```
0010:
0020:
0030:
0040:
0050:
0060:                   ************************************
0070:                   ***** INPUT AND OUTPUT ROUTINES *****
0080:                   ***** FOR THE  MICRO-ADE SYSTEM *****
0090:                   ************************************
0100:
0110:
0120: 2DC5              IO      ORG     $2DC5   77.06.24
0130:
0140: 2DC5              BLO     *       $0010   POINTER TO WORKSPACE
0150: 2DC5              N       *       $0015   LINE NUMBER
0160: 2DC5              SALX    *       $0060   FILE EXECUTION ADDRESS
0170: 2DC5              SAHX    *       $0061
0180: 2DC5              ID      *       $0062   FILE ID
0190: 2DC5              PMODE   *       $0063   PAGE MODE FLAG
0200: 2DC5              COUNTL  *       $0064   LINE COUNT
0210:
0220: 2DC5              GANG    *       $00F0   CWRITE PULSER
0230: 2DC5              TIC     *       $00F1   CWRITE TIMER
0240: 2DC5              COUNT   *       $00F2   CWRITE COUNTER
0250: 2DC5              TMP     *       $00F3   TEMPORARY STORAGE
0260: 2DC5              YTMP    *       $00F4     "         "
0270: 2DC5              XTMP    *       $00F5     "         "
0280: 2DC5              TRIB    *       $00FE   CYCLE COUNTER
0290:
0300: 2DC5              BUFFER  *       $0100   INPUT/OUTPUT BUFFER
0310:
0320: 2DC5              RESTRT  *       $2031   EDITOR WARM ENTRY ADDRESS
0330:
0340:
0350:                   KIM ROM AND PIA ADDRESSES
0360:
0370: 2DC5              SBD     *       $1742   PIA LOCATION
0380: 2DC5              CHKL    *       $17E7   CHECKSUM
0390: 2DC5              CHKH    *       $17E8
0400: 2DC5              VEB     *       $17EC   VOLATILE EXECUTION BLOCK
0410: 2DC5              SAL     *       $17F5   TAPE START ADDRESS
0420: 2DC5              SAH     *       $17F6
0430: 2DC5              EAL     *       $17F7   TAPE END ADDRESS
0440: 2DC5              EAH     *       $17F8
0450: 2DC5              INTVEB  *       $1932   INIT VEB SUBROUTINE
0460: 2DC5              CHKT    *       $194C   CHECK SUM SUBROUTINE
0470: 2DC5              INCVEB  *       $19EA   INCREMENT VEB SUB
0480: 2DC5              RDBYT   *       $19F3   READ BYTE SUBROUTINE
0490: 2DC5              RDCHT   *       $1A24   READ CHAR SUBROUTINE
0500: 2DC5              RDBIT   *       $1A41   READ BIT SUBROUTINE
0510: 2DC5              INIT    *       $1E8C   RESET ALL PIAS
0520:
0530:
0540:
0550:
0560:
```

```
0570:
0580:
0590:                   ******** INPUT AND OUTPUT ROUTINES ********
0600:
0610:                   SUBROUTINE TO PRINT THE CURRENT LINE NUMBER
0620:
0630: 2DC5 A5 16        NOUT    LDA   N       +01 GET HI N
0640: 2DC7 A6 15                LDX   N           GET LO N...PRINT THEM
0650:
0660:                   SUB TO PRINT 2 HEX BYTES
0670:                   FIRST BYTE IS IN A
0680:                   SECOND BYTE IS IN X
0690:
0700: 2DC9 20 CD 2D     HEXAX   JSR   HEXOUT PRINT ACCUMULATOR
0710: 2DCC 8A                   TXA           GET BYTE IN X ... PRINT IT
0720:
0730:                   SUBROUTINE TO PRINT 1 HEX BYTE
0740:                   INPUT IS IN ACCUMULATOR
0750:
0760: 2DCD 48           HEXOUT  PHA           SAVE INPUT
0770: 2DCE 4A                   LSRA          GET
0780: 2DCF 4A                   LSRA          UPPER
0790: 2DD0 4A                   LSRA          NYBBLE
0800: 2DD1 4A                   LSRA
0810: 2DD2 20 D8 2D             JSR   HEX     PRINT UPPER NYBBLE
0820: 2DD5 68                   PLA           GET INPUT BACK
0830: 2DD6 29 0F                ANDIM $0F     GET LOWER NYBBLE
0840:
0850:                   SUBROUTINE TO PRINT 1 HEX CHARACTER
0860:                   INPUT CHAR IS IN ACCUMULATOR
0870:
0880: 2DD8 C9 0A        HEX     CMPIM $0A     LETTER OR NUMBER?
