

```

0010: *****
0020:
0030: TAPE COPYING PROGRAM
0040: ;
0050: HJC OTTEN JULY 1981 V1.0
0060: JULY 1982 V1.1
0070: ;
0080: BASED ON SUPERDUPE FROM THE FIRST BOOK OF KIM
0090: AND THE MICRO ADE CASSETTE ROUTINES
0100: ;
0110: COPY WILL REPRODUCE A TAPE COMPLETELY AUTOMATIC
0120: WHEN READ AND WRITE RECORDERS ARE CONNECTED
0130: WITH THE MOTOR CONTROLS OF MICRO ADE
0140: ;
0150: DURING READ OPERATIONS IT IS POSSIBLE TO STOP THE
0160: COPYING BY MAKING PA7 OF PIA 1700 HIGH
0170: REMOVE THIS BIT TESTS IF NOT AVAILABLE
0180: AT 0249, 02D6 AND 02E9
0190: OR CHANGE FOR KIM TTY BREAK THE PIA LOCATION TO 1740
0200: AND THE BRANCH FOLLOWING TO BMI
0210: ;
0220: INSERT REMINDED READ AND WRITE CASSETTES
0230: AND START AT 0200
0240: ;
0250: FIRST THE WRITE CASSETTE IS RUN PAST THE LEADER
0260: THEN A FILE IS READ , IF FOUND THE ID AND STARTADDRESS
0270: ARE DISPLAYED . NEXT THE FILE IS WRITTEN.
0280: IF READY THE DISPLAY LINE LOOKS LIKE
0290: ID= 01 START= $3600 COPY !
0300: THE DATA FROM THE FILE IS STORED IN A BUFFER AT $2000
0310: WHERE ENOUGH RAM MUST BE AVAILABLE
0320: ;
0330: THIS PROGRAM CAN EASILY BE MODIFIED TO
0340: A CHECKING AND DIRECTORY OF ALL FILES ON A TAPE
0350: BY REPLACING THE JUMP
0360: 02D0 4C 2A 02 JMP CREAD
0370: THAT SKIPS THE WRITING OF A FILE
0380: ;
0390: *****
0400: ;
0410: 2000 COPY ORG $2000 ;
0420: ;
0430: DEFINES :
0440: ;
0450: 2000 BUFFER * $2300 ;
0460: 2000 SYNC * $16 ; SYNCHRONISATION CHARACTER
0470: 2000 EOD * $2F ; END OF DATA
0480: 2000 EOF * $04 ; END OF FILE
0490: 2000 SOD * $2A ; START OF DATA
0500: 2000 FF * $0C ; FORM FEED
0510: ;
0520: ; ZERO PAGE USE
0530: 2000 HELPL * $00DD ;
0540: 2000 HELPH * $00DE ;
0550: 2000 CHKCP * $00DF ; CHECK OR COPY FLAG
0560: ;

```



```

0570: 2000      RPTL  *      $00E0 ; READ POINTER IN BUFFER
0580: 2000      RPTRH *      $00E1 ;
0590: 2000      WPTL  *      $00E2 ; WRITE POINTER IN BUFFER
0600: 2000      WPTRH *      $00E3 ;
0610: 2000      STRPTL *      $00ED ; POINTER IN PRINT STRING
0620: 2000      STRPTH *      $00EE ;
0630: 2000      CC     *      $00F1 ; COUNT FIELDS IN WRITE
0640: 2000      COUNT *      $00F2 ;
0650: 2000      CHKSUM *      $00F6 ; CHECKSUM
0660: 2000      CHKHI  *      $00F7 ;
0670: 2000      GANG   *      $00F5 ; USED BY WRITE
0680: 2000      ID      *      $00F9 ; ID OF FILE
0690: 2000      STADRL *      $00FA ; STARTADDRESS OF FILE
0700: 2000      STADRH *      $00FB ;
0710: 2000      STARTA *      $00FC ; CONSTANT FOR INDEXED WRITE
0720: 2000      TRIB   *      $00FE ; USED BY WRITE
0730:           ;
0740:           ; PIA LOCATIES
0750:           ;
0760: 2000      PAD     *      $1700 ; BREAKKEY IN BIT 7 -
