855 FC9B 8C1701 STY TEMP
0856 FC9E 0D1701 ORA TEMP
0857 FCA1 CD0901 CMP KEY
0858 FCA4 F009 BEQ SC2
0859 FCA6 8D0901 STA KEY
0860 FCA9 A9F5 SC1 LDA #UDBNC

```

```

0861 FCAB 8D1601 STA DBCNTR
0862 FCAC 60 RTS
0863 FCAD EE1601 SC2 INC DBCNTR
0864 FCBE 60 SCEND RTS
0865 FCB3
0866 FCB3 02 ROW . BYTE $02, $04, $08
0867 FCB6 10 . BYTE $10, $20, $30
0868 FCB9

```

 ### CLOCK ROUTINE ###

```

0870 FCB9 CE1401 XCLOCK: DEC DIV1
0871 FCBC D051 BNE CLKEND
0872 FCBE A905 LDA #5
0873 FCC0 8D1401 STA DIV1 ; 10 MSEC
0874 FCC3 AD0701 LDA TIMER
0875 FCC6 D008 BNE XCL1 ; DECREMENT TIMER UNTIL '0000'
0876 FCC8 AD0801 LDA TIMER+1
0877 FCCB F006 BEQ XCL2
0878 FCCD CE0801 DEC TIMER+1
0879 FCC0 CE0701 XCL1 DEC TIMER
0880 FCD3 CE1501 XCL2 DEC DIV2
0881 FCD6 D037 BNE CLKEND
0882 FCD8 A964 LDA #100
0883 FCDA 8D1501 STA DIV2
0884 FCDD F8 SED
0885 FCDE A001 LDY #1
0886 FCE0 A200 LDX #0
0887 FCE2 18 CLC
0888 FCE3 98 TYA
0889 FCE4 6D0601 ADC SECNDS
0890 FCE7 8D0601 STA SECNDS
0891 FCFA C960 CMP #$60
0892 FCEC 3021 BMI CLKEND
0893 FCEE 18 CLC
0894 FCEF 8E0601 STX SECNDS
0895 FCF2 98 TYA
0896 FCF3 6D0501 ADC MINUTS
0897 FCF6 8D0501 STA MINUTS
0898 FCF9 C960 CMP #$60
0899 FCFB 3012 BMI CLKEND
0900 FCFD 18 CLC
0901 FCFE 8E0501 STX MINUTS
0902 FDB1 98 TYA
0903 FDB2 6D0401 ADC HOURS
0904 FDB5 8D0401 STA HOURS
0905 FDB8 C924 CMP #$24
0906 FDBA 3003 BMI CLKEND
0907 FDBC 8E0401 STX HOURS
0908 FDBF D8 CLKEND CLD
0909 FDB1 60 RTS

```

0910 FD11

DISPLAY 7 SEGMENTS CODE

0912	FD11	3F	CHASET	. BYTE \$3F, \$06, \$5B	
0913	FD14	4F		. BYTE \$4F, \$66, \$6D	
0914	FD17	7D		. BYTE \$7D, \$07, \$7F	
0915	FD1A	6F		. BYTE \$6F, \$77, \$7C	
0916	FD1D	39		. BYTE \$39, \$5E, \$79	
0917	FD20	71		. BYTE \$71	
0918	FD21	00		. BYTE \$00	; BLANK
0919	FD22	80		. BYTE \$80	; DOT
0920	FD23	48		. BYTE \$48	; EQUAL
0921	FD24				
0922	FD24		NCHARS:	==*-CHASET	
0923	FD24				

 ### CASSETTE DUMP ###

```

0925 FD24      ;
0926 FD24      ; *****
0927 FD24      ; *****
0928 FD24      ; ***** DUMP MEMORY TO TAPE *****
0929 FD24      ; ***** IN KIM-6502 FORMAT *****
0930 FD24      ; *****
0931 FD24      ; *****
0932 FD24      ;
0933 FD24      ;
0934 FD24      F3700      =138      ; FOR 3700 HERTZ
0935 FD24      F2400      =207      ; FOR 2400 HERTZ
0936 FD24      ;
0937 FD24      ;

```

 ### DUMP MAIN ROUTINE ###