0890: 2DDA 18                   CLC           CALCULATE ASCII
0900: 2DDB 30 02                BMI   HEXA    IF IT IS A NUMBER!
0910: 2DDD 69 07                ADCIM $07     ADD 7 TO LETTER
0920: 2DDF 69 30        HEXA    ADCIM $30     AND 30 TO BOTH
0930: 2DE1 D0 0D                BNE   OUTCH   THEN PRINT IT
0940:
0950:                   SUBROUTINE TO PRINT A BACKSPACE
0960:                   IF YOUR TERMINAL CAN'T-CHANGE THE 5F TO
0970:                   ANOTHER CHARACTER TO INDICATE DELETES
0980:
0990: 2DE3 A9 5F        BACKSP LDAIM $5F      BACKSPACE CHARACTER
1000: 2DE5 D0 09                BNE   OUTCH   PRINT IT
1010:
1020:                   SUBROUTINE TO PRINT CARRIAGE RETURN
1030:                   AND LINE FEED
1040:
1050: 2DE7 A9 0D        CRLF    LDAIM $0D     GET CR CHARACTER
1060: 2DE9 D0 05                BNE   OUTCH   AND PRINT IT
ID=02

0010:
0020:                   SUBROUTINE TO PRINT  2 HEX BYTES
0030:                   FOLLOWED IMMEDIATEDLY BY A SPACE
0040:
0050: 2DEB 20 C9 2D     HEXSP   JSR   HEXAX   PRINT 2 HEX BYTES
0060:
```

```
0070:                   SUBROUTINE TO PRINT A SPACE
0080:
0090: 2DEE A9 20        OUTSP  LDAIM '       LOAD SPACE IN A
0100:
0110:                   SUBROUTINE TO PRINT AN ASCII CHARACTER
0120:                   INPUT CHARACTER IS IN THE ACCUMULATOR
0130:
0140: 2DF0 84 F4        OUTCH  STY   YTMP    HIDE Y
0150: 2DF2 86 F5               STX   XTMP    AND X
0160: 2DF4 C9 0D               CMPIM $0D     IS THIS A CARRIAGE RETURN?
0170: 2DF6 D0 1E               BNE   NOCR     SKIP LF IF NOT
0180:
0190: 2DF8 A6 63               LDX   PMODE   CHECK PAGE MODE FLAG
0200: 2DFA D0 13               BNE   NOPG     SKIP IF NOT ON
0210: 2DFC E6 64               INC   COUNTL  ADD 1 TO LINES PRINTED
0220: 2DFE D0 0F               BNE   NOPG     SKIP IF NOT END OF SCREEN
0230:
0240: 2E00 20 2B 2E            JSR   INCH     PAUSE UNTIL INPUT OF ANY KEY
0250: 2E03 C9 1B               CMPIM $1B     WAS ESCAPE KEY ENTERED?
