APPENDIX M

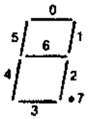
VERSION 1.1 MONITOR ENHANCEMENTS

SY1.1, the second release of the SUPERMON monitor program, includes several enhancements and fixes to SY1.0.

All entry point addresses in the main monitor program remain unchanged from version SY1.0. In the cassette routines, however, only LOADT and DUMPT can be relied upon to be at the old addresses.

The most useful enhancements come in the cassette software. First, the high speed tape leader time is now variable. It can be changed by modifying location TAPDEL, A630. TAPDEL is interpreted in units of 1.5 seconds of SYNC characters.

While reading a cassette file, its file ID will be displayed in non-decoded form on the leftmost LED. The display may be interpreted as follows:



where the numbers alongside the segments indicate the binary bit which corresponds to that segment.

Example: File 01

File 02

File 54

File FE

File 03

During the SYNC search (while the "S." is lit on the display), the tape load operation may be interrupted by depressing the CR button on the hex keypad and turning off the recorder. The message Er 8C will be displayed.

The high speed write waveform is now variable. For information on using this feature, see technical note #72-SSC.

The error count after a paper tape load operation is displayed. The first digit of the error number indicates the number of invalid characters of checksums encountered in the tape, and the second digit indicates the number of memory errors (invalid readbacks). In each case, the digit 'F' indicates greater than or equal to 15 errors.

APPENDIX N

SYM I/O SOFTWARE

Using the SYM's character I/O routines in your software is easy and provides great flexibility.

The I/O devices to be discussed are:

- CRT	An RS-232 serial ASCII terminal
- TTY	A 20mA loop serial ASCII device
- HKB	The 25 key ASCII keyboard on the SYM
- DSP	The 6 digit 7 segment ASCII display on the SYM
- SCP	The oscilloscope driver circuitry on the SYM
- UIN	A user provided special input device
- UOUT	A user provided special output device

The software modules are:

- VEC	I/O Vectors in Monitor RAM
- TIO	Serial Terminal drivers in ROM
- HDOUT, HKEY HKB and DSP	Drivers in ROM
- SCD	SCP drivers in RAM
- INCHR, OUTCHR	Character I/O subroutines in ROM
- UIO	User I/O drivers in RAM
- MON	The monitor program itself, exclusive of I/O

The easiest to use method of CHAR I/O is used by MON, this method is explained first.

Whenever the monitor wants to output a character, it places the ASCII value of the character in A and calls OUTCHR. The character is output and monitor resumes. That character may have gone out to a CRT, TTY, or both, DSP, SCP, or UOUT.

The vector OUTVEC controlled which device received the output by containing the address of the driver for the receiving device. If the output device is a CRT or TTY, then OUTVEC contains the address of ("points to") a driver called TOUT (Terminal OUT). When the DSP is the output device, OUTVEC points to HDOUT. The decision of which output driver to point to is made initially by monitor at sign on time. If, after RESET, a key on HKB is pressed, OUTVEC is left pointing to HDOUT (and INVEC pointing to HKEY). If a "1" or "Q" is pressed on the CRT, then OUTVEC is loaded with the address of TOUT. The I/O Vectors will not be changed again until RESET or the user loads them himself. If you wish to output characters to the device that you signed in on, then place the character in A and JSR to OUTCHR. Output will be disabled if the OUTPUT DISABLE FLAG (bit 6 of TECHO) is set.

When monitor needs to input a character, it calls INCHR. Control is vectored thru INVEC so that input is obtained from the device, CRT or HKB, that was signed in on. Several things occur to the character as it is being input. If it is lower case alphabetic, it is changed to upper case. Bit 7 is cleared. If the character is Control 0, ASCII ØF, then the OUTPUT DISABLE FLAG is toggled, and another character is obtained. The ASCII CHAR is returned in A. Both INCHR and OUTCHR are transparent to the X and Y registers.

When you connect a UIN or UOUT to the SYM, you must also write the I/O driver for it. If UIN is an unencoded keyboard, for instance, you must connect it to a 6522 VIA's I/O ports, and write the driver that will configure the VIA, scan the keys, debounce them, and return the ASCII character in A. To use this UIN, sign in on HKB or CRT, then change INVEC to point to the UIN input driver subroutine. The MON command SD is provided for this purpose. Now whenever MON or BASIC or a user program calls INCHR, the character will be obtained from UIN. Similarly, if the output device is to be an oscilloscope, then the scope driver subroutine provided in this manual should be installed in RAM and OUTVEC (if necessary) and SCNVEC should be pointed to this subroutine. All characters sent to OUTCHR will appear on the scope.

While HKB and DSP are in use, the following vectors are used: INVEC, OUTVEC, INSVEC (points to the routine that checks the input device for a BREAK condition), and SCNVEC (lights up the display or oscilloscope, etc. while waiting for an input key).

By vectoring the I/O on the SYM, any I/O devices can be used by all the software by calling INCHR and OUTCHR. If you wish to input lower case letters, then JSR INVEC directly or duplicate the code in bytes 8A1B thru 8A40 in RAM and change bytes 8A23 thru 8A2C to NOP's. Note that placing I/O routines in RAM can cause difficulties during Trace or Debug because the I/O routine used to print the Trace may itself get traced.

If your software needs special capability, you may choose to call MON I/O subroutines directly. For instance, lighting up the DSP to do a "video" game like ping-pong requires that you display some patterns that may not have ASCII equivalents. In this case the six locations of DISBUF should be poked to display the desired pattern and SCAND repeatedly called to light up the display. The subroutine ACCESS can be called to unwrite-protect the MON RAM. To control the game the HKB keys "4" and "CR" are especially useful. The subroutines KYSTAT and LRNKEY will tell you whether any key is down or not, and if so, which key is down, in ASCII. When an oscilloscope is used for output, the subroutine TEXT can do the bookkeeping for keeping the last 32 CHARS in SCPBUF ready for use.

Since all character I/O on the SYM can be in ASCII, even with the HKB and DSP, your software can use the HKB and DSP, CRT, TTY, SCP, or special I/O devices, without having to be rewritten. MON, for instance, doesn't even "know" whether the HKB and DSP or a CRT are being used for I/O.

If you have used other systems, maybe the KIM, for instance, some differences should be apparent. More I/O capability exists on the SYM than is typical. While no two systems handle I/O alike, most of the I/O routines accomplish the same task. In the KIM, the display can be lighted with hexadecimal characters, alas with no decimal points, by storing the hex in three bytes 00F9, 00FA, and 00FB. In the SYM the same thing is done by calling OUTBYT three times with the hex bytes in A. Or call OUTCHR up to 6 times, or store the 7 segment codes in DISBUF. To get a keycode from the HKB use: HKEY, GETKEY, GK, LRNKEY, and/or KYSTAT. Note that if you call GK, not only is the ASCII code returned in A, but X contains a "key code". The key codes for Ø thru F are ØØ thru ØF, the hex nibble equivalents, very handy. Since MON is 85% subroutines, the potential to save time and code by calling MON subroutines is unusually large. The program listing in the back of the manual provides the details of how all subroutines work.