0770: 2000      PBD     *      $1702 ; PB2 = READ, PB3 = WRITE MOTOR CONTROL
0780: 2000      PBDD    *      $1703 ;
0790: 2000      KPAD    *      $1740 ; LEDDISPLAY
0800: 2000      KPADD    *      $1741 ;
0810: 2000      KPBD     *      $1742 ; CASSETTE I/O
0820: 2000      KPBDD    *      $1743 ;
0830: 2000      TIMERT   *      $1744 ; TIMER
0840: 2000      TIMER   *      $1747 ;
0850:           ;
0860:           ; KIM MONITOR ROUTINES
0870:           ;
0880: 2000      RDBIT    *      $1A41 ; READ BIT FROM TAPE
0890: 2000      RDBTK    *      $19F3 ; READ BYTE FROM TAPE
0900: 2000      RDCHTK   *      $1A24 ; READ CHARACTER FROM TAPE (2 BYTE)
0910: 2000      PACKT    *      $1A00 ; PACK ASCII TO HEX
0920: 2000      CHK      *      $1F91 ; COMPUTE CHECKSUM IN F6,F7
0930: 2000      CHKT     *      $194C ; COMPUTE CHECKSUM IN 17E7,E8
0940: 2000      INIT     *      $1E8C ; INIT KIM PIA'S
0950: 2000      SPACE    *      $1E9E ; PRINT SPACE ON TTY
0960: 2000      OUTCH    *      $1EA0 ; PRINT ACCU ON TTY
0970: 2000      PRTBYT   *      $1E3B ; PRINT ACCU AS TWO HEX CHARACTERS ON TTY
0980: 2000      CRLF     *      $1E2F ; PRINT CR + LF ON TTY
0990: 2000      MNITOR   *      $1C00 ; KIM MONITOR ENTRY POINT
1000:           ;
1010:           *****
1020:           ;
1030:           MAIN FLOW OF PROGRAM :
1040:           ;
1050:           PROGRAM COPY
1060:           ;
1070:           INITCOPY
1080:           REPEAT
1090:           READ FILE TO BUFFER
1100:           WRITE BUFFER TO FILE
1110:           UNTIL FOREVER
1120:           ;

```

PARALLEL KEYBOARD


```

1130:      END
1140:      ;
1150:      ;*****
1160:      ;
1170:      PROCEDURES :
1180:      ;
1190:      2000 D8      START CLD      ; PROCEDURE INITCOPY
1200:      2001 A9 0C      LDAIM #0C      ; INIT CASSETTE MOTOR CONTROL
1210:      2003 8D 03 17      STA PBDD      ;
1220:      2006 A9 FF      LDAIM #FF      ;
1230:      2008 8D 02 17      STA PBD      ;
1240:      200B A9 0C      LDAIM FF      ; CLEARSCREEN
1250:      200D 20 A0 1E      JSR OUTCH      ;
1260:      2010 A2 FF      LDXIM #FF      ;
1270:      2012 CA      NEX DEX      ;
1280:      2013 D0 FD      BNE NEX      ;
1290:      2015 A2 84      LDXIM HELLOM      ; PRINTSTRING ('TAPE COPY')
1300:      2017 A0 22      LDVIM HELLOM /
1310:      2019 20 4F 22      JSR PRTSTR      ;
1320:      201C A2 A0      LDXIM ANDMSG      ;
1330:      201E A0 22      LDVIM ANDMSG /
1340:      2020 20 4F 22      JSR PRTSTR      ;
1350:      2023 A2 8E      LDXIM CHKMSG      ;
1360:      2025 A0 22      LDVIM CHKMSG /
1370:      2027 20 4F 22      JSR PRTSTR      ;
1380:      202A A2 9B      LDXIM URSMSG      ;
1390:      202C A0 22      LDVIM URSMSG /