0260: 2E05 D0 04               BNE   ON       IF NOT CONTINUE IN PAGE MODE
0270: 2E07 A9 FF               LDAIM $FF     TURN OFF PAGE MODE
0280: 2E09 85 63               STA   PMODE
0290:
0300: 2E0B A2 F0        ON     LDXIM $F0     RESET LINE COUNTER
0310: 2E0D 86 64               STX   COUNTL  TO -16
0320:
0330: 2E0F A9 0A        NOPG   LDAIM $0A     PRINT A LINE FEED
0340: 2E11 20 16 2E            JSR   NOCR     (REMOVE IF YOUR TERMINAL HAS AUT
0350:
0360: 2E14 A9 0D               LDAIM $0D     THIS WAS A CR, REMEMBER
0370: 2E16 20 A0 2E     NOCR   JSR   OUTPUT  SO PRINT IT
0380:
0390:                   ROUTINE TO TEST FOR BREAK DURING I/O
0400:
0410: 2E19 2C 40 17     BRKTST BIT   $1740   TEST INPUT PORT OF PIA
0420: 2E1C 10 05               BPL   BREAK    IF BIT 7=0
0430:
0440: 2E1E A6 F5               LDX   XTMP    SEEK HIDDEN X
0450: 2E20 A4 F4               LDY   YTMP    AND HIDDEN Y
0460: 2E22 60                  RTS           AND ITS ALL OVER
0470:
0480: 2E23 2C 40 17     BREAK  BIT   $1740   WAIT UNTIL KEY
0490: 2E26 10 FB               BPL   BREAK    IS RELEASED
0500: 2E28 4C 31 20            JMP   RESTRT  THEN GO TO EDITOR
0510:
0520:                   ROUTINE TO INPUT AN ASCII CHARACTER
0530:                   RETURNS IT IN ACCUMULATOR
0540:
0550: 2E2B 86 F5        INCH   STX   XTMP    HIDE X
0560: 2E2D 84 F4               STY   YTMP    AND Y
0570: 2E2F 20 9D 2E            JSR   INPUT   CALL USER INPUT ROUTINE
0580: 2E32 29 7F               ANDIM $7F     STRIP PARITY BIT
0590: 2E34 C9 0D               CMPIM $0D     WAS INPUT A RETURN
0600: 2E36 D0 07               BNE   NOCRIN  IF NOT ITS OK
0610:
0620: 2E38 A9 0A               LDAIM $0A     PRINT A LF WITH CR INPUT
```

```
0630: 2E3A 20 A0 2E          JSR   OUTPUT SKIP THIS IF AUTO LF ON TERMINAL
0640: 2E3D A9 0D             LDAIM $0D    REPLACE CR AGAIN FOR RTS
0650: 2E3F D0 D8      NOCRIN BNE   BRKTST RETURN VIA BREAK TEST
0660:
ID=03

0010:
0020:                      SUBROUTINE TO FILL BUFFER FROM
0030:                      KEYBOARD INPUT
0040:
0050: 2E41 A0 00      BUFIN  LDYIM $00    RESET BUFFER COUNTER
0060: 2E43 20 2B 2E   INB    JSR   INCH    GET CHARACTER INPUT
0070: 2E46 C9 7F             CMPIM $7F    WAS IT A DELETE?
0080: 2E48 D0 07             BNE   ONIN    IF NOT, CARRY ON
0090:
0100: 2E4A 20 E3 2D          JSR   BACKSP PRINT A BACKSPACE
0110: 2E4D 88                DEY          BACK UP IN BUFFER
0120: 2E4E 10 F3             BPL   INB     AND GET IT RIGHT THIS TIME
0130: 2E50 00                BRK          ERROR--BACKED UP TOO FAR!
0140:
0150: 2E51 C9 5C      ONIN   CMPIM $5C    (BACKSLASH) WILL
0160: 2E53 D0 06             BNE   OKB     DELETE WHOLE LINE
0170: 2E55 20 E7 2D          JSR   CRLF    PRINT RETURN AND LF
0180: 2E58 4C 41 2E          JMP   BUFIN   AND START OVER
0190:
0200: 2E5B C9 05      OKB    CMPIM $05    WAS IT A CTL-E?