SUPERMON Monitor Listing

```
LINE # LOC     CODE      LINE

0002  0000              ;
0003  0000              ;*****
0004  0000              ;***** COPYRIGHT 1979 SYNERTEK SYSTEMS CORPORATION
0005  0000              ;***** VERSION 2  4/13/79   "SY1.1"
0006  0000                     *=$A600          ;SYS RAM (ECHOED AT TOP OF MEM)
0007  A600              SCPBUF *=*+$20          ;SCOPE BUFFER LAST 32 CHRS
0008  A620              RAM    =*               ;DEFAULT BLK FILLS STARTING HERE
0009  A620              JTABLE *=*+$10          ; 8JUMPS - ABS ADDR, LO HI ORDER
0010  A630              TAPDEL *=*+1            ;HS TAPE DELAY
0011  A631              KMBDRY *=*+1            ;KIM TAPE READ BOUNDARY
0012  A632              HSBDRY *=*+1            ;HS TAPE READ BOUNDARY
0013  A633              SCR3   *=*+1            ;RAM SCRATCH LOCS 3-F
0014  A634              SCR4   *=*+1
0015  A635              TAPET1 *=*+1            ;HS TAPE 1/2 BIT TIME
0016  A636              SCR6   *=*+1
0017  A637              SCR7   *=*+1
0018  A638              SCR8   *=*+1
0019  A639              SCR9   *=*+1
0020  A63A              SCRA   *=*+1
0021  A63B              SCRB   *=*+1
0022  A63C              TAPET2 *=*+1            ;HS TAPE 1/2 BIT TIME
0023  A63D              SCRD   *=*+1
0024  A63E              RC     =SCRD
0025  A63E              SCRE   *=*+1
0026  A63F              SCRF   *=*+1
0027  A640              DISBUF *=*+5            ;DISPLAY BUFFER
0028  A645              RDIG   *=*+1            ;RIGHT MOST DIGIT OF DISPLAY
0029  A646                     *=*+3            ;NOT USED
0030  A649              PARNR  *=*+1            ;NUMBER OF PARMS RECEIVED
0031  A64A              ;
0032  A64A              ; 3 16 BIT PARMS, LO HI ORDER
0033  A64A              ; PASSED TO EXECUTE BLOCKS
0034  A64A              ;
0035  A64A              P3L    *=*+1
0036  A64B              P3H    *=*+1
0037  A64C              P2L    *=*+1
0038  A64D              P2H    *=*+1
0039  A64E              P1L    *=*+1
0040  A64F              P1H    *=*+1
0041  A650              PADBIT *=*+1            ;PAD BITS FOR CARRIAGE RETURN
0042  A651              SDBYT  *=*+1            ;SPEED BYTE FOR TERMINAL I/O
0043  A652              ERCNT  *=*+1            ; ERROR COUNT  (MAX $FF)
0044  A653              ; BIT 7 = ECHO /NO ECHO, BIT 6 = CTL O TOGGLE SW
0045  A653              TECHO  *=*+1            ;TERMINAL ECHO FLAG
0046  A654              ; BIT7 =CRT IN, 6 =TTY IN, 5 = TTY OUT, 4 = CRT OUT
0047  A654              TOUTFL *=*+1            ;OUTPUT FLAGS
0048  A655              KSHFL  *=*+1            ;KEYBOARD SHIFT FLAG
0049  A656              TV     *=*+1            ;TRACE VELOCITY (0=SINGLE STEP)
0050  A657              LSTCOM *=*+1            ;STORE LAST MONITOR COMMAND
0051  A658              MAXRC  *=*+1            ;MAX REC LENGTH FOR MEM DUMP
0052  A659              ;
0053  A659              ; USER REG'S FOLLOW
0054  A659              ;
0055  A659              PCLR   *=*+1            ;PROG CTR
0056  A65A              PCHR   *=*+1
```

```
LINE # LOC     CODE        LINE

0057   A65B                SR      *=*+1             ;STACK
0058   A65C                FR      *=*+1             ;FLAGS
0059   A65D                AR      *=*+1             ;AREG
0060   A65E                XR      *=*+1             ;XREG
0061   A65F                YR      *=*+1             ;YREG
0062   A660                ;
0063   A660                ; I/O VECTORS FOLLOW
0064   A660                ;
0065   A660                INVEC   *=*+3             ;IN CHAR
0066   A663                OUTVEC *=*+3              ;OUT CHAR
0067   A666                INSVEC *=*+3              ;IN STATUS
0068   A669                URSVEC *=*+3              ;UNRECOGNIZED SYNTAX VECTOR
0069   A66C                URCVEC *=*+3              ;UNRECOGNIZED CMD/ERROR VECTOR
0070   A66F                SCNVEC *=*+3              ;SCAN ON-BOARD DISPLAY
0071   A672                ;
0072   A672                ; TRACE, INTERRUPT VECTORS
0073   A672                ;
0074   A672                EXEVEC *=*+2              ; EXEC CMD ALTERNATE INVEC
0075   A674                TRCVEC *=*+2              ;TRACE
0076   A676                UBRKVC *=*+2              ;USER BRK AFTER MONITOR
0077   A678                UBRKV   =UBRKVC
0078   A678                UIRQVC *=*+2              ;USER NON-BRK IRQ AFTER MONITOR
0079   A67A                UIRQV   =UIRQVC
0080   A67A                NMIVEC *=*+2              ;NMI
0081   A67C                RSTVEC *=*+2              ;RESET
0082   A67E                IRQVEC *=*+2              ;IRQ
0083   A680                ;
0084   A680                ;
0085   A680                ;I/O REG DEFINITIONS
0086   A680                PADA    =$A400            ;KEYBOARD/DISPLAY
0087   A680                PBDA    =$A402            ;SERIAL I/O
0088   A680                OR3A=$AC01                ;WP, DBON, DBOFF
0089   A680                DDR3A=OR3A+2              ;DATA DIRECTION FOR SAME
0090   A680                OR1B=$A000
0091   A680                DDR1B=$A002
0092   A680                PCR1    =$A00C            ; POR/TAPE REMOTE
0093   A680                ;
0094   A680                ; MONITOR MAINLINE
0095   A680                ;
0096   A680                       *=$8000
0097   8000   4C 7C 8B     MONITR JMP MONENT         ;INIT S, CLD, GET ACCESS
0098   8003   20 FF 80     WARM   JSR GETCOM         ;GET COMMAND + PARMS  (0-3)
0099   8006   20 4A 81            JSR DISPAT         ;DISPATCH CMD,PARMS TO EXEC BLKS
0100   8009   20 71 81            JSR ERMSG          ;DISP ER MSG IF CARRY SET
0101   800C   4C 03 80            JMP WARM           ;AND CONTINUE
0102   800F                ;
0103   800F                ; TRACE AND INTERRUPT ROUTINES
0104   800F                ;
0105   800F   08           IRQBRK PHP                ;IRQ OR BRK ?
0106   8010   48                  PHA
0107   8011   8A                  TXA
0108   8012   48                  PHA
0109   8013   BA                  TSX
0110   8014   BD 04 01            LDA $104,X         ;PICK UP FLAGS
0111   8017   29 10               AND #$10
```

```
LINE # LOC     CODE        LINE

0112   8019   F0 07                 BEQ DETIRQ
0113   801B   68           DETBRK FLA                    ;BRK
0114   801C   AA                  TAX
0115   801D   68                  PLA
0116   801E   28                  PLP
0117   801F   6C F6 FF            JMP ($FFF6)
0118   8022   68           DETIRQ PLA                    ;IRQ (NON BRK)
0119   8023   AA                  TAX
0120   8024   68                  PLA
0121   8025   28                  PLP
0122   8026   6C F8 FF            JMP ($FFF8)
0123   8029   20 86 8B     SVIRQ  JSR ACCESS             ;SAVE REGS AND DISPLAY CODE
0124   802C   38                  SEC
0125   802D   20 64 80            JSR SAVINT
0126   8030   A9 31               LDA #'1
0127   8032   4C 53 80            JMP IDISP
0128   8035   08           USRENT PHP                    ;USER ENTRY
0129   8036   20 86 8B            JSR ACCESS
0130   8039   38                  SEC
0131   803A   20 64 80            JSR SAVINT
0132   803D   EE 59 A6            INC PCLR
0133   8040   D0 03               BNE *+5
0134   8042   EE 5A A6            INC PCHR
0135   8045   A9 33               LDA #'3
0136   8047   4C 53 80            JMP IDISP
0137   804A   20 86 8B     SVBRK  JSR ACCESS
0138   804D   18                  CLC
0139   804E   20 64 80            JSR SAVINT
0140   8051   A9 30               LDA #'0
0141   8053                ; INTRPT CODES  0 = BRK
0142   8053                ;               1 = IRQ
0143   8053                ;               2 = NMI
0144   8053                ;               3 = USER ENTRY
0145   8053   48           IDISP  PHA                    ;OUT PC, INTRPT CODE (FROM A)
0146   8054   20 D3 80            JSR DBOFF              ;STOP NMI'S
0147   8057   20 4D 83            JSR CRLF
0148   805A   20 37 83            JSR OPCCOM
0149   805D   68                  PLA
0150   805E   20 47 8A            JSR OUTCHR
0151   8061   4C 03 80            JMP WARM
0152   8064   8D 5D A6     SAVINT STA AR                 ;SAVE USER REGS AFTER INTRPT
0153   8067   8E 5E A6            STX XR
0154   806A   8C 5F A6            STY YR
0155   806D   BA                  TSX
0156   806E   D8                  CLD
0157   806F   BD 04 01            LDA $104,X
0158   8072   69 FF               ADC #$FF
0159   8074   8D 59 A6            STA PCLR
0160   8077   BD 05 01            LDA $105,X
0161   807A   69 FF               ADC #$FF
0162   807C   8D 5A A6            STA PCHR
0163   807F   BD 03 01            LDA $103,X
0164   8082   8D 5C A6            STA FR
0165   8085   BD 02 01            LDA $102,X
0166   8088   9D 05 01            STA $105,X
```

```
LINE # LOC     CODE       LINE

0167  808B  BD 01 01             LDA $101,X
0168  808E  9D 04 01             STA $104,X
0169  8091  E8                   INX
0170  8092  E8                   INX
0171  8093  E8                   INX
0172  8094  9A                   TXS
0173  8095  E8                   INX
0174  8096  E8                   INX
0175  8097  8E 5B A6             STX SR
0176  809A  60                   RTS
0177  809B  20 86 8B     SVNMI   JSR ACCESS         ;TRACE IF TV NE 0
0178  809E  38                   SEC
0179  809F  20 64 80             JSR SAVINT
0180  80A2  20 D3 80             JSR DBOFF          ;STOP NMI'S
0181  80A5  AD 56 A6             LDA TV
0182  80A8  D0 05                BNE TVNZ
0183  80AA  A9 32                LDA #'2
0184  80AC  4C 53 80             JMP IDISP
0185  80AF  20 37 83     TVNZ    JSR OPCCOM         ;TRACE WITH DELAY
0186  80B2  AD 5D A6             LDA AR
0187  80B5  20 4A 83             JSR OBCRLF         ;DISPLAY ACC
0188  80B8  20 5A 83             JSR DELAY
0189  80BB  90 10                BCC TRACON         ;STOP IF KEY ENTERED
0190  80BD  4C 03 80             JMP WARM
0191  80C0  20 86 8B     TRCOFF  JSR ACCESS         ;DISABLE NMIS
0192  80C3  38                   SEC
0193  80C4  20 64 80             JSR SAVINT
0194  80C7  20 D3 80             JSR DBOFF
0195  80CA  6C 74 A6             JMP (TRCVEC)       ;AND GO TO SPECIAL TRACE
0196  80CD  20 E4 80     TRACON  JSR DBON           ;ENABLE NMI'S
0197  80D0  4C FD 83             JMP GO1ENT+3       ;AND RESUME (NO WRITE PROT)
0198  80D3  AD 01 AC     DBOFF   LDA OR3A           ;PULSE DEBUG OFF
0199  80D6  29 DF                AND #$DF
0200  80D8  09 10                ORA #$10
0201  80DA  8D 01 AC             STA OR3A
0202  80DD  AD 03 AC             LDA DDR3A
0203  80E0  09 30                ORA #$30
0204  80E2  D0 0F                BNE DBNEW-3        ;RELEASE FLIP FLOP SO KEY WORKS
0205  80E4  AD 01 AC     DBON    LDA OR3A           ;PULSE DEBUG ON
0206  80E7  29 EF                AND #$EF
0207  80E9  09 20                ORA #$20
0208  80EB  8D 01 AC             STA OR3A
0209  80EE  AD 03 AC             LDA DDR3A
0210  80F1  09 30                ORA #$30
0211  80F3  8D 03 AC             STA DDR3A
0212  80F6  AD 03 AC     DBNEW   LDA DDR3A          ;RELEASE FLIP FLOP
0213  80F9  29 CF                AND #$CF
0214  80FB  8D 03 AC             STA DDR3A
0215  80FE  60                   RTS
0216  80FF               ;
0217  80FF               ; GETCOM - GET COMMAND AND 0-3 PARMS
0218  80FF               ;
0219  80FF  20 4D 83     GETCOM  JSR CRLF
0220  8102  A9 2E                LDA #'.        ;PROMPT
0221  8104  20 47 8A             JSR OUTCHR
```

```
LINE # LOC     CODE       LINE

0222  8107  20 1B 8A    GETC1  JSR INCHR
0223  810A  F0 F3              BEQ GETCOM       ;CARRIAGE RETURN?
0224  810C  C9 7F              CMP #$7F         ;DELETE?
0225  810E  F0 F7              BEQ GETC1
0226  8110  C9 00              CMP #0           ;NULL?
0227  8112  F0 F3              BEQ GETC1
0228  8114              ; L,S,U NEED TO BE HASHED 2 BYTES TO ONE
0229  8114  C9 53              CMP #'S
0230  8116  F0 1B              BEQ HASHUS
0231  8118  C9 55              CMP #'U
0232  811A  F0 17              BEQ HASHUS
0233  811C  C9 4C              CMP #'L
0234  811E  F0 0F              BEQ HASHL
0235  8120  8D 57 A6    STOCOM STA LSTCOM
0236  8123  20 42 83           JSR SPACE
0237  8126  20 08 82           JSR PSHOVE       ;ZERO PARMS
0238  8129  20 08 82           JSR PSHOVE
0239  812C  4C 20 82           JMP PARM        ;AND GO GET PARMS
0240  812F  A9 01       HASHL  LDA #$01         ;HASH LOAD CMDS TO ONE BYTE
0241  8131  10 02              BPL HASHUS+2
0242  8133  0A          HASHUS ASL A            ;HASH 'USER' CMDS TO ONE BYTE A
0243  8134  0A                 ASL A            ;U0 = $14 THRU U7 =$1B
0244  8135  8D 57 A6           STA LSTCOM
0245  8138  20 1B 8A           JSR INCHR        ;GET SECOND
0246  813B  F0 C2              BEQ GETCOM       ;CARRIAGE RETURN?
0247  813D  18                 CLC
0248  813E  6D 57 A6           ADC LSTCOM
0249  8141  29 0F              AND #$0F
0250  8143  09 10              ORA #$10
0251  8145  10 D9              BPL STOCOM
0252  8147  FF                 .BYT $FF,$FF,$FF ;NOT USED
0252  8148  FF
0252  8149  FF
0253  814A              ;
0254  814A              ;DISPATCH TO EXEC BLK 0PARM, 1PARM, 2PARM, OR 3PARM
0255  814A              ;
0256  814A  C9 0D       DISPAT CMP #$0D         ;C/R IF OK ELSE URSVEC
0257  814C  D0 20              BNE HIPN
0258  814E  AD 57 A6           LDA LSTCOM
0259  8151  AE 49 A6           LDX PARNR
0260  8154  D0 03              BNE M12
0261  8156  4C 95 83           JMP BZPARM       ;0 PARM BLOCK
0262  8159  E0 01       M12    CPX #$01
0263  815B  D0 03              BNE M13
0264  815D  4C DA 84           JMP B1PARM       ;1PARM BLOCK
0265  8160  E0 02       M13    CPX #$02
0266  8162  D0 03              BNE M14
0267  8164  4C 19 86           JMP B2PARM      ;2 PARM BLOCK
0268  8167  E0 03       M14    CPX #$03
0269  8169  D0 03              BNE HIPN
0270  816B  4C 14 87           JMP B3PARM      ;3 PARM BLOCK
0271  816E  6C 6A A6    HIPN   JMP (URSVEC+1)   ;ELSE UNREC SYNTAX VECTOR
0272  8171              ;
0273  8171              ; ERMSG - PRINT ACC IN HEX IF CARRY SET
0274  8171              ;
```

```
LINE # LOC     CODE       LINE

0275   8171   90 44      ERMSG  BCC M15
0276   8173   48                PHA
0277   8174   20 4D 83          JSR CRLF
0278   8177   A9 45             LDA #'E
0279   8179   20 47 8A          JSR OUTCHR
0280   817C   A9 52             LDA #'R
0281   817E   20 47 8A          JSR OUTCHR
0282   8181   20 42 83          JSR SPACE
0283   8184   68                PLA
0284   8185   4C FA 82          JMP OUTBYT
0285   8188              ;
0286   8188              ; SAVER - SAVE ALL REG'S + FLAGS ON STACK
0287   8188              ; RETURN WITH F,A,X,Y UNCHANGED
0288   8188              ; STACK HAS          FLAGS,A,X,Y PUSHED
0289   8188   08         SAVER  PHP                 ;
0290   8189   48                PHA          ;
0291   818A   48                PHA          ;
0292   818B   48                PHA
0293   818C   08                PHP
0294   818D   48                PHA
0295   818E   8A                TXA
0296   818F   48                PHA
0297   8190   BA                TSX
0298   8191   BD 09 01          LDA $0109,X
0299   8194   9D 05 01          STA $0105,X
0300   8197   BD 07 01          LDA $0107,X
0301   819A   9D 09 01          STA $0109,X
0302   819D   BD 01 01          LDA $0101,X
0303   81A0   9D 07 01          STA $0107,X
0304   81A3   BD 08 01          LDA $0108,X
0305   81A6   9D 04 01          STA $0104,X
0306   81A9   BD 06 01          LDA $0106,X
0307   81AC   9D 08 01          STA $0108,X
0308   81AF   98                TYA
0309   81B0   9D 06 01          STA $0106,X
0310   81B3   68                PLA
0311   81B4   AA                TAX
0312   81B5   68                PLA
0313   81B6   28                PLP
0314   81B7   60         M15    RTS
0315   81B8              ; RESTORE EXCEPT A,F
0316   81B8   08         RESXAF PHP
0317   81B9   BA                TSX
0318   81BA   9D 04 01          STA $0104,X
0319   81BD   28                PLP
0320   81BE              ; RESTORE EXCEPT F
0321   81BE   08         RESXF  PHP
0322   81BF   68                PLA
0323   81C0   BA                TSX
0324   81C1   9D 04 01          STA $0104,X
0325   81C4              ; RESTORE ALL 100%
0326   81C4   68         RESALL PLA
0327   81C5   A8                TAY
0328   81C6   68                PLA
0329   81C7   AA                TAX
```

```
LINE # LOC     CODE        LINE

0330  81C8  68                  PLA
0331  81C9  28                  PLP
0332  81CA  60                  RTS
0333  81CB             ;
0334  81CB             ; MONITOR UTILITIES
0335  81CB             ;
0336  81CB  C9 20      ADVCK    CMP #$20        ;SPACE?
0337  81CD  F0 02               BEQ M1
0338  81CF  C9 3E               CMP #'>       ;FWD ARROW?
0339  81D1  38         M1       SEC
0340  81D2  60                  RTS
0341  81D3  20 FA 82   OBCMIN JSR OUTBYT        ;OUT BYTE, OUT COMMA, IN BYTE
0342  81D6  20 3A 83   COMINB JSR COMMA         ;OUT COMMA, IN BYTE
0343  81D9  20 1B 8A   INBYTE JSR INCHR
0344  81DC  20 75 82            JSR ASCNIB
0345  81DF  B0 14               BCS OUT4
0346  81E1  0A                  ASL A
0347  81E2  0A                  ASL A
0348  81E3  0A                  ASL A
0349  81E4  0A                  ASL A
0350  81E5  8D 33 A6            STA SCR3
0351  81E8  20 1B 8A            JSR INCHR
0352  81EB  20 75 82            JSR ASCNIB
0353  81EE  B0 11               BCS OUT2
0354  81F0  0D 33 A6            ORA SCR3
0355  81F3  18         GOOD     CLC
0356  81F4  60                  RTS
0357  81F5  C9 3A      OUT4     CMP #':         ;COLON ?
0358  81F7  D0 05               BNE OUT1
0359  81F9  20 1B 8A            JSR INCHR
0360  81FC  D0 F5               BNE GOOD        ;CARRIAGE RETURN?
0361  81FE  B8         OUT1     CLV
0362  81FF  50 03               BVC CRCHK
0363  8201  2C 04 82   OUT2     BIT CRCHK
0364  8204  C9 0D      CRCHK    CMP #$0D        ;CHECK FOR C/R
0365  8206  38                  SEC
0366  8207  60                  RTS
0367  8208  A2 10      PSHOVE LDX #$10          ;PUSH PARMS DOWN