1400:      202E 20 4F 22      JSR PRTSTR      ;
1410:      2031 20 2F 1E      JSR CRLF      ;
1420:      2034 20 2F 1E      JSR CRLF      ;
1430:      2037 A2 8E      LDXIM CHKMSG      ; PRINT ('CHECK OR COPY')
1440:      2039 A0 22      LDVIM CHKMSG /
1450:      203B 20 4F 22      JSR PRTSTR      ;
1460:      203E A2 94      LDXIM ONLMSG      ;
1470:      2040 A0 22      LDVIM ONLMSG /
1480:      2042 20 4F 22      JSR PRTSTR      ;
1490:      2045 A9 00      LDAIM #00      ; CHKCOP FLAG = COPY
1500:      2047 85 DF      STA CHKCOP      ;
1510:      2049 20 68 22      JSR GETCH      ; WAIT FOR ANSWER
1520:      204C 48      PHA      ;
1530:      204D 20 A0 1E      JSR OUTCH      ; ECHO CHARACTER
1540:      2050 20 2F 1E      JSR CRLF      ;
1550:      2053 68      PLA      ;
1560:      2054 C9 59      CMPIM <Y      ;
1570:      2056 D0 0E      BNE NOCHK      ; IF CHAR <> Y THEN COPY
1580:      2058 A9 FF      LDAIM #FF      ;
1590:      205A 85 DF      STA CHKCOP      ;
1600:      205C A2 8E      LDXIM CHKMSG      ; PRINT CHECK
1610:      205E A0 22      LDVIM CHKMSG /
1620:      2060 20 4F 22      JSR PRTSTR      ;
1630:      2063 4C 7C 20      JMP CREAD      ;
1640:      2066 A2 88      NOCHK LDXIM COPYMS      ;
1650:      2068 A0 22      LDVIM COPYMS /
1660:      206A 20 4F 22      JSR PRTSTR      ; PRINT COPY
1670:      206D A9 F7      LDAIM #F7      ; TURN WRITE CASSETTE MOTOR ON
1680:      206F 8D 02 17      STA PBD      ;

```



```

1690: 2072 A2 10          LDXIM $10      ; DELAY (LEADERCASSETTE)
1700: 2074 20 3F 22      JSR  DELAY      ;
1710: 2077 A9 FF          LDAIM $FF      ; TURN WRITE CASSETTE MOTOR OFF
1720: 2079 8D 02 17      STA  PBD       ; END INITCOPY
1730:
1740: 207C A9 23          CREAD LDAIM BUFFER / PROCEDURE READ FILE TO BUFFER
1750: 207E 85 E1          STA  RPTRH      ; READPTR := BUFFERSTART
1760: 2080 85 E3          STA  WPTRH      ; WRITEPTR := BUFFERSTART
1770: 2082 A9 00          LDAIM $00      ; CHECKSUM := 0
1780: 2084 85 F6          STA  CHKSUM     ;
1790: 2086 85 F7          STA  CHKHI     ;
1800: 2088 A9 00          LDAIM BUFFER     ;
1810: 208A 85 E0          STA  RPTRL     ;
1820: 208C 85 E2          STA  WPTRL     ;
1830: 208E A9 FB          LDAIM $FB      ; TURN READ CASSETTEMOTOR ON
1840: 2090 8D 02 17      STA  PBD       ;
1850: 2093 A9 13          LDAIM $13      ; INIT CASSETTE I/O
1860: 2095 8D 42 17      STA  KPBD      ;
1870: 2098 A9 7F          LDAIM $7F      ; INIT LEDDISPLAY
1880: 209A 8D 41 17      STA  KPADD      ;
1890:
1900: 209D 2C 00 17      SYN  BIT  PAD      ; REPEAT
1910: 20A0 10 03          BPL  GOON      ; WHILE NOT BYTE = SYNC DO
1920: 20A2 4C 5D 21      JMP  MONRET    ; READ(BYTE)