0210: 2E5D F0 11             BEQ   OVR     YES--GO TO OVR FUNCTION
0220: 2E5F 99 00 01          STAAY BUFFER JUST AN ORDINARY CHARACTER TO
0230:
0240: 2E62 C9 0D             CMPIM $0D    WAS THIS THE END?
0250: 2E64 F0 09             BEQ   ENDBU   YES, SO GET OUT OF HERE
0260: 2E66 C8                INY          INCREMENT POINTER
0270: 2E67 C0 3A             CPYIM $3A    ALLOW ONLY 58 CHARS +6 PROMPT=6
0280: 2E69 30 D8             BMI   INB     STILL SOME ROOM FOR MORE
0290: 2E6B A9 0D             LDAIM $0D    FORCE CR TO END LINE
0300: 2E6D D0 EC             BNE   OKB     PRINT IT AND PUT IN BUFFER
0310: 2E6F 60         ENDBU  RTS          ALL DONE
0320:
0330: 2E70 20 E3 2D   OVR    JSR   BACKSP CANCEL THE CTL CHAR (THIS MAY N
0340:                                      NECESSARY ON SOME TERMINALS)
0350: 2E73 B9 00 01   OVX    LDAAY BUFFER GET CHARACTER IN BUFFER
0360: 2E76 C9 0D             CMPIM $0D    IS IT THE END?
0370: 2E78 F0 C9             BEQ   INB     IF SO--GO GET NEW ADDITION
0380: 2E7A 20 F0 2D          JSR   OUTCH   SHOW HIM WHAT IT IS
0390: 2E7D C8                INY          ON TO NEXT CHARACTER
0400: 2E7E C0 3A             CPYIM $3A    BUT DON'T GET CARRIED AWAY! BUF
0410: 2E80 D0 F1             BNE   OVX     KEEP GOING                    L
0420: 2E82 F0 BF             BEQ   INB     LET HIM FIX IT UP
0430:
0440:                      SUBROUTINE TO PRINT THE BUFFER
0450:
0460: 2E84 A0 00      PRBUF  LDYIM $00    RESET THE POINTER
0470: 2E86 B9 00 01   PRNTB  LDAAY BUFFER GET A CHARACTER
0480: 2E89 48                PHA          HIDE IT TEMPORARILY
0490: 2E8A 20 F0 2D          JSR   OUTCH   PRINT IT
0500: 2E8D 68                PLA          SEEK IT BACK
0510: 2E8E C8                INY          POINT TO NEXT CHARACTER
0520: 2E8F C9 0D             CMPIM $0D    WAS THAT THE END OF THE BUFFER?
```

```
0530: 2E91 D0 F3            BNE   PRNTB  NO, THERE MUST BE MORE
0540: 2E93 60               RTS          YES, SO RETURN
0550:
0560:                     **** ********* *********
0570:                   ***** USER SPECIFIED ADDRESSES *****
0580:                     **** ********* *********
0590:
0600:
0610: 2E94 4C 00 1A  PACKT  JMP   $1A00  A KIM SUBROUTINE TO PACK ASCII IN
0620:
0630: 2E97 4C AF 2E  READ   JMP   CREAD  THE CASSETTE READ SUBROUTINE
0640:
0650: 2E9A 4C 35 2F  WRITE  JMP   CWRITE THE CASSETTE WRITE SUBROUTINE
0660:
0670: 2E9D 4C 5A 1E  INPUT  JMP   $1E5A  THE KEYBOARD INPUT ROUTINE
0680:
0690: 2EA0 4C A0 1E  OUTPUT JMP   $1EA0  THE PRINTER OUTPUT ROUTINE
0700:
0710:                   DEFINITION OF SOURCE LOCATION
0720:
0730: 2EA3 35           SOURCM =     $35    SOURCE - 1
0740: 2EA4 36           SOURCE =     $36    SOURCE AREA OV MEMORY STARTS HERE
0750: 2EA5 40           SOURCF =     $40    AND ENDS JUST BELOW HERE
0760:
0770:                   DEFINITION OF SYMBOL TABLE LOCATION
0780:
0790: 2EA6 30           SYMBOL =     $30    SYMBOL TABLE STARTS HERE
0800: 2EA7 36           SYMF   =     $36    AND JUST BELOW HERE
0810:
0820:                   DEFINITION OF OBJECT LOCATION
0830:
0840: 2EA8 02           OBJECT =     $02    THE OBJECT WILL BE ASSEMBLED TO U
0850:
ID=04


0010:
0020:                   ************************************************
0030:                   ***** KIM CASSETTE READ AND WRITE ROUTINES *****
0040:                   ************************************************
0050:
0060:                   CASSETTE READ ERROR (INCORRECT ID)
0070:
0080: 2EA9 20 CD 2D  ERID   JSR   HEXOUT PRINT THE WRONG ID,
0090: 2EAC 20 EE 2D         JSR   OUTSP  SPACE AND THEN START OVER
0100:
0110:                   ***** CASSETTE READ SUBROUTINE *****
0120:
0130:                   VERY SIMILAR TO THE READ ROUTINE
0140:                   IN THE KIM ROM.
