

AIM 65

MICROCOMPUTER

**MONITOR
PROGRAM
LISTING**

AIM 65



Rockwell International


```

LINE # LOC CODE LINE
0025 0000 ; *****
0026 0000 ; * USER 6522 ADDRESSES (A000-A00F) *
0027 0000 ; *****
0028 0000 ; *=$A000
0029 A000 UIDRB *=$+1 ; DATA REG B
0030 A001 UIDRAH *=$+1 ; DATA REG A
0031 A002 UIDRBL *=$+1 ; DATA DIR REG B
0032 A003 UIDRA *=$+1 ; DATA DIR REG A
0033 A004 UT1L *=$+1 ; TIMER 1 COUNTER LOW
0034 A005 UT1CH *=$+1 ; TIMER 1 COUNTER HIGH
0035 A006 UT1LL *=$+1 ; TIMER 1 LATCH LOW
0036 A007 UT1LH *=$+1 ; TIMER 1 LATCH HIGH
0037 A008 UT2L *=$+1 ; TIMER 2 LATCH & COUNTER LOW
0038 A009 UT2H *=$+1 ; TIMER 2 COUNTER HIGH
0039 A00A USR *=$+1 ; SHIFT REGISTER
0040 A00B UACR *=$+1 ; AUX CONTROL REGISTER
0041 A00C UPCR *=$+1 ; PERIPHERAL CONTROL REGISTER
0042 A00D UIFR *=$+1 ; INTERRUPT FLAG REGISTER
0043 A00E UIER *=$+1 ; INTERRUPT ENABLE REGISTER
0044 A00F UIDRA *=$+1 ; DATA REGISTER A
    
```

```

LINE # LOC CODE LINE
0046 A010 ASSEM =$D000 ; ASSEMBLER ENTRY
0047 A010 BASIEN =$B000 ; BASIC ENTRY (COLD)
0048 A010 BASIRE =$R003 ; BASIC REENTRY (WARM)

; MONITOR RAM
; TEXT EDITOR EQUATES (PAGE 0)
; OVERLAPS TABUF2+50 (TAPE OUTPUT BUFFER $AD-$FF)
*=$00DF
NOWLN *=$+2 ; CURRENT LINE
BOTLN *=$+2 ; LAST ACTIVE , SO FAR
TEXT *=$+2 ; LIMITS OF BUFFER (START)
END *=$+2 ; LIMITS OF BUFFER (END)
SAVE *=$+2 ; USED BY REPLACE
OLDLEN *=$+1 ; ORIG LENGTH
LENGTH *=$+1 ; NEW LENGTH
STRING *=$+20 ; FIND STRING

*=$0100
; BREAKPOINTS AND USER I/O HANDLERS
BRK *=$+8 ; BRK LOCATIONS
UIN *=$+2 ; USER INPUT HANDLER (VECTOR)
UOUT *=$+2 ; USER OUTPUT HANDLER (VECTOR)

; UNUSED KEYS TO GO TO USER ROUTINE
KEYF1 *=$+3 ; USER PUTS A JMP INSTRUCTION TO...
KEYF2 *=$+3 ; GO TO HIS ROUTINE ON EITHER KEY...
KEYF3 *=$+3 ; ENTRY

; EQUATES FOR DISASSEMBLER (PAGE 1)
*=$0116 ; SAME AS TAPE BUFFER I/O (TABUFF)
FORMA *=$+1
LMNEM *=$+1
RMNEM *=$+14

; EQUATES FOR MNEMONIC ENTRY
MOVAD *=$+8
TYPE *=$+2
TMASK1 =MOVAD+1
TMASK2 =MOVAD+1
CH *=$+3
ADFLD *=$+20
HISTH =$A42E ; SHARE WITH NAME & HIST
BYTESM =HISTH+1
TEMPX =HISTH+3
TEMPA =HISTH+5
OPCODE =HISTH+6
CODFLG =HISTH+9

; *****
; * 6532 ADDRESSES (A400-A7FF) *
; *****
*=$A400
MONRAM *=$
; JUMP VECTORS
IRQV4 *=$+2 ; IRQ AFTER MONITOR (NO BRK)
    
```

```

LINE # LOC CODE LINE
0101 A402 NMIV2 **+2 ;NMI
0102 A404 IRQV2 **+2 ;IRQ

; I/O DEVICES
0104 A406 ;I/O DEVICES
0105 A406 DILINK **+2 ;DISPL LINKAGE (TO ECHO TO DISP)
0106 A408 TSPEED **+1 ;TAPE SPEED (C7,5B,5A)
0107 A409 GAP **+1 ;TIMING GAP BETWEEN BLOCKS
0108 A40A ;END OF USER ALTERABLE LOCATIONS
0109 A40A NPUL **+1 ;# OF HALF PULSES...
0110 A40B TIMG **+3 ;FOR TAPE
0111 A40E REGF **+1 ;REGS FLG FOR SINGLE STEP MODE
0112 A40F DISFLG **+1 ;DISASSEM FLG FOR SINGLE STEP MODE
0113 A410 BKFLG **+1 ;ENABLE OR DIS BREAKPOINTS
0114 A411 PRIFLG **+1 ;ENABLE OR DIS PRINTER
0115 A412 INFLG **+1 ;INPUT DEVICE
0116 A413 OUTFLG **+1 ;OUTPUT DEVICE
0117 A414 HISTP **+1 ;HISTORY PTR (SINGLE STEP) (Y)
0118 A415 CURPO2 **+1 ;DISPLAY POINTER
0119 A416 CURPOS **+1 ;PRINTER POINTER
0120 A417 CNTH30 **+1 ;BAUD RATE &...
0121 A418 CNTL30 **+1 ;DELAY FOR TTY
0122 A419 COUNT **+1 ;# OF LINES (0-99)
0123 A41A S1 **+2 ;START ADDRESS
0124 A41C ADDR **+2 ;END ADDRESS
0125 A41E CKSUM **+2 ;CHECKSUM
0126 A420 S2 =BKS+6 ;VERTICAL COUNT (ONLY ON DUMP)

; MONITOR REGISTERS
0128 A420 ; MONITOR REGISTERS
0129 A420 SAVFS **+1 ;STATUS
0130 A421 SAVA **+1 ;ACCUM
0131 A422 SAVX **+1 ;X REG
0132 A423 SAVY **+1 ;Y REG
0133 A424 SAVS **+1 ;STACK POINTER
0134 A425 SAVPC **+2 ;PROGR COUNTER

; WORK AREAS FOR PAGE ZERO SIMULATION
0136 A427 ; WORK AREAS FOR PAGE ZERO SIMULATION
0137 A427 ;SIMULATE LDA (NNNN),Y ;WHERE NNNN IS ABSOLUTE
0138 A427 STIY **+3 ;STA NM,Y
0139 A42A CPIY **+3 ;CMP NM,Y OR LDA NM,Y
0140 A42D **+1 ;RTS
0141 A42E LDYI =CPIY ;LDA NM,Y

; VARIABLES FOR TAPE
0143 A42E ; VARIABLES FOR TAPE
0144 A42E NAME **+6 ;FILE NAME
0145 A434 TAPIN **+1 ;IN FLG (TAPE 1 OR 2)
0146 A435 TAPOUT **+1 ;OUT FLG (TAPE 1 OR 2)
0147 A436 TAPTR **+1 ;TAPE BUFF POINTER
0148 A437 TAPTR2 **+1 ;TAPE OUTPUT BUFF PTR
0149 A438 HIST =NAME ;FOUR LAST ADDR + NEXT (SINGL STEP)
0150 A438 BLK =*0115 ;BLOCK COUNT
0151 A438 TABUFF =*0116 ;TAPE BUFFER (I/O)
0152 A438 BLKO =*0168 ;OUTPUT BLOCK COUNT
0153 A438 TABUF2 =*00AD ;OUTPUT BUFF WHEN ASSEMB (PAGO)
0154 A438 DIBUFF **+40 ;DISPLAY BUFFER

```

```

LINE # LOC CODE LINE
0156 A460 ;VARIABLES USED IN PRINTING
0157 A460 IBUFM **+20 ;PRINTER BUFFER
0158 A474 IDIR **+1 ;DIRECTION == 0=>+, FF=>-
0159 A475 ICOL **+1 ;COLUMN LEFTMOST=0,RIGHTMOST=4
0160 A476 IOFFST **+1 ;OFFSET 0=LEFT DGT,1=RIGHT DGT
0161 A477 IDOT **+1 ;# OF LAST DOT ENCOUNTERED
0162 A478 IOU7L **+1 ;LOWER 8 OUTPUTS(8 COLS ON RIGHT)
0163 A479 IOU7U **+1 ;UPPER 2 DIGITS
0164 A47A IBITL **+1 ;1 BIT MSK FOR CURRENT OUTPUT
0165 A47B IBITU **+1 ;1 BIT MSK FOR CURRENT ROW
0166 A47C IMASK **+1 ;MSK FOR CURRENT ROW
0167 A47D JUMP **+2 ;INDIR & ADDR OF TABL FOR CURR ROW

; VARIABLES FOR KEYBOARD
0169 A47F ; VARIABLES FOR KEYBOARD
0170 A47F ROLLFL **+1 ;SAVE LAST STROBE FOR ROLLOVER
0171 A480 KMASK =CFIY ;TO MASK OFF CTRL OR SHIFT
0172 A480 STBKEY =CFIY+1 ;STROBE KEY (1-8 COLUMNS)

; I/O ASSIGNMENT
0174 A480 ; I/O ASSIGNMENT
0175 A480 *=$A480
0176 A480 DRA2 **+1 ;DATA REG A
0177 A481 DDRA2 **+1 ;DATA DIR REG A
0178 A482 DRB2 **+1 ;DATA REG B
0179 A483 DDRB2 **+1 ;DATA DIR REG B

; WRITE EDGE DETECT CONTROL (NOT USED BECAUSE KB)
0181 A484 ; WRITE EDGE DETECT CONTROL (NOT USED BECAUSE KB)
0182 A484 *=$A484
0183 A484 DNPA7 **+1 ;DISABLE PA7 INT ,NEG EDG DET
0184 A485 DFP A7 **+1 ;DIS PA7 INT ,POS EDGE DETE
0185 A486 ENPA7 **+1 ;ENA PA7 INT ,NEG EDG DET
0186 A487 EPPA7 **+1 ;ENA PA7 INT ,POS EDG DET

; READ AND CLEAR INTERRUPT
0188 A488 ; READ AND CLEAR INTERRUPT
0189 A488 *=$A485
0190 A485 RINT **+1 ;BIT 7=TIMER FLG , BIT 6=PA7 FLG

; TIMER INTERRUPT
0192 A486 ; TIMER INTERRUPT
0193 A486 *=$A494
0194 A494 ;WRITE COUNT TO INTERVAL TIMER
0195 A494 ;INTERRUPT DISABLE FOR THESE ADDRS
0196 A494 DIV1 **+1 ;DIV BY 1 (DISABLE);ADD 8 TO ENA
0197 A495 DIV8 **+1 ;DIV BY 8 (DIS) ; ADD 8 TO ENA
0198 A496 DIV64 **+1 ;DIV BY 64 (DIS) ; ADD 8 TO ENA
0199 A497 DI1024 **+1 ;DIV BY 1024 (DIS) ; ADD 8 TO ENA

0201 A498 ; *****
0202 A498 ; * 6522 ADDRESSES (MONIT) (A800-ARFF) *
0203 A498 ; *****
0204 A498 *=$A800
0205 A800 DRB **+1 ;DATA REG B
0206 A801 DRAH **+1 ;DATA REG A
0207 A802 DDRB **+1 ;DATA DIR REG B
0208 A803 DDRA **+1 ;DATA DIR REG A
0209 A804 T1L **+1 ;TIMER 1 COUNTER LOW
0210 A805 T1CH **+1 ;TIMER 1 COUNTER HIGH