1930: 20A5 20 41 1A      GOON JSR  RDBIT    ; IF BREAK EXIT TO MONITOR
1940: 20A8 46 F9          LSR  ID       ; ENDWHILE
1950: 20AA 05 F9          ORA  ID       ; READ(BYTE)
1960: 20AC 85 F9          STA  ID       ; UNTIL BYTE = START OF DATA
1970: 20AE 8D 40 17      STA  KPAD      ;
1980: 20B1 C9 16          TST  CMPIM $16  ;
1990: 20B3 D0 E8          BNE  SYN       ;
2000: 20B5 20 68 21      JSR  RDCHT    ;
2010: 20B8 8D 40 17      STA  KPAD      ;
2020: 20BB C9 2A          CMPIM $2A  ;
2030: 20BD D0 F2          BNE  TST       ;
2040: 20BF 20 55 21      JSR  RDBYT    ;
2050: 20C2 85 F9          STA  ID       ; ID := READ(BYTE)
2060: 20C4 A2 FE          LDXIM $FE      ; STARTADDRESS := READ(ADDRESS)
2070: 20C6 20 55 21      ADDR JSR  RDBYT    ;
2080: 20C9 95 FC          STAZX STARTA  ;
2090: 20CB 20 91 1F      JSR  CHK       ;
2100: 20CE E8             INX          ;
2110: 20CF 30 F5          BMI  ADDR      ;
2120:
2130: 20D1 A2 02          BYTE LDXIM $02  ; WHILE NOT BYTE = END OF DATA DO
2140: 20D3 20 68 21      DUBL JSR  RDCHT    ; BUFFER(RPOINTER) := READ(CHARACTER)
2150: 20D6 C9 2F          CMPIM EOD  ; CHECKSUM := CHECKSUM + CHARACTER
2160: 20D8 F0 15          BEQ  WIND     ; RPOINTER := RPOINTER + 1
2170: 20DA 20 00 1A      JSR  PACKET    ; ENDWHILE
2180: 20DD D0 73          BNE  ELNK      ;
2190: 20DF CA            DEX          ;
2200: 20E0 D0 F1          BNE  DUBL      ;
2210: 20E2 81 E0          STAIR RPTRL    ;
2220: 20E4 20 91 1F      JSR  CHK       ;
2230: 20E7 E6 E0          INC  RPTRL     ;
2240: 20E9 D0 02          BNE  OVER      ;

```



```

2250: 20EB E6 E1      INC  RPTRH  ;
2260: 20ED D0 E2      OVER BNE  BYTE  ;
2270: 20EF 20 55 21   WIND JSR  RDBYT  ; RCHECKSUM := READ(CHECKSUM)
2280: 20F2 C5 F7      CMP   CHKHI  ;
2290: 20F4 D0 5C      BNE  ELNK   ;
2300: 20F6 20 55 21   JSR  RDBYT  ;
2310: 20F9 C5 F6      CMP   CHKSUM ;
2320: 20FB 08          PHP      ;
2330: 20FC A9 FF      LDAIM $FF  ; TURN READ CASSETTE MOTOR OFF
2340: 20FE 8D 02 17   STA  PBD   ;
2350: 2101 28          PLP      ; IF RCHECKSUM <> CHECKSUM GREAD FILE TO BUF
2360: 2102 D0 4E      BNE  ELNK   ;
2370: 2104 20 2F 1E   JSR  CRLF  ; PRINTSTRING ('ID=')
2380: 2107 A2 76      LDXIM IDMES  ; PRINT (ID)
2390: 2109 A0 22      LDYIM IDMES  /
2400: 210B 20 4F 22   JSR  PRTSTR ;
2410: 210E A5 F9      LDA  ID     ;
2420: 2110 20 3B 1E   JSR  PRTBYT ;
2430: 2113 A2 7A      LDXIM STMES  ; PRINTSTRING (' START=')
2440: 2115 A0 22      LDYIM STMES  /
2450: 2117 20 4F 22   JSR  PRTSTR ; PRINT (STARTADDRESS)
2460: 211A A5 FB      LDA  STADRH ;
2470: 211C 20 3B 1E   JSR  PRTBYT ;
2480: 211F A5 FA      LDA  STADRL ;
2490: 2121 20 3B 1E   JSR  PRTBYT ;