0150:                   THE LED DISPLAY OF INCOMING DATA ADAPTED
0160:                   FROM VUTAPE, A PROGRAM BY JIM BUTTERFIELD
0170:                   FROM THE KIM-1 USER NOTES.
0180:
 190:
0200: 2EAF AD 02 17  CREAD  LDA   $1702  TURN ON CASSETTE #1 BY
0210: 2EB2 29 FB            ANDIM $FB    CHANGING BIT 2 TO
0220: 2EB4 8D 02 17         STA   $1702  ZERO IN PIA PORT B
0230:
```

```
0240: 2EB7 A9 7F           LDAIM $7F    TURN ON THE KIM LED DISPLAY
0250: 2EB9 8D 41 17        STA   $1741  BY SETTING THE DD REG
0260:
0270: 2EBC D8              CLD          JUST TO MAKE SURE
0280:
0290: 2EBD A9 8D           LDAIM $8D    SET UP VEB
0300: 2EBF 8D EC 17        STA   VEB    TO SAVE DATA
0310: 2EC2 20 32 19        JSR   INTVER (IN KIM ROM)
0320:
0330: 2EC5 A9 13           LDAIM $13    TURN ON INPUT PORT FROM CASSETTE
0340: 2EC7 8D 42 17        STA   SBD
0350:
0360: 2ECA 20 41 1A  SYNC  JSR   RDBIT  START READING A BIT AT A TIME
0370:
0380: 2ECD 46 F3           LSRZ  TMP    SHIFT IT INTO TMP
0390: 2ECF 05 F3           ORAZ  TMP
0400: 2ED1 85 F3           STAZ  TMP    AND SAVE IT
0410: 2ED3 8D 40 17        STA   $1740  PLACE IT ON THE LED
0420:
0430: 2ED6 C9 16     TST   CMPIM $16    IS IT A SYNC CHARACTER?
0440: 2ED8 D0 F0           BNE   SYNC   IF NOT, KEEP TRYING
0441:
0450: 2EDA 20 24 1A        JSR   RDCHT  IN SYNC, READ A CHARACTER
0460: 2EDD 8D 40 17        STA   $1740  DISPLAY IT ON LED
0470: 2EE0 C9 2A           CMPIM $2A    IS IT THE START OF DATA?
0480: 2EE2 D0 F2           BNE   TST    IF NOT, LOOP AGAIN
0481:
0490: 2EE4 20 F3 19        JSR   RDBYT  READ THE TAPE ID
0500: 2EE7 C5 62           CMP   ID     IS THIS THE RIGHT TAPE?