```

MONITOR VARIABLES PA00-J001A.....PAGE 0006

```

LINE # LOC CODE LINE
0211 A806 T1LL **+1 ;TIMER 1 LATCH LOW
0212 A807 T1LH **+1 ;TIMER 1 LATCH HIGH
0213 A808 T2L **+1 ;TIMER 2 LATCH & COUNTER LOW
0214 A809 T2H **+1 ;TIMER 2 COUNTER HIGH
0215 A80A SR **+1 ;SHIFT REGISTER
0216 A80B ACR **+1 ;AUX CONTROL REGISTER
0217 A80C PCR **+1 ;PERIPHERAL CONTROL REGISTER
0218 A80D IFR **+1 ;INTERRUPT FLAG REGISTER
0219 A80E IER **+1 ;INTERRUPT ENABLE REGISTER
0220 A80F DRA **+1 ;DATA REGISTER A

0222 A810 ;DEFINE I/O CONTROL FOR PCR (CA1,CA2,CB1,CB2)
0223 A810 DATIN =%0E ;DATA IN CA2=1
0224 A810 DATOUT =%0C ;DATA OUT CA2=0
0225 A810 PRST =%00 ;PRINT START (CB1) ,NEG DETEC
0226 A810 SP12 =%01 ;STROBE P1,P2 (CA1) ,POS DETEC
0227 A810 MON =%0C ;MOTOR ON (CB2=0)
0228 A810 MOFF =%E0
0229 A810 ;MSKS TO OBTAIN EACH INTERRUPT
0230 A810 MPRST =%10 ;INT FLG FOR CB1
0231 A810 MSP12 =%02 ;INT FLG FOR CA1
0232 A810 MT2 =%20 ;INT FLG FOR T2

0234 A810 ;DEFINE I/O CONTROL FOR ACR (TIMERS,SR)
0235 A810 PRIME =1700 ; PRINTING TIME =1.7 MSEC
0236 A810 DEBTIM =5000 ; DEBOUNCE TIME (5 MSEC)
0237 A810 T2I =%00 ;T2 AS ONE SHOT (PRI,KB,TTY,TAPE)
0238 A810 T1I =%00 ;T1 AS ONE SHOT,PR7 DIS (TAPES)
0239 A810 T1FR =%C0 ;T1 IN FREE RUNNING (TAPE)

0241 A810 ; *****
0242 A810 ; * DISPLAY (AC00-AFFF) *
0243 A810 ; *****
0244 A810 ; REGISTERS FOR DISPLAY (6520)
0245 A810 *=%AC00
0246 AC00 RA **+1 ;REGISTER A
0247 AC01 CRA **+1 ;CONTROL REG A
0248 AC02 RB **+1 ;REG B
0249 AC03 CRB **+1 ;CONTROL REG B

0251 AC04 ;CHR 00-03 ENA BY %AC04-AC07
0252 AC04 ;CHR 04-07 ENA BY %AC08-AC0B
0253 AC04 ;CHR 08-11 ENA BY %AC10-AC13
0254 AC04 ;CHR 12-15 ENA BY %AC20-AC23
0255 AC04 ;CHR 16-19 ENA BY %AC40-AC43

0257 AC04 NULLC =%FF
0258 AC04 CR =%0D
0259 AC04 LF =%0A
0260 AC04 ESCAPE =%1B
0261 AC04 RUB =%08
0262 AC04 ERS =%BD
0263 AC04 .FILE A1

```

MONITOR MESSAGES PA00-J001A.....PAGE 0007

```

LINE # LOC CODE LINE
0265 AC04 ; E=ENTER EDITOR
0266 AC04 ; T=RE-ENTER EDITOR TO RE-EDIT SOURCE
0267 AC04 ; R=SHOW REGISTERS
0268 AC04 ; M=DISPLAY MEMORY
0269 AC04 ; =SHOW NEXT 4 ADDRESSES
0270 AC04 ; G=GO AT CURRENT P.C. (COUNT)
0271 AC04 ; /=ALTER CURRENT MEMORY
0272 AC04 ; L=LOAD OBJECT
0273 AC04 ; D=DUMP OBJECT
0274 AC04 ; N=ASSEMBLE
0275 AC04 ; *=ALTER P.C.
0276 AC04 ; A=ALTER ACCUMULATOR
0277 AC04 ; X=ALTER X REGISTER
0278 AC04 ; Y=ALTER Y REGISTER
0279 AC04 ; P=ALTER PROCESSOR STATUS
0280 AC04 ; S=ALTER STACK POINTER
0281 AC04 ; B=SET BREAK ADDR
0282 AC04 ; ?=SHOW BREAK ADDRESSES
0283 AC04 ; #=CLEAR BREAK ADDRESSES
0284 AC04 ; H=SHOW TRACE HISTORY STACK
0285 AC04 ; V=TOGGLE REGISTER PRINT WITH DIS.
0286 AC04 ; Z=TOGGLE DISASSEMBLER TRACE
0287 AC04 ; \=TURN ON/OFF PRINTER
0288 AC04 ; =ADV PAPER
0289 AC04 ; I=MNEMONIC ENTRY
0290 AC04 ; K=DISASSEMBLE MEMORY
0291 AC04 ; 1=TOGGLE TAPE 1 CONTRL (ON OR OFF)
0292 AC04 ; 2=TOGGLE TAPE 2 CONTRL
0293 AC04 ; 3=VERIFY CKSUM FOR TAPES
0294 AC04 ; 4=ENABLE BREAKS
0295 AC04 ; 5=BASIC ENTRY (COLD)
0296 AC04 ; 6=BASIC REENTRY (WARM)
0297 AC04 ;
0298 AC04 ; FOLLOWING KEYS ARE UNUSED BUT 'HOOKS'
0299 AC04 ; ARE PROVIDED IN LOCATIONS 010C-0114
0300 AC04 ;
0301 AC04 ; KEYF1,KEYF2,KEYF3

0303 AC04 *=%E000
0304 E000 ;ALL MSGS HAVE MSB=1 OF LAST CHAR TO END IT
0305 E000 46 52 M1 .BYT 'FROM',ERS
0305 E004 BD
0306 E005 54 4F M3 .BYT 'TO',ERS
0306 E007 BD
0307 E008 20 2A M4 .BYT ' **** PS AA XX YY S',%D3
0307 E01B D3
0308 E01C 4D 4F M5 .BYT 'MORE',%BF
0308 E020 BF
0309 E021 4F 4E M6 .BYT 'ON',%A0 ;'ON '
0309 E023 A0
0310 E024 4F 46 M7 .BYT 'OF',%C6 ;'OFF '
0310 E026 C6
0311 E027 42 52 M8 .BYT 'BR',%CB ;'BRK'
0311 E029 CB