2500: 2124 A2 7F      LDXIM ENMSG  ; PRINT ('END = ')
2510: 2126 A0 22      LDYIM ENMSG  /
2520: 2128 20 4F 22   JSR  PRTSTR ;
2530: 212B 38          SEC      ;
2540: 212C A5 E0      LDA  RPTRL  ; HELP := READPOINTER - BEGINBUFFER
2550: 212E E9 00      SBCIM BUFFER ;
2560: 2130 85 DD      STA  HELPL  ;
2570: 2132 A5 E1      LDA  RPTRH  ;
2580: 2134 E9 23      SBCIM BUFFER /
2590: 2136 85 DE      STA  HELPH  ;
2600: 2138 18          CLC      ; HELP := HELP + STARTADDRESS
2610: 2139 A5 DD      LDA  HELPL  ;
2620: 213B 65 FA      ADC  STADRL  ;
2630: 213D 85 DD      STA  HELPL  ;
2640: 213F A5 DE      LDA  HELPH  ;
2650: 2141 65 FB      ADC  STADRH  ;
2660: 2143 20 3B 1E   JSR  PRTBYT ;
2670: 2146 A5 DD      LDA  HELPL  ;
2680: 2148 20 3B 1E   JSR  PRTBYT ;
2690: 214B A5 DF      LDA  CHKCOP  ;
2700: 214D D0 03      BNE  ELNK   ; IF CHECKCOPY FLAG <> THEN CHECK ONLY
2710: 214F 4C 70 21   JMP  CWRITE ;
2720: 2152 4C 7C 20   ELNK JMP  CREAD  ; END READ FILE TO BUFFER
2730:                ;
2740:                ; SUBROUTINE READBYTE
2750:                ;
2760: 2155 2C 00 17   RDBYT BIT  PAD   ; PROCEDURE READ(BYTE)
2770: 2158 30 03      BMI  MONRET ; IF BREAKKEY THEN EXIT TO MONITOR
2780:                ;
2790: 215A 4C F3 19   JMP  RDBTK  ; ENDF
2800: 215D A9 FF      MONRET LDAIM $FF ; READ BYTE FROM CASSETTE

```



```

2810: 215F 8D 02 17      STA  PBD      : END READ BYTE
2820: 2162 20 8C 1E      JSR  INIT      :
2830: 2165 4C 00 1C      JMP  MNITOR    :
2840:
2850: 2168 2C 00 17  RDCHT BIT  PAD      : PROCEDURE READ(CHARACTER)
2860: 216B 30 F0          BMI  MONRET    : IF BREAKKEY THEN EXIT TO MONITOR
2870: 216D 4C 24 1A      JMP  RDCHTK    : READ CHARACTER FROM TAPE
2880:
2890: 2170 A9 F7  CWRITE LDAIM $F7      : PROCEDURE WRITE BUFFER TO FILE
2900: 2172 8D 02 17      STA  PBD      : TURN WRITE CASSETTE MOTOR ON
2910: 2175 A2 08          LDXIM $08      :
2920: 2177 20 3F 22      JSR  DELAY      : DELAY (FILEGAP)
2930: 217A A9 27          LDAIM $27      : INIT WRITE CASSETTE I/O
2940: 217C 85 F5          STA  GANG      :
2950: 217E A9 BF          LDAIM $BF      :
2960: 2180 8D 43 17      STA  KPBDD     :
2970: 2183 A2 FF          LDXIM $FF      : COUNT := 255
2980: 2185 A9 16          LDAIM SYNC     : WHILE COUNT > 0 DO
2990: 2187 20 E2 21      JSR  NWRITE    : WRITE(SYNCHRONIZATION CHARACTER)
3000: 218A A9 2A          LDAIM SOD      : COUNT := COUNT - 1
3010: 218C 20 05 22      JSR  OUTCHT    : ENDWHILE
3020: 218F A5 F9          LDA  ID      : WRITE (START OF DATA )
3030: 2191 20 F1 21      JSR  OUTBT     : WRITE (ID)
3040: 2194 A5 FA          LDA  STADRL   : WRITE (STARTADDRESS)
3050: 2196 20 F1 21      JSR  OUTBT     :
3060: 2199 A5 FB          LDA  STADRH   :