0368  820A  0E 4A A6   PRM10    ASL P3L
0369  820D  2E 4B A6            ROL P3H
0370  8210  2E 4C A6            ROL P2L
0371  8213  2E 4D A6            ROL P2H
0372  8216  2E 4E A6            ROL P1L
0373  8219  2E 4F A6            ROL P1H
0374  821C  CA                  DEX
0375  821D  D0 EB               BNE PRM10
0376  821F  60                  RTS
0377  8220  20 88 81   PARM     JSR SAVER       ;GET PARMS - RETURN ON C/R OR ERR
0378  8223  A9 00               LDA #0
0379  8225  8D 49 A6            STA PARNR
0380  8228  8D 33 A6            STA SCR3
0381  822B  20 08 82   PM1      JSR PSHOVE
0382  822E  20 1B 8A   PARFIL JSR INCHR
0383  8231  C9 2C               CMP #',         ;VALID DELIMITERS - ,
0384  8233  F0 04               BEQ M21
```

```
LINE # LOC     CODE        LINE

0385  8235  C9 2D                CMP #'-
0386  8237  D0 11                BNE M22
0387  8239  A2 FF       M21      LDX #$FF
0388  823B  8E 33 A6             STX SCR3
0389  823E  EE 49 A6             INC PARNR
0390  8241  AE 49 A6             LDX PARNR
0391  8244  E0 03                CPX #$03
0392  8246  D0 E3                BNE PM1
0393  8248  F0 1D                BEQ M24
0394  824A  20 75 82    M22      JSR ASCNIB
0395  824D  B0 18                BCS M24
0396  824F  A2 04                LDX #4
0397  8251  0E 4A A6    M23      ASL P3L
0398  8254  2E 4B A6             ROL P3H
0399  8257  CA                   DEX
0400  8258  D0 F7                BNE M23
0401  825A  0D 4A A6             ORA P3L
0402  825D  8D 4A A6             STA P3L
0403  8260  A9 FF                LDA #$FF
0404  8262  8D 33 A6             STA SCR3
0405  8265  D0 C7                BNE PARFIL
0406  8267  2C 33 A6    M24      BIT SCR3
0407  826A  F0 03                BEQ M25
0408  826C  EE 49 A6             INC PARNR
0409  826F  C9 0D       M25      CMP #$0D
0410  8271  18                   CLC
0411  8272  4C B8 81             JMP RESXAF
0412  8275  C9 0D       ASCNIB   CMP #$0D        ;C/R?
0413  8277  F0 19                BEQ M29
0414  8279  C9 30                CMP #'0
0415  827B  90 0C                BCC M26
0416  827D  C9 47                CMP #'G
0417  827F  B0 08                BCS M26
0418  8281  C9 41                CMP #'A
0419  8283  B0 08                BCS M27
0420  8285  C9 3A                CMP #':
0421  8287  90 06                BCC M28
0422  8289  C9 30       M26      CMP #'0
0423  828B  38                   SEC            ;CARRY SET - NON HEX
0424  828C  60                   RTS
0425  828D  E9 37       M27      SBC #$37
0426  828F  29 0F       M28      AND #$0F
0427  8291  18                   CLC
0428  8292  60          M29      RTS
0429  8293  EE 4A A6    INCP3    INC P3L         ;INCREMENT P3 (16 BITS)
0430  8296  D0 03                BNE *+5
0431  8298  EE 4B A6             INC P3H
0432  829B  60                   RTS
0433  829C  AE 4D A6    P2SCR    LDX P2H         ;MOVE P2 TO FE,FF
0434  829F  86 FF                STX $FF
0435  82A1  AE 4C A6             LDX P2L
0436  82A4  86 FE                STX $FE
0437  82A6  60                   RTS
0438  82A7  AE 4B A6    P3SCR    LDX P3H         ;MOVE P3 TO FE,FF
0439  82AA  86 FF                STX $FF
```

```
LINE # LOC      CODE        LINE

0440   82AC   AE 4A A6             LDX P3L
0441   82AF   86 FE                STX $FE
0442   82B1   60                   RTS
0443   82B2   E6 FE        INCCMP  INC $FE          ;INCREM FE,FF, COMPARE TO P3
0444   82B4   D0 14                BNE COMPAR
0445   82B6   E6 FF                INC $FF
0446   82B8   D0 10        WRAP    BNE COMPAR       ;TEST FOR WRAP AROUND
0447   82BA   2C BD 82             BIT EXWRAP
0448   82BD   60           EXWRAP  RTS
0449   82BE   A5 FE        DECCMP  LDA $FE          ;DECREM FE,FF AND COMPARE TO P3
0450   82C0   D0 06                BNE M32
0451   82C2   A5 FF                LDA $FF
0452   82C4   F0 F2                BEQ WRAP
0453   82C6   C6 FF                DEC $FF
0454   82C8   C6 FE        M32     DEC $FE
0455   82CA   20 88 81     COMPAR  JSR SAVER        ;COMPARE FE,FF TO P3
0456   82CD   A5 FF                LDA $FF
0457   82CF   CD 4B A6             CMP P3H
0458   82D2   D0 05                BNE EXITCP
0459   82D4   A5 FE                LDA $FE
0460   82D6   CD 4A A6             CMP P3L
0461   82D9   B8           EXITCP  CLV
0462   82DA   4C BE 81             JMP RESXF
0463   82DD   08           CHKSAD  PHP              ;16 BIT CKSUM IN SCR6,7
0464   82DE   48                   PHA
0465   82DF   18                   CLC
0466   82E0   6D 36 A6             ADC SCR6
0467   82E3   8D 36 A6             STA SCR6
0468   82E6   90 03                BCC M33
0469   82E8   EE 37 A6             INC SCR7
0470   82EB   68           M33     PLA
0471   82EC   28                   PLP
0472   82ED   60                   RTS
0473   82EE   AD 59 A6     OUTPC   LDA PCLR         ;OUTPUT PC
0474   82F1   AE 5A A6             LDX PCHR
0475   82F4   48           OUTXAH  PHA
0476   82F5   8A                   TXA
0477   82F6   20 FA 82             JSR OUTBYT
0478   82F9   68                   PLA
0479   82FA   48           OUTBYT  PHA              ;OUTPUT 2 HEX DIGS FROM A
0480   82FB   48                   PHA
0481   82FC   4A                   LSR A
0482   82FD   4A                   LSR A
0483   82FE   4A                   LSR A
0484   82FF   4A                   LSR A
0485   8300   20 44 8A             JSR NBASOC
0486   8303   68                   PLA
0487   8304   20 44 8A             JSR NBASOC
0488   8307   68                   PLA
0489   8308   60                   RTS
0490   8309   29 0F        NIBASC  AND #$0F         ;NIBBLE IN A TO ASCII IN A
0491   830B   C9 0A                CMP #$0A      ;LINE FEED
0492   830D   B0 04                BCS NIBALF
0493   830F   69 30                ADC #$30
0494   8311   90 02                BCC EXITNB
```

```
LINE # LOC     CODE        LINE

0495   8313   69 36       NIBALF ADC #$36
0496   8315   60          EXITNB RTS
0497   8316   20 4D 83    CRLFSZ JSR CRLF          ;PRINT CRLF, FF, FE
0498   8319   A6 FF       OUTSZ  LDX $FF
0499   831B   A5 FE              LDA $FE
0500   831D   4C F4 82           JMP OUTXAH        [illegible] X,A
0501   8320   A9 3F       OUTQM  LDA #'?
0502   8322   4C 47 8A           JMP OUTCHR
0503   8325   20 3A 83    OCMCK  JSR COMMA         ;OUT COMMA, CKSUM LO
0504   8328   AD 36 A6           LDA SCR6
0505   832B   4C FA 82           JMP OUTBYT
0506   832E   A9 00       ZERCK  LDA #0            ;INIT CHECKSUM
0507   8330   8D 36 A6           STA SCR6
0508   8333   8D 37 A6           STA SCR7
0509   8336   60                 RTS
0510   8337   20 EE 82    OPCCOM JSR OUTPC         ;PC OUT, COMMA OUT
0511   833A   48          COMMA  PHA               ;COMMA OUT
0512   833B   A9 2C              LDA #',
0513   833D   D0 06              BNE SPCP3
0514   833F   20 42 83    SPC2   JSR SPACE         ;2 SPACES OUT
0515   8342   48          SPACE  PHA               ;1 SPACE OUT
0516   8343   A9 20              LDA #$20      ;SPACE
0517   8345   20 47 8A    SPCP3  JSR OUTCHR
0518   8348   68                 PLA
0519   8349   60                 RTS
0520   834A   20 FA 82    OBCRLF JSR OUTBYT        ;BYTE OUT, CRLF OUT
0521   834D   48          CRLF   PHA
0522   834E   A9 0D              LDA #$0D
0523   8350   20 47 8A           JSR OUTCHR
0524   8353   A9 0A              LDA #$0A      ;LINE FEED
0525   8355   20 47 8A           JSR OUTCHR
0526   8358   68                 PLA
0527   8359   60                 RTS
0528   835A   AE 56 A6    DELAY  LDX TV            ;DELAY DEPENDS ON TV
0529   835D   20 88 81    DL1    JSR SAVER
0530   8360   A9 FF              LDA #$FF
0531   8362   8D 39 A6           STA SCR9
0532   8365   8D 38 A6           STA SCR8
0533   8368   0E 38 A6    DLY1   ASL SCR8          ;(SCR9,8)=FFFF-2**X
0534   836B   2E 39 A6           ROL SCR9
0535   836E   CA                 DEX
0536   836F   D0 F7              BNE DLY1
0537   8371   20 03 89    DLY2   JSR IJSCNV        ;SCAN DISPLAY
0538   8374   20 86 83           JSR INSTAT        ;SEE IF KEY DOWN
0539   8377   B0 0A              BCS DLY0
0540   8379   EE 38 A6           INC SCR8          ;SCAN 2**X+1 TIMES
0541   837C   D0 03              BNE *+5
0542   837E   EE 39 A6           INC SCR9
0543   8381   D0 EE              BNE DLY2
0544   8383   4C BE 81    DLY0   JMP RESXF
0545   8386               ; INSTAT - SEE IF KEY DOWN, RESULT IN CARRY
0546   8386               ; KYSTAT, TSTAT RETURN IMMEDIATELY W/STATUS
0547   8386               ; INSTAT WAITS FOR RELEASE
0548   8386   20 92 83    INSTAT JSR INJISV
0549   8389   90 06              BCC INST2
```

```
LINE # LOC     CODE       LINE

0550   838B  20 92 83   INST1   JSR INJISV
0551   838E  B0 FB              BCS INST1
0552   8390  38                 SEC
0553   8391  60         INST2   RTS
0554   8392  6C 67 A6   INJISV JMP (INSVEC+1)
0555   8395             ;
0556   8395             ;
0557   8395             ; *** EXECUTE BLOCKS BEGIN HERE
0558   8395             ;
0559   8395             BZPARM=*
0560   8395             ; ZERO PARM COMMANDS
0561   8395             ;
0562   8395  C9 52      REGZ    CMP #'R          ;DISP REGISTERS
0563   8397  D0 5A              BNE GOZ          ;PC,S,F,A,X,Y
0564   8399  20 4D 83   RGBACK JSR CRLF
0565   839C  A9 50              LDA #'P
0566   839E  20 47 8A           JSR OUTCHR
0567   83A1  20 42 83           JSR SPACE
0568   83A4  20 EE 82           JSR OUTPC
0569   83A7  20 D6 81           JSR COMINB
0570   83AA  B0 13              BCS NH3
0571   83AC  8D 34 A6           STA SCR4
0572   83AF  20 D9 81           JSR INBYTE
0573   83B2  B0 0B              BCS NH3
0574   83B4  8D 59 A6           STA PCLR
0575   83B7  AD 34 A6           LDA SCR4
0576   83BA  8D 5A A6           STA PCHR
0577   83BD  90 09              BCC M34
0578   83BF  D0 02      NH3     BNE NOTCR
0579   83C1  18         EXITRG CLC
0580   83C2  60         EXRGP1 RTS
0581   83C3  20 CB 81   NOTCR   JSR ADVCK
0582   83C6  D0 FA              BNE EXRGP1
0583   83C8  A0 00      M34     LDY #0
0584   83CA  C8         M35     INY
0585   83CB  C0 06              CPY #6
0586   83CD  F0 CA              BEQ RGBACK
0587   83CF  20 4D 83           JSR CRLF
0588   83D2  B9 99 8F   NXTRG   LDA RGNAM-1,Y    ;GET REG NAME
0589   83D5             ; OUTPUT 3 SPACES TO LINE UP DISPLAY
0590   83D5  20 47 8A           JSR OUTCHR
0591   83D8  20 42 83           JSR SPACE
0592   83DB  20 3F 83           JSR SPC2
0593   83DE  B9 5A A6           LDA PCHR,Y
0594   83E1  20 D3 81           JSR OBCMIN
0595   83E4  B0 05              BCS M36
0596   83E6  99 5A A6           STA PCHR,Y
0597   83E9  90 DF              BCC M35
0598   83EB  F0 D4      M36     BEQ EXITRG
0599   83ED  20 CB 81           JSR ADVCK
0600   83F0  F0 D8              BEQ M35
0601   83F2  60                 RTS
0602   83F3  C9 47      GOZ     CMP #'G
0603   83F5  D0 20              BNE LPZB
0604   83F7  20 4D 83   GO2     JSR CRLF
```

```
LINE # LOC     CODE       LINE

0605   83FA   20 9C 8B   GO1ENT JSR NACCES       ;WRITE PROT MONITOR RAM
0606   83FD   AE 5B A6          LDX SR           ;RESTORE REGS
0607   8400   9A                TXS
0608   8401   AD 5A A6          LDA PCHR
0609   8404   48                PHA
0610   8405   AD 59 A6          LDA PCLR
0611   8408   48         NR10   PHA
0612   8409   AD 5C A6          LDA FR
0613   840C   48                PHA
0614   840D   AC 5F A6          LDY YR
0615   8410   AE 5E A6          LDX XR
0616   8413   AD 5D A6          LDA AR
0617   8416   40                RTI
0618   8417   C9 11      LPZB   CMP #$11         ;LOAD PAPER TAPE
0619   8419   F0 03             BEQ *+5
0620   841B   4C A7 84          JMP DEPZ
0621   841E   20 88 81          JSR SAVER
0622   8421   20 4D 83          JSR CRLF
0623   8424   A9 00             LDA #0
0624   8426   8D 52 A6          STA ERCNT
0625   8429   20 2E 83   LPZ    JSR ZERCK
0626   842C   20 1B 8A   LP1    JSR INCHR
0627   842F   C9 3B             CMP #$3B         ;SEMI COLON
0628   8431   D0 F9             BNE LP1
0629   8433   20 A1 84          JSR LDBYTE
0630   8436   B0 56             BCS TAPERR
0631   8438   D0 09             BNE NUREC
0632   843A   AD 52 A6          LDA ERCNT        ;ERRORS ?
0633   843D   F0 01             BEQ *+3
0634   843F   38         EXITLP SEC
0635   8440   4C B8 81          JMP RESXAF
0636   8443   8D 3D A6   NUREC  STA RC
0637   8446   20 A1 84          JSR LDBYTE
0638   8449   B0 43             BCS TAPERR
0639   844B   85 FF             STA $FF
0640   844D   20 A1 84          JSR LDBYTE
0641   8450   B0 D7             BCS LPZ
0642   8452   85 FE             STA $FE
0643   8454   20 A1 84   MORED  JSR LDBYTE
0644   8457   B0 35             BCS TAPERR
0645   8459   A0 00             LDY #0
0646   845B   91 FE             STA ($FE),Y
0647   845D   D1 FE             CMP ($FE),Y
0648   845F   F0 0C             BEQ LPGD
0649   8461   AD 52 A6          LDA ERCNT
0650   8464   29 0F             AND #$0F
0651   8466   C9 0F             CMP #$0F
0652   8468   F0 03             BEQ *+5
0653   846A   EE 52 A6          INC ERCNT
0654   846D   20 B2 82   LPGD   JSR INCCMP
0655   8470   CE 3D A6          DEC RC
0656   8473   D0 DF             BNE MORED
0657   8475   20 D9 81          JSR INBYTE
0658   8478   B0 14             BCS TAPERR
0659   847A   CD 37 A6          CMP SCR7
```

```
LOC     CODE        LINE

847D  D0 0C               BNE BADDY
847F  20 D9 81            JSR INBYTE
8482  B0 0A               BCS TAPERR
8484  CD 36 A6            CMP SCR6
8487  F0 A0               BEQ LPZ
8489  D0 03               BNE TAPERR      ;(ALWAYS)
848B  20 D9 81     BADDY  JSR INBYTE
848E  AD 52 A6     TAPERR LDA ERCNT
8491  29 F0               AND #$F0
8493  C9 F0               CMP #$F0
8495  F0 92               BEQ LPZ
8497  AD 52 A6            LDA ERCNT
849A  69 10               ADC #$10
849C  8D 52 A6            STA ERCNT
849F  D0 88               BNE LPZ
84A1  20 D9 81     LDBYTE JSR INBYTE
84A4  4C DD 82            JMP CHKSAD
84A7  C9 44        DEPZ   CMP #'D         ;DEPOSIT, 0 PARM - USE (OLD)
84A9  D0 03               BNE MEMZ
84AB  4C E1 84            JMP NEWLN
84AE  C9 4D        MEMZ   CMP #'M         ;MEM, 0 PARM - USE (OLD)
84B0  D0 03               BNE VERZ
84B2  4C 17 85            JMP NEWLOC
84B5  C9 56        VERZ   CMP #'V         ;VERIFY, 0 PARM - USE (OLD)
84B7  D0 0D               BNE L1ZB        ; ... DO 8 BYTES (LIKE VER 1 PARM)
84B9  A5 FE               LDA $FE
84BB  8D 4A A6            STA P3L
84BE  A5 FF               LDA $FF
84C0  8D 4B A6            STA P3H
84C3  4C 9A 85            JMP VER1+4
84C6  C9 12        L1ZB   CMP #$12        ;LOAD KIM, ZERO PARM
84C8  D0 05               BNE L2ZB
84CA  A0 00               LDY #0          ;MODE = KIM
84CC  4C 78 8C     L1J    JMP LENTRY      ;GO TO CASSETTE ROUTINE
84CF  C9 13        L2ZB   CMP #$13        ;LOAD HS, ZERO PARM
84D1  D0 04               BNE EZPARM
84D3  A0 80               LDY #$80        ;MODE = HS
84D5  D0 F5               BNE L1J         ;(ALWAYS)
84D7  6C 6D A6     EZPARM JMP (URCVEC+1)  ;ELSE UNREC COMMAND
84DA               B1PARM=*
84DA               ;
84DA               ; 1 PARAMETER COMMAND EXEC BLOCKS
84DA               ;
84DA  C9 44        DEP1   CMP #'D         ;DEPOSIT, 1 PARM
84DC  D0 32               BNE MEM1
84DE  20 A7 82            JSR P3SCR
84E1  20 16 83     NEWLN  JSR CRLFSZ
84E4  A0 00               LDY #0
84E6  A2 08               LDX #8
84E8  20 42 83     DEPBYT JSR SPACE
84EB  20 D9 81            JSR INBYTE
84EE  B0 11               BCS NH41
84F0  91 FE               STA ($FE),Y
84F2  D1 FE               CMP ($FE),Y     ;VERIFY
84F4  F0 03               BEQ DEPN
```

```
LINE # LOC     CODE        LINE

0715   84F6   20 20 83             JSR OUTQM         ;TYPE "?" IF NG
0716   84F9   20 B2 82    DEPN     JSR INCCMP
0717   84FC   CA                   DEX
0718   84FD   D0 E9                BNE DEPBYT
0719   84FF   F0 E0                BEQ NEWLN
0720   8501   F0 0B       NH41     BEQ DEPEC
0721   8503   C9 20                CMP #$20          ;SPACE = FWD
0722   8505   D0 4C                BNE DEPES
0723   8507   70 F0                BVS DEPN
0724   8509   20 42 83             JSR SPACE
0725   850C   10 EB                BPL DEPN
0726   850E   18          DEPEC    CLC
0727   850F   60                   RTS
0728   8510   C9 4D       MEM1     CMP #'M           ;MEMORY, 1 PARM
0729   8512   D0 65                BNE GO1
0730   8514   20 A7 82             JSR P3SCR
0731   8517   20 16 83    NEWLOC   JSR CRLFSZ
0732   851A   20 3A 83             JSR COMMA
0733   851D   A0 00                LDY #0
0734   851F   B1 FE                LDA ($FE),Y
0735   8521   20 D3 81             JSR OBCMIN
0736   8524   B0 11                BCS NH42
0737   8526   A0 00                LDY #0
0738   8528   91 FE                STA ($FE),Y
0739   852A   D1 FE                CMP ($FE),Y       ;VERIFY MEM
0740   852C   F0 03                BEQ NXTLOC
0741   852E   20 20 83             JSR OUTQM         ;TYPE ? AND CONTINUE
0742   8531   20 B2 82    NXTLOC   JSR INCCMP
0743   8534   18                   CLC
0744   8535   90 E0                BCC NEWLOC
0745   8537   F0 3E       NH42     BEQ EXITM1
0746   8539   50 04                BVC *+6
0747   853B   C9 3C                CMP #'<
0748   853D   F0 D8                BEQ NEWLOC
0749   853F   C9 20                CMP #$20        ;SPACE ?
0750   8541   F0 EE                BEQ NXTLOC
0751   8543   C9 3E                CMP #'>
0752   8545   F0 EA                BEQ NXTLOC
0753   8547   C9 2B                CMP #'+
0754   8549   F0 10                BEQ LOCP8
0755   854B   C9 3C                CMP #'<
0756   854D   F0 06                BEQ PRVLOC
0757   854F   C9 2D                CMP #'-
0758   8551   F0 16                BEQ LOCM8
0759   8553   38          DEPES    SEC
0760   8554   60                   RTS
0761   8555   20 BE 82    PRVLOC   JSR DECCMP        ;BACK ONE BYT
0762   8558   18                   CLC
0763   8559   90 BC                BCC NEWLOC
0764   855B   A5 FE       LOCP8    LDA $FE           ;GO FWD 8 BYTES
0765   855D   18                   CLC