```

| LINE # | LOC | CODE | LINE | |
|--------|------|----------|-------|--------------------------------------------------|
| 0312 | E02A | 49 4E | M9 | .BYT 'IN',EQS |
| 0312 | E02C | BD | | |
| 0313 | E02D | 4F 55 54 | M10 | .BYT 'OUT',EQS |
| 0313 | E030 | BD | | |
| 0314 | E031 | 20 4D | M11 | .BYT ' MEM FAIL', \$A0 |
| 0314 | E03A | A0 | | |
| 0315 | E03B | 20 50 | M12 | .BYT ' PRINTER DOW', \$CE |
| 0315 | E047 | CE | | |
| 0316 | E048 | 20 53 | TMSG0 | .BYT ' SRCH' |
| 0317 | E04D | 20 46 | TMSG1 | .BYT ' F',EQS |
| 0317 | E04F | BD | | |
| 0318 | E050 | 54 | TMSG2 | .BYT 'T',EQS |
| 0318 | E051 | BD | | |
| 0319 | E052 | A0 | TMSG3 | .BYT \$A0,\$C5,\$D2,\$D2 ;PRINT ' ERROR' ,MSB=1 |
| 0319 | E053 | C5 | | |
| 0319 | E054 | D2 | | |
| 0319 | E055 | D2 | | |
| 0320 | E056 | CF | | .BYT \$CF,\$D2,\$A0,\$A0,\$A0,\$A0,\$A0,\$A0,',' |
| 0320 | E057 | D2 | | |
| 0320 | E058 | A0 | | |
| 0320 | E059 | A0 | | |
| 0320 | E05A | A0 | | |
| 0320 | E05B | A0 | | |
| 0320 | E05C | A0 | | |
| 0320 | E05D | A0 | | |
| 0320 | E05E | 3B | | |
| 0321 | E05F | 41 | TMSG5 | .BYT 'A',EQS |
| 0321 | E060 | BD | | |
| 0322 | E061 | 42 4C | TMSG6 | .BYT 'BLK=', \$A0 |
| 0322 | E065 | A0 | | |
| 0323 | E066 | A0 | TMSG7 | .BYT \$A0,\$C0,\$CF,\$C1,\$C4,',' |
| 0323 | E067 | CC | | |
| 0323 | E068 | CF | | |
| 0323 | E069 | C1 | | |
| 0323 | E06A | C4 | | |
| 0323 | E06B | 3B | | |
| 0324 | E06C | 45 44 | EMSG1 | .BYT 'EDITO', \$D2 ;EDITOR MESSAGES |
| 0324 | E071 | B2 | | |
| 0325 | E072 | 45 4E | EMSG2 | .BYT 'EN', \$C4 |
| 0325 | E074 | C4 | | |

| LINE # | LOC | CODE | LINE | |
|--------|------|----------|-------|-------------------------------------------------------|
| 0327 | E075 | | | ; VECTORS COME HERE FIRST AFTER JUMP THRU FFFA-FFFF |
| 0328 | E075 | 6C 02 A4 | NMIV1 | JMP (NMIV2) ;NMIV2 IS A VECTOR TO NMIV3 |
| 0329 | E078 | 6C 04 A4 | IRQV1 | JMP (IRQV2) ;IRQV2 IS A VECTOR TO IRQV3 |
| 0331 | E07B | | | ; SINGLE STEP ENTRY POINT (NMI) |
| 0332 | E07B | BD 21 A4 | NMIV3 | STA SAVA ;SAVE ACCUM |
| 0333 | E07E | 68 | | FLA |
| 0334 | E07F | BD 20 A4 | | STA SAVPS ;SAVE PROCESSOR STATUS |
| 0335 | E082 | D8 | | CLD |
| 0336 | E083 | BE 22 A4 | | STX SAVX ;SAVE X |
| 0337 | E086 | BC 23 A4 | | STY SAVY |
| 0338 | E089 | 68 | | FLA |
| 0339 | E08A | BD 25 A4 | | STA SAVPC ;PROGRAM COUNTER |
| 0340 | E08D | 68 | | FLA |
| 0341 | E08E | BD 26 A4 | | STA SAVPC+1 |
| 0342 | E091 | BA | | TSX |
| 0343 | E092 | BE 24 A4 | | STX SAVS ;GET STACK PTR & SAVE IT |
| 0344 | E095 | | | ;TRACE THE ADDRESS |
| 0345 | E095 | AC 14 A4 | | LDY HISTP ;GET POINTER TO HISTORY STACK |
| 0346 | E098 | AD 26 A4 | | LDA SAVPC+1 ;SAVE HALT ADDR IN HISTORY STACK |
| 0347 | E09B | 99 2E A4 | | STA HIST,Y |
| 0348 | E09E | AD 25 A4 | | LDA SAVPC |
| 0349 | E0A1 | 99 2F A4 | | STA HIST+1,Y |
| 0350 | E0A4 | 20 8B E6 | | JSR NHIS ;UPDATE POINTER |
| 0351 | E0A7 | AD 10 A4 | | LDA BKFLG ;SOFT BREAKS ON? |
| 0352 | E0AA | F0 08 | | BEQ NMIS ;NO ,DONT CHCK BRKPOINT LIST |
| 0353 | E0AC | 20 6B E7 | | JSR CKR ;CHECK BREAKPOINT LIST |
| 0354 | E0AF | 90 03 | | BCC NMIS ;DID NOT HIT BREAKPOINT |
| 0355 | E0B1 | 4C 7F E1 | NM14 | JMP IRQ2 ;HIT A BREAK-TRAP TO MONITOR |
| 0356 | E0B4 | 20 90 E7 | NM15 | JSR DONE ;COUNT =0 ? |
| 0357 | E0B7 | F0 F8 | | BEQ NM14 ;YES,TRAP TO MONITOR |
| 0358 | E0B9 | 20 07 E9 | | JSR RCHEK ;CHK IF HE WANTS TO INTERR |
| 0359 | E0BC | 4C 6D E2 | | JMP GOBK ;NOT DONE-RESUME EXECUTION |
| 0361 | E0BF | | | ; POWER UP AND RESET ENTRY POINT (RST TRANSFERS HERE) |
| 0362 | E0BF | D8 | RSET | CLD ;CLEAR DEC MODE |
| 0363 | E0C0 | 78 | | SEI ;DISABLE INTERRUPT |
| 0364 | E0C1 | A2 FF | | LDX #\$FF ;INIT STACK PTR |
| 0365 | E0C3 | 9A | | TXS |
| 0366 | E0C4 | BE 24 A4 | | STX SAVS ;ALSO INIT SAVED STACK PTR |
| 0367 | E0C7 | | | ; INITIALIZE 6522 |
| 0368 | E0C7 | A2 0E | | LDX #14 |
| 0369 | E0C9 | BD 43 E7 | RS1 | LDA INTAB1,X ;PB1-PB0,PA7-PA0 FOR PRNTR |
| 0370 | E0CC | 9D 00 A8 | | STA DRB,X ;PB2=TT0,PB6=TTI |
| 0371 | E0CF | CA | | DEX ;PB4-PB5=TAFE CONTROL,PB7=DATA |
| 0372 | E0D0 | 10 F7 | | BPL RS1 ;PB3 =SWITCH KB/TTY |
| 0373 | E0D2 | | | ; INITIALIZE 6532 |
| 0374 | E0D2 | A2 03 | | LDX #3 ;PORTS USED FOR KB |
| 0375 | E0D4 | BD 52 E7 | RS2 | LDA INTAB2,X ;PA0-PA7 AS OUTPUT |
| 0376 | E0D7 | 9D 80 A4 | | STA DKA2,X ;PB0-PB7 AS INPUT |
| 0377 | E0DA | CA | | DEX |
| 0378 | E0DB | 10 F7 | | BPL RS2 |
| 0379 | E0DD | | | ; INITIALIZE MONITOR RAM (6532) |
| 0380 | E0DD | AD 56 E7 | | LDA INTAB3 ;CHECK IF NMIV2 HAS BEEN CHANGED |

```

LINE # LOC      CODE      LINE
0381 E0E0 CD 02 A4      CMP NMIV2      ; IF IT HAS THEN ASSUME A COLD
0382 E0E3 D0 0C      BNE RS3A      ; START AND INITIALIZE EVERYTHING
0383 E0E5 AD 57 E7      LDA INTAR3+1
0384 E0E8 CD 03 A4      CMP NMIV2+1
0385 E0E9 D0 04      BNE RS3A
0386 E0ED A2 10      LDX #16      ; THEY ARE EQUAL , IT'S A WARM RESET
0387 E0EF D0 02      BNE RS3
0388 E0F1 A7 00      RS3A LDX #0      ; INIT EVERYTHING (POWER UP)
0389 E0F3 BD 56 E7      RS3  LDA INTAR3,X
0390 E0F6 9D 02 A4      STA NMIV2,X
0391 E0F9 EB          INX
0392 E0FA E0 15      CFX #21
0393 E0FC E0FC 90 F5      BCC RS3
0394 E0FE          ; INITIALIZE DISPLAY (6520)
0395 E0FE A9 00      LDA #0      ; SET CONTR REG FOR DATA DIR REG
0396 E100 A2 01      LDX #1
0397 E102 20 13 E1      JSR SETREG
0398 E105 A9 FF      LDA #$FF      ; SET DATA DIR REG FOR OUTPUT
0399 E107 CA          DEX
0400 E108 20 13 E1      JSR SETREG
0401 E10B A9 04      LDA #$04      ; SET CONTR REG FOR PORTS
0402 E10D EB          INX
0403 E10E 20 13 E1      JSR SETREG
0404 E111 D0 07      BNE RS3B
0405 E113 9D 00 AC      SETREG STA RA,X
0406 E116 9D 02 AC      STA RB,X
0407 E119 60          RTS
0408 E11A 58          RS3B CLI      ; CLEAR INTERRUPT

0410 E11B          ; KB/TTY SWITCH TEST AND BIT RATE MEASUREMENT
0411 E11B A9 08      LDA #$08
0412 E11D 2C 00 AB      RS4 BIT DRB      ; PB3=SWITCH KB/TTY
0413 E120 D0 22      BNE RS7      ; A*M ,PB6-> V (OVERFLOW FLG)
0414 E122 70 F9      BVS RS4      ; BRNCH ON KB
0415 E124 A9 FF      LDA #$FF      ; START BIT=PB6=0?
0416 E126 8D 09 AB      STA T2H      ; YES , INITIALIZE TIMER T2
0417 E129 2C 00 AB      RS5 BIT DRB
0418 E12C 50 FB      RVC RS5      ; END OF START BIT ?
0419 E12E AD 09 AB      LDA T2H      ; NO , WAIT UNTIL PB6 BACK TO 1
0420 E131 49 FF      EOR #$FF      ; STORE TIMING
0421 E133 8D 17 A4      STA CNTH30      ; COMPLEMENT
0422 E136 AD 08 AB      LDA T2L
0423 E139 49 FF      EOR #$FF
0424 E13B 20 7C FE      JSR PATCH1
0425 E13E 20 13 EA      RS6 JSR CRLW      ; ADJUST IT
0426 E141 4C 72 FF      JMP FAT21      ; CLEAR DISPLAY
0427 E144 A2 13 F1      RS7 LDX #19      ; CLEAR HARWARE CURSORS
0428 E146 BA          RS8 TXA
0429 E147 48          PHA
0430 E148 A9 00      LDA #0
0431 E14A 20 7B EF      JSR OUTDD1
0432 E14D 68          PLA
0433 E14E AA          TAX
0434 E14F CA          DEX
0435 E150 10 F4      BPL RS8
    
```

```

LINE # LOC      CODE      LINE
0436 E152 30 EA      BMI RS6

0438 E154          ; BRK INSTR (00) OR IRQ ENTRY POINT
0439 E154 8D 21 A4      IRQV3 STA SAVA
0440 E157 68          PLA
0441 E158 48          PHA
0442 E159 29 10      AND #$10      ; GET STATUS
0443 E15B D0 06      BNE IRQ1      ; SEE IF 'BRK' , ISOLATE B FLG
0444 E15D AD 21 A4      LDA SAVA      ; TRAP WAS CAUSED BY 'BRK' INSTRUC
0445 E160 6C 00 A4      JMP (IRQV4)    ; TRAP CAUSED BY IRQ SO TRANSFER
0446 E163          ; IS 'BRK' INSTR , SHOW PC & DATA
0447 E163          ; PC IS OFF BY ONE , SO ADJUST IT
0448 E163 68          IRQ1 PLA
0449 E164 8D 20 A4      STA SAVPS      ; SAVE PROCESSOR STATUS
0450 E167 8E 22 A4      STX SAVX
0451 E16A 8C 23 A4      STY SAVY
0452 E16D D8          CLD
0453 E16E 68          PLA
0454 E16F 38          SEC          ; PROGR CNTR
0455 E170 E9 01      SBC #1        ; SUBTRACT ONE FROM RETURN ADDR
0456 E172 8D 25 A4      STA SAVPC
0457 E175 68          PLA
0458 E176 E9 00      SRC #0
0459 E178 8D 26 A4      STA SAVPC+1
0460 E17B BA          TSX          ; GET STACK PTR & SAVE IT
0461 E17C 8E 24 A4      STX SAVS
0462 E17F          ; SHOW PC AND DATA
0463 E17F 20 61 F4      IRQ2 JSR REGQ      ; SHOW NEXT INTRUCTION & CONTINUE
    
```