0510: 2EE9 D0 BE           BNE   ERID   PRINT IT IF WRONG
0511:
0520: 2EEB 20 F3 19        JSR   RDBYT  READ THE START ADDRESS
0530: 2EEE 20 4C 19        JSR   CHKT   INCLUDE IT IN CHECKSUM
0540: 2EF1 8D ED 17        STA   VEB    +01 AND SAVE IT IN VEB
0550: 2EF4 20 F3 19        JSR   RDBYT  READ THE HI PART OF ADDRESS
0560: 2EF7 20 4C 19        JSR   CHKT   INCLUDE IN SUM
0570: 2EFA 8D EE 17        STA   VEB    +02 SET IT UP IN VEB
0571:
0572:
0580: 2EFD A2 02     LOADIT LDXIM $02    START TO LOAD DATA AS
0590: 2EFF 20 24 1A  READIT JSR   RDCHT  ASCII CHARACTERS
0600: 2F02 C9 2F           CMPIM '/     END OF DATA SYMBOL
0610: 2F04 F0 14           BEQ   ENDRD  SO WIND IT UP
0611:
0620: 2F06 20 94 2E        JSR   PACKT  PACK THE ASCII INTO HEX
0630: 2F09 D0 BF           BNE   SYNC   ERROR IN CHARACTER READ NOT = HE
0640: 2F0B CA              DEX          COUNT TO TWO
0650: 2F0C D0 F1           BNE   READIT READ SECOND HALF
0651:
0660: 2F0E 20 4C 19        JSR   CHKT   ADD TO CHECKSUM
0670: 2F11 20 EC 17        JSR   VEB    STORE VIA VEB
0680: 2F14 20 EA 19        JSR   INCVEB INCREMENT STORE ADDRESS
0690: 2F17 4C FD 2E        JMP   LOADIT AND READ NEXT BYTE
0691:
0700: 2F1A 20 F3 19  ENDRD JSR   RDBYT  READ CHECKSUM FROM TAPE
0710: 2F1D CD E7 17        CMP   CHKL   COMPARE TO CALCULATED
```

```
0720: 2F20 D0 A8            BNE   SYNC   AND START OVER IF WRONG
0730: 2F22 20 F3 19         JSR   RDBYT  GET SECOND HALF OF SUM
0740: 2F25 CD E8 17         CMP   CHKH   AND DO THE SAME
0750: 2F28 D0 A0            BNE   SYNC   WITH IT
0751:
0760: 2F2A AD 02 17  OKRD   LDA   $1702  TURN OFF CASSETTE
0770: 2F2D 09 04            ORAIM $04    BY SETTING BIT 2
0780: 2F2F 8D 02 17         STA   $1702  OF THE PORT
0790: 2F32 4C 8C 1E         JMP   INIT   RETURN VIA INIT (RESET ALL PORTS)
0791:
ID=05

0010:                ***** KIM CASSETTE WRITE SUBROUTINE *****
0020:
0030:                ADAPTED FROM SUPERTAPE BY JIM BUTTERFIELD
0040:                AS PUBLISHED IN KIM-1 USER NOTES (V 1,N 2)
 050:
0060:
0070: 2F35 AD 02 17  CWRITE LDA   $1702  TURN ON CASSETTE #2
0080: 2F38 29 F7            ANDIM $F7    BY SETTING BIT 3 = 0
0090: 2F3A 8D 02 17         STA   $1702  IN PIA PORT B
0100:
0110: 2F3D A9 AD            LDAIM $AD    SET UP
0120: 2F3F 8D EC 17         STA   VEB    VEB FOR SAVE
0130: 2F42 20 32 19         JSR   INTVEB
0140:
0150: 2F45 A9 27            LDAIM $27    SET FLAG
0160: 2F47 85 F0            STAZ  GANG   FOR SBD LATER
0170:
0180: 2F49 A9 BF            LDAIM $BF    TURN ON
0190: 2F4B 8D 43 17         STA   $1743  OUTPUT TO CASSETTE
0200:
0210: 2F4E A2 F0            LDXIM $F0    SEND 240 SYNC PULSES (OPTIMUM # D
0220:                                     ON RECORDER START/STOP TIME)