0766   855E   69 08                ADC #$08
0767   8560   85 FE                STA $FE
0768   8562   90 02                BCC M42
0769   8564   E6 FF                INC $FF
```

```
LINE # LOC     CODE        LINE

0770   8566   18           M42    CLC
0771   8567   90 AE               BCC NEWLOC
0772   8569   A5 FE        LOCM8  LDA $FE            ; GO BACKWD 8 BYTES
0773   856B   38                  SEC
0774   856C   E9 08               SBC #$08
0775   856E   85 FE               STA $FE
0776   8570   B0 02               BCS M43
0777   8572   C6 FF               DEC $FF
0778   8574   18           M43    CLC
0779   8575   90 A0               BCC NEWLOC
0780   8577   18           EXITM1 CLC
0781   8578   60                  RTS
0782   8579   C9 47        GO1    CMP #'G            ; GO, 1 PARM (RTRN ADDR ON STK)
0783   857B   D0 19               BNE VER1        ; ...  PARM IS ADDR TO GO TO
0784   857D   20 4D 83            JSR CRLF
0785   8580   20 9C 8B            JSR NACCES         ;WRITE PROT MONITR RAM
0786   8583   A2 FF               LDX #$FF           ;PUSH RETURN ADDR
0787   8585   9A                  TXS
0788   8586   A9 7F               LDA #$7F
0789   8588   48                  PHA
0790   8589   A9 FF               LDA #$FF
0791   858B   48                  PHA
0792   858C   AD 4B A6            LDA P3H
0793   858F   48                  PHA
0794   8590   AD 4A A6            LDA P3L
0795   8593   4C 08 84            JMP NR10
0796   8596   C9 56        VER1   CMP #'V            ;VERIFY, 1 PARM (8 BYTES, CKSUM)
0797   8598   D0 1A               BNE JUMP1
0798   859A   AD 4A A6            LDA P3L
0799   859D   8D 4C A6            STA P2L
0800   85A0   18                  CLC
0801   85A1   69 07               ADC #$07
0802   85A3   8D 4A A6            STA P3L
0803   85A6   AD 4B A6            LDA P3H
0804   85A9   8D 4D A6            STA P2H
0805   85AC   69 00               ADC #0
0806   85AE   8D 4B A6            STA P3H
0807   85B1   4C 40 86            JMP VER2+4
0808   85B4   C9 4A        JUMP1  CMP #'J            ;JUMP (JUMP TABLE IN SYS RAM)
0809   85B6   D0 1F               BNE L11B
0810   85B8   AD 4A A6            LDA P3L
0811   85BB   C9 08               CMP #8             ; 0-7 ONLY VALID
0812   85BD   B0 26               BCS JUM2
0813   85BF   20 9C 8B            JSR NACCES         ;WRITE PROT SYS RAM
0814   85C2   0A                  ASL A
0815   85C3   A8                  TAY
0816   85C4   A2 FF               LDX #$FF           ;INIT STK PTR
0817   85C6   9A                  TXS
0818   85C7   A9 7F               LDA #$7F           ;PUSH COLD RETURN
0819   85C9   48                  PHA
0820   85CA   A9 FF               LDA #$FF
0821   85CC   48                  PHA
0822   85CD   B9 21 A6            LDA JTABLE+1,Y     ;GET ADDR FROM TABLE
0823   85D0   48                  PHA                ;PUSH ON STACK
0824   85D1   B9 20 A6            LDA JTABLE,Y
```

```
LINE # LOC     CODE        LINE

0825   85D4  4C 08 84              JMP NR10         ;LOAD UP USER REG'S AND RTI
0826   85D7  C9 12        L11B     CMP #$12         ;LOAD KIM FMT, 1 PARM
0827   85D9  D0 14                 BNE L21B
0828   85DB  A0 00                 LDY #0           ;MODE = KIM
0829   85DD  AD 4A A6     L11C     LDA P3L
0830   85E0  C9 FF                 CMP #$FF         ; ID MUST NOT BE FF
0831   85E2  D0 02                 BNE *+4
0832   85E4  38                    SEC
0833   85E5  60           JUM2     RTS
0834   85E6  20 08 82              JSR PSHOVE       ;FIX PARM POSITION
0835   85E9  20 08 82     L11D     JSR PSHOVE
0836   85EC  4C 78 8C              JMP LENTRY
0837   85EF  C9 13        L21B     CMP #$13         ;LOAD TAPE, HS FMT, 1 PARM
0838   85F1  D0 04                 BNE WPR1B
0839   85F3  A0 80                 LDY #$80         ;MODE = HS
0840   85F5  D0 E6                 BNE L11C
0841   85F7  C9 57        WPR1B    CMP #'W          ;WRITE PROT USER RAM
0842   85F9  D0 1B                 BNE E1PARM
0843   85FB  AD 4A A6              LDA P3L          ; FIRST DIG IS 1K ABOVE 0,
0844   85FE  29 11                 AND #$11         ; SECOND IS 2K ABOVE 0
0845   8600  C9 08                 CMP #8           ; THIRD IS 3K ABOVE 0.
0846   8602  2A                    ROL A
0847   8603  4E 4B A6              LSR P3H
0848   8606  2A                    ROL A
0849   8607  0A                    ASL A
0850   8608  29 0F                 AND #$0F
0851   860A  49 0F                 EOR #$0F         ;0 IS PROTECT
0852   860C  8D 01 AC              STA OR3A
0853   860F  A9 0F                 LDA #$0F
0854   8611  8D 03 AC              STA DDR3A
0855   8614  18                    CLC
0856   8615  60                    RTS
0857   8616  4C 27 88     E1PARM JMP CALC3
0858   8619               B2PARM=*
0859   8619               ;
0860   8619               ; 2 PARAMETER EXEC BLOCKS
0861   8619               ;
0862   8619  C9 10        STD2     CMP #$10         ;STORE DOUBLE BYTE
0863   861B  D0 12                 BNE MEM2
0864   861D  20 A7 82              JSR P3SCR
0865   8620  AD 4D A6              LDA P2H
0866   8623  A0 01                 LDY #1
0867   8625  91 FE                 STA ($FE),Y
0868   8627  88                    DEY
0869   8628  AD 4C A6              LDA P2L
0870   862B  91 FE                 STA ($FE),Y
0871   862D  18                    CLC
0872   862E  60                    RTS
0873   862F  C9 4D        MEM2     CMP #'M          ;CONTINUE MEM SEARCH W/OLD PTR
0874   8631  D0 09                 BNE VER2
0875   8633  AD 4C A6              LDA P2L
0876   8636  8D 4E A6              STA P1L
0877   8639  4C 08 88              JMP MEM3C
0878   863C  C9 56        VER2     CMP #'V          ;VERIFY MEM W/CHKSUMS , 2 PARM
0879   863E  D0 48                 BNE L12B
```

```
LINE # LOC     CODE        LINE

0880   8640   20 9C 82          JSR P2SCR
0881   8643   20 2E 83          JSR ZERCK
0882   8646   20 16 83   VADDR  JSR CRLFSZ
0883   8649   A2 08             LDX #8
0884   864B   20 42 83   V2     JSR SPACE
0885   864E   A0 00             LDY #0
0886   8650   B1 FE             LDA ($FE),Y
0887   8652   20 DD 82          JSR CHKSAD
0888   8655   20 FA 82          JSR OUTBYT
0889   8658   20 B2 82          JSR INCCMP
0890   865B   70 11             BVS V1
0891   865D   F0 02             BEQ *+4
0892   865F   B0 0D             BCS V1
0893   8661   CA                DEX
0894   8662   D0 E7             BNE V2
0895   8664   20 25 83   VOCK   JSR OCMCK
0896   8667   20 86 83          JSR INSTAT
0897   866A   90 DA             BCC VADDR
0898   866C   18                CLC
0899   866D   60                RTS
0900   866E   20 BE 82   V1     JSR DECCMP
0901   8671   E0 08             CPX #8
0902   8673   F0 03             BEQ *+5
0903   8675   E8                INX
0904   8676   10 F6             BPL V1
0905   8678   20 25 83          JSR OCMCK
0906   867B   20 4D 83          JSR CRLF
0907   867E   20 42 83          JSR SPACE
0908   8681   AE 37 A6          LDX SCR7
0909   8684   20 F4 82          JSR OUTXAH
0910   8687   60                RTS
0911   8688   C9 12      L12B   CMP #$12         ;LOAD KIM FMT TAPE, 2 PARMS
0912   868A   D0 0C             BNE SP2B
0913   868C   AD 4C A6   L12C   LDA P2L
0914   868F   C9 FF             CMP #$FF         ;ID MUST BE FF
0915   8691   D0 F4             BNE L12B-1       ;ERR
0916   8693   A0 00             LDY #0           ;MODE = HS
0917   8695   4C E9 85          JMP L11D
0918   8698   C9 1C      SP2B   CMP #$1C         ;SAVE PAPER TAPE, 2 PARMS
0919   869A   D0 75             BNE E2PARM
0920   869C   18                CLC
0921   869D   20 88 81          JSR SAVER
0922   86A0   20 9C 82          JSR P2SCR
0923   86A3   20 FA 86   SP2C   JSR DIFFZ
0924   86A6   B0 03             BCS SP2D
0925   86A8   4C C4 81   SPEXIT JMP RESALL
0926   86AB   20 4D 83   SP2D   JSR CRLF
0927   86AE   CD 58 A6          CMP MAXRC
0928   86B1   90 05             BCC SP2E
0929   86B3   AD 58 A6          LDA MAXRC
0930   86B6   B0 02             BCS SP2F
0931   86B8   69 01      SP2E   ADC #1
0932   86BA   8D 3D A6   SP2F   STA RC
0933   86BD   A9 3B             LDA #$3B         ;SEMI COLON
0934   86BF   20 47 8A          JSR OUTCHR
```

```
LINE # LOC     CODE        LINE

0935   86C2   AD 3D A6            LDA RC
0936   86C5   20 F4 86            JSR SVBYTE
0937   86C8   A5 FF               LDA $FF
0938   86CA   20 F4 86            JSR SVBYTE
0939   86CD   A5 FE               LDA $FE
0940   86CF   20 F4 86            JSR SVBYTE
0941   86D2   A0 00        MORED2 LDY #$00
0942   86D4   B1 FE               LDA ($FE),Y
0943   86D6   20 F4 86            JSR SVBYTE
0944   86D9   20 86 83            JSR INSTAT      ;STOP IF KEY DEPRESSED
0945   86DC   B0 CA               BCS SPEXIT
0946   86DE   20 B2 82            JSR INCCMP
0947   86E1   70 C5               BVS SPEXIT
0948   86E3   CE 3D A6            DEC RC
0949   86E6   D0 EA               BNE MORED2
0950   86E8   AE 37 A6            LDX SCR7
0951   86EB   AD 36 A6            LDA SCR6
0952   86EE   20 F4 82            JSR OUTXAH
0953   86F1   18                  CLC
0954   86F2   90 AF               BCC SP2C
0955   86F4   20 DD 82     SVBYTE JSR CHKSAD
0956   86F7   4C FA 82            JMP OUTBYT
0957   86FA   20 2E 83     DIFFZ  JSR ZERCK
0958   86FD   AD 4A A6     DIFFL  LDA P3L
0959   8700   38                  SEC
0960   8701   E5 FE               SBC $FE
0961   8703   48                  PHA
0962   8704   AD 4B A6            LDA P3H
0963   8707   E5 FF               SBC $FF
0964   8709   F0 04               BEQ DIFF1
0965   870B   68                  PLA
0966   870C   A9 FF               LDA #$FF
0967   870E   60                  RTS
0968   870F   68           DIFF1  PLA
0969   8710   60           DIFFL2 RTS
0970   8711   4C 27 88     E2PARM JMP CALC3       ;MAY BE CALC OR EXEC
0971   8714                B3PARM=*
0972   8714                ;
0973   8714                ; 3 PARAMETER COMMAND EXECUTE BLOCKS
0974   8714                ;
0975   8714   C9 46        FILL3  CMP #'F         ;FILL MEM
0976   8716   D0 21               BNE BLK3
0977   8718   20 9C 82            JSR P2SCR
0978   871B   A9 00               LDA #0
0979   871D   8D 52 A6            STA ERCNT       ;ZERO ERROR COUNT
0980   8720   AD 4E A6            LDA P1L
0981   8723   A0 00        F1     LDY #0
0982   8725   91 FE               STA ($FE),Y
0983   8727   D1 FE               CMP ($FE),Y     ;VERIFY
0984   8729   F0 03               BEQ F3
0985   872B   20 C1 87            JSR BRTT        ;INC ERCNT (UP TO FF)
0986   872E   20 B2 82     F3     JSR INCCMP
0987   8731   70 7C               BVS B1
0988   8733   F0 EE               BEQ F1
0989   8735   90 EC               BCC F1
```

```
LINE # LOC      CODE        LINE

0990   8737  B0 76       F2     BCS B1            ;(ALWAYS)
0991   8739  C9 42       BLK3   CMP #'B           ;BLOCK MOVE (OVERLAP OK)
0992   873B  F0 03              BEQ *+5
0993   873D  4C CD 87           JMP S13B
0994   8740  A9 00              LDA #0
0995   8742  8D 52 A6           STA ERCNT
0996   8745  20 9C 82           JSR P2SCR
0997   8748  AD 4E A6           LDA P1L
0998   874B  85 FC              STA $FC
0999   874D  AD 4F A6           LDA P1H
1000   8750  85 FD              STA $FD
1001   8752  C5 FF              CMP $FF           ;WHICH DIRECTION TO MOVE?
1002   8754  D0 06              BNE *+8
1003   8756  A5 FC              LDA $FC
1004   8758  C5 FE              CMP $FE
1005   875A  F0 53              BEQ B1            ;16 BITS EQUAL THEN FINISHED
1006   875C  B0 14              BCS B2            ;MOVE DEC'NG
1007   875E  20 B7 87    BLP    JSR BMOVE         ;MOVE INC'NG
1008   8761  E6 FC              INC $FC
1009   8763  D0 02              BNE *+4
1010   8765  E6 FD              INC $FD
1011   8767  20 B2 82           JSR INCCMP
1012   876A  70 43              BVS B1
1013   876C  F0 F0              BEQ BLP
1014   876E  90 EE              BCC BLP
1015   8770  B0 3D              BCS B1
1016   8772  A5 FC       B2     LDA $FC           ;CALC VALS FOR MOVE DEC'NG
1017   8774  18                 CLC
1018   8775  6D 4A A6           ADC P3L
1019   8778  85 FC              STA $FC
1020   877A  A5 FD              LDA $FD
1021   877C  6D 4B A6           ADC P3H
1022   877F  85 FD              STA $FD
1023   8781  38                 SEC
1024   8782  A5 FC              LDA $FC
1025   8784  E5 FE              SBC $FE
1026   8786  85 FC              STA $FC
1027   8788  A5 FD              LDA $FD
1028   878A  E5 FF              SBC $FF
1029   878C  85 FD              STA $FD
1030   878E  20 A7 82           JSR P3SCR
1031   8791  AD 4C A6           LDA P2L
1032   8794  8D 4A A6           STA P3L
1033   8797  AD 4D A6           LDA P2H
1034   879A  8D 4B A6           STA P3H
1035   879D  20 B7 87    BLP1   JSR BMOVE         ;MOVE DEC'NG
1036   87A0  A5 FC              LDA $FC
1037   87A2  D0 02              BNE *+4
1038   87A4  C6 FD              DEC $FD
1039   87A6  C6 FC              DEC $FC
1040   87A8  20 BE 82           JSR DECCMP
1041   87AB  70 02              BVS B1
1042   87AD  B0 EE              BCS BLP1
1043   87AF  AD 52 A6    B1     LDA ERCNT         ;FINISHED, TEST ERCNT
1044   87B2  38                 SEC
```

```
LINE # LOC     CODE       LINE

1045   87B3   D0 01              BNE *+3
1046   87B5   18                 CLC
1047   87B6   60                 RTS
1048   87B7   A0 00      BMOVE   LDY #0          ;MOVE 1 BYT + VER
1049   87B9   B1 FE              LDA ($FE),Y
1050   87BB   91 FC              STA ($FC),Y
1051   87BD   D1 FC              CMP ($FC),Y
1052   87BF   F0 0B              BEQ BRT
1053   87C1   AC 52 A6   BRTT    LDY ERCNT       ;INC ERCNT, DONT PASS FF
1054   87C4   C0 FF              CPY #$FF
1055   87C6   F0 04              BEQ *+6
1056   87C8   C8                 INY
1057   87C9   8C 52 A6           STY ERCNT
1058   87CC   60         BRT     RTS
1059   87CD   C9 1D      S13B    CMP #$1D        ;SAVE KIM FMT TAPE, 3 PARMS
1060   87CF   D0 15              BNE S23B
1061   87D1   A0 00              LDY #$0         ;MODE = KIM
1062   87D3   AD 4E A6   S13C    LDA P1L
1063   87D6   D0 02              BNE *+4         ;ID MUST NOT = 0
1064   87D8   38                 SEC
1065   87D9   60                 RTS
1066   87DA   C9 FF              CMP #$FF        ;ID MUST NOT = FF
1067   87DC   D0 02              BNE *+4
1068   87DE   38         S1NG    SEC
1069   87DF   60                 RTS
1070   87E0   20 93 82           JSR INCP3       ;USE END ADDR + 1
1071   87E3   4C 87 8E           JMP SENTRY
1072   87E6   C9 1E      S23B    CMP #$1E        ;SAVE HS FMT TAPE, 3 PARMS
1073   87E8   D0 04              BNE L23P
1074   87EA   A0 80              LDY #$80        ;MODE = HS
1075   87EC   D0 E5              BNE S13C        ;(ALWAYS)
1076   87EE   C9 13      L23P    CMP #$13        ;LOAD HS, 3 PARMS
1077   87F0   D0 0F              BNE MEM3
1078   87F2   AD 4E A6           LDA P1L
1079   87F5   C9 FF              CMP #$FF        ;ID MUST BE FF
1080   87F7   D0 E5              BNE S1NG        ;ERR RETURN
1081   87F9   20 93 82           JSR INCP3       ;USE END ADDR + 1
1082   87FC   A0 80              LDY #$80        ;MODE = HS
1083   87FE   4C 78 8C           JMP LENTRY
1084   8801   C9 4D      MEM3    CMP #'M         ;MEM 3 SEARCH - BYTE
1085   8803   D0 22              BNE CALC3
1086   8805   20 9C 82           JSR P2SCR
1087   8808   AD 4E A6   MEM3C   LDA P1L
1088   880B   A0 00              LDY #0
1089   880D   D1 FE              CMP ($FE),Y
1090   880F   F0 0B              BEQ MEM3E       ;FOUND SEARCH BYTE?
1091   8811   20 B2 82   MEM3D   JSR INCCMP      ;NO, INC BUFFER ADDR
1092   8814   70 04              BVS MEM3EX      ;WRAP AROUND?
1093   8816   F0 F0              BEQ MEM3C
1094   8818   90 EE              BCC MEM3C
1095   881A   18         MEM3EX  CLC
1096   881B   60                 RTS             ;SEARCHED TO BOUND
1097   881C   20 17 85   MEM3E   JSR NEWLOC      ;FOUND SEARCH BYTE
1098   881F   90 05              BCC MEM3F
1099   8821   C9 47              CMP #'G         ;ENTERED G?
```

```
LINE # LOC     CODE        LINE

       8823   F0 EC                BEQ MEM3D
       8825   38                   SEC
       8826   60           MEM3F   RTS
       8827   C9 43        CALC3   CMP #'C CALCULATE, 1, 2 OR 3 PARMS
       8829   D0 26                BNE EXE3         ;RESULT = P1+P2-P3
       882B   20 4D 83     C1      JSR CRLF
       882E   20 42 83             JSR SPACE
       8831   18                   CLC
       8832   AD 4E A6             LDA P1L
       8835   6D 4C A6             ADC P2L
       8838   A8                   TAY
       8839   AD 4F A6             LDA P1H
       883C   6D 4D A6             ADC P2H
       883F   AA                   TAX
       8840   38                   SEC
       8841   98                   TYA
       8842   ED 4A A6             SBC P3L
       8845   A8                   TAY
       8846   8A                   TXA
       8847   ED 4B A6             SBC P3H
       884A   AA                   TAX
       884B   98                   TYA
       884C   20 F4 82             JSR OUTXAH
       884F   18                   CLC
       8850   60                   RTS
       8851   C9 45        EXE3    CMP #'E          ;EXECUTE FROM RAM, 1-3 PARMS
       8853   D0 57                BNE E3PARM
       8855                ; SEE IF VECTOR ALREADY MOVED
       8855   AD 62 A6             LDA INVEC+2      ;INVEC MOVED TO SCRA, SCRB
       8858                ; HI BYTE OF EXEVEC MUST BE DIFFERENT FROM INVEC
       8858   CD 73 A6             CMP EXEVEC+1     ;$FA, $FB USED AS RAM PTR
       885B   F0 15                BEQ PTRIN
       885D   8D 3B A6             STA SCRA+1       ;SAVE INVEC IN SCRA,B
       8860   AD 61 A6             LDA INVEC+1
       8863   8D 3A A6             STA SCRA
       8866   AD 72 A6             LDA EXEVEC       ;PUT ADDR OF RIN IN INVEC
       8869   8D 61 A6             STA INVEC+1
       886C   AD 73 A6             LDA EXEVEC+1
       886F   8D 62 A6             STA INVEC+2
       8872   AD 4B A6     PTRIN   LDA P3H          ;INIT RAM PTR IN $FA, $FB
       8875   85 FB                STA $FB