| LINE # | LOC | CODE | LINE |
|--------|------|----------|---------------------------------------------------|
| 0465 | E1B2 | | ; THIS ROUTINE WILL GET A CHR WITH 'C ' FROM |
| 0466 | E1B2 | | ; KB/TTY & THEN WILL GO TO THE RESPECTIVE COMMAND |
| 0467 | E1B2 | 4C 59 FF | START JMP PAT19 ; CLEAR DEC MODE & (CR) |
| 0468 | E1B5 | A9 BC | STA1 LDA #*BC ; 'C' CHR WITH MSB=1 FOR DISP |
| 0469 | E1B7 | 20 7A E9 | JSR OUTPUT |
| 0470 | E1B8 | 20 96 FE | JSR RED1 ; GET CHR & ECHO FROM KB/TTY |
| 0471 | E1B0 | 48 | PHA |
| 0472 | E1B8 | A9 3E | LDA #') |
| 0473 | E190 | 20 7A E9 | JSR OUTPUT |
| 0474 | E193 | 68 | PLA ; SCAN LIST OF CMDS FOR ENTERED CHR |
| 0475 | E194 | A2 20 | LDX #MCNT ; COUNT OF COMMANDS |
| 0476 | E196 | DD C4 E1 | MCM2 CMP COMB, X ; CHECK NEXT COMMAND IN LIST |
| 0477 | E199 | F0 11 | BEQ MCM3 ; MATCH , SO PROCESS THIS COMMAND |
| 0478 | E198 | CA | DEX |
| 0479 | E19C | 10 FB | BPL MCM2 |
| 0480 | E19E | | ; IS BAD COMMAND |
| 0481 | E19E | 20 D4 E7 | JSR RM |
| 0482 | E1A1 | D8 | COMIN CLD |
| 0483 | E1A2 | 20 FE E8 | JSR LL |
| 0484 | E1A5 | AE 24 A4 | LDX SAVS |
| 0485 | E1A8 | 9A | TXS |
| 0486 | E1A9 | 4C 82 E1 | JMP START |
| 0487 | E1AC | | ; HAVE VALID COMMAND |
| 0488 | E1AC | BA | MCM3 TXA ; CONVERT TO WORD (MULT BY 2) |
| 0489 | E1AD | 0A | ASL A ; 2 BYTES (ADDR) |
| 0490 | E1AE | AA | TAX |
| 0491 | E1AF | BD E5 E1 | LDA MONCOM, X ; GET ADDRESS OF COMMAND PROCESSOR |
| 0492 | E1B2 | BD 7D A4 | STA JUMP |
| 0493 | E1B5 | BD E6 E1 | LDA MONCOM+1; X |
| 0494 | E1B8 | BD 7E A4 | STA JUMP+1 |
| 0495 | E1BB | 20 C1 E1 | JSR JMPR ; CMD PROCESSORS CAN EXIT WITH 'RTS' |
| 0496 | E1BE | 4C 82 E1 | JMP START |
| 0497 | E1C1 | 6C 7D A4 | JMPR JMP (JUMP) ; GO TO COMMAND |
| 0499 | E1C4 | | ; VALID COMMANDS |
| 0500 | E1C4 | | MCNT=32 ; COUNT |
| 0501 | E1C4 | 45 54 | COMB .BYT 'ETRMG/LDN*AXYPS' |
| 0502 | E1D4 | 42 3F | .BYT 'B?#HUVZIK123456CJ', #5E |
| 0502 | E1E4 | 5E | |
| 0504 | E1E5 | 39 F6 | MONCOM .WOR EDIT, REENTR, REG, MEM, GO |
| 0504 | E1E7 | CF F6 | |
| 0504 | E1E9 | 27 E2 | |
| 0504 | E1EB | 48 E2 | |
| 0504 | E1ED | 61 E2 | |
| 0505 | E1EF | A0 E2 | .WOR CHNGG, LOAD, DUMP, ASSEM, CGPC, CGA |
| 0505 | E1F1 | E6 E2 | |
| 0505 | E1F3 | 3B E4 | |
| 0505 | E1F5 | 00 D0 | |
| 0505 | E1F7 | D4 E5 | |
| 0505 | E1F9 | EE E5 | |
| 0506 | E1FB | F2 E5 | .WOR CGX, CGY, CGPS, CGS, NXT5, BRKA |
| 0506 | E1FD | F6 E5 | |
| 0506 | E1FF | EA E5 | |