3070: 219B 20 F1 21      JSR  OUTBT     :
3080: 219E A0 00  DATA  LDVIM $00      : WHILE WPOINTER < RPOINTER DO
3090: 21A0 B1 E2          LDAIV WPTL    : WRITE (BUFFER(WPOINTER)0)
3100: 21A2 20 F1 21      JSR  OUTBT     : WPOINTER := WPOINTER + 1
3110: 21A5 E6 E2          INC  WPTL    : ENDWHILE
3120: 21A7 D0 02          BNE  SAMP     :
3130: 21A9 E6 E3          INC  WPTRH    :
3140: 21AB A5 E2  SAMP  LDA  WPTL     :
3150: 21AD C5 E0          CMP  RPTL     :
3160: 21AF A5 E3          LDA  WPTRH    :
3170: 21B1 E5 E1          SBC  RPTRH    :
3180: 21B3 90 E9          BCC  DATA    :
3190: 21B5 A9 2F          LDAIM EOD     : WRITE (END OF DATA )
3200: 21B7 20 05 22      JSR  OUTCHT    :
3210: 21BA A5 F7          LDA  CHKHI    : WRITE (CHECKSUM)
3220: 21BC 20 F1 21      JSR  OUTBT     :
3230: 21BF A5 F6          LDA  CHKSUM    :
3240: 21C1 20 F1 21      JSR  OUTBT     :
3250: 21C4 A2 02          LDXIM $02     :
3260: 21C6 A9 04          LDAIM $04     : WRITE (END OF FILE)
3270: 21C8 20 E2 21      JSR  NWRITE    :
3280: 21CB A9 FF          LDAIM $FF     : TURN WRITE CASSETTE MOTOR OFF
3290: 21CD 8D 02 17      STA  PBD      :
3300: 21D0 20 8C 1E      JSR  INIT      :
3310: 21D3 A2 88          LDXIM COPYMS : PRINT ('COPY !')
3320: 21D5 A0 22          LDVIM COPYMS /
3330: 21D7 20 4F 22      JSR  PRTSTR    :
3340: 21DA A9 21          LDAIM '!'     :
3350: 21DC 20 A0 1E      JSR  OUTCH    :
3360: 21DF 4C 7C 20      JMP  CREAD    : END WRITE BUFFER TO FILE

```



```

3370:      ;
3380:      ; SUBROUTINE NWRITE
3390:      ;
3400: 21E2 86 F1      NWRITE STX    CC      ; PROCEDURE NWRITE (N,CHARACTER)
3410: 21E4 48          HICA  PHA      ;   WHILE N > 0 DO
3420: 21E5 20 05 22    JSR    OUTCHT  ;   WRITE (CHARACTER)
3430: 21E8 68          PLA      ;   N := N - 1
3440: 21E9 C6 F1      DEC    CC      ;   ENDWHILE
3450: 21EB D0 F7      BNE    HICA      ; END NWRITE
3460: 21ED 60          RTS      ;
3470:      ;
3480:      ; SUBROUTINE OUTBTC
3490:      ;
3500: 21EE 20 4C 19    OUTBTC JSR    CHKT  ; PROCEDURE WRITE(BYTE) AS TWO ASCII
3510: 21F1 48          OUTBT  PHA      ;
3520: 21F2 4A          LSRA      ;   CHECKSUM := CHECKSUM + BYTE
3530: 21F3 4A          LSRA      ;   CONVERT LEFT NIBBLE OF BYTE TO ASCII
3540: 21F4 4A          LSRA      ;   WRITE(ASCII)
3550: 21F5 4A          LSRA      ;   CONVERT RIGHT NIBBLE OF BYTE TO ASCII
3560: 21F6 20 FA 21    JSR    HEXT      ;   WRITE(ASCII)
3570: 21F9 68          PLA      ; END WRITE BYTE
3580: 21FA 29 0F      HEXT  ANDIM $0F  ;
3590: 21FC C9 0A      CMPIM $0A      ;
3600: 21FE 18          CLC      ;
3610: 21FF 30 02      BMI    HEXAT      ;
3620: 2201 69 07      ADCIM $07      ;