0230: 2F50 A9 16            LDAIM $16    SYNC CHARACTER
0240: 2F52 20 A3 2F         JSR   HIC    OUTPUT X TIMES
0250:
0260: 2F55 A9 2A            LDAIM $2A    SEND START OF DATA CHAR
0270: 2F57 20 C6 2F         JSR   OUTCHT
0280:
0290: 2F5A A5 62            LDA   ID     GET ID
0300: 2F5C 20 B2 2F         JSR   OUTBT  AND SEND AS A BYTE
0310:
0320: 2F5F A5 60            LDAZ  SALX   SEND EXECUTION ADDRESS
0330: 2F61 20 AF 2F         JSR   OUTBTC WITH CHECKSUM CALCULATION
0340: 2F64 A5 61            LDAZ  SAHX   HI PART TOO
0350: 2F66 20 AF 2F         JSR   OUTBTC
0360:
0370: 2F69 20 EC 17  DUMPTA JSR   VEB    GET A BYTE OF MEMORY
0380: 2F6C 20 AF 2F         JSR   OUTBTC SEND AND CHECKSUM IT
0390: 2F6F 20 EA 19         JSR   INCVER POINT TO NEXT BYTE
0400: 2F72 AD ED 17         LDA   VEB    +01 CHECK FOR END
0410: 2F75 CD F7 17         CMP   EAL    AGAINST EAL
0420: 2F78 AD EE 17         LDA   VEB    +02 AND
0430: 2F7B ED F8 17         SBC   EAH    EAH
0440: 2F7E 90 E9            BCC   DUMPTA AGAIN IF NOT END
0450:
0460: 2F80 A9 2F            LDAIM $      SEND END OF DATA CHAR
```

```
0470: 2F82 20 C6 2F          JSR   OUTCHT AS CHAR
0480:
0490: 2F85 AD E7 17          LDA   CHKL   SEND
0500: 2F88 20 B2 2F          JSR   OUTBT  CHECKSUM
0510: 2F8B AD E8 17          LDA   CHKH   LO AND
0520: 2F8E 20 B2 2F          JSR   OUTBT  HI
0530: 2F91 A2 02             LDXIM $02    AND SEND 2
0540: 2F93 A9 04             LDAIM $04    EOT CHARS
0550: 2F95 20 A3 2F          JSR   HIC
0560:
0570: 2F98 AD 02 17          LDA   $1702  TURN OFF CASSETTE
0580: 2F9B 09 08             ORAIM $08    BY SETTING BIT 3
0590: 2F9D 8D 02 17          STA   $1702  OF THE CONTROL PORT
0600: 2FA0 4C 8C 1E          JMP   INIT   RESET ALL PORTS
0610:
0620:                  SUBROUTINE TO SEND X CHARACTERS TO TAPE
0630:
0640: 2FA3 86 F1       HIC   STXZ  TIC    SAVE THE COUNT
0650: 2FA5 48          HICA  PHA          AND THE CHARACTER
0660: 2FA6 20 C6 2F          JSR   OUTCHT SEND THE CHAR
0670: 2FA9 68                PLA          AND GET IT BACK
0680: 2FAA C6 F1             DECZ  TIC    TO SEND AGAIN
0690: 2FAC D0 F7             BNE   HICA   UNTIL COUNT = 0
0700: 2FAE 60                RTS
0710:
0720:                  SUB TO SEND CHARACTER WITH CHECKSUM CALCULATION
0730:
0740: 2FAF 20 4C 19    OUTBTC JSR  CHKT   ADD CHAR TO SUM
0750:
0760:                  SUB TO SEND BYTE AS TWO ASCII CHARS
0770:
0780: 2FB2 48          OUTBT PHA          SAVE BYTE
0790: 2FB3 4A                LSRA         GET
0800: 2FB4 4A                LSRA         UPPER
0810: 2FB5 4A                LSRA         NYBBLE
0820: 2FB6 4A                LSRA
0830: 2FB7 20 BB 2F          JSR   HEXT   AND SEND IT