| LINE # | LOC | CODE | LINE |
|--------|------|-------|---------------------------------------------|
| 0506 | E201 | FA E5 | |
| 0506 | E203 | 0D E6 | |
| 0506 | E205 | 1B E6 | |
| 0507 | E207 | 4D E6 | .WOR SHOW, CLRBB, SHIS, REGT, TRACE |
| 0507 | E209 | FE E6 | |
| 0507 | E20B | 65 E6 | |
| 0507 | E20D | D9 E6 | |
| 0507 | E20F | DD E6 | |
| 0508 | E211 | 9E F8 | .WORD MNEENT, KDISA, TOGTA1, TOGTA2, VECKSM |
| 0508 | E213 | 0A E7 | |
| 0508 | E215 | BD E6 | |
| 0508 | E217 | CB E6 | |
| 0508 | E219 | 94 E6 | |
| 0509 | E21B | E5 E6 | .WORD BRKK, BASIEN, BASIRE |
| 0509 | E21D | 00 B0 | |
| 0509 | E21F | 03 B0 | |
| 0510 | E221 | | ; USER DEFINED FUNCTIONS |
| 0511 | E221 | 0C 01 | .WOR KEYF1, KEYF2, KEYF3 |
| 0511 | E223 | 0F 01 | |
| 0511 | E225 | 12 01 | |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|--------------------------------------------------|
| 0513 | E227 | | ;***** R COMMAND--DISPLAY REGISTERS ***** |
| 0514 | E227 | 20 13 EA | REG JSR CRLW ;CLEAR DISP IF KB |
| 0515 | E22A | A0 08 | LDY #M4-M1 ;MESSAG & (CR) |
| 0516 | E22C | 20 AF E7 | JSR KEP |
| 0517 | E22F | 20 24 EA | JSR CRCK |
| 0518 | E232 | 20 3E E8 | REG1 JSR BLANK |
| 0519 | E235 | A0 09 | LDY #SAVPC-ADDR ;OUTPUT PGR CNTR (SAVPC+1,SAVPC) |
| 0520 | E237 | 20 DD E2 | JSR WRITAD |
| 0521 | E23A | A9 20 | LDA #CSAVPS ;NOW THE OTHER 5 REGS |
| 0522 | E23C | 8D 1C A4 | STA ADDR |
| 0523 | E23F | A9 A4 | LDA #CSAVPS |
| 0524 | E241 | 8D 1D A4 | STA ADDR+1 |
| 0525 | E244 | A2 05 | LDX #5 ;COUNT |
| 0526 | E246 | D0 07 | BNE MEM1 ;SHARE CODE |
| 0528 | E248 | | ;***** M COMMAND--DISPLAY MEMORY ***** |
| 0529 | E248 | 20 AE EA | MEM JSR ADDR ;GET START ADDRESS IN ADDR |
| 0530 | E24B | B0 13 | RCS MEM3 |
| 0531 | E24D | A2 04 | MEIN LDX #4 |
| 0532 | E24F | A0 00 | MEM1 LDY #0 |
| 0533 | E251 | 20 3E E8 | MEM2 JSR BLANK |
| 0534 | E254 | A9 1C | LDA #CADDR |
| 0535 | E256 | 20 58 E8 | JSR LDAY ;LOAD CONTENTS OF CURR LOCATION |
| 0536 | E259 | 20 46 EA | JSR NUMA ;AND DISPLAY IT AS 2 HEX DIGITS |
| 0537 | E25C | C8 | INY |
| 0538 | E25D | CA | DEX ;DECR COUNTER |
| 0539 | E25E | D0 F1 | BNE MEM2 |
| 0540 | E260 | 60 | MEM3 RTS ;GET NEXT COMMAND |
| 0542 | E261 | | ;***** G COMMAND--RESTART PROCESSOR ***** |
| 0543 | E261 | 20 37 E8 | GO JSR PSL1 ;"/ |
| 0544 | E264 | 20 85 E7 | JSR GCNT ;GET COUNT |
| 0545 | E267 | 20 F0 E9 | JSR CRLF |
| 0546 | E26A | 4C 86 E2 | JMP GOBK1 ;RESUME EXECUTION |
| 0547 | E26D | AD 0E A4 | GOBK LDA REGF ;DISPLAY REGISTERS ? |
| 0548 | E270 | F0 06 | REQ GOBK0 ;NO, BRANCH |
| 0549 | E272 | 20 32 E2 | JSR REG1 ;SHOW THE SIX REG |
| 0550 | E275 | 20 24 EA | JSR CRCK ;(CR) |
| 0551 | E278 | 20 07 E9 | GOBK0 JSR RCHEK ;SEE IF HE WANTS TO INTERRUPT |
| 0552 | E27B | AD 0F A4 | LDA DISFLG ;DISASSEMBLE CURRENT INSTR ? |
| 0553 | E27E | F0 06 | REQ GOBK1 ;NO, BRANCH |
| 0554 | E280 | 20 6C F4 | JSR DISASM ;DISASM THIS INSTRUCTION |
| 0555 | E283 | 20 13 EA | JSR CRLW |
| 0556 | E286 | AE 24 A4 | GOBK1 LDX SAVS ;RESTORE SAVED REGS FOR RTI |
| 0557 | E289 | 9A | TXS |
| 0558 | E28A | AC 23 A4 | LDY SAVY |
| 0559 | E28D | AE 22 A4 | LDX SAVX |
| 0560 | E290 | AD 26 A4 | LDA SAVPC+1 |
| 0561 | E293 | 48 | PHA ;PUT PC ON STACK |
| 0562 | E294 | AD 25 A4 | LDA SAVPC |
| 0563 | E297 | 48 | PHA |
| 0564 | E298 | AD 20 A4 | LDA SAVPS ;STATUS ALSO |
| 0565 | E29B | 48 | PHA |
| 0566 | E29C | AD 21 A4 | LDA SAVA |
| 0567 | E29F | 40 | RTI ;AND AWAY WE GO... |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|-----------------------------------------------------------|
| 0569 | E2A0 | | ;***** / COMMAND--ALTER MEMORY ***** |
| 0570 | E2A0 | 20 3E E8 | CHNGG JSR BLANK |
| 0571 | E2A3 | 20 DB E2 | JSR WRITAZ ;WRITE ADDR |
| 0572 | E2A6 | 20 3E E8 | CHNG1 JSR BLANK |
| 0573 | E2A9 | 20 5D EA | JSR RD2 ;GET VALUE |
| 0574 | E2AC | 90 0A | BCC CH2 ;ISN'T SKIP OR DONE |
| 0575 | E2AE | C9 20 | CMP #' |
| 0576 | E2B0 | D0 13 | BNE CH3 ;NOT BLANK SO MUST BE DONE |
| 0577 | E2B2 | | ;SKIP THIS LOCATION |
| 0578 | E2B2 | 20 3E E8 | JSR BLANK |
| 0579 | E2B5 | 4C C0 E2 | JMP CH4 |
| 0580 | E2B8 | | ;IS ALTER |
| 0581 | E2B8 | 20 78 E8 | CH2 JSR SADDR ;STORE ENTERED VALUE INTO MEMORY |
| 0582 | E2B8 | F0 03 | REQ CH4 ;NO ERROR IN STORE |
| 0583 | E2BD | 4C 33 E8 | JMP MEMERR ;MEMORY WRITE ERROR |
| 0584 | E2C0 | CB | CH4 INY |
| 0585 | E2C1 | C0 04 | CFY #4 |
| 0586 | E2C3 | D0 E1 | BNE CHNG1 ;GO AGAIN |
| 0587 | E2C5 | | ;HAVE DONE LINE OR HAVE (CR) |
| 0588 | E2C5 | 20 CD E2 | CH3 JSR NXTADD ;UPDATE THE ADDRESS |
| 0589 | E2C8 | A9 0D | LDA #CR ;CLEAR DISPL |
| 0590 | E2CA | 4C E9 FE | JMP PATC10 ;ONLY ONE (CR) & BACK TO MONITOR |
| 0592 | E2CD | 98 | NXTADD TYA ;ADD Y TO ADDR+1, ADDR |
| 0593 | E2CE | 18 | CLC |
| 0594 | E2CF | 6D 1C A4 | AIC ADDR |
| 0595 | E2D2 | 8D 1C A4 | STA ADDR |
| 0596 | E2D5 | 90 03 | BCC NXTA1 |
| 0597 | E2D7 | EE 1D A4 | INC ADDR+1 |
| 0598 | E2DA | 60 | NXTA1 RTS |
| 0600 | E2DB | | ;WRITE CURRENT VALUE OF ADDR |
| 0601 | E2DB | | ;PART OF / & SPACE COMM |
| 0602 | E2DB | A0 00 | WRITAZ LDY #0 |
| 0603 | E2DD | B9 1D A4 | WRITAD LDA ADDR+1, Y |
| 0604 | E2E0 | BE 1C A4 | LDX ADDR, Y |
| 0605 | E2E3 | 4C 42 EA | JMP WRAX |
| 0607 | E2F6 | | ;***** L COMMAND--GENERAL LOAD ***** |
| 0608 | E2F6 | | ;LOAD OBJECT FROM TTY, USER, TAPE OR TAPE IN KIM-1 FORMAT |
| 0609 | E2F6 | 20 48 E8 | LOAD JSR WHEREI ;WHERE INPUT |
| 0610 | E2F9 | | ;GET ' ' , # OF BYTES AND SA |
| 0611 | E2E9 | 20 93 E9 | LOAD1 JSR INALL ;GET FIRST CHAR |
| 0612 | E2EC | C9 3B | CMP #' ;LOOK FOR BEGINNING |
| 0613 | E2EF | D0 F9 | BNE LOAD1 ;IGNOKE ALL CHARS BEFORE ' ;' |
| 0614 | E2F0 | 20 4D E8 | JSR CLRCK ;CLEAR CHECKSUM |
| 0615 | E2F3 | 20 4B E5 | JSR CHEKAR ;READ RECORD LENGTH |
| 0616 | E2F6 | AA | TAX ;SAVE IN X THE # BYTES |
| 0617 | E2F7 | 20 4B E5 | JSR CHEKAR ;READ UPPER HALF OF ADDRESS |
| 0618 | E2FA | 8D 1D A4 | STA ADDR+1 |
| 0619 | E2FD | 20 4B E5 | JSR CHEKAR ;READ LOWER HALF OF ADDRESS |
| 0620 | E300 | 8D 1C A4 | STA ADDR |
| 0621 | E303 | 8A | TXA |
| 0622 | E304 | F0 1B | REQ LOAD4 ;LAST RECORD (RECORD LENGTH=0) |

```

LINE # LOC CODE LINE
0623 E306 ;GET DATA
0624 E306 20 FD E3 LOAD2 JSR RBYTE ;READ NEXT BYTE OF DATA
0625 E309 20 13 E4 JSR STBYTE ;STORE AT LOC (ADDR+1,ADDR)
0626 E30C CA DEX ;DECR RECORD LENGTH
0627 E30D D0 F7 BNE LOAD2
0628 E30F ;COMPARE CKSUM
0629 E30F 20 FD E3 JSR RBYTE ;READ UPPER HALF OF CKSUM
0630 E312 CD 1F A4 CMP CKSUM+1 ;COMPARE TO COMPUTED VALUE
0631 E315 D0 6E BNE CKERR ;CKSUM ERROR
0632 E317 20 FD E3 JSR RBYTE ;READ LOWER HALF OF CHECKSUM
0633 E31A CD 1E A4 CMP CKSUM
0634 E31D D0 66 BNE CKERR
0635 E31F F0 C8 BEQ LOAD1 ;UNTIL LAST RECORD
0636 E321 A2 05 LOAD4 LDX #5 ;READ 4 MORE ZEROS
0637 E323 20 FD E3 LOAD5 JSR RBYTE
0638 E326 CA DEX
0639 E327 D0 FA BNE LOAD5
0640 E329 20 93 E9 JSR INALL ;READ LAST (CR)
0641 E32C 4C 20 E5 JMP DU13 ;SET DEFAULT DEV & GO BACK

0643 E32F ;LOAD ROUTINE FROM TAPE BY BLOCKS
0644 E32F ;CHECK FOR RIGHT FILE & LOAD FIRST BLOCK
0645 E32F A9 00 LOADTA LDA #*00 ;CLEAR BLOCK COUNT
0646 E331 8D 15 01 STA BLK
0647 E334 20 53 ED JSR TIBY1 ;LOAD BUFFER WITH A BLOCK
0648 E337 CA DEX ;SET X=0
0649 E338 8E 15 A4 STX CURPO2 ;CLEAR DISPLAY PTR
0650 E338 8D 16 01 LDA TABUFF,X ;BLK COUNT SHOULD BE ZERO
0651 E33E D0 EF BNE LOADTA ;NO, READ ANOTHER BLOCK
0652 E340 E8 INX
0653 E341 ;AFTER FIRST BLOCK OUTPUT FILE NAME
0654 E341 EE 11 A4 INC PRIFLG ;SO DO NOT GO TO PRINT.
0655 E344 A0 48 LDY #MSG0-M1 ;PRINT "F="
0656 E346 20 AF E7 JSR KEP
0657 E349 8D 16 01 LOAD1A LDA TABUFF,X ;OUTPUT FILE NAME
0658 E34C 20 7A E9 JSR OUTPUT ;ONLY TO DISPLAY
0659 E34F E8 INX
0660 E350 E0 06 CPX #6
0661 E352 D0 F5 BNE LOAD1A
0662 E354 20 3E E8 JSR BLANK
0663 E357 A0 61 LDY #MSG6-M1 ;PRINT "BLK= "
0664 E359 20 AF E7 JSR KEP
0665 E35C CE 11 A4 DEC PRIFLG ;RESTORE PRINTR FLG
0666 E35F 20 8D ED JSR ADDBK1 ;JUST OUTPUT BLK CNT
0667 E362 A2 01 LDX #1 ;RESTORE X
0668 E364 ;CHECK IF FILE IS CORRECT
0669 E364 8D 16 01 LOADT2 LDA TABUFF,X ;NOW CHCK FILE NAME
0670 E367 D0 2D A4 CMP NAME-1,X
0671 E36A D0 C3 BNE LOADTA ;IF NO FILENAME GET
0672 E36C E8 INX ;ANOTHER BLOCK
0673 E36D E0 06 CPX #6 ;FILENAME=5 CHRS
0674 E36F D0 F3 BNE LOADT2
0675 E371 8E 36 A4 STX TAP1R ;SAVE TAPE BUFF PTR
0676 E374 EE 11 A4 INC PRIFLG ;OUTPUT MSG ONLY TO DISPLAY
    
```

```

LINE # LOC CODE LINE
0677 E377 A9 00 LDA #0 ;CLEAR DISPLAY POINTER
0678 E379 8D 15 A4 STA CURPO2
0679 E37C A0 66 LDY #MSG7-M1 ;PRINT "LOAD " WITHOUT CLR DISPL
0680 E37E 20 96 E3 JSR CKER1
0681 E381 CE 11 A4 DEC PRIFLG
0682 E384 60 RTS

0684 E385 ;LINE CKSUM ERROR
0685 E385 20 8E E3 CKERR JSR CKER0 ;SUBR SO MNEM ENTRY CAN USE IT
0686 E388 20 8B E2 JSR WRITAZ ;WRITE ADDR
0687 E38B 4C A1 E1 JMP COMIN
0688 E38E 20 FE E8 CKER0 JSR LL ;SET DEFAULT DEVICES
0689 E391 20 24 EA JSR CRCK ;(CR)
0690 E394 A0 52 CKER00 LDY #MSG3-M1 ;PRINT "ERROR"
0691 E396 89 00 E0 CKER1 LDA M1,Y ;DONT CLR DISPLAY TO THE RIGHT
0692 E399 C9 3B CMP #' ;
0693 E39B F0 06 BEQ CKER2
0694 E39D 20 7A E9 JSR OUTPUT ;ONLY TO TERMINAL
0695 E3A0 C8 INY
0696 E3A1 D0 F3 BNE CKER1
0697 E3A3 60 CKER2 RTS

0699 E3A4 ;LOAD ROUTINE FROM TAPE WITH KIM-1 FORMAT
0700 E3A4 20 4D EB LOADK1 JSR CLRCK ;CLEAR CKSUM
0701 E3A7 20 EA ED LOADK1 JSR TAISSET ;SET TAPE FOR INPUT
0702 E3AA 20 29 EE LOADK2 JSR GETTAP ;READ CHARACTER FROM TAPE
0703 E3AD C9 2A CMP #'* ;BEGINNING OF FILE?
0704 E3AF F0 06 BEQ LOADK3 ;YES,BRNCH
0705 E3B1 C9 14 CMP #'*16 ;IF NOT * SHOULD BE SYN
0706 E3B3 D0 F2 BNE LOADK1
0707 E3B5 F0 F3 BEQ LOADK2
0708 E3B7 20 FD E3 LOADK3 JSR RBYTE ;READ ID FROM TAPE
0709 E3BA 8D 21 A1 STA SAVA ;SAVE ID
0710 E3BD ;NOW GET ADDR TO DISPLAY
0711 E3BD ;& COMPARE ID AFTERWARDS
0712 E3BD 20 4B E5 JSR CHEKAR ;GET START ADDR LOW
0713 E3C0 8D 1C A4 STA ADDR
0714 E3C3 20 4B E5 JSR CHEKAR ;GET START ADDR HIGH
0715 E3C6 8D 1D A4 STA ADDR+1
0716 E3C9 20 25 E4 JSR GETID ;ID FROM HIM
0717 E3CC CD 21 A4 CMP SAVA ;DO IDS MATCH?
0718 E3CF D0 D3 BNE LOADK1 ;NO ,GET ANOTHER FILE
0719 E3D1 A2 02 LOADK5 LDX #*02 ;GET 2 CHARS
0720 E3D3 20 29 EE LOADK6 JSR GETTAP ;1 CHAR FROM TAPE
0721 E3D6 C9 2F CMP #' / ;LAST CHAR ?
0722 E3D8 F0 0E BEQ LOADK7 ;YES,BRNCH
0723 E3DA 20 84 EA JSR PACK ;CONVERT TO HEX
0724 E3DD 80 A6 BCS CKERR ;NOT HEX CHAR SO ERROR
0725 E3DF CA DEX
0726 E3E0 D0 F1 BNE LOADK6
0727 E3E2 20 13 E4 JSR STRYTE ;STORE & CHCK MEM FAIL
0728 E3E5 4C D1 E3 JMP LOADK5 ;NEXT
0729 E3E8 20 FD E3 LOADK7 JSR RBYTE ;END OF DATA CMP CKSUM
0730 E3EB CD 1E A4 CMP CKSUM ;LOW
    
```