       8877   AD 4A A6             LDA P3L
       887A   85 FA                STA $FA
       887C   18                   CLC
       887D   60                   RTS
       887E   20 88 81     RIN     JSR SAVER        ;GET INPUT FROM RAM
       8881   A0 00                LDY #$0          ;RAM PTR IN $FA, $FB
       8883   B1 FA                LDA ($FA),Y
       8885   F0 12                BEQ RESTIV       ;IF 00 BYTE, RESTORE INVEC
       8887   E6 FA                INC $FA
       8889   D0 02                BNE *+4
       888B   E6 FB                INC $FB
       888D   2C 53 A6             BIT TECHO        ;ECHO CHARS IN ?
       8890   10 03                BPL *+5
       8892   20 47 8A             JSR OUTCHR
```

```
LINE # LOC      CODE        LINE

1155   8895   18                  CLC
1156   8896   4C B8 81            JMP RESXAF
1157   8899   AD 3A A6     RESTIV LDA SCRA         ;RESTORE INVEC
1158   889C   8D 61 A6            STA INVEC+1
1159   889F   AD 3B A6            LDA SCRA+1
1160   88A2   8D 62 A6            STA INVEC+2
1161   88A5   18                  CLC
1162   88A6   20 1B 8A            JSR INCHR
1163   88A9   4C B8 81            JMP RESXAF
1164   88AC   6C 6D A6     E3PARM JMP (URCVEC+1)   ;... ELSE UNREC CMD
1165   88AF                ; ***
1166   88AF                ; *** HEX KEYBOARD I/O
1167   88AF                ; ***
1168   88AF   20 88 81     GETKEY JSR SAVER        ;FIND KEY
1169   88B2   20 CF 88            JSR GK
1170   88B5   C9 FE               CMP #$FE
1171   88B7   D0 13               BNE EXITGK
1172   88B9   20 CF 88            JSR GK
1173   88BC   8A                  TXA
1174   88BD   0A                  ASL A
1175   88BE   0A                  ASL A
1176   88BF   0A                  ASL A
1177   88C0   0A                  ASL A
1178   88C1   8D 3E A6            STA SCRE
1179   88C4   20 CF 88            JSR GK
1180   88C7   8A                  TXA
1181   88C8   18                  CLC
1182   88C9   6D 3E A6            ADC SCRE
1183   88CC   4C B8 81     EXITGK JMP RESXAF
1184   88CF   A9 00        GK     LDA #0
1185   88D1   8D 55 A6            STA KSHFL
1186   88D4   20 03 89     GK1    JSR IJSCNV        ;SCAN KB
1187   88D7   F0 FB               BEQ GK1
1188   88D9   20 2C 89            JSR LRNKEY       ;WHAT KEY IS IT?
1189   88DC   F0 F6               BEQ GK1
1190   88DE   48                  PHA
1191   88DF   8A                  TXA
1192   88E0   48                  PHA
1193   88E1   20 72 89            JSR BEEP
1194   88E4   20 23 89     GK2    JSR KEYQ
1195   88E7   D0 FB               BNE GK2         ;Z=1 IF KEY DOWN
1196   88E9   20 9B 89            JSR NOBEEP       ;DELAY (DEBOUNCE) W/O BEEP
1197   88EC   20 23 89            JSR KEYQ
1198   88EF   D0 F3               BNE GK2
1199   88F1   68                  PLA
1200   88F2   AA                  TAX
1201   88F3   68                  PLA
1202   88F4   C9 FF               CMP #$FF        ;IF SHIFT, SET FLAG + GET NEXT KEY
1203   88F6   D0 07               BNE EXITG
1204   88F8   A9 19               LDA #$19
1205   88FA   8D 55 A6            STA KSHFL
1206   88FD   D0 D5               BNE GK1
1207   88FF   60           EXITG  RTS
1208   8900   20 C1 89     HDOUT  JSR OUTDSP        ;CHAR OUT, SCAN KB
1209   8903   6C 70 A6     IJSCNV JMP (SCNVEC+1)
```

```
LINE # LOC      CODE       LINE

       8906   A9 09       SCAND   LDA #$9          ;SCAN DISPLAY FROM DISBUF
       8908   20 A5 89            JSR CONFIG
       890B   A2 05               LDX #5
       890D   A0 00       SC1     LDY #0
       890F   BD 40 A6            LDA DISBUF,X
       8912   8C 00 A4            STY PADA
       8915   8E 02 A4            STX PBDA
       8918   8D 00 A4            STA PADA
       891B   A0 10               LDY #$10
       891D   88          SC2     DEY
       891E   D0 FD               BNE SC2
       8920   CA                  DEX
       8921   10 EA               BPL SC1
       8923   20 A3 89    KEYQ    JSR KSCONF       ; KEY DOWN ? (YES THEN Z=1)
       8926   AD 00 A4            LDA PADA
       8929   49 7F               EOR #$7F
       892B   60                  RTS
       892C   29 3F       LRNKEY  AND #$3F         ;DETERMINE WHAT KEY IS DOWN
       892E   8D 3F A6            STA SCRF
       8931   A9 05               LDA #$05
       8933   20 A5 89            JSR CONFIG
       8936   AD 02 A4            LDA PBDA
       8939   29 07               AND #$07
       893B   49 07               EOR #$07
       893D   D0 05               BNE LK1
       893F   2C 00 A4            BIT PADA
       8942   30 1A               BMI NOKEY
       8944   C9 04       LK1     CMP #$04
       8946   90 02               BCC LK2
       8948   A9 03               LDA #$03
       894A   0A          LK2     ASL A
       894B   0A                  ASL A
       894C   0A                  ASL A
       894D   0A                  ASL A
       894E   0A                  ASL A
       894F   0A                  ASL A
       8950   18                  CLC
       8951   6D 3F A6            ADC SCRF
       8954   A2 19               LDX #$19
       8956   DD D6 8B    LK3     CMP SYM,X
       8959   F0 05               BEQ FOUND
       895B   CA                  DEX
       895C   10 F8               BPL LK3
       895E   E8          NOKEY   INX
       895F   60                  RTS
       8960   8A          FOUND   TXA
       8961   18                  CLC
       8962   6D 55 A6            ADC KSHFL
       8965   AA                  TAX
       8966   BD EF 8B            LDA ASCII,X
       8969   60                  RTS
       896A   20 23 89    KYSTAT  JSR KEYQ         ;KEY DOWN? RETURN IN CARRY
       896D   18                  CLC
       896E   F0 01               BEQ *+3
       8970   38                  SEC
```

```
LINE # LOC     CODE        LINE

       8971   60                   RTS
       8972   20 88 81    BEEP     JSR SAVER        ;DELAY (BOUNCE) W/BEEP
       8975   A9 0D       BEEPP3   LDA #$0D
       8977   20 A5 89    BEEPP5   JSR CONFIG
       897A   A2 70                LDX #$70       ;DURATION CONSTANT
       897C   A9 08       BE1      LDA #8
       897E   8D 02 A4             STA PBDA
       8981   20 95 89             JSR BE2
       8984   A9 06                LDA #6
       8986   8D 02 A4             STA PBDA
       8989   20 95 89             JSR BE2
       898C   CA                   DEX
       898D   D0 ED                BNE BE1
       898F   20 A3 89             JSR KSCONF
       8992   4C C4 81             JMP RESALL
       8995   A0 1A       BE2      LDY #$1A
       8997   88          BE3      DEY
       8998   D0 FD                BNE BE3
       899A   60                   RTS
       899B   20 88 81    NOBEEP   JSR SAVER        ;DELAY W/O BEEP
       899E   A9 01                LDA #$01
       89A0   4C 77 89             JMP BEEPP5       ;(BNE BEEPP5, $FF)
       89A3   A9 01       KSCONF   LDA #$1          ;CONFIGURE FOR KEYBOARD
       89A5   20 88 81    CONFIG   JSR SAVER        ;CONFIGURE I/O FROM TABLE VAL
       89A8   A0 01                LDY #$01
       89AA   AA                   TAX
       89AB   BD C8 8B    CON1     LDA VALSP2,X
       89AE   99 02 A4             STA PBDA,Y
       89B1   BD C6 8B             LDA VALS,X
       89B4   99 00 A4             STA PADA,Y
       89B7   CA                   DEX
       89B8   88                   DEY
       89B9   10 F0                BPL CON1
       89BB   4C C4 81             JMP RESALL
       89BE   20 AF 88    HKEY     JSR GETKEY       ;GET KEY FROM KB AND ECHO ON KB
       89C1   20 88 81    OUTDSP   JSR SAVER        ;DISPLAY OUT
       89C4   29 7F                AND #$7F
       89C6   C9 07                CMP #$07       ;BELL?
       89C8   D0 03                BNE NBELL
       89CA   4C 75 89             JMP BEEPP3       ;YES - BEEP
       89CD   20 06 8A    NBELL    JSR TEXT         ;PUSH INTO SCOPE BUFFER
       89D0   C9 2C                CMP #$2C       ;COMMA?
       89D2   D0 0A                BNE OUD1
       89D4   AD 45 A6             LDA RDIG
       89D7   09 80                ORA #$80       ;TURN ON DECIMAL PT
       89D9   8D 45 A6             STA RDIG
       89DC   D0 25                BNE EXITOD
       89DE   A2 3A       OUD1     LDX #$3A
       89E0   DD EE 8B    OUD2     CMP ASCIM1,X
       89E3   F0 05                BEQ GETSGS
       89E5   CA                   DEX
       89E6   D0 F8                BNE OUD2
       89E8   F0 19                BEQ EXITOD
       89EA   BD 28 8C    GETSGS   LDA SEGSM1,X     ;GET CORR SEG CODE FROM TABLE
       89ED   C9 F0                CMP #$F0
```

```
LINE # LOC     CODE       LINE

1375  8A5D  85 F9               STA $F9
1376  8A5F  AD 02 A4    LOOK    LDA PBDA          ;FIND LEADING EDGE
1377  8A62  2D 54 A6            AND TOUTFL
1378  8A65  38                  SEC
1379  8A66  E9 40               SBC #$40
1380  8A68  90 F5               BCC LOOK
1381  8A6A  20 E9 8A    TIN     JSR DLYH          ;TERMINAL BIT
1382  8A6D  AD 02 A4            LDA PBDA
1383  8A70  2D 54 A6            AND TOUTFL
1384  8A73  38                  SEC
1385  8A74  E9 40               SBC #$40          ;OR BITS 6,7 (TTY,CRT)
1386  8A76  2C 53 A6            BIT TECHO         ;ECHO BIT?
1387  8A79  10 06               BPL DMY1
1388  8A7B  20 D4 8A            JSR OUT
1389  8A7E  4C 87 8A            JMP SAVE
1390  8A81  A0 07       DMY1    LDY #7
1391  8A83  88          TLP1    DEY
1392  8A84  D0 FD               BNE TLP1
1393  8A86  EA                  NOP
1394  8A87  66 F9       SAVE    ROR $F9
1395  8A89  20 E9 8A            JSR DLYH
1396  8A8C  48                  PHA               ;TIMING
1397  8A8D  B5 00               LDA 0,X
1398  8A8F  68                  PLA
1399  8A90  90 D8               BCC TIN
1400  8A92  20 E9 8A            JSR DLYH
1401  8A95  18                  CLC
1402  8A96  20 D4 8A            JSR OUT
1403  8A99  A5 F9               LDA $F9
1404  8A9B  49 FF               EOR #$FF
1405  8A9D  4C B8 81            JMP RESXAF
1406  8AA0  85 F9       TOUT    STA $F9           ;TERMINAL CHR OUT
1407  8AA2  20 88 81            JSR SAVER
1408  8AA5  20 E9 8A            JSR DLYH          ;DELAY 1/2 BIT TIME
1409  8AA8  A9 30               LDA #$30          ;SET FOR OUTPUT
1410  8AAA  8D 03 A4            STA PBDA+1        ;DATA DIRECTION
1411  8AAD  A5 F9               LDA $F9           ;RECOVER CHR DATA
1412  8AAF  A2 0B               LDX #$0B          ;START BIT,8DATA, 3STOPS
1413  8AB1  49 FF               EOR #$FF          ;INVERT DATA
1414  8AB3  38                  SEC               ;START BIT
1415  8AB4  20 D4 8A    OUTC    JSR OUT           ;OUTPUT BIT FROM CARRY
1416  8AB7  20 E6 8A            JSR DLYF          ;WAIT FULL BIT TIME
1417  8ABA  A0 06               LDY #$06
1418  8ABC  88          PHAKE   DEY
1419  8ABD  D0 FD               BNE PHAKE
1420  8ABF  EA                  NOP
1421  8AC0  4A                  LSR A
1422  8AC1  CA                  DEX
1423  8AC2  D0 F0               BNE OUTC
1424  8AC4  A5 F9               LDA $F9
1425  8AC6  C9 0D               CMP #$0D          ;CARRIAGE RETURN?
1426  8AC8  F0 04               BEQ GOPAD         ;YES-PAD IT
1427  8ACA  C9 0A               CMP #$0A          ;PAD LINE FEED TOO
1428  8ACC  D0 03               BNE LEAVE
1429  8ACE  20 32 8B    GOPAD   JSR PAD
```

```
LINE # LOC      CODE        LINE

1430   8AD1   4C C4 81    LEAVE   JMP RESALL
1431   8AD4   48          OUT     PHA                 ;TERMINAL BIT OUT
1432   8AD5   AD 02 A4            LDA PBDA
1433   8AD8   29 0F               AND #$0F
1434   8ADA   90 02               BCC OUTONE
1435   8ADC   09 30               ORA #$30
1436   8ADE   2D 54 A6    OUTONE  AND TOUTFL          ;MASK OUTPUT
1437   8AE1   8D 02 A4            STA PBDA
1438   8AE4   68                  PLA
1439   8AE5   60                  RTS
1440   8AE6               ;
1441   8AE6   20 E9 8A    DLYF    JSR DLYH            ;DELAY FULL
1442   8AE9   08          DLYH    PHP                 ;DELAY HALF
1443   8AEA   48                  PHA
1444   8AEB   8A                  TXA
1445   8AEC   48                  PHA
1446   8AED   98                  TYA
1447   8AEE   AE 51 A6            LDX SDBYT
1448   8AF1   A0 03       DLYX    LDY #3
1449   8AF3   88          DLYY    DEY
1450   8AF4   D0 FD               BNE DLYY
1451   8AF6   CA                  DEX
1452   8AF7   D0 F8               BNE DLYX
1453   8AF9   A8                  TAY
1454   8AFA   68                  PLA
1455   8AFB   AA                  TAX
1456   8AFC   68                  PLA
1457   8AFD   28                  PLP
1458   8AFE   60                  RTS
1459   8AFF   A9 00       BAUD    LDA #0              ;DETERMINE BAUD RATE ON PB7
1460   8B01   A8                  TAY
1461   8B02   AD 02 A4    SEEK    LDA PBDA
1462   8B05   0A                  ASL A
1463   8B06   B0 FA               BCS SEEK
1464   8B08   20 27 8B    CLEAR   JSR INK
1465   8B0B   90 FB               BCC CLEAR
1466   8B0D   20 27 8B    SET     JSR INK
1467   8B10   B0 FB               BCS SET
1468   8B12   8C 51 A6            STY SDBYT
1469   8B15   BD 63 8C    DEAF    LDA DECPTS,X
1470   8B18   CD 51 A6            CMP SDBYT
1471   8B1B   B0 07               BCS AGAIN
1472   8B1D   BD 69 8C            LDA STDVAL,X        ;LOAD CLOSEST STD VALUE
1473   8B20   8D 51 A6            STA SDBYT
1474   8B23   60                  RTS
1475   8B24   E8          AGAIN   INX
1476   8B25   10 EE               BPL DEAF
1477   8B27   C8          INK     INY
1478   8B28   A2 1C               LDX #$1C
1479   8B2A   CA          INK1    DEX
1480   8B2B   D0 FD               BNE INK1
1481   8B2D   AD 02 A4            LDA PBDA
1482   8B30   0A                  ASL A
1483   8B31   60                  RTS
1484   8B32   AE 50 A6    PAD     LDX PADBIT          ;PAD CARRIAGE RETURN OR LF
```

```
LINE # LOC     CODE        LINE

1485   8B35  20 E6 8A   PAD1   JSR DLYF         ;WITH EXTRA STOP BITS
1486   8B38  CA                DEX
1487   8B39  D0 FA             BNE PAD1
1488   8B3B  60                RTS
1489   8B3C  20 A3 89   TSTAT  JSR KSCONF       ;SEE IF BREAK KEY DOWN
1490   8B3F  AD 02 A4          LDA PBDA
1491   8B42  2D 54 A6          AND TOUTFL
1492   8B45  38                SEC
1493   8B46  E9 40             SBC #$40
1494   8B48  60                RTS
1495   8B49  FF                .BYT $FF         ;NOT USED
1496   8B4A             ; ***
1497   8B4A             ; *** RESET - TURN OFF POR, INIT SYS RAM, ENTER MONITOR
1498   8B4A             ; ***
1499   8B4A             ;
1500   8B4A  A2 FF      RESET  LDX #$FF
1501   8B4C  9A                TXS              ;INIT STACK PTR
1502   8B4D  A9 CC      POR    LDA #$CC
1503   8B4F  8D 0C A0          STA PCR1         ;DISABLE POR, TAPE OFF
1504   8B52  A9 04             LDA #4
1505   8B54  48                PHA
1506   8B55  28                PLP              ;INIT P, DISABLE IRQ DURING DFTXFR
1507   8B56  20 86 8B          JSR ACCESS       ;UN WRITE PROT SYS RAM
1508   8B59  A2 5F      DFTXFR LDX #$5F         ;INIT SYS RAM (EXCPT SCPBUF)
1509   8B5B  BD A0 8F          LDA DFTBLK,X
1510   8B5E  9D 20 A6          STA RAM,X
1511   8B61  CA                DEX
1512   8B62  10 F7             BPL DFTXFR+2
1513   8B64  A9 07      NEWDEV LDA #7           ;CHANGE DEVC/BAUD RATE
1514   8B66  20 47 8A          JSR OUTCHR       ;BEEP
1515   8B69  20 A3 89   SWITCH JSR KSCONF       ;KEYBOARD OR TERMINAL?
1516   8B6C  20 26 89   SWLP   JSR KEYQ+3
1517   8B6F  D0 0B             BNE MONENT
1518   8B71  2C 02 A4          BIT PBDA
1519   8B74  10 F6             BPL SWLP
1520   8B76  20 B7 8B          JSR VECSW        ;SWITCH VECTORS
1521   8B79  20 FF 8A          JSR BAUD
1522   8B7C  A2 FF      MONENT LDX #$FF         ;MONITOR ENTRY
1523   8B7E  9A                TXS
1524   8B7F  D8                CLD
1525   8B80  20 86 8B          JSR ACCESS       ;UNWRITE PROT MONITOR RAM
1526   8B83  4C 03 80          JMP WARM
1527   8B86  20 88 81   ACCESS JSR SAVER        ;UN WRITE PROT SYS RAM
1528   8B89  AD 01 AC          LDA OR3A
1529   8B8C  09 01             ORA #1
1530   8B8E  8D 01 AC   ACC1   STA OR3A
1531   8B91  AD 03 AC          LDA DDR3A
1532   8B94  09 01             ORA #1
1533   8B96  8D 03 AC          STA DDR3A
1534   8B99  4C C4 81          JMP RESALL
1535   8B9C  20 88 81   NACCES JSR SAVER        ;WRITE PROT SYS RAM
1536   8B9F  AD 01 AC          LDA OR3A
1537   8BA2  29 FE             AND #$FE
1538   8BA4  18                CLC
1539   8BA5  90 E7             BCC ACC1
```

```
LINE # LOC      CODE        LINE

1540  8BA7   20 86 8B    TTY     JSR ACCESS        ;UN WRITE PROT RAM
1541  8BAA   A9 D5               LDA #$D5          ;110 BAUD
1542  8BAC   8D 51 A6            STA SDBYT
1543  8BAF   AD 54 A6            LDA TOUTFL
1544  8BB2   09 40               ORA #$40
1545  8BB4   8D 54 A6            STA TOUTFL
1546  8BB7   20 86 8B    VECSW   JSR ACCESS        ;UN WRITE PROT RAM
1547  8BBA   A2 08               LDX #$8
1548  8BBC   BD 6F 8C    SWLP2   LDA TRMTBL,X
1549  8BBF   9D 60 A6            STA INVEC,X
1550  8BC2   CA                  DEX
1551  8BC3   10 F7               BPL SWLP2
1552  8BC5   60                  RTS
1553  8BC6               ;
1554  8BC6               ;***
1555  8BC6               ;*** TABLES (I/O CONFIGURATIONS, KEY CODES, ASCII CODES)
1556  8BC6               ;***
1557  8BC6   00          VALS    .BYT $00,$80,$08,$37 ;KB SENSE, A=1
1557  8BC7   80
1557  8BC8   08
1557  8BC9   37
1558  8BCA   00                  .BYT $00,$7F,$00,$30 ;KB LRN, A=5
1558  8BCB   7F
1558  8BCC   00
1558  8BCD   30
1559  8BCE   00                  .BYT $00,$FF,$00,$3F ;SCAN DSP, A=9
1559  8BCF   FF
1559  8BD0   00
1559  8BD1   3F
1560  8BD2   00                  .BYT $00,$00,$07,$3F ;BEEP, A=D
1560  8BD3   00
1560  8BD4   07
1560  8BD5   3F
1561  8BD6               VALSP2 =VALS+2
1562  8BD6               SYM     =*                ;KEY CODES RETURNED BY LRNKEY
1563  8BD6               TABLE=*
1564  8BD6   01                  .BYT $01          ;0/U0
1565  8BD7   41                  .BYT $41          ;1/U1
1566  8BD8   81                  .BYT $81          ;2/U2