```

0939 FD24 78      DUMP:      SEI
0940 FD25 2030FD      JSR DMPOPN
0941 FD28 207DFD      JSR DMPMEM
0942 FD2B 209BFD      JSR DMPCLS
0943 FD2E 58          CLI
0944 FD2F 60          RTS
0945 FD30      ;
0946 FD30      ; ### OPEN FOR DUMP ###
0947 FD30 2037FD DMPOPN      JSR DMPINI
0948 FD33 2052FD      JSR DMPHED
0949 FD36 60          RTS
0950 FD37      ;
0951 FD37      ; ### INIT HARDWARE FOR DUMP ###
0952 FD37 A9C0      DMPINI      LDA #C0      ; T1 FREE-RUNNING, PB7 PULS
0953 FD39 8D0BC0      STA ACR
0954 FD3C A9C0      LDA #C0      ; CB2 LOW
0955 FD3E 8D0CC0      STA PCR
0956 FD41 A908      LDA #DSPOFF      ; DISPLAY OFF
0957 FD43 8D01C0      STA DRA
0958 FD46 A900      LDA #0
0959 FD48 8D1E01      STA CHKSML
0960 FD4B 8D1F01      STA CHKSMH
0961 FD4E 8D05C0      STA T1CH
0962 FD51 60          RTS
0963 FD52      ;
0964 FD52      ; ### DUMP HEADER ###
0965 FD52 A264      DMPHED      LDX #100
0966 FD54 8E1701      STX TEMP
0967 FD57 A916      LDA #16
0968 FD59 2007FD NXTSNC      JSR DMPACC      ; PRE-AMBLE
0969 FD5C CE1701      DEC TEMP

```

```

0970 FD5F D0F8      BNE NXTSNC
0971 FD61
0972 FD61 A92A      LDA #'*
0973 FD63 20D7FD    JSR DMPACC      ; TRIGGER CHR
0974 FD66 AD0001    LDA ID
0975 FD69 20CAFD    JSR DMPBYT      ; ID. NUMBER
0976 FD6C A5F8      LDA ADL
0977 FD6E 20B8FD    JSR ADDCK      ; START ADDRESS
0978 FD71 20CAFD    JSR DMPBYT
0979 FD74 A5F9      LDA ADH
0980 FD76 20B8FD    JSR ADDCK
0981 FD79 20CAFD    JSR DMPBYT
0982 FD7C 60        RTS
0983 FD7D
0984 FD7D      ; ### DUMP BODY (MEMORY) ###
0985 FD7D A000      DMPMEM    LDY #0
0986 FD7F B1F8      LDA (ADL),Y
0987 FD81 20B8FD    JSR ADDCK
0988 FD84 20CAFD    JSR DMPBYT
0989 FD87 E6F8      INC ADL
0990 FD89 D002      BNE DM1
0991 FD8B E6F9      INC ADH
0992 FD8D 38        DM1      SEC
0993 FD8E AD0201    LDA EAL
0994 FD91 E5F8      SBC ADL
0995 FD93 AD0301    LDA EAH
0996 FD96 E5F9      SBC ADH
0997 FD98 B0E3      BCS DMPMEM
0998 FD9A 60        RTS
0999 FD9B
1000 FD9B      ; ### DUMP END-RECORD ###
1001 FD9B A92F      DMPCLS    LDA #' /
1002 FD9D 20D7FD    JSR DMPACC
1003 FDA0 AD1E01    LDA CHKSMH
1004 FDA3 20CAFD    JSR DMPBYT      ; CHECKSUM
1005 FDA6 AD1F01    LDA CHKSMH
1006 FDA9 20CAFD    JSR DMPBYT
1007 FDAC A904      LDA #04
1008 FDAE 20D7FD    JSR DMPACC      ; 'EOT'
1009 FDB1 20D7FD    JSR DMPACC
1010 FDB4 20B5FB    JSR INITRM      ; SET-UP FOR DISPLAY
1011 FDB7 60        RTS
1012 FDB8
1013 FDB8      ; ### ADD BYTE TO CHECKSUM ###
1014 FDB8 48        ADDCK     PHA
1015 FDB9 18        CLC
1016 FDBA 6D1E01    ADC CHKSMH
1017 FDBD 8D1E01    STA CHKSMH
1018 FDC0 AD1F01    LDA CHKSMH
1019 FDC3 6900      ADC #0
1020 FDC5 8D1F01    STA CHKSMH
1021 FDC8 68        PLA

```