COMMANDS

PA00-J001A.....PAGE 0018

```

LINE # LOC      CODE      LINE
0731 E3EE D0 95          BNE CKERR
0732 E3F0 20 F0 E3      JSR RRYTE
0733 E3F3 C0 1F A4      CMP CKSUM+1 ;HIGH
0734 E3F6 D0 8D          BNE CKERR
0735 E3F8 68            PLA ;CORRECT RTN INSTEAD OF WHEREI
0736 E3F9 68            PLA
0737 E3FA 4C 20 E5      JMP DU13 ;TELL HIM & GO BACK TO COMMAN

0739 E3FD ;GET 2 ASCII CHRS INTO 1 BYTE
0740 E3FD ;FOR TAPE (T) GET ONLY ONE HEX CHR
RBYTE LDA INFLG ;INPUT DEVICE
0742 E400 C9 54          CMP #'T
0743 E402 D0 03          BNE RRYT1
0744 E404 4C 93 E9      JMP INALL ;ONLY ONE BYTE FOR T (INPUT DEV)
0745 E407 20 93 E9      RBYT1 JSR INALL
0746 E40A 20 84 EA      JSR PACK
0747 E40D 20 93 E9      JSR INALL
0748 E410 4C 84 EA      JMP PACK

0750 E413 ;STORE AND CHECK MEMORY FAIL
0751 E413 20 4E E5      STBYTE JSR CHEKA ;ADD TO CKSUM
0752 E416 A0 00          LDY #0
0753 E418 20 78 EB      JSR SADDR ;STORE AND CHCK
0754 E41B F0 03          BEQ *+5
0755 E41D 4C 33 EB      JMP MEMERR ;MEMORY WRITE ERROR
0756 E420 A0 01          LDY #1
0757 E422 4C D0 E2      JMP NXTADD ;INC ADDR+1, ADDR BY 1

0759 E425 ;GET ID FROM LAST 2 CHR OF FILENAM
0760 E425 A2 04          GETID LDX #4 ;SEE WHAT HE GAVE US
0761 E427 BD 2E A4      GID1 LDA NAME,X ;GET LAST 2 CHARS
0762 E42A CA            DEX
0763 E42B C9 20          CMP #'20 ;(SPACE) ?
0764 E42D F0 F8          BEQ GID1
0765 E42F BD 2E A4      LDA NAME,X ;CONVERT TO BINARY
0766 E432 20 84 EA      JSR PACK
0767 E435 BD 2F A4      LDA NAME+1,X
0768 E438 4C 84 EA      JMP PACK ;ID IS IN STIY

0770 E43B ;***** D COMMAND-GENERAL DUMP *****
0771 E43B ;TO TTY, PRINTR, USER, X ,TAPE, TAKIM-1
0772 E43B AD 10 A4      DUMP LDA BKFLG ;SAVE IT TO USE IT
0773 E43E 4B            PHA
0774 E43F A9 00          LDA #00
0775 E441 BD 10 A4      STA BKFLG
0776 E444 20 24 EA      DU1 JSR CRCK ;<CR>
0777 E447 20 A3 E7      DU0 JSR FROM ;GET START ADDR
0778 E44A B0 FB          BCS DU0 ;IN CASE OF ERROR DO IT AGAIN
0779 E44C 20 3E EB      JSR BLANK
0780 E44F 20 10 F9      JSR ADRES1 ;TRANSFER ADDR TO S1
0781 E452 20 A7 E7      DU1B JSR TO ;GET END ADDR
0782 E455 B0 FB          BCS DU1B
0783 E457 20 13 EA      JSR CRLOW
0784 E45A AD 10 A4      LDA BKFLG ;EXECUTE WHEREO ONLY ONCE
    
```