1567  8BD9   C1                  .BYT $C1          ;3/U3
1568  8BDA   02                  .BYT $02          ;4/U4
1569  8BDB   42                  .BYT $42          ;5/U5
1570  8BDC   82                  .BYT $82          ;6/U6
1571  8BDD   C2                  .BYT $C2          ;7/U7
1572  8BDE   04                  .BYT $04          ;8/JMP
1573  8BDF   44                  .BYT $44          ;9/VER
1574  8BE0   84                  .BYT $84          ;A/ASCII
1575  8BE1   C4                  .BYT $C4          ;B/BLK MOV
1576  8BE2   08                  .BYT $08          ;C/CALC
1577  8BE3   48                  .BYT $48          ;D/DEP
1578  8BE4   88                  .BYT $88          ;E/EXEC
1579  8BE5   C8                  .BYT $C8          ;F/FILL
1580  8BE6   10                  .BYT $10          ;CR/SD
1581  8BE7   50                  .BYT $50          ;-/+
1582  8BE8   90                  .BYT $90          ;>/<
```

```
LINE # LOC     CODE       LINE

1583   8BE9  D0                  .BYT $D0          ;SHIFT
1584   8BEA  20                  .BYT $20          ;GO/LP
1585   8BEB  60                  .BYT $60          ;REG/SP
1586   8BEC  A0                  .BYT $A0          ;MEM/WP
1587   8BED  00                  .BYT $00          ;L2/L1
1588   8BEE  40                  .BYT $40          ;S2/S1
1589   8BEF               ASCIM1 =*-1
1590   8BEF               ASCII  =*                ;ASCII CODES AND HASH CODES
1591   8BEF  30                  .BYT $30          ;ZERO
1592   8BF0  31                  .BYT $31          ;ONE
1593   8BF1  32                  .BYT $32          ;TWO
1594   8BF2  33                  .BYT $33          ;THREE
1595   8BF3  34                  .BYT $34          ;FOUR
1596   8BF4  35                  .BYT $35          ;FIVE
1597   8BF5  36                  .BYT $36          ;SIX
1598   8BF6  37                  .BYT $37          ;SEVEN
1599   8BF7  38                  .BYT $38          ;EIGHT
1600   8BF8  39                  .BYT $39          ;NINE
1601   8BF9  41                  .BYT $41          ;A
1602   8BFA  42                  .BYT $42          ;B
1603   8BFB  43                  .BYT $43          ;C
1604   8BFC  44                  .BYT $44          ;D
1605   8BFD  45                  .BYT $45          ;E
1606   8BFE  46                  .BYT $46          ;F
1607   8BFF  0D                  .BYT $0D          ;CR
1608   8C00  2D                  .BYT $2D          ;DASH
1609   8C01  3E                  .BYT $3E          ;>
1610   8C02  FF                  .BYT $FF          ;SHIFT
1611   8C03  47                  .BYT $47          ;G
1612   8C04  52                  .BYT $52          ;R
1613   8C05  4D                  .BYT $4D          ;M
1614   8C06  13                  .BYT $13          ;L2
1615   8C07  1E                  .BYT $1E          ;S2
1616   8C08               ; KB UPPER CASE
1617   8C08  14                  .BYT $14          ;U0
1618   8C09  15                  .BYT $15          ;U1
1619   8C0A  16                  .BYT $16          ;U2
1620   8C0B  17                  .BYT $17          ;U3
1621   8C0C  18                  .BYT $18          ;U4
1622   8C0D  19                  .BYT $19          ;U5
1623   8C0E  1A                  .BYT $1A          ;U6
1624   8C0F  1B                  .BYT $1B          ;U7
1625   8C10  4A                  .BYT $4A          ;J
1626   8C11  56                  .BYT $56          ;V
1627   8C12  FE                  .BYT $FE          ;ASCII
1628   8C13  42                  .BYT $42          ;B
1629   8C14  43                  .BYT $43          ;C
1630   8C15  44                  .BYT $44          ;D
1631   8C16  45                  .BYT $45          ;E
1632   8C17  46                  .BYT $46          ;F
1633   8C18  10                  .BYT $10          ;SD
1634   8C19  2B                  .BYT $2B          ;+
1635   8C1A  3C                  .BYT $3C          ;<
1636   8C1B  00                  .BYT $00          ;SHIFT
1637   8C1C  11                  .BYT $11          ;LP
```

```
LINE # LOC     CODE       LINE

1638  8C1D  1C                  .BYT $1C         ;SP
1639  8C1E  57                  .BYT $57         ;W
1640  8C1F  12                  .BYT $12         ;L1
1641  8C20  1D                  .BYT $1D         ;S1
1642  8C21  2E                  .BYT $2E         ;.
1643  8C22  20                  .BYT $20         ;BLANK
1644  8C23  3F                  .BYT $3F         ;?
1645  8C24  50                  .BYT $50         ;P
1646  8C25  07                  .BYT $07         ;BELL
1647  8C26  53                  .BYT $53         ;S
1648  8C27  58                  .BYT $58         ;X
1649  8C28  59                  .BYT $59         ;Y
1650  8C29                ; SEGMENT CODES FOR ON-BOARD DISPLAY
1651  8C29                SEGSM1 =*-1
1652  8C29  3F                  .BYT $3F         ;ZERO
1653  8C2A  06                  .BYT $06         ;ONE
1654  8C2B  5B                  .BYT $5B         ;TWO
1655  8C2C  4F                  .BYT $4F         ;THREE
1656  8C2D  66                  .BYT $66         ;FOUR
1657  8C2E  6D                  .BYT $6D         ;FIVE
1658  8C2F  7D                  .BYT $7D         ;SIX
1659  8C30  07                  .BYT $07         ;SEVEN
1660  8C31  7F                  .BYT $7F         ;EIGHT
1661  8C32  67                  .BYT $67         ;NINE
1662  8C33  77                  .BYT $77         ;A
1663  8C34  7C                  .BYT $7C         ;B
1664  8C35  39                  .BYT $39         ;C
1665  8C36  5E                  .BYT $5E         ;D
1666  8C37  79                  .BYT $79         ;E
1667  8C38  71                  .BYT $71         ;F
1668  8C39  F0                  .BYT $F0         ;CR
1669  8C3A  40                  .BYT $40         ;DASH
1670  8C3B  70                  .BYT $70         ;>
1671  8C3C  00                  .BYT $00         ;SHIFT
1672  8C3D  6F                  .BYT $6F         ;G
1673  8C3E  50                  .BYT $50         ;R
1674  8C3F  54                  .BYT $54         ;M
1675  8C40  38                  .BYT $38         ;L2
1676  8C41  6D                  .BYT $6D         ;S2
1677  8C42  01                  .BYT $01         ;U0
1678  8C43  08                  .BYT $08         ;U1
1679  8C44  09                  .BYT $09         ;U2
1680  8C45  30                  .BYT $30         ;U3
1681  8C46  36                  .BYT $36         ;U4
1682  8C47  5C                  .BYT $5C         ;U5
1683  8C48  63                  .BYT $63         ;U6
1684  8C49  03                  .BYT $03         ;U7
1685  8C4A  1E                  .BYT $1E         ;J
1686  8C4B  72                  .BYT $72         ;V
1687  8C4C  77                  .BYT $77         ;A
1688  8C4D  7C                  .BYT $7C         ;B
1689  8C4E  39                  .BYT $39         ;C
1690  8C4F  5E                  .BYT $5E         ;D
1691  8C50  79                  .BYT $79         ;E
1692  8C51  71                  .BYT $71         ;F
```

```
LINE # LOC     CODE        LINE

1693   8C52  6D                    .BYT $6D        ;SD
1694   8C53  76                    .BYT $76        ;+
1695   8C54  46                    .BYT $46        ;<
1696   8C55  00                    .BYT $00        ;SHIFT
1697   8C56  38                    .BYT $38        ;LP
1698   8C57  6D                    .BYT $6D        ;SP
1699   8C58  1C                    .BYT $1C        ;W
1700   8C59  38                    .BYT $38        ;L1
1701   8C5A  6D                    .BYT $6D        ;S1
1702   8C5B  80                    .BYT $80        ;.
1703   8C5C  00                    .BYT $00        ;SPACE
1704   8C5D  53                    .BYT $53        ;?
1705   8C5E  73                    .BYT $73        ;P
1706   8C5F  49                    .BYT $49        ;BELL
1707   8C60  6D                    .BYT $6D        ;S
1708   8C61  64                    .BYT $64        ;X
1709   8C62  6E                    .BYT $6E        ;Y
1710   8C63  97            DECPTS .BYT $97,$3D,$1F,$10,$08,$00 ; TO DETERMINE BAUD RATE
1710   8C64  3D
1710   8C65  1F
1710   8C66  10
1710   8C67  08
1710   8C68  00
1711   8C69  D5 4C         STDVAL .DBY $D54C,$2410,$0601    ;STD VALS FOR BAUD RATES
1711   8C6B  24 10
1711   8C6D  06 01
1712   8C6F                ; 110,300,600,1200,2400,4800 BAUD
1713   8C6F  4C 58 8A      TRMTBL JMP INTCHR       ;ALTERNATE VCTRS FOR TIO
1714   8C72  4C A0 8A             JMP TOUT
1715   8C75  4C 3C 8B             JMP TSTAT
1716   8C78                ;
1717   8C78                ;
1718   8C78                ;***
1719   8C78                ;*** DEFAULT TABLE
1720   8C78                ;***
1721   8C78                       *=$8FA0
1722   8FA0                DFTBLK=*
1723   8FA0  00 C0                .WORD $C000      ;BASIC  *** JUMP TABLE
1724   8FA2  A7 8B                .WORD TTY
1725   8FA4  64 8B                .WORD NEWDEV
1726   8FA6  00 00                .WORD $0000      ;PAGE ZERO
1727   8FA8  00 02                .WORD $0200
1728   8FAA  00 03                .WORD $0300
1729   8FAC  00 C8                .WORD $C800
1730   8FAE  00 D0                .WORD $D000
1731   8FB0  04                   .BYT $04         ;TAPE DELAY (9.0 SEC)
1732   8FB1  2C                   .BYT $2C         ;KIM TAPE BOUNDARY
1733   8FB2  46                   .BYT $46         ;HS TAPE BOUNDARY
1734   8FB3  00                   .BYT $00,$00     ;SCR3,SCR4
1734   8FB4  00
1735   8FB5  33                   .BYT $33         ;HS TAPE FIRST 1/2 BIT
1736   8FB6  00                   .BYT $00,$00     ;SCR6,SCR7
1736   8FB7  00
1737   8FB8  00                   .BYT $00,$00,$00,$00 ;SCR8-SCRB
1737   8FB9  00
```

```
LINE # LOC      CODE        LINE

1737   8FBA   00
1737   8FBB   00
1738   8FBC   5A                      .BYT $5A        ;HS TAPE SECOND 1/2 BIT
1739   8FBD   00                      .BYT $00,$00,$00 ;SCRD-SCRF
1739   8FBE   00
1739   8FBF   00
1740   8FC0   00                      .BYT $00,$00,$6D,$6E,$86,$06 ;DISP BUFFER (SY1.1)
1740   8FC1   00
1740   8FC2   6D
1740   8FC3   6E
1740   8FC4   86
1740   8FC5   06
1741   8FC6   00                      .BYT $00,$00,$00 ;NOT USED
1741   8FC7   00
1741   8FC8   00
1742   8FC9   00                      .BYT $00        ;PARNR
1743   8FCA   00 00                   .DBYT $0000,$0000,$0000 ;PARMS
1743   8FCC   00 00
1743   8FCE   00 00
1744   8FD0   01                      .BYT $01        ;PADBIT
1745   8FD1   4C                      .BYT $4C        ;SDBYT
1746   8FD2   00                      .BYT $00        ;ERCNT
1747   8FD3   80                      .BYT $80        ;TECHO
1748   8FD4   B0                      .BYT $B0        ;TOUTFL
1749   8FD5   00                      .BYT $00        ;KSHFL
1750   8FD6   00                      .BYT $00        ;TV
1751   8FD7   00                      .BYT $00        ;LSTCOM
1752   8FD8   10                      .BYT $10        ;MAXRC
1753   8FD9   4A 8B                   .WORD RESET     ;USER REG'S
1754   8FDB   FF                      .BYT $FF        ;STACK
1755   8FDC   00                      .BYT $00        ;FLAGS
1756   8FDD   00                      .BYT $00        ;A
1757   8FDE   00                      .BYT $00        ;X
1758   8FDF   00                      .BYT $00        ;Y
1759   8FE0                   ; VECTORS
1760   8FE0   4C BE 89              JMP HKEY        ;INVEC
1761   8FE3   4C 00 89              JMP HDOUT       ;OUTVEC
1762   8FE6   4C 6A 89              JMP KYSTAT      ;INSVEC
1763   8FE9   4C D1 81              JMP M1          ;UNRECOGNIZED SYNTAX (ERROR)
1764   8FEC   4C D1 81              JMP M1          ;UNRECOGNIZED COMMAND (ERROR)
1765   8FEF   4C 06 89              JMP SCAND       ;SCNVEC
1766   8FF2   7E 88                 .WORD RIN       ;IN PTR FOR EXEC FROM RAM
1767   8FF4   C0 80                 .WORD TRCOFF    ;USER TRACE VECTOR
1768   8FF6   4A 80                 .WORD SVBRK     ;BRK
1769   8FF8   29 80                 .WORD SVIRQ     ;USER IRQ
1770   8FFA   9B 80                 .WORD SVNMI     ;NMI
1771   8FFC   4A 8B                 .WORD RESET     ;RESET
1772   8FFE   0F 80                 .WORD IRQBRK    ;IRQ

1774   9000                   LENTRY =$8C78
1775   9000                   SENTRY =$8C78+$20F
1776   9000                   RGNAM  =$8F9A           ;REGISTER NAME PATCH
```

```
LINE # LOC     CODE       LINE

1778  9000                 .END

ERRORS = 0000 <0000>
```

```
SYMBOL   VALUE  LINE DEFINED        CROSS-REFERENCES

ACCESS   8B86       1527   123  129  137  177  191 1507 1525 1540 1546
ACC1     8B8E       1530  1539
ADVCK    81CB        336   581  599
AGAIN    8B24       1475  1471
AR       A65D         59   152  186  616
ASCII    8BEF       1590  1259
ASCIM1   8BEE       1589  1313
ASCNIB   8275        412   344  352  394
BADDY    848B        666   660
BAUD     8AFF       1459  1521
BEEP     8972       1266  1193
BEEPP3   8975       1267  1304
BEEPP5   8977       1268  1286
BE1      897C       1270  1277
BE2      8995       1280  1272 1275
BE3      8997       1281  1282
BLK3     8739        991   976
BLP      875E       1007  1013 1014
BLP1     879D       1035  1042
BMOVE    87B7       1048  1007 1035
BRT      87CC       1058  1052
BRTT     87C1       1053   985
BZPARM   8395        559   261
B1       87AF       1043   987  990 1005 1012 1015 1041
B1PARM   84DA        699   264
B2       8772       1016  1006
B2PARM   8619        858   267
B3PARM   8714        971   270
CALC3    8827       1103   857  970 1085
CHKSAD   82DD        463   676  887  955
CLEAR    8B08       1464  1465
COMINB   81D6        342   569
COMMA    833A        511   342  503  732
COMPAR   82CA        455   444  446
CONFIG   89A5       1288  1211 1230 1268
CON1     89AB       1291  1297
CRCHK    8204        364   362  363
CRLF     834D        521   147  219  277  497  564  587  604  622  784  905
                           926 1105
CRLFSZ   8316        497   706  731  882
C1       882B       1105  ****
DBNEW    80F6        212   204
DBOFF    80D3        198   146  180  194
DBON     80E4        205   196
DDR1B    A002         91  ****
DDR3A    AC03         89   202  209  211  212  214  854 1531 1533
DEAF     8B15       1469  1476
DECCMP   82BE        449   761  899 1040
DECPTS   8C63       1710  1469
DELAY    835A        528   188
DEPBYT   84E8        709   718
DEPEC    850E        726   720
DEPES    8553        759   722
DEPN     84F9        716   714  723  725
```

```
SYMBOL  VALUE  LINE DEFINED      CROSS-REFERENCES

DEPZ     84A7        677  620
DEP1     84DA        703  ****
DETBRK   801B        113  ****
DETIRQ   8022        118   112
DFTBLK   8FA0       1722  1509
DFTXFR   8B59       1508  1512
DIFFL    86FD        958  ****
DIFFL2   8710        969  ****
DIFFZ    86FA        957   923
DIFF1    870F        968   964 1495
DISBUF   A640         27  1214 1323 1324
DISPAT   814A        256    99
DLYF     8AE6       1441  1416 1485
DLYH     8AE9       1442  1381 1395 1400 1408 1441
DLYO     8383        544   539
DLYX     8AF1       1448  1452
DLYY     8AF3       1449  1450
DLY1     8368        533   536
DLY2     8371        537   543
DL1      835D        529  ****
DMY1     8A81       1390  1387
ERCNT    A652         43   624  632  649  653  667  671  673  979  995 1043
                          1053 1057
ERMSG    8171        275   100
EXEVEC   A672         74  1130 1135 1137
EXE3     8851       1125  1104
EXITCP   82D9        461   458
EXITG    88FF       1207  1203
EXITGK   88CC       1183  1171
EXITLP   843F        634  ****
EXITM1   8577        780   745
EXITNB   8315        496   494
EXITOD   8A03       1330  1311 1317 1320
EXITRG   83C1        579   598
EXRGP1   83C2        580   582
EXWRAP   82BD        448   447
EZPARM   84D7        698   695
E1PARM   8616        857   842
E2PARM   8711        970   919
E3PARM   88AC       1164  1126
FILL3    8714        975  ****
FOUND    8960       1255  1250
FR       A65C         58   164  612
F1       8723        981   988  989
F2       8737        990  ****
F3       872E        986   984
GETCOM   80FF        219    98  223  246
GETC1    8107        222   225  227
GETKEY   88AF       1168  1299
GETSGS   89EA       1318  1314
GK       88CF       1184  1169 1172 1179
GK1      88D4       1186  1187 1189 1206
GK2      88E4       1194  1195 1198
GOOD     81F3        355   360
GOPAD    8ACE       1429  1426
GO2      83F7        604  ****
GOZ      83F3        602   563
```

```
SYMBOL  VALUE  LINE DEFINED        CROSS-REFERENCES

G01      8579        782   729
G01ENT   83FA        605   197
HASHL    812F        240   234
HASHUS   8133        242   230  232  241
HDOUT    8900       1208  1754
HIPN     816E        271   257  269
HKEY     89BE       1299  1753
IDISP    8053        145   127  136  184
IJSCNV   8903       1209   537 1186
INBYTE   81D9        343   572  657  661  666  675  710
INCCMP   82B2        443   654  716  742  889  946  986 1011 1091
INCHR    8A1B       1348   222  245  252  343  351  359  382  626 1162 1362
INCP3    8293        429  1070 1081
INJINV   8A41       1365  1349
INJISV   8392        554   548  550
INJOUV   8A55       1372  1370
INK      8B27       1477  1464 1466
INK1     8B2A       1479  1480
INRT1    8A2D       1356  1352 1354
INRT2    8A3C       1363  1357
INSTAT   8386        548   538  896  944
INST1    838B        550   551
INST2    8391        553   549
INSVEC   A666         67   554
INTCHR   8A58       1373  1713
INVEC    A660         65  1128 1133 1136 1138 1158 1160 1365 1549
IRQBRK   800F        105  1765
IRQVEC   A67E         82  ****
JTABLE   A620          9   822  824
JUMP1    85B4        808   797
JUM2     85E5        833   812
KEYQ     8923       1223  1194 1197 1261 1516
KSCONF   89A3       1287  1223 1278 1489 1515
KSHFL    A655         48  1185 1205 1257
KYSTAT   896A       1261  1755
LDBYTE   84A1        675   629  637  640  643
LEAVE    8AD1       1430  1428
LENTRY   8C78       1716   693  836 1083
LK1      8944       1237  1234
LK2      894A       1240  1238
LK3      8956       1249  1252
LOCM8    8569        772   758
LOCP8    855B        764   754
LOOK     8A5F       1376  1380