3630: 2203 69 30      HEXAT  ADCIM $30  ;
3640:      ;
3650: 2205 A0 08      OUTCHT LDVIM $08  ; PROCEDURE WRITE (BYTE)
3660: 2207 84 F2      STY    COUNT      ;   COUNT :=8
3670: 2209 A0 02      TRY    LDVIM $02  ;   WHILE COUNT > 0 DO
3680: 220B 84 FE      STY    TRIB      ;   SEND 3 PULSES 3700 HZ
3690: 220D BE 3B 22    ZON    LDXAY NPUL ;   IF MSB BYTE = 1 THEN
3700: 2210 48          PHA      ;   SEND 3 PULSES 3700 HZ
3710: 2211 2C 47 17    ZONA  BIT    TIMER ;   ELSE
3720: 2214 10 FB      BPL    ZONA      ;   SEND 2 PULSES 2400 HZ
3730: 2216 B9 3C 22    LDAAV TIMG      ;
3740: 2219 8D 44 17    STA    TIMERT    ;   ENDIF
3750: 221C A5 F5      LDA    GANG      ;   SEND 2 PULSES 2400 HZ
3760: 221E 49 80      EORIM $80      ;   SHIFT BYTE LEFT
3770: 2220 8D 42 17    STA    KPBD      ;   ENDWHILE
3780: 2223 85 F5      STA    GANG      ; END WRITE BYTE
3790: 2225 CA          DEX      ;
3800: 2226 D0 E9      BNE    ZONA      ;
3810: 2228 68          PLA      ;
3820: 2229 C6 FE      DEC    TRIB      ;
3830: 222B F0 05      BEQ    SETZ      ;
3840: 222D 30 07      BMI    ROUT      ;
3850: 222F 4A          LSRA      ;
3860: 2230 90 DB      BCC    ZON      ;
3870: 2232 A0 00      SETZ   LDVIM $00  ;
3880: 2234 F0 D7      BEQ    ZON      ;
3890: 2236 C6 F2      ROUT   DEC    COUNT ;
3900: 2238 D0 CF      BNE    TRY      ;
3910: 223A 60          RTS      ;
3920:      ;

```



```

3930:                ; TIMING TABLE
3940:                ;
3950: 223B 02      NPUL  =    $02    ; NUMBER OF 2400 HZ PULSES
3960: 223C 03      TIMG  =    $03    ; TIMER COUNT
3970: 223D 03              =    $03    ; NUMBER OF 3700 PULSES
3980: 223E 7E              =    $7E    ; TIMER COUNT
3990:                ;
4000:                ; DELAY ROUTINE
4010:                ;
4020: 223F A0 FF      DELAY LDYIM $FF    ; PROCEDURE DELAY (NR)
4030: 2241 A9 FF      YLOOP LDAIM $FF    ;   DELAY TIME * NR
4040: 2243 38      ALOOP SEC              ; END DELAY
4050: 2244 E9 01              SBCIM $01    ;
4060: 2246 D0 FB      BNE  ALOOP          ;
4070: 2248 88              DEY              ;
4080: 2249 D0 F6      BNE  YLOOP          ;
4090: 224B CA              DEX              ;
4100: 224C D0 F1      BNE  DELAY          ;
4110: 224E 60              RTS              ;
4120:                ;
4130:                ; SUBROUTINE PRINT STRING
4140:                ;
4150: 224F 86 ED      PRTSTR STX  STRPTL  ; PROCEDURE PRINTSTRING (STRINGPTR)
4160: 2251 84 EE      STY  STRPTH          ;
4170: 2253 A0 00      PNEXT LDYIM $00      ;   WHILE NOT END OF STRING DO
4180: 2255 B1 ED      LDAIY STRPTL          ;   PRINT (MEMORY (STRINGPTR))
4190: 2257 48              PHA              ;   STRINGPTR := STRINGPTR + 1