COMMANDS

PA00-J001A.....PAGE 0019

```

LINE # LOC      CODE      LINE
0785 E45D D0 0E          BNE DU1A
0786 E45F 20 71 E8      JSR WHEREO ;WHICH DEV (OUTFLG)
0787 E462 A9 00          LDA #0
0788 E464 BD 06 01      STA S2 ;CLEAR RECORD COUNT
0789 E467 BD 07 01      STA S2+1
0790 E46A EE 10 A4      INC BKFLG ;SET FLG
0791 E46D ;CHKC OUTPUT DEV
DU1A LDA OUTFLG
0792 E46D AD 13 A4      DU1A LDA OUTFLG
0793 E470 C9 4B          CMP #'K ;TAPE FOR KIM?
0794 E472 D0 04          BNE *+6
0795 E474 68            PLA ;FULL FLG
0796 E475 4C 87 E5      JMP DUMPKI ;YES, GO OUTPUT WHOLE FILE
0797 E478 A0 01          LDY #1 ;OUTPUT ONE MORE BYTE
0798 E47A 20 C0 E2      JSR NXTADD
0799 E47D 20 F0 E9      DU2 JSR CRLF
0800 E480 20 07 E9      JSR RCHEK ;SEE IF HE WANTS TO INTERRUPT
0801 E483 ;CALCULATE # OF BYTES YET TO BE DUMPED
0802 E483 20 4D EB      JSR CRCK ;CLEAR CKSUM
0803 E486 AD 1C A4      LDA ADDR ;END ADDRESS-CURRENT ADDRESS
0804 E489 3B            SEC
0805 E48A ED 1A A4      SBC S1
0806 E48D 4B            PHA ;# OF BYTES LOW
0807 E48E AD 1D A4      LDA ADDR+1
0808 E491 ED 1B A4      SBC S1+1
0809 E494 D0 09          BNE DU6 ;# OF BYTES HIGH
0810 E496 ;SEE IF 24 OR MORE BYTES TO GO
0811 E496 68            PLA ;# BYTES HIGH WAS ZERO
0812 E497 F0 42          BEQ DU10 ;ARE DONE
0813 E499 C9 18          CMP #24 ;# BYTES > 24 ?
0814 E49B 90 05          BCC DU8 ;NO ,ONLY OUTPUT REMAINING BYTES
0815 E49D B0 01          BCS DU7 ;YES ,24 BYTES IN NEXT RECORD
0816 E49F 68            DU6 PLA
0817 E4A0 A9 18          DU7 LDA #24
0818 E4A2 ;OUTPUT ";", # OF BYTES AND SA
0819 E4A2 4B            DU8 PHA
0820 E4A3 20 BA E9      JSR SEMI ;SEMICOLON
0821 E4A6 68            PLA
0822 E4A7 BD 19 A4      STA COUNT ;SAVE # OF BYTES
0823 E4AA 20 38 E5      JSR OUTCK ;OUTPUT # OF BYTES
0824 E4AD AD 1B A4      LDA S1+1 ;OUTPUT ADDRESS
0825 E4B0 20 38 E5      JSR OUTCK
0826 E4B3 AD 1A A4      LDA S1
0827 E4B6 20 38 E5      JSR OUTCK
0828 E4B9 ;OUTPUT DATA
0829 E4B9 20 31 E5      DU9 JSR OUTCKS ;GET CHAR SPEC BY S1 (NO PAG 0)
0830 E4BC A9 00          LDA #0 ;CLEAR DISP PTR
0831 E4BE BD 15 A4      STA CURPD2
0832 E4C1 20 5D E5      JSR ADIS1 ;INCR S1+1,S1
0833 E4C4 CE 19 A4      DEC COUNT ;DECREMENT BYTE COUNT
0834 E4C7 D0 F0          BNE DU9 ;NOT DONE WITH THIS RECORD
0835 E4C9 ;OUTPUT CKSUM
0836 E4C9 AD 1F A4      LDA CKSUM+1
0837 E4CC 20 3B E5      JSR OUTCK1 ;WITHOUT CHEKA
0838 E4CF AD 1E A4      LDA CKSUM
0839 E4D2 20 3B E5      JSR OUTCK1
    
```

COMMANDS

PA00-J001A.....PAGE 0020

| LINE # | LOC | CODE | LINE | |
|--------|------|----------|-----------------------------------------|-------------------------------------|
| 0840 | E4D5 | 20 66 E5 | JSR INCS2 | ; INC VERTICAL COUNT |
| 0841 | E4DB | 4C 7D E4 | JMP DU2 | ; NEXT RECORD |
| 0842 | E4DR | | | |
| 0843 | E4DR | A0 1C | ; ALL DONE | |
| 0844 | E4DI | 20 70 E9 | DU10 LDY #M5-M1 | ; PRINT *MORE ?* |
| 0845 | E4E0 | C9 59 | JSR KEPR | ; OUTPUT MSG AND GET AN ANSWER |
| 0846 | E4E2 | D0 03 | CMP #'Y | |
| 0847 | E4E4 | 4C 44 E4 | RNE *+5 | |
| 0848 | E4E7 | 68 | JMP DU1 | ; DUMP MORE DATA |
| 0849 | E4EB | 8D 10 A4 | FLA | ; RESTORE FLG |
| 0850 | E4EB | | STA BKFLG | |
| 0851 | E4EB | | ; OUTPUT LAST RECORD | |
| 0851 | E4E8 | 20 66 E5 | JSR INCS2 | |
| 0852 | E4EE | 20 BA E9 | JSR SEMI | ; OUTPUT ' ; ' |
| 0853 | E4F1 | A2 02 | LIX #2 | |
| 0854 | E4F3 | A9 00 | LDA #0 | ; OUTPUT # OF BYTES (0=LAST RECORD) |
| 0855 | E4F5 | 20 3B E5 | JSR OUTCK1 | |
| 0856 | E4F8 | AD 07 01 | DU10A LDA S2+1 | ; OUTPUT RECORD COUNT |
| 0857 | E4F8 | 20 3B E5 | JSR OUTCK1 | ; CHECKCUM IS THE SAME |
| 0858 | E4FE | AD 06 01 | LDA S2 | |
| 0859 | E501 | 20 3B E5 | JSR OUTCK1 | |
| 0860 | E504 | CA | DEX | |
| 0861 | E505 | D0 F1 | RNE DU10A | |
| 0862 | E507 | 20 F0 E9 | JSR CRLF | |
| 0863 | E50A | | ; CLOSE TAPE BLOCK IF ACTIVE | |
| 0864 | E50A | AD 13 A4 | DU11 LDA OUTFLG | |
| 0865 | E50D | C9 54 | CMP #'T | |
| 0866 | E50F | D0 0F | RNE DU13 | ; NO ,BRANCH |
| 0867 | E511 | AD 37 A4 | DU12 LDA TAPTR2 | ; TAP OUTPUT BUFF PTR |
| 0868 | E514 | C9 01 | CMP #1 | ; BECAUSE FIRST ONE IS BLK CNT |
| 0869 | E516 | F0 08 | REQ DU13 | ; NO DATA TO WRITE |
| 0870 | E518 | A9 00 | LDA #0 | ; FILL REST BUFF ZEROS |
| 0871 | E51A | 20 8B F1 | JSR TOBYTE | ; OUTPUT TO BUFF |
| 0872 | E51D | 4C 11 E5 | JMP DU12 | ; FINISH THIS BLOCK |
| 0873 | E520 | 20 13 EA | DU13 JSR CRLW | |
| 0874 | E523 | 18 | CLC | ; ENABL INTERR |
| 0875 | E524 | A9 00 | LDA #T1I | ; T1 FROM FREE RUNNING TO 1 SHOT |
| 0876 | E526 | 8D 0B AB | STA ACR | |
| 0877 | E529 | A9 34 | DU14 LDA #*34 | ; SET BOTH TAPES ON |
| 0878 | E52B | 8D 00 AB | STA IRB | |
| 0879 | E52E | 4C FE E8 | JMP LL | |
| 0881 | E531 | | ; GET CHAR SPECIFIED BY START ADDR (S1) | |
| 0882 | E531 | A9 1A | OUTCKS LDA #*S1 | |
| 0883 | E533 | A0 00 | LDY #0 | |
| 0884 | E535 | 20 5B EB | JSR LDAY | |
| 0886 | E538 | | ; ADD TO CHECKSUM AND PRINT | |
| 0887 | E538 | 20 4E E5 | OUTCK JSR CHEKA | ; CHCKSUM |
| 0888 | E53B | 48 | OUTCK1 PHA | |
| 0889 | E53C | AD 13 A4 | LDA OUTFLG | ; IF TAPE DO NOT CNVRT |
| 0890 | E53F | C9 54 | CMP #'T | ; TO TWO ASCII CHRS |
| 0891 | E541 | D0 04 | RNE OUTCK2 | |
| 0892 | E543 | 68 | FLA | |
| 0893 | E544 | 4C 8B F1 | JMP TOBYTE | ; OUTPUT TO TAP BUFF |
| 0894 | E547 | 68 | OUTCK2 FLA | |