```

1022 FDC9 60          RTS
1023 FDCA          ;
1024 FDCA          ; ### DUMP ONE BYTE ###
1025 FDCA 200DFE DMPBYT JSR HXASCH ; HEX --> ASCII HIGH-BYTE
1026 FDCD 20D7FD      JSR DMPACC
1027 FDD0 2016FE      JSR HXASCL ; HEX --> ASCII LOW-BYTE
1028 FDD3 20D7FD      JSR DMPACC
1029 FDD6 60          RTS
1030 FDD7          ;
1031 FDD7          ; ### DUMP ACCUMULATOR ###
1032 FDD7 A208 DMPACC LDX #8
1033 FDD9 48          PHA
1034 FDDA 48          PHA
1035 FDDB 18          NXTBIT CLC
1036 FDDC 20EFFD      JSR CPULSE ; 3700 HZ
1037 FDDF 68          PLA
1038 FDE0 4A          LSR A
1039 FDE1 48          PHA
1040 FDE2 20EFFD      JSR CPULSE ; 3700 OR 2400 HZ
1041 FDE5 38          SEC
1042 FDE6 20EFFD      JSR CPULSE ; 2400 HZ
1043 FDE9 0A          DEX
1044 FDEA D0EF        BNE NXTBIT
1045 FDEC 68          PLA
1046 FDED 68          PLA
1047 FDEE 60          RTS
1048 FDEF          ;
1049 FDEF          ; ### MAKE A PULSE ###
1050 FDEF A900 CPULSE LDA #0 ; BIT IS IN CARRY
1051 FDF1 2A          ROL A
1052 FDF2 A9          TAY
1053 FDF3 B909FE      LDA FREQ.Y
1054 FDF6 8D06C0      STA T1LL
1055 FDF9 B90BFE      LDA NPLS.Y
1056 FDFC A8          TAY
1057 FDFD 2C0DC0 WAITPL BIT IFR
1058 FE00 50FB        BVC WAITPL
1059 FE02 AD04C0      LDA T1CL ; CLEAR INT. FLAG
1060 FE05 88          DEY
1061 FE06 D0F5        BNE WAITPL
1062 FE08 60          RTS
1063 FE09          ;
1064 FE09 88          FREQ .BYTE F3700-2
1065 FE0A 0D          .BYTE F2400-2
1066 FE0B 12          NPLS .BYTE 18 ; NUMBER OF HALF-PULSES
1067 FE0C 0C          .BYTE 12
1068 FE0D          ;
1069 FE0D          ; ### CONVERSION HEX --> ASCII ###
1070 FE0D 8D1801 HXASCH STA TEMP+1 ; ENTRY FOR HIGH-BYTE
1071 FE10 4A          LSR A
1072 FE11 4A          LSR A
1073 FE12 4A          LSR A

```

1074	FE13	4A	LSR A	
1075	FE14	1005	BPL HEXASC	
1076	FE16	AD1801	HXASCL	LDA TEMP+1 ; ENTRY FOR LOW-BYTE
1077	FE19	290F	AND ##0F	
1078	FE1B		HEXASC	
1079	FE1B	C90A	CMP ##0A	
1080	FE1D	9002	BCC HA1	
1081	FE1F	6906	ADC #6	
1082	FE21	6930	HA1	ADC ##30
1083	FE23	60	RTS	
1084	FE24			

 ### CASSETTE LOAD ###

```

1086 FE24      ;
1087 FE24      ; *****
1088 FE24      ; *****
1089 FE24      ; *****
1090 FE24      ; ***** LOAD MEMORY FROM TAPY *****
1091 FE24      ; ***** IN 'KIM-6502' FORMAT *****
1092 FE24      ; *****
1093 FE24      ; *****
1094 FE24      ; *****
1095 FE24      ;
1096 FE24      ;
  
```

 ### LOAD MAIN ROUTINE ###