LPGD     846D        654   648
LPZ      8429        625   641  664  670  674
LPZB     8417        618   603
LP1      842C        626   628
LRNKEY   892C       1227  1188
LSTCOM   A657         50   235  244  248  258
L1J      84CC        693   697
L1ZB     84C6        690   684
L11B     85D7        826   809
L11C     85DD        829   840
L11D     85E9        835   917
L12B     8688        911   879  915
L12C     868C        913  ****
```

```
SYMBOL   VALUE   LINE DEFINED        CROSS-REFERENCES

L2ZB     84CF          694   691
L21B     85EF          837   827
L23P     87EE         1076  1073
MAXRC    A658           51   927  929
MEMZ     84AE          680   678
MEM1     8510          728   704
MEM2     862F          873   863
MEM3     8801         1084  1077
MEM3C    8808         1087   877 1093 1094
MEM3D    8811         1091  1100
MEM3E    881C         1097  1090
MEM3EX   881A         1095  1092
MEM3F    8826         1102  1098
MONENT   8B7C         1522    97 1517
MONITR   8000           97  ****
MORED    8454          643   656
MORED2   86D2          941   949
M1       81D1          339   337 1757
M12      8159          262   260
M13      8160          265   263
M14      8167          268   266
M15      81B7          314   275
M21      8239          387   384
M22      824A          394   386
M23      8251          397   400
M24      8267          406   393   395
M25      826F          409   407
M26      8289          422   415   417
M27      828D          425   419
M28      828F          426   421
M29      8292          428   413
M32      82C8          454   450
M33      82EB          470   468
M34      83C8          583   577
M35      83CA          584   597   600
M36      83EB          598   595
M42      8566          770   768
M43      8574          778   776
NACCES   8B9C         1535   605   785   813
NBASOC   8A44         1366   485   487   591
NBELL    89CD         1305  1303
NEWDEV   8B64         1513  1725
NEWLN    84E1          706   679   719
NEWLOC   8517          731   682   744   748   763   771   779 1097
NH3      83BF          578   570   573
NH41     8501          720   711
NH42     8537          745   736
NIBALF   8313          495   492
NIBASC   8309          490  1366
NMIVEC   A67A           80  ****
NOBEEP   899B         1284  1196
NOKEY    895E         1253  1236
NOTCR    83C3          581   578
NR10     8408          611   795   825
NUREC    8443          636   631
NXTLOC   8531          742   740   750   752
NXTRG    83D2          588  ****
```

```
SYMBOL  VALUE  LINE DEFINED        CROSS-REFERENCES

OBCMIN   81D3        341   594  735
OBCRLF   834A        520   187
OCMCK    8325        503   895  904
OPCCOM   8337        510   148  185
OR1B     A000         90  ****
OR3A     AC01         88    89  198  201  205  208  852 1528 1530 1536
OUD1     89DE       1312  1307
OUD2     89E0       1313  1316
OUD3     89F4       1323  1327
OUT      8AD4       1431  1388 1402 1415
OUTBYT   82FA        479   284  341  477  505  520  888  956
OUTC     8AB4       1415  1423
OUTCHR   8A47       1367   150  221  279  281  502  517  523  525  566  589
                           934 1154 1514
OUTDSP   89C1       1300  1208
OUTONE   8ADE       1436  1434
OUTPC    82EE        473   510  568
OUTQM    8320        501   715  741
OUTSZ    8319        498  ****
OUTVEC   A663         66  1372
OUTXAH   82F4        475   500  908  952 1122
OUT1     81FE        361   358
OUT2     8201        363   353
OUT4     81F5        357   345
PAD      8B32       1484  1429
PADA     A400         86  1215 1217 1224 1235 1294
PADBIT   A650         41  1484
PAD1     8B35       1485  1487
PARFIL   822E        382   405
PARM     8220        377   239
PARNR    A649         30   259  379  389  390  408
PBDA     A402         87  1216 1231 1271 1274 1292 1376 1382 1410 1432 1437
                          1461 1481 1490 1518
PCHR     A65A         56   134  162  474  576  593  596  608
PCLR     A659         55   132  159  473  574  610
PCR1     A00C         92  1503
PHAKE    8ABC       1418  1419
PM1      822B        381   392
POR      8B4D       1502  ****
PRM10    820A        368   375
PRVLOC   8555        761   756
PSHOVE   8208        367   237  238  381  834  835
PTRIN    8872       1139  1131
P1H      A64F         40   373  999 1111
P1L      A64E         39   372  876  980  997 1062 1078 1087 1108
P2H      A64D         38   371  433  804  865 1033 1112
P2L      A64C         37   370  435  799  869  875  913 1031 1109
P2SCR    829C        433   880  922  977  996 1086
P3H      A64B         36   369  398  431  438  457  688  792  803  806  847
                           962 1021 1034 1119 1139
P3L      A64A         35   368  397  401  402  429  440  460  686  794  798
                           802  810  829  843  958 1018 1032 1116 1141
P3SCR    82A7        438   705  730  864 1030
RAM      A620          8  1510
RC       A63D         24   636  655  932  935  948
RDIG     A645         28  1308 1310 1329
REGZ     8395        562  ****
```

```
SYMBOL   VALUE   LINE DEFINED         CROSS-REFERENCES

RESALL   81C4         326    925 1279 1298 1330 1371 1430 1534
RESET    8B4A        1500   1746 1764
RESTIV   8899        1157   1148
RESXAF   81B8         316    411 1156 1163 1183 1364 1405
RESXF    81BE         321    462  544  635
RGBACK   8399         564    586
RIN      887E        1145   1759
RSTVEC   A67C          81   ****
SAVE     8A87        1394   1389
SAVER    8188         289    377  455  529  621  921 1145 1168 1266 1284 1288
                            1300 1348 1367 1373 1407 1527 1535

SAVINT   8064         152    125  131  139  179  193
SCAND    8906        1210   1758
SCNVEC   A66F          70   1209
SCPBUF   A600           7   1335 1336 1342
SCRA     A63A          20   1132 1134 1157 1159
SCRB     A63B          21   ****
TAPET2   A63C          22   ****
SCRD     A63D          23     24
SCRE     A63E          25   1178 1182
SCRF     A63F          26   1228 1247
TAPDEL   A630          10   ****
KMBDRY   A631          11   ****
HSBDRY   A632          12   ****
SCR3     A633          13    350  354  380  388  404  406
SCR4     A634          14    571  575
TAPET1   A635          15   ****
SCR6     A636          16    466  467  504  507  663  951
SCR7     A637          17    469  508  659  907  950
SCR8     A638          18    532  533  540
SCR9     A639          19    531  534  542
SC1      890D        1213   1222
SC2      891D        1219   1220
SDBYT    A651          42   1447 1468 1470 1473 1542
SEEK     8B02        1461   1463
SEGSM1   8C28        1651   1318
SENTRY   8E87        1717   1071
SET      8B0D        1466   1467
SPACE    8342         515    236  282  514  567  709  724  884  906 1106
SPCP3    8345         517    513
SPC2     833F         514    592
SPEXIT   86A8         925    945  947
SP2B     8698         918    912
SP2C     86A3         923    954
SP2D     86AB         926    924
SP2E     86B8         931    928
SP2F     86BA         932    930
SR       A65B          57    175  606
STDVAL   8C69        1711   1472
STD2     8619         862   ****
STOCOM   8120         235    251
SVBRK    804A         137   1761
SVBYTE   86F4         955    936  938  940  943
SVIRQ    8029         123   1762
SVNMI    809B         177   1763
SWITCH   8B69        1515   ****
SWLP     8B6C        1516   1519
```

```
SYMBOL   VALUE   LINE DEFINED         CROSS-REFERENCES

SWLP2     8BBC          1548  1551
SYM       8BD6          1562  1249
S1NG      87DE          1068  1080
S13B      87CD          1059   993
S13C      87D3          1062  1075
S23B      87E6          1072  1060
TABLE     8BD6          1563  ****
TAPERR    848E           667   630   638   644   658   662   665
TECHO     A653            45  1152  1358  1360  1368  1386
TEXT      8A06          1331  1305
TIN       8A6A          1381  1399
TLP1      8A83          1391  1392
TOUT      8AA0          1406  1714
TOUTFL    A654            47  1377  1383  1436  1543  1545
TRACON    80CD           196   189
TRCOFF    80C0           191  1760
TRCVEC    A674            75   195
TRMTBL    8C6F          1713  1548
TSTAT     8B3C          1489  1715
TTY       8BA7          1540  1724
TV        A656            49   181   528
TVNZ      80AF           185   182
TXTMOV    8A0B          1335  1338
UBRKV     A676            77  ****
UBRKVC    A676            76    77
UIRQV     A678            79  ****
UIRQVC    A678            78    79
URCVEC    A66C            69   698  1164
URSVEC    A669            68   271
USRENT    8035           128  ****
VADDR     8646           882   897
VALS      8BC6          1557  1293  1561
VALSP2    8BC8          1561  1291
VECSW     8BB7          1546  1520
VERZ      84B5           683   681
VER1      8596           796   689   783
VER2      863C           878   807   874
VOCK      8664           895  ****
V1        866D           900   890   892   903
V2        864B           884   894
WARM      8003            98   101   151   190  1526
WFR1B     85F7           841   838
WRAP      82B8           446   452
XR        A65E            60   153   615
YR        A65F            61   154   614
ZERCK     832E           506   625   881   957
```

```
LINE # LOC     CODE        LINE

0002  0000                 ;****** VERSION 2 4/13/79 'SY1.1'
0003  0000                 ;****** COPYRIGHT 1978 SYNERTEK SYSTEMS CORPORATION
0004  0000                 ;******
0005  0000                 BDRY   =$F8           ;0/1 BDRY FOR READ TIMING
0006  0000                 OLD    =$F9           ;HOLD PREV INPUT LEVEL IN GETTR
0007  0000                 CHAR   =$FC           ;CHAR ASSY AND DISASSY
0008  0000                 MODE   =$FD           ;BIT7=1 IS HS, 0 IS KIM
0009  0000                                      ;... BIT6=1 - IGNORE DATA
0010  0000                 BUFADL =$FE           ;RUNNING BUFFER ADR
0011  0000                 BUFADH =$FF
0012  0000                 TAPDEL =$A630         ;HI SPEED TAPE DELAY
0013  0000                 KMBDRY =$A631         ;KIM READ BDRY
0014  0000                 HSBDRY =$A632         ;HS READ BDRY
0015  0000                 TAPET1 =$A635         ;HS FIRST 1/2 BIT
0016  0000                 TAPET2 =$A63C         ;HS SECOND 1/2 BIT
0017  0000                 SCR6   =$A636         ;SCR 6
0018  0000                 SCR7   =$A637         ;SCR 7
0019  0000                 SCR8   =$A638         ;SCR 8
0020  0000                 SCR9   =$A639         ;SCR 9

0022  0000                        *=$A64A
0023  A64A                 EAL    *=*+1          ;P3L - END ADDR +1 (LO)
0024  A64B                 EAH    *=*+1          ;P3H -  (HI)
0025  A64C                 SAL    *=*+1          ;P2L - START ADDR  (LO)
0026  A64D                 SAH    *=*+1          ;P2H -  (HI)
0027  A64E                 ID     *=*+1          ;P1L -  ID

0029  A64F                 EOT    = $04
0030  A64F                 SYN    = $16
0031  A64F                 TPBIT  =%1000         ;BIT 3 IS ENABLE/DISABLE TO DECODER
0032  A64F                 FRAME  =$FF           ;ERROR MSG # FOR FRAME ERROR
0033  A64F                 CHECK  =$CC           ;ERROR # FOR CHECKSUM ERROR
0034  A64F                 LSTCHR =$2F           ;LAST CHAR NOT '/'
0035  A64F                 NONHEX =$FF           ;NON HEX CHAR IN KIM REC

0037  A64F                 ACCESS =$8B86         ;UNWRITE PROTECT SYSTEM RAM
0038  A64F                 P2SCR  =$829C         ;MOVE P2 TO $FF,$FE IN PAGE ZERO
0039  A64F                 ZERCK  =$832E         ;MOVE ZERO TO CHECK SUM
0040  A64F                 CONFIG =$89A5         ;CONFIGURE I/O

0042  A64F                 ; I/O - TAPE ON/OFF IS CB2 ON VIA 1 (A000)
0043  A64F                 ;       TAPE IN IS PB6 ON VIA 1 (A000)
0044  A64F                 ;       TAPE OUT IS CODE 7 TO DISPLAY DECODER, THRU 6532,
0045  A64F                 ;             PB0-PB3 (A400)

0047  A64F                 VIAACR =$A00B
0048  A64F                 VIAPCR =$A00C         ;CONTROL CB2 TAPE ON/OFF, POR
0049  A64F                 TPOUT  =$A402
0050  A64F                 TAPOUT =TPOUT
0051  A64F                 DDROUT =$A403
0052  A64F                 TAPIN  =$A000
0053  A64F                 DDRIN  =$A002
0054  A64F                 TIMER  =$A406         ;6532 TIMER READ
0055  A64F                 TIM8   =$A415         ;6532 TIMER SET (8US)
0056  A64F                 DDRDIG =$A401
```

```
 LOC      CODE        LINE

 A64F                DIG     =$A400

 A64F                ; LOADT ENTER W/ID IN PARM 2, MODE IN [Y]

 A64F                        *=$8C78
 8C78  20 A9 8D      LOADT   JSR START          ;INITIALIZE
 8C7B  20 52 8D      LOADT2  JSR SYNC           ;GET IN SYNC
 8C7E  20 E1 8D      LOADT4  JSR RDCHTX
 8C81  C9 2A                 CMP #'*            ;START OF DATA?
 8C83  F0 06                 BEQ LOAD11
 8C85  C9 16                 CMP #SYN           ;NO - SYN?
 8C87  D0 F2                 BNE LOADT2         ;IF NOT, RESTART SYNC SEARCH
 8C89  F0 F3                 BEQ LOADT4         ;IF YES, KEEP LOOKING FOR *

 8C8B  06 FD         LOAD11  ASL MODE           ;GET MODE IN A, CLEAR BIT6
 8C8D  6A                    ROR A
 8C8E  85 FD                 STA MODE
 8C90  20 26 8E              JSR RDBYTX         ;READ ID BYTE ON TAPE
 8C93  8D 00 A4              STA DIG            ;DISPLAY ON LED (NOT DECODED)
 8C96  CD 4E A6              CMP ID             ;COMPARE WITH REQUESTED ID
 8C99  F0 29                 BEQ LOADT5         ;LOAD IF EQUAL
 8C9B  AD 4E A6              LDA ID             ;COMPARE WITH 0
 8C9E  C9 00                 CMP #0
 8CA0  F0 22                 BEQ LOADT5         ;IF 0, LOAD ANYWAY
 8CA2  C9 FF                 CMP #$FF           ;COMPARE WITH FF
 8CA4  F0 07                 BEQ LOADT6         ;IF FF, USE REQUEST SA TO LOAD

 8CA6  24 FD                 BIT MODE           ;UNWANTED RECORD. KIM OR HS?
 8CA8  30 16                 BMI HWRONG
 8CAA  4C 7B 8C              JMP LOADT2         ;IF KIM, RESTART SEARCH

 8CAD                ; SA (&EA IF USED) COME FROM REQUEST. DISCARD TAPE VALUES
 8CAD                ;    (BUFAD ALREADY SET TO SA BY 'START')
 8CAD                ;
 8CAD  20 74 8E      LOADT6  JSR RDCHK          ;GET SAL FROM TAPE
 8CB0  20 74 8E              JSR RDCHK          ;GET SAH FROM TAPE
 8CB3  24 FD                 BIT MODE           ;HS OR KIM?
 8CB5  10 52                 BPL LOADT7         ;IF KIM, START READING DATA
 8CB7  20 74 8E              JSR RDCHK          ;HS. GET EAH, EAL FROM TAPE
 8CBA  20 74 8E              JSR RDCHK          ; ... BUT IGNORE
 8CBD  4C DE 8C              JMP LT7H           ;START READING HS DATA

 8CC0                ; SA (& EA IF USED) COME FROM TAPE. SA REPLACES BUFAD

 8CC0  A9 C0         HWRONG  LDA #$C0           ;READ THRU TO GET TO NEXT REC
 8CC2  85 FD                 STA MODE           ;BUT DON'T CHECK CKSUM, NO FRAME ERR

 8CC4  20 74 8E      LOADT5  JSR RDCHK          ;GET SAL FROM TAPE
 8CC7  85 FE                 STA BUFADL         ;PUT IN BUF START L
 8CC9  20 74 8E              JSR RDCHK          ;SAME FOR SAH
 8CCC  85 FF                 STA BUFADH
 8CCE                ;(SAL - H STILL HAVE REQUEST VALUE)
 8CCE  24 FD                 BIT MODE           ;HS OR KIM?
 8CD0  10 37                 BPL LOADT7         ;IF KIM, START READING RECORD
 8CD2  20 74 8E              JSR RDCHK          ;HS. GET & SAVE EAL,EAH
```

```
LINE # LOC     CODE       LINE

0112  8CD5  8D 4A A6           STA EAL
0113  8CD8  20 74 8E           JSR RDCHK
0114  8CDB  8D 4B A6           STA EAH

0116  8CDE                ; READ HS DATA

0118  8CDE  20 E5 8D      LT7H   JSR RDBYTH      ;GET NEXT BYTE
0119  8CE1  A6 FE                LDX BUFADL      ;CHECK FOR END OF DATA + 1
0120  8CE3  EC 4A A6             CPX EAL
0121  8CE6  D0 07                BNE LT7HA
0122  8CE8  A6 FF                LDX BUFADH
0123  8CEA  EC 4B A6             CPX EAH
0124  8CED  F0 14                BEQ LT7HB
0125  8CEF  20 77 8E      LT7HA  JSR CHKT        ;NOT END, UPDATE CHECKSUM
0126  8CF2  24 FD                BIT MODE        ;WRONG RECORD?
0127  8CF4  70 04                BVS LT7HC       ;IF SO, DONT STORE BYTE
0128  8CF6  A0 00                LDY #0          ;STORE BYTE
0129  8CF8  91 FE                STA (BUFADL),Y
0130  8CFA  E6 FE         LT7HC  INC BUFADL      ;BUMP BUFFER ADDR
0131  8CFC  D0 E0                BNE LT7H
0132  8CFE  E6 FF                INC BUFADH      ;CARRY
0133  8D00  4C DE 8C             JMP LT7H

0135  8D03  C9 2F         LT7HB  CMP #'/         ;EA, MUST BE '/'
0136  8D05  D0 29                BNE LCERR       ;LAST CHAR NOT '/'
0137  8D07  F0 15                BEQ LOADT8      ;(ALWAYS)

0139  8D09                ; READ KIM DATA

0141  8D09  20 2A 8E      LOADT7 JSR RDBYT
0142  8D0C  B0 26                BCS LDT7A       ;NONHEX OR LAST CHAR?
0143  8D0E  20 77 8E             JSR CHKT        ;UPDATE CHECKSUM (PACKED BYTE)
0144  8D11  A0 00                LDY #0          ;STORE BYTE
0145  8D13  91 FE                STA (BUFADL),Y
0146  8D15  E6 FE                INC BUFADL      ;BUMP BUFFER ADR
0147  8D17  D0 F0                BNE LOADT7      ;CARRY?
0148  8D19  E6 FF                INC BUFADH
0149  8D1B  4C 09 8D             JMP LOADT7

0151  8D1E                ; TEST CHECKSUM & FINISH

0153  8D1E                LOADT8 =*
0154  8D1E  20 26 8E      LT8A   JSR RDBYTX      ;CHECK SUM
0155  8D21  CD 36 A6             CMP SCR6
0156  8D24  D0 16                BNE CKERR
0157  8D26  20 26 8E             JSR RDBYTX
0158  8D29  CD 37 A6             CMP SCR7
0159  8D2C  D0 0E                BNE CKERR       ;CHECK SUM ERROR
0160  8D2E  F0 11                BEQ OKEXIT      ;(ALWAYS)

0162  8D30  A9 2F         LCERR  LDA #LSTCHR     ;LAST CHAR IS NOT '/'
0163  8D32  D0 0A                BNE NGEXIT      ;(ALWAYS)

0165  8D34  C9 2F         LDT7A  CMP #'/         ;LAST OR NONHEX?
0166  8D36  F0 E6                BEQ LOADT8      ;LAST
```

```
LINE # LOC     CODE        LINE

0167   8D38                FRERR                  ;FRAMING ERROR
0168   8D38   A9 FF        NHERR  LDA #NONHEX     ;KIM ONLY, NON HEX CHAR READ
0169   8D3A   D0 02               BNE NGEXIT      ;(ALWAYS)

0171   8D3C   A9 CC        CKERR  LDA #CHECK      ;CHECKSUM ERROR