4200: 2258 20 A0 1E    JSR  OUTCH          ;   ENDWHILE
4210: 225B 68              PLA              ;
4220: 225C 30 09      BMI  LAST            ;
4230: 225E E6 ED      INC  STRPTL          ;
4240: 2260 D0 02      BNE  NEXTS           ;
4250: 2262 E6 EE      INC  STRPTH          ;
4260: 2264 4C 53 22    NEXTS JMP  PNEXT     ;
4270: 2267 60              LAST  RTS        ;
4280:                ;
4290:                ; SUBROUTINE GET CHARACTER
4300:                ;
4310: 2268 2C 00 17    GETCH BIT  PAD      ; PROCEDURE GETCHARACTER ( CHAR )
4320: 226B 10 FB      BPL  GETCH          ;   WAIT FOR STROBE
4330: 226D 2C 00 17    GWAIT BIT  PAD      ;   WAIT FOR END OF STROBE
4340: 2270 30 FB      BMI  GWAIT          ;
4350: 2272 AD 00 17    LDA  PAD          ;   GET CHARACTER
4360: 2275 60              RTS            ;   END GETCHARACTER
4370:                ;
4380:                ; STRING DATA
4390:                ;
4400: 2276 49      IDMES  =    'I
4410: 2277 44              =    'D
4420: 2278 3D              =    '='
4430: 2279 A0              =    $A0
4440:                ;
4450: 227A 20      STMES  =    '
4460: 227B 53              =    'S
4470: 227C 54              =    'T
4480: 227D 3D              =    '='

```

PARAL
KEYB
INPU


```
4490: 227E A0          =      $A0
4500:                ;
4510: 227F 20      ENMSG  =      '/'
4520: 2280 45          =      '/E
4530: 2281 4E          =      '/N
4540: 2282 3D          =      '/=
4550: 2283 A0          =      $A0
4560:                ;
4570: 2284 54      HELLOM =      '/T
4580: 2285 41          =      '/A
4590: 2286 50          =      '/P
4600: 2287 45          =      '/E
4610: 2288 20      COPYMS =      '/'
4620: 2289 43          =      '/C
4630: 228A 4F          =      '/O
4640: 228B 50          =      '/P
4650: 228C 59          =      '/Y
4660: 228D A0          =      $A0
4670:                ;
4680:                ;
4690: 228E 43      CHKMSG =      '/C
4700: 228F 48          =      '/H
4710: 2290 45          =      '/E
4720: 2291 43          =      '/C
4730: 2292 4B          =      '/K
4740: 2293 A0          =      $A0
4750: 2294 4F      ONLMSG =      '/O
4760: 2295 4E          =      '/N
4770: 2296 4C          =      '/L
4780: 2297 59          =      '/Y
4790: 2298 20          =      '/'
4800: 2299 3F          =      '/?
4810: 229A A0          =      $A0
4820: 229B 56      URSMSG =      '/U
4830: 229C 31          =      '/1
4840: 229D 2E          =      '/.
4850: 229E 31          =      '/1
4860: 229F A0          =      $A0
4870:                ;
4880: 22A0 41      ANDMSG =      '/A
4890: 22A1 4E          =      '/N
4900: 22A2 44          =      '/D
4910: 22A3 A0          =      $A0
```