```

1098 FE24 78      LOAD: SEI
1099 FE25 A908     LDA #DSPOFF      ; DISPLAY OFF
1100 FE27 8D01C0   STA DRA
1101 FE2A A9E0     LDA #$E0        ; CB2 HIGH
1102 FE2C 8D0CC0   STA PCR
1103 FE2F A97F     LDA #$7F        ; PB7 FOR INPUT
1104 FE31 8D02C0   STA DDRB
1105 FE34 A900     LDA #0
1106 FE36 8D0BC0   STA ACR        ; T2 ONE-SHOT
1107 FE39 8D1E01   STA CHKSML     ; CLEAR CHECKSUM
1108 FE3C 8D1F01   STA CHKSMH
1109 FE3F      ;
1110 FE3F      ; ### READ THE SYNC-BYTE'S ###
1111 FE3F 2008FF   LODSNC JSR RDBIT
1112 FE42 6A      ROR A
1113 FE43 C916     CMP #$16
1114 FE45 D0F8     BNE LODSNC
1115 FE47 A909     LDA #9
1116 FE49 8D1801   STA TEMP+1
1117 FE4C 20FCFE   LS1 JSR LDACC
1118 FE4F C916     CMP #$16
1119 FE51 D0EC     BNE LODSNC
1120 FE53 CE1801   DEC TEMP+1
1121 FE56 10F4     BPL LS1
1122 FE58      ;
1123 FE58      ; ### FIND TRIGGER-WORD ###
1124 FE58 20FCFE   LODSTR JSR LDACC
1125 FE5B C92A     CMP #'*
1126 FE5D F006     BEQ LS2
1127 FE5F C916     CMP #$16
1128 FE61 F0F5     BEQ LODSTR
1129 FE63 D0DA     BNE LODSNC
1130 FE65 20D1FE   LS2 JSR RDBYTE
  
```

```

1131 FE68 8D0101 STA TAPID ; LEAVE CURRENT ID.
1132 FE6B CD0001 CMP ID
1133 FE6E F01A BEQ LODSAD ; LOAD EVERY FILE
1134 FE70 AD0001 LDA ID
1135 FE73 C900 CMP #0
1136 FE75 F013 BEQ LODSAD
1137 FE77 C9FF CMP #$FF ; ID=$FF /RELOCATE & LOAD
1138 FE79 D0C4 BNE LODSNC ; FIND NEXT FILE
1139 FE7B
1140 FE7B 20D1FE JSR RDBYTE
1141 FE7E 20B8FD JSR ADDCK
1142 FE81 20D1FE JSR RDBYTE
1143 FE84 20B8FD JSR ADDCK
1144 FE87 4C9AFE JMP LODDAT
1145 FE8A
1146 FE8A ; ### GET START ADDRESS ###
1147 FE8A 20D1FE LODSAD JSR RDBYTE
1148 FE8D 20B8FD JSR ADDCK
1149 FE90 85F8 STA ADL
1150 FE92 20D1FE JSR RDBYTE
1151 FE95 20B8FD JSR ADDCK
1152 FE98 85F9 STA ADH
1153 FE9A
1154 FE9A ; ### LOAD DATA TO MEM ###
1155 FE9A 20D1FE LODDAT JSR RDBYTE
1156 FE9D B00F BCS LODEND ; *EOF*
1157 FE9F 20B8FD JSR ADDCK
1158 FEA2 A000 LDY #0
1159 FEA4 91F8 STA (ADL),Y
1160 FEA6 E6F8 INC ADL
1161 FEA8 D0F0 BNE LODDAT
1162 FEAR E6F9 INC ADH
1163 FEAC D0EC BNE LODDAT
1164 FEAE
1165 FEAE ; ### TEST CHECKSUM ###
1166 FEAE 20D1FE LODEND JSR RDBYTE
1167 FEB1 CD1E01 CMP CHKSM1
1168 FEB4 D00F BNE LE1
1169 FEB6 20D1FE JSR RDBYTE
1170 FEB9 CD1F01 CMP CHKSMH
1171 FEBC D007 BNE LE1
1172 FEBE A900 LDA #0
1173 FEC0 8D1301 STA TGLKEY
1174 FEC3 F005 BEQ LE2
1175 FEC5 A980 LE1 LDA #$80
1176 FEC7 8D1301 STA TGLKEY
1177 FECA AD08C0 LE2 LDA T2LL ; CLEAR T2 INT. FOR "SST"
1178 FECD 20B5FB JSR INITRM ; SET VIA FOR TERMINAL
1179 FED0 60 RTS
1180 FED1
1181 FED1 ; ### READ ONE BYTE ###
1182 FED1 20E7FE RDBYTE JSR RDHEX

```