0840: 2FBA 68                PLA          RETURN BYTE
0850:
0860:                  SUBROUTINE TO SEND ONE HEX CHAR AS ASCII
0870:
0880: 2FBB 29 0F       HEXT  ANDIM $0F    CLEAN UP DATA
0890: 2FBD C9 0A             CMPIM $0A    CHANGE TO ASCII
0900: 2FBF 18                CLC          BY ADDING
0910: 2FC0 30 02             BMI   HEXAT
0920: 2FC2 69 07             ADCIM $07    37 TO A...F
0930: 2FC4 69 30       HEXAT ADCIM $30    AND 30 TO 0...9
ID=06

0010:
0020:                  SUBROUTINE TO SEND ONE 8 BIT BYTE
0030:
0040: 2FC6 A0 08       OUTCHT LDYIM $08   EIGHT BIT COUNT
0050: 2FC8 84 F2             STYZ  COUNT
0060: 2FCA A0 02       TRY   LDYIM $02    START AT
0070: 2FCC 84 FE             STYZ  TRIB   3600 HERTZ
0080: 2FCE BE FC 2F    ZON   LDXAY NPUL   NUMBER OF HALF CYCLES
0090: 2FD1 48                PHA          SAVE THE CHAR
```

```
0100:
0110: 2FD2 2C 47 17  ZONA    BIT   $1747  WAIT FOR END OF CYCLE
0120: 2FD5 10 FB             BPL   ZONA   IN TIGHT LOOP
0130:
0140: 2FD7 B9 FD 2F          LDAAY TIMG   SET UP TIMER
0150: 2FDA 8D 44 17          STA   $1744  FOR THIS PULSE
0160:
0170: 2FDD A5 F0             LDAZ  GANG   CHANGE STATE
0180: 2FDF 49 80             EORIM $80    OF OUTPUT
0190: 2FE1 8D 42 17          STA   $1742  PORT
0200:
0210: 2FE4 85 F0             STAZ  GANG   AND SAVE STATE
0220: 2FE6 CA                DEX          DONE ALL CYCLES?
0230: 2FE7 D0 E9             BNE   ZONA   NO-THEN SEND ANOTHER
0240:
0250: 2FE9 68                PLA          RETRIEVE BYTE
 260: 2FEA C6 FE             DECZ  TRIB   ONE MORE GONE
0270: 2FEC F0 05             BEQ   SETZ   THE LAST ONE, TOO
0280: 2FEE 30 07             BMI   ROUT   EVEN THE LAST ONE WENT
0290:
0300: 2FF0 4A                LSRA         ANOTHER BIT TO THE CARRY
0310: 2FF1 90 DB             BCC   ZON    IF IT IS NOT SET
0320: 2FF3 A0 00     SETZ    LDYIM $00    SWITCH TO 2400 HZ
0330: 2FF5 F0 D7             BEQ   ZON    ALWAYS
0340:
0350: 2FF7 C6 F2     ROUT    DECZ  COUNT  ONE BIT SENT
0360: 2FF9 D0 CF             BNE   TRY    BUT MORE TO GO
0370: 2FFB 60                RTS          ALL OVER, GO HOME
0380:
0390:                TIMING TABLE
0400:
0410: 2FFC 02        NPUL    =     $02    TWO PULSES
0420: 2FFD C3        TIMG    =     $C3    THE RIGHT TIME
0430: 2FFE 03                =     $03    3 PULSES
0440: 2FFF 7E                =     $7E    AND ENOUGH TIME
 450:
0460:                IF YOUR RECORDER CANNOT HANDLE THIS SPEED,
0470:                YOU CAN SLOW DOWN BY CHANGING NPUL AND NPUL+02
0480:                TO ONE OF THE FOLLOWING:      03       04
0490:                                              06       09
0500:                (THIS IS THE KIM ROM SPEED)   0C       12
ID=
```