0173   8D3E   38           NGEXIT SEC             ;ERROR INDICATOR TO MONITOR IS CARRY
0174   8D3F   B0 01               BCS EXIT        ;(ALWAYS)

0176   8D41   18           OKEXIT CLC             ;NO ERROR

0178   8D42   24 FD        EXIT   BIT MODE
0179   8D44   50 08               BVC EX10        ;READING WRONG REC?
0180   8D46   A0 80               LDY #$80
0181   8D48   4C 78 8C            JMP LOADT       ;RESTART SEARCH

0183   8D4B   68           USRREQ PLA             ;USER REQUESTS EXIT
0184   8D4C   68                  PLA
0185   8D4D   38                  SEC
0186   8D4E   A2 CC        EX10   LDX #$CC
0187   8D50   D0 69               BNE STTC        ;STOP TAPE, RETURN
0188   8D52   AD 02 A0     SYNC   LDA DDRIN       ;CHANGE DATA DIRECTION
0189   8D55   29 BF               AND #$BF
0190   8D57   8D 02 A0            STA DDRIN
0191   8D5A   A9 00               LDA #0
0192   8D5C   8D 0B A0            STA VIAACR
0193   8D5F   AD 31 A6            LDA KMBDRY      ;SET UP BOUNDARY
0194   8D62   24 FD               BIT MODE
0195   8D64   10 03               BPL SY100
0196   8D66   AD 32 A6            LDA HSBDRY
0197   8D69   85 F8        SY100  STA BDRY
0198   8D6B   A9 6D               LDA #$6D
0199   8D6D   8D 00 A4            STA DIG         ;INDICATE NO SYNC ON LEDS
0200   8D70   A5 FD               LDA MODE        ;TURN ON OUT OF SYNC MODE
0201   8D72   09 40               ORA #$40        ;BIT6
0202   8D74   85 FD               STA MODE
0203   8D76   A9 7F        SYNC5  LDA #$7F        ;TEST FOR CR DOWN ON HKB
0204   8D78   8D 01 A4            STA DDRDIG
0205   8D7B   2C 00 A4            BIT DIG
0206   8D7E   10 CB               BPL USRREQ      ;CR KEY DOWN - EXIT (ERROR)
0207   8D80   20 9F 8D            JSR SYNBIT
0208   8D83   66 FC               ROR CHAR
0209   8D85   A5 FC               LDA CHAR
0210   8D87   C9 16               CMP #SYN
0211   8D89   D0 EB               BNE SYNC5
0212   8D8B   A2 0A        SYNC10 LDX #10         ;NOW MAKE SURE CAN GET 10 SYNS
0213   8D8D   20 E1 8D            JSR RDCHTX
0214   8D90   C9 16               CMP #SYN
0215   8D92   D0 E2               BNE SYNC5
0216   8D94   CA                  DEX
0217   8D95   D0 F6               BNE SYNC10+2
0218   8D97   8E 00 A4            STX DIG         ;TURN OFF DISPLAY
0219   8D9A   CA                  DEX             ;X=$FF
0220   8D9B   8E 01 A4            STX DDRDIG
0221   8D9E   60                  RTS
```

```
LINE # LOC      CODE        LINE

0222   8D9F                 ;SYNBIT - GET BIT IN SYN SEARCH. IF HS, ENTER WITH
0223   8D9F                 ;  TIMER STARTED BY PREV BIT. BIT RETURNED IN CARRY.

0225   8D9F  24 FD          SYNBIT BIT MODE          ;KIM OR HS?
0226   8DA1  10 69                 BPL RDBITK        ;KIM
0227   8DA3  20 CA 8D       SYB10  JSR GETTR         ;HS
0228   8DA6  B0 22                 BCS GETTR         ;IF SHORT, GET NEXT TRANS
0229   8DA8  60                    RTS               ;BIT IS ZERO

0231   8DA9  84 FD          START  STY MODE          ;MODE PARM PASSED IN [Y]
0232   8DAB  20 86 8B              JSR ACCESS        ;FIX BASIC WARM START BUG
0233   8DAE  A9 09                 LDA #9
0234   8DB0  20 A5 89              JSR CONFIG        ;PARTIAL I/O CONFIGURATION
0235   8DB3  20 2E 83              JSR ZERCK         ;ZERO THE CHECK SUM
0236   8DB6  20 9C 82              JSR P2SCR         ;MOVE SA TO FE,FF IN PAGE ZERO
0237   8DB9  A2 EC                 LDX #$EC
0238   8DBB  8E 0C A0       STTC   STX VIAPCR        ;TAPE ON
0239   8DBE  60                    RTS

0241   8DBF                 ; GETTR - GET TRANSITION TIME FROM 6532 CLOCK
0242   8DBF                 ; DESTROYS A,Y

0244   8DBF  A9 00          KGETTR LDA #0            ;KIM GETTR - GET FULL CYCLE
0245   8DC1  85 F9                 STA OLD           ;FORCE GETTR POLARITY
0246   8DC3  AD 00 A0       KG100  LDA TAPIN         ;WAIT TIL INPUT LO
0247   8DC6  29 40                 AND #$40
0248   8DC8  D0 F9                 BNE KG100

0250   8DCA  A0 FF          GETTR  LDY #$FF
0251   8DCC  AD 00 A0       NOTR   LDA TAPIN
0252   8DCF  29 40                 AND #$40
0253   8DD1  C5 F9                 CMP OLD
0254   8DD3  F0 F7                 BEQ NOTR          ;NO CHANGE
0255   8DD5  85 F9                 STA OLD
0256   8DD7  AD 06 A4              LDA TIMER
0257   8DDA  8C 15 A4              STY TIM8          ;RESTART CLOCK
0258   8DDD  18                    CLC
0259   8DDE  65 F8                 ADC BDRY
0260   8DE0  60                    RTS

0262   8DE1  24 FD          RDCHTX BIT MODE          ;READ HS OR KIM CHARACTER
0263   8DE3  10 7A                 BPL RDCHT         ;KIM

0265   8DE5                 ; RDBYTH - READ HS BYTE
0266   8DE5                 ; Y DESTROYED, BYTE RETURNED IN CHAR AND A
0267   8DE5                 ; TIME FROM ONE CALL TO NEXT MUST BE LESS THAN
0268   8DE5                 ;    START BIT TIME (TIMER STILL RUNNING)

0270   8DE5  8E 38 A6       RDBYTH STX SCR8          ;SAVE X
0271   8DE8  A2 08                 LDX #8
0272   8DEA  20 CA 8D              JSR GETTR         ;GET START BIT TIME
0273   8DED  B0 14                 BCS RDBH90        ;IF NOT 0, FRAMING ERR
0274   8DEF  20 CA 8D       RDBH10 JSR GETTR         ;GET BIT IN CARRY
0275   8DF2  90 04                 BCC RDASSY
0276   8DF4  20 CA 8D              JSR GETTR         ;BIT IS ONE, WAIT HALF CYC
```

```
LINE # LOC     CODE        LINE

0277  8DF7  38                SEC             ;MAKE SURE '1'
0278  8DF8  66 FC      RDASSY ROR CHAR
0279  8DFA  CA                DEX
0280  8DFB  D0 F2             BNE RDBH10
0281  8DFD  A5 FC             LDA CHAR        ;GET IN ACC
0282  8DFF  AE 38 A6          LDX SCR8        ;RESTORE X
0283  8E02  60                RTS
0284  8E03  24 FD      RDBH90 BIT MODE        ;NO ERR IF NOT IN SYNC
0285  8E05  70 F8             BVS RDBH90-4    ;OR READING WRONG REC
0286  8E07  68                PLA             ;FIX STACK
0287  8E08  68                PLA
0288  8E09  4C 38 8D          JMP FRERR

0290  8E0C             ; RDBITK - READ KIM BIT - X,Y,A DESTROYED, BIT RETURNED IN (

0292  8E0C  20 BF 8D   RDBITK JSR KGETTR      ;WAIT FOR LF
0293  8E0F  B0 FB             BCS RDBITK
0294  8E11  20 BF 8D          JSR KGETTR      ;GET SECOND
0295  8E14  B0 F6             BCS RDBITK
0296  8E16  A2 00             LDX #0
0297  8E18  E8         RDB100 INX             ;COUNT LF FULL CYCLES
0298  8E19  20 BF 8D          JSR KGETTR
0299  8E1C  90 FA             BCC RDB100
0300  8E1E  20 BF 8D          JSR KGETTR      ;GET SECOND
0301  8E21  90 F5             BCC RDB100
0302  8E23  E0 08             CPX #$08        ;GET BIT TO CARRY
0303  8E25  60                RTS

0305  8E26  24 FD      RDBYTX BIT MODE        ;READ HS OR KIM BYTE
0306  8E28  30 BB             BMI RDBYTH      ;HS

0308  8E2A  20 5F 8E   RDBYT  JSR RDCHT       ;READ KIM BYTE INTO CHAR AND A
0309  8E2D  C9 2F             CMP #'/         ;READ ONE CHAR IF LAST
0310  8E2F  F0 2C             BEQ PACKT3      ;SET CARRY AND RETURN
0311  8E31  20 3C 8E          JSR PACKT
0312  8E34  B0 26             BCS RDRTN       ;NON HEX CHAR?
0313  8E36  AA                TAX             ;SAVE MSD
0314  8E37  20 5F 8E          JSR RDCHT
0315  8E3A  86 FC             STX CHAR        ;MOVE MSD TO CHAR
0316  8E3C             ; AND FALL INTO PACKT AGAIN

0318  8E3C             ;PACKT - ASCII HEX TO 4 BITS
0319  8E3C             ;INPUT IN A, OUTPUT IN CHAR AND A, CARRY SET = NON HEX

0321  8E3C  C9 30      PACKT  CMP #$30        ;LT '0'?
0322  8E3E  90 1D             BCC PACKT3
0323  8E40  C9 47             CMP #$47        ;GT 'F' ?
0324  8E42  B0 19             BCS PACKT3
0325  8E44  C9 40             CMP #$40        ;A-F?
0326  8E46  F0 15             BEQ PACKT3      ;40 NOT VALID
0327  8E48  90 03             BCC PACKT1
0328  8E4A  18                CLC
0329  8E4B  69 09             ADC #9
0330  8E4D  2A         PACKT1 ROL A           ;GET LSD INTO LEFT NIBBLE
0331  8E4E  2A                ROL A
```

```
LINE # LOC     CODE        LINE

0386   8E8F   A2 01               LDX #1          ;KIM DELAY CONSTANT (OUTER)
0387   8E91   A4 FD               LDY MODE        ;128 KIM, 0 HS
0388   8E93   10 03               BPL DUMPT1      ;KIM - DO 128 SYNS
0389   8E95   AE 30 A6            LDX TAPDEL      ;HS INITIAL DELAY (OUTER)
0390   8E98   8A           DUMPT1 TXA
0391   8E99   48                  PHA
0392   8E9A   A9 16        DMPT1A LDA #SYN
0393   8E9C   20 0A 8F            JSR OUTCTX
0394   8E9F   88                  DEY
0395   8EA0   D0 F8               BNE DMPT1A      ;INNER LOOP (HS OR KIM)
0396   8EA2   68                  PLA
0397   8EA3   AA                  TAX
0398   8EA4   CA                  DEX
0399   8EA5   D0 F1               BNE DUMPT1
0400   8EA7   A9 2A               LDA #'*         ;WRITE START
0401   8EA9   20 0A 8F            JSR OUTCTX

0403   8EAC   AD 4E A6            LDA ID          ;WRITE ID
0404   8EAF   20 3F 8F            JSR OUTBTX

0406   8EB2   AD 4C A6            LDA SAL         ;WRITE SA
0407   8EB5   20 3C 8F            JSR OUTBCX
0408   8EB8   AD 4D A6            LDA SAH
0409   8EBB   20 3C 8F            JSR OUTBCX

0411   8EBE                ;
0412   8EBE   24 FD               BIT MODE        ;KIM OR HS?
0413   8EC0   10 0C               BPL DUMPT2

0415   8EC2   AD 4A A6            LDA EAL         ;HS. WRITE EA
0416   8EC5   20 3C 8F            JSR OUTBCX
0417   8EC8   AD 4B A6            LDA EAH
0418   8ECB   20 3C 8F            JSR OUTBCX

0420   8ECE   A5 FE        DUMPT2 LDA BUFADL      ;CHECK FOR LAST BYTE
0421   8ED0   CD 4A A6            CMP EAL
0422   8ED3   D0 25               BNE DUMPT4
0423   8ED5   A5 FF               LDA BUFADH
0424   8ED7   CD 4B A6            CMP EAH
0425   8EDA   D0 1E               BNE DUMPT4

0427   8EDC   A9 2F               LDA #'/         ;LAST. WRITE "/"
0428   8EDE   20 0A 8F            JSR OUTCTX
0429   8EE1   AD 36 A6            LDA SCR6        ;WRITE CHECK SUM
0430   8EE4   20 3F 8F            JSR OUTBTX
0431   8EE7   AD 37 A6            LDA SCR7
0432   8EEA   20 3F 8F            JSR OUTBTX

0434   8EED   A9 04               LDA #EOT        ;WRITE TWO EOT'S
0435   8EEF   20 3F 8F            JSR OUTBTX
0436   8EF2   A9 04               LDA #EOT
0437   8EF4   20 3F 8F            JSR OUTBTX

0439   8EF7                DT3E   = * (SET "OK" MARK)
0440   8EF7   4C 41 8D            JMP OKEXIT
```

```
LINE # LOC     CODE        LINE

0442   8EFA   A0 00       DUMPT4 LDY #0          ;GET BYTE
0443   8EFC   B1 FE              LDA (BUFADL),Y
0444   8EFE   20 3C 8F           JSR OUTBCX      ;WRITE IT W/CHK SUM
0445   8F01   E6 FE              INC BUFADL      ;BUMP BUFFER ADDR
0446   8F03   D0 C9              BNE DUMPT2
0447   8F05   E6 FF              INC BUFADH      ;CARRY
0448   8F07   4C CE 8E           JMP DUMPT2
0449   8F0A   24 FD       OUTCTX BIT MODE        ;HS OR KIM?
0450   8F0C   10 48              BPL OUTCHT      ;KIM

0452   8F0E               ;  OUTBTH - NO CLOCK
0453   8F0E               ; A,X DESTROYED
0454   8F0E               ; MUST RESIDE ON ONE PAGE - TIMING CRITICAL
0455   8F0E   A2 09       OUTBTH LDX #9          ;8 BITS + START BIT
0456   8F10   8C 39 A6           STY SCR9
0457   8F13   85 FC              STA CHAR
0458   8F15   AD 02 A4           LDA TAPOUT      ;GET PREV LEVEL
0459   8F18   46 FC       GETBIT LSR CHAR
0460   8F1A   49 08              EOR #TPBIT
0461   8F1C   8D 02 A4           STA TAPOUT      ;INVERT LEVEL
0462   8F1F               ; *** HERE STARTS FIRST HALF CYCLE
0463   8F1F   AC 35 A6           LDY TAPET1
0464   8F22   88          A416   DEY             ;TIME FOR THIS LOOP IS 5Y-1
0465   8F23   D0 FD              BNE A416
0466   8F25   90 12              BCC NOFLIP      ;NOFLIP IF BIT ZERO
0467   8F27   49 08              EOR #TPBIT      ;BIT IS ONE - INVERT OUTPUT
0468   8F29   8D 02 A4           STA TAPOUT
0469   8F2C               ; *** END OF FIRST HALF CYCLE
0470   8F2C   AC 3C A6    B416   LDY TAPET2
0471   8F2F   88          B416B  DEY             ;LENGTH OF LOOP IS 5Y-1
0472   8F30   D0 FD              BNE B416B
0473   8F32   CA                 DEX
0474   8F33   D0 E3              BNE GETBIT      ;GET NEXT BIT (LAST IS 0 START BIT)
0475   8F35   AC 39 A6           LDY SCR9        ; (BY 9 BIT LSR)
0476   8F38   60                 RTS
0477   8F39   EA          NOFLIP NOP             ;TIMING
0478   8F3A   90 F0              BCC B416        ;(ALWAYS)
0479   8F3C               ;
0480   8F3C   20 77 8E    OUTBCX JSR CHKT        ;WRITE HS OR KIM BYTE & CKSUM
0481   8F3F   24 FD       OUTBTX BIT MODE        ;WRITE HS OR KIM BYTE
0482   8F41   30 CB              BMI OUTBTH      ;HS

0484   8F43               ;OUTBTC - OUTPUT ONE KIM BYTE

0486   8F43               OUTBTC =*
0487   8F43   A8          OUTBT  TAY             ;SAVE DATA BYTE
0488   8F44   4A                 LSR A
0489   8F45   4A                 LSR A
0490   8F46   4A                 LSR A
0491   8F47   4A                 LSR A
0492   8F48   20 4B 8F           JSR HEXOUT      ;MORE SIG DIGIT
0493   8F4B               ; FALL INTO HEXOUT

0495   8F4B   29 0F       HEXOUT AND #$0F        ;CVT LSD OF [A] TO ASCII, OUTPUT
0496   8F4D   C9 0A              CMP #$0A
```

```
LINE # LOC     CODE        LINE

0497  8F4F  18                    CLC
0498  8F50  30 02                 BMI HEX1
0499  8F52  69 07                 ADC #$07
0500  8F54  69 30         HEX1    ADC #$30

0502  8F56                ; OUTCHT - OUTPUT ASCII CHAR (KIM)
0503  8F56                ; CLOCK NOT USED
0504  8F56                ; X,Y PRESERVED
0505  8F56                ; MUST RESIDE ON ONE PAGE - TIMING CRITICAL

0507  8F56  8E 38 A6      OUTCHT STX SCR8         ;PRESERVE X
0508  8F59  8C 39 A6             STY SCR9         ;DITTO Y
0509  8F5C  85 FC                STA CHAR
0510  8F5E  A9 FF                LDA #$FF         ;USE FF W/SHIFTS TO COUNT BITS
0511  8F60  48            KIMBIT PHA              ;SAVE BIT CTR
0512  8F61  AD 02 A4             LDA TPOUT        ;GET CURRENT OUTPUT LEVEL
0513  8F64  46 FC                LSR CHAR         ;GET DATA BIT IN CARRY
0514  8F66  A2 12                LDX #18          ;ASSUME 'ONE'
0515  8F68  B0 02                BCS HF
0516  8F6A  A2 24                LDX #36          ;BIT IS ZERO
0517  8F6C  A0 19         HF     LDY #25
0518  8F6E  49 08                EOR #TPBIT       ;INVERT OUTPUT
0519  8F70  8D 02 A4             STA TPOUT
0520  8F73  88            HFP1   DEY              ;PAUSE FOR 138 USEC
0521  8F74  D0 FD                BNE HFP1
0522  8F76  CA                   DEX              ;COUNT HALF CYCS OF HF
0523  8F77  D0 F3                BNE HF
0524  8F79  A2 18         LF     LDX #24          ;ASSUME BIT IS ONE
0525  8F7B  B0 02                BCS LF20
0526  8F7D  A2 0C                LDX #12          ;BIT IS ZERO
0527  8F7F  A0 27         LF20   LDY #39
0528  8F81  49 08                EOR #TPBIT       ;INVERT OUTPUT
0529  8F83  8D 02 A4             STA TPOUT
0530  8F86  88            LFP1   DEY              ;PAUSE FOR 208 USEC
0531  8F87  D0 FD                BNE LFP1
0532  8F89  CA                   DEX              ;COUNT HALF CYCS
0533  8F8A  D0 F3                BNE LF20
0534  8F8C  68                   PLA              ;RESTORE BIT CTR
0535  8F8D  0A                   ASL A            ;DECREMENT IT
0536  8F8E  D0 D0                BNE KIMBIT       ;FF SHIFTED 8X = 00
0537  8F90  AE 38 A6             LDX SCR8
0538  8F93  AC 39 A6             LDY SCR9
0539  8F96  98                   TYA              ;RESTORE DATA BYTE
0540  8F97  60                   RTS


0542  8F98  FF                   .BYT $FF,$FF     ;NOT USED
0542  8F99  FF



0544  8F9A                ; REGISTER NAME PATCH
0545  8F9A                       *=$8F9A
0546  8F9A  53                   .BYT 'S'
```

```
LINE # LOC      CODE        LINE

0547   8F9B  46                    .BYT 'F'
0548   8F9C  41                    .BYT 'A'
0549   8F9D  58                    .BYT 'X'
0550   8F9E  59                    .BYT 'Y'
0551   8F9F  01                    .BYT $01
0552   8FA0                ;
0553   8FA0                        .END

ERRORS = 0000 <0000>
```

```
SYMBOL TABLE

SYMBOL   VALUE

A416     8F22   ACCESS   8B86   B416     8F2C   B416B    8F2F
BDRY     00F8   BUFADH   00FF   BUFADL   00FE   CHAR     00FC
CHECK    00CC   CHKT     8E77   CHKT10   8E84   CKERR    8D3C
CONFIG   89A5   DDRDIG   A401   DDRIN    A002   DDROUT   A403
DIG      A400   DMPT1A   8E9A   DT3E     8EF7   DUMPT    8E87
DUMPT1   8E98   DUMPT2   8ECE   DUMPT4   8EFA   EAH      A64B
EAL      A64A   EOT      0004   EX10     8D4E   EXIT     8D42
FRAME    00FF   FRERR    8D38   GETBIT   8F18   GETTR    8DCA
HEX1     8F54   HEXOUT   8F4B   HF       8F6C   HFP1     8F73
HSBDRY   A632   HWRONG   8CC0   ID       A64E   KBITS    8E63
KG100    8DC3   KGETTR   8DBF   KIMBIT   8F60   KMBDRY   A631
LCERR    8D30   LDT7A    8D34   LF       8F79   LF20     8F7F
LFP1     8F86   LOAD11   8C8B   LOADT    8C78   LOADT2   8C7B
LOADT4   8C7E   LOADT5   8CC4   LOADT6   8CAD   LOADT7   8D09
LOADT8   8D1E   LSTCHR   002F   LT7H     8CDE   LT7HA    8CEF
LT7HB    8D03   LT7HC    8CFA   LT8A     8D1E   MODE     00FD
NGEXIT   8D3E   NHERR    8D38   NOFLIP   8F39   NONHEX   00FF
NOTR     8DCC   OKEXIT   8D41   OLD      00F9   OUTBCX   8F3C
OUTBT    8F43   OUTBTC   8F43   OUTBTH   8F0E   OUTBTX   8F3F
OUTCHT   8F56   OUTCTX   8F0A   P2SCR    829C   PACKT    8E3C
PACKT1   8E4D   PACKT2   8E53   PACKT3   8E5D   RDASSY   8DF8
RDB100   8E18   RDBH10   8DEF   RDBH90   8E03   RDBITK   8E0C
RDBYT    8E2A   RDBYTH   8DE5   RDBYTX   8E26   RDCHK    8E74
RDCHT    8E5F   RDCHTX   8DE1   RDRTN    8E5C   SAH      A64D
SAL      A64C   SCR6     A636   SCR7     A637   SCR8     A638
SCR9     A639   START    8DA9   STTC     8DBB   SY100    8D69
SYB10    8DA3   SYN      0016   SYNBIT   8D9F   SYNC     8D52
SYNC10   8D8B   SYNC5    8D76   TAPDEL   A630   TAPET1   A635
TAPET2   A63C   TAPIN    A000   TAPOUT   A402   TIM8     A415
TIMER    A406   TPBIT    0008   TPOUT    A402   USRREQ   8D4B
VIAACR   A00B   VIAPCR   A00C   ZERCK    832E
END OF ASSEMBLY

<
```