```

1183 FED4 B010          BCS RDB1          ; *EOF*
1184 FED6 0A          ASL A
1185 FED7 0A          ASL A
1186 FED8 0A          ASL A
1187 FED9 0A          ASL A
1188 FEDA 8D1701       STA TEMP
1189 FEDD 20E7FE       JSR RDHEX
1190 FEE0 B004          BCS RDB1          ; *EOF*
1191 FEE2 0D1701       ORA TEMP
1192 FEE5 18          CLC
1193 FEE6 60          RDB1          RTS
1194 FEE7          ;
1195 FEE7          ; ### READ A HEX NUMBER ###
1196 FEE7 20FCFE RDHEX JSR LDACC
1197 FEER C92F          CMP #'/
1198 FEED F00C          BEQ RH2          ; /*EOF*/
1199 FEEF 38          SEC
1200 FEEF E930          SBC #$30
1201 FEF1 C90A          CMP #$0A
1202 FEF3 3004          BMI RH1
1203 FEF5 38          SEC
1204 FEF6 E907          SBC #$07
1205 FEF8 18          CLC
1206 FEF9 60          RH1          RTS          ; RETURN WITH HEX NUMBER
1207 FEFA 38          RH2          SEC
1208 FEFB 60          RTS          ; RETURN WITH *EOF* MARKER
1209 FEFC          ;
1210 FEFC          ; ### READ 8 BITS TO ACC ###
1211 FEFC A900          LDACC          LDA #0
1212 FEFE A207          LDX #7
1213 FF00 2008FF LDA1 JSR RDBIT
1214 FF03 6A          ROR A
1215 FF04 CA          DEX
1216 FF05 10F9          BPL LDA1
1217 FF07 60          RTS
1218 FF08          ;
1219 FF08          ; ### READ ONE BIT ###
1220 FF08 48          RDBIT          PHA
1221 FF09 2010FF RB0 JSR GETFRQ
1222 FF0C 10FB          BPL RB0          ; WAIT FOR 2400 HZ
1223 FF0E A000          LDY #0
1224 FF10 2010FF RB1 JSR GETFRQ
1225 FF13 1003          BPL RB2          ; WAIT FOR 3700 HZ
1226 FF15 C8          INY
1227 FF16 10F8          BPL RB1
1228 FF18 C009          RB2          CPY #9          ; BIT IS IN CARRY
1229 FF1A 68          PLA
1230 FF1B 60          RTS
1231 FF1C          ;
1232 FF1C          ; ### TIME PULSE LENGTH ###
1233 FF1C 2C00C0 GETFRQ BIT DRB
1234 FF1F 30FB          BMI GETFRQ          ; WAIT TILL LOW

```

```

1235 FF21 2C00C0 GF0 BIT DRB
1236 FF24 10FB BPL GF0 ; WAIT TILL HIGH
1237 FF26 A991 LDA #145
1238 FF28 8D08C0 STA T2LL
1239 FF2B A900 LDA #0
1240 FF2D 8D09C0 STA T2CH
1241 FF30 AD0DC0 GF1 LDA IFR
1242 FF33 2920 AND #20
1243 FF35 F0F9 BEQ GF1 ; GET IN CENTER
1244 FF37 2C00C0 BIT DRB ; AND TEST THE LINE
1245 FF3A 60 RTS
1246 FF3B ;
1247 FF3B ;
1248 FF3B ; ##### RESET & INTERRUPT VECTORS #####
1249 FF3B ;
1250 FF3B ;
1251 FFF9 ; *=ROMBAS-7+800 ; FOR 2K (P)ROM
1252 FFF9 FA .BYT $FA
1253 FFFA 18FC .WOR NMI, RESET, IRQHDL
1254 0000 ;
1255 0000 ;
1256 0000 ;
1257 0000 ;
1258 0000 ; .OPT NOOBJECT
1259 0000 ;
1260 0000 ; ##### LINK IRQ TO THE 65-SYSTEM VECTOR'S #####
1261 0000 ;
1262 0000 ; *=200 ; FOR DEBUG-PACKAGE LINKING
1263 0200 ECFB .WOR IRQHDL, NMI, IRQHDL
1264 0206 ;
1265 0206 ;
1266 0206 ; .END

```

ERRORS: 0000

<0000>