COMMANDS

PA00-J001A.....PAGE 0021

| LINE # | LOC | CODE | LINE | |
|--------|------|----------|----------------------------------------------------|-----------------------------------|
| 0895 | E54B | 4C 46 EA | JMP NUMA | ; TWO ASCII REPRE |
| 0897 | E54B | 20 FD E3 | CHEKAR JSR RBYTE | ; TWO ASCII CHR---> 1 BYTE |
| 0898 | E54E | 48 | CHEKA PHA | ; ADD TO CHECKSUM |
| 0899 | E54F | 18 | CLC | |
| 0900 | E550 | 6D 1E A4 | ADC CKSUM | |
| 0901 | E553 | 8D 1E A4 | STA CKSUM | |
| 0902 | E556 | 90 03 | BCC *+5 | |
| 0903 | E558 | EE 1F A4 | INC CKSUM+1 | |
| 0904 | E55B | 68 | FLA | |
| 0905 | E55C | 60 | RTS | |
| 0907 | E55D | | ; ADD ONE TO START ADDR (S1) | |
| 0908 | E55D | EE 1A A4 | ADD\$1 INC S1 | |
| 0909 | E560 | D0 03 | RNE ADD1 | |
| 0910 | E562 | EE 1B A4 | INC S1+1 | |
| 0911 | E565 | 60 | ADD1 RTS | |
| 0913 | E566 | EE 06 01 | INCS2 INC S2 | ; INCR VERTICAL COUNT |
| 0914 | E569 | D0 03 | RNE *+5 | |
| 0915 | E56B | EE 07 01 | INC S2+1 | |
| 0916 | E56E | 60 | RTS | |
| 0918 | E56F | | ; OPEN A FILE FOR OUTPUT TO TAPE BY BLOCKS | |
| 0919 | E56F | | ; OUTPUT FILENAME GIVEN BY JSR WHERED TO TAPE BUFF | |
| 0920 | E56F | A2 00 | DUMPTA LDX #0 | ; INITIALIZE TAPTR |
| 0921 | E571 | 8A | TXA | ; TO OUTPUT |
| 0922 | E572 | 8E 68 01 | STX BLK0 | ; BLOCK COUNTER |
| 0923 | E575 | 8E 37 A4 | STX TAPTR2 | ; TAP OUTPUT BUFF PTR |
| 0924 | E578 | 20 8B F1 | JSR TOBYTE | ; TWO START OF FILE CHRS |
| 0925 | E57B | 8D 2E A4 | DUMPT1 LDA NAME,X | ; OUTPUT FILENAME |
| 0926 | E57E | 20 8B F1 | JSR TOBYTE | |
| 0927 | E581 | E8 | INX | |
| 0928 | E582 | E0 05 | CFX #5 | ; 5 FILENAME CHRS ? |
| 0929 | E584 | D0 F5 | RNE DUMPT1 | |
| 0930 | E586 | 60 | RTS | |
| 0932 | E587 | | ; DUMP ROUTINE TO TAPE WITH KIM-1 FORMAT | |
| 0933 | E587 | 20 1D F2 | DUMPKI JSR TAOSET | ; SET TAPE FOR OUTPUT |
| 0934 | E58A | A9 2A | LDA #'* | ; TO EITHER 1 OR 2 |
| 0935 | E58C | 20 4A F2 | JSR OUTTAP | ; DIRECTLY TO TAPE |
| 0936 | E58F | | ; ID FROM LAST 2 CHRS OF FILENAME | |
| 0937 | E58F | 20 25 E4 | JSR GETID | |
| 0938 | E592 | 20 3B E5 | JSR OUTCK1 | |
| 0939 | E595 | 20 4D EB | JSR CLRCK | |
| 0940 | E598 | | ; STARTING ADDR | |
| 0941 | E598 | AD 1A A4 | LDA S1 | |
| 0942 | E59B | 20 3B E5 | JSR OUTCK | ; WITH CHCKSUM |
| 0943 | E59E | AD 1B A4 | LDA S1+1 | |
| 0944 | E5A1 | 20 3B E5 | JSR OUTCK | |
| 0945 | E5A4 | | ; OUTPUT DATA | |
| 0946 | E5A4 | 20 31 E5 | DUK2 JSR OUTCKS | ; OUTPUT CHR SPECIFIED BY S1+1,S1 |
| 0947 | E5A7 | 20 5D E5 | JSR ADD\$1 | ; INCREM S1+1,S1 |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|--------------------------------------------------|
| 0948 | E5AA | AD 1A A4 | LDA S1 ;CHKCK FOR LAST BYTE |
| 0949 | E5AD | CD 1C A4 | CMP ADDR ;LSB OF END ADDR |
| 0950 | E5B0 | AD 1B A4 | LDA S1+1 |
| 0951 | E5B3 | ED 1D A4 | SBC ADDR+1 |
| 0952 | E5B6 | 90 EC | BCC DUK2 ;NEXT CHR |
| 0953 | E5B8 | | ;NOW SEND END CHR */* |
| 0954 | E5B8 | A9 2F | LDA #2F |
| 0955 | E5BA | 20 4A F2 | JSR OUTTAP ;DIRECTLY TO TAPE |
| 0956 | E5BD | | ;CHECKSUM |
| 0957 | E5BD | AD 1E A4 | LDA CKSUM |
| 0958 | E5C0 | 20 46 EA | JSR NUMA ;ASCII REPRES |
| 0959 | E5C3 | AD 1F A4 | LDA CKSUM+1 |
| 0960 | E5C6 | 20 46 EA | JSR NUMA |
| 0961 | E5C9 | | ;TWO EOT CHRS |
| 0962 | E5C9 | A9 04 | LDA #*04 |
| 0963 | E5CB | 20 4A F2 | JSR OUTTAP |
| 0964 | E5CE | 20 4A F2 | JSR OUTTAP |
| 0965 | E5D1 | | ;TURN TAPES ON |
| 0966 | E5D1 | 4C 20 E5 | JMP DU13 |
| 0968 | E5D4 | | ;***** * COMMAND-ALTER PROGRAM COUNTER ***** |
| 0969 | E5D4 | 20 AE EA | CGFC JSR ADDRIN ;ADDR (=ADDRESS ENTERED) FROM KB |
| 0970 | E5D7 | 20 DD E5 | CGPC0 JSR CGPC1 ;TRANSFER ADDR TO SAVFC |
| 0971 | E5DA | 4C 13 EA | JMP CRL0W |
| 0972 | E5DD | AD 1D A4 | CGPC1 LDA ADDR+1 ;THIS WAY MNEMONICS CAN USE IT |
| 0973 | E5E0 | 8D 26 A4 | STA SAVFC+1 |
| 0974 | E5E3 | AD 1C A4 | LDA ADDR |
| 0975 | E5E6 | 8D 25 A4 | STA SAVFC |
| 0976 | E5E9 | 60 | RTS |
| 0978 | E5EA | | ;***** P COMMAND-ALTER PROCESSOR STATUS ***** |
| 0979 | E5EA | A2 00 | CGPS LDX #0 |
| 0980 | E5EC | F0 0E | BEG CGALL |
| 0982 | E5EE | | ;***** A COMMAND-ALTER ACCUMULATOR ***** |
| 0983 | E5EE | A2 01 | CGA LDX #1 |
| 0984 | E5F0 | D0 0A | BNE CGALL |
| 0986 | E5F2 | | ;***** X COMMAND-ALTER X REGISTER ***** |
| 0987 | E5F2 | A2 02 | CGX LDX #2 |
| 0988 | E5F4 | D0 06 | BNE CGALL |
| 0990 | E5F6 | | ;***** Y COMMAND-ALTER Y REGISTER ***** |
| 0991 | E5F6 | A2 03 | CGY LDX #3 |
| 0992 | E5F8 | D0 02 | BNE CGALL |
| 0994 | E5FA | | ;***** S COMMAND-ALTER STACK POINTER ***** |
| 0995 | E5FA | A2 04 | CGS LDX #4 |
| 0996 | E5FC | 20 DB E7 | CGALL JSR EQUAL ;PRINT PROMPT |
| 0997 | E5FF | 20 5D EA | JSR RDU2 ;GET VALUE FROM KEYBOARD |
| 0998 | E602 | B0 04 | BCS GOERR |
| 0999 | E604 | 9D 20 A4 | STA SAVFS,X |
| 1000 | E607 | 60 | RTS |
| 1001 | E608 | 20 D4 E7 | GOERR JSR RM |
| 1002 | E60B | D0 EF | BNE CGALL |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|-----------------------------------------------------|
| 1004 | E60D | | ;***** (SPACE) COMMAND-SHOW NEXT 5 MEMORY LOC ***** |
| 1005 | E60D | 20 3E EB | NXT5 JSR BLANK |
| 1006 | E610 | A0 04 | LDY #4 ;UPDATE ADDR FROM |
| 1007 | E612 | 20 CD E2 | JSR NXTADD ;(M)=XXXX |
| 1008 | E615 | 20 DB E2 | JSR WRITAZ ;OUTPUT ADDRESS |
| 1009 | E618 | 4C 4D E2 | JMP MEIN ;DISPLAY CONTENTS OF NEXT 4 LOCS |
| 1011 | E61B | | ;***** B COMMAND-SET BREAKPOINT ADDR ***** |
| 1012 | E61B | A0 27 | BRKA LDY #MB-M1 ;PRINT 'BRK' |
| 1013 | E61D | 20 AF E7 | JSR KEP |
| 1014 | E620 | 20 37 EB | BRK1 JSR PSL1 ;PRINT '/' |
| 1015 | E623 | 20 73 E9 | JSR REBOUT ;GET BREAK NUMBER |
| 1016 | E626 | 3B | SEC |
| 1017 | E627 | E9 30 | SBC #*30 ;0 THRU 3 |
| 1018 | E629 | 30 04 | BMI BKERR ;CHARACTER < '0' -ILLEGAL |
| 1019 | E62B | C9 04 | CMP #4 ;FOUR BRK POINTS |
| 1020 | E62D | 30 05 | BMI BKOK ;0 < CHARACTER < 4 -OK |
| 1021 | E62F | 20 D4 E7 | BKERR JSR RM ;ERROR |
| 1022 | E632 | D0 EC | BNE BRK1 ;ALLOW REENTRY OF BREAK NUMBER |
| 1023 | E634 | 0A | BKOK ASL A ;*2 TO FORM WORD OFFSET |
| 1024 | E635 | 48 | FHA ;SAVE IT |
| 1025 | E636 | 20 AE EA | JSR ADDRIN ;GET ADDRESS FOR BREAKPOINT |
| 1026 | E639 | 68 | FLA |
| 1027 | E63A | B0 10 | BCS BK02 ;BAD ADDRESS ENTERED |
| 1028 | E63C | 20 3D FF | JSR FATC18 ;(CCR) & CLR BUFFERS |
| 1029 | E63F | AA | TAX ;# OF BRK |
| 1030 | E640 | AD 1C A4 | LDA ADDR ;STORE ENTERED ADDR IN BRKPT LIST |
| 1031 | E643 | 9D 00 01 | STA BKS,X |
| 1032 | E646 | AD 1D A4 | LDA ADDR+1 |
| 1033 | E649 | 9D 01 01 | STA BKS+1,X |
| 1034 | E64C | 60 | RTS ;ALL DONE |
| 1036 | E64D | | ;***** ? COMMAND-SHOW CURRENT BREAKPOINTS ***** |
| 1037 | E64D | A0 00 | SHOW LDY #0 |
| 1038 | E64F | 20 13 EA | JSR CRL0W |
| 1039 | E652 | 20 3E EB | SH1 JSR BLANK |
| 1040 | E655 | BE 00 01 | LDX BKS,Y ;ADDRESS OF NEXT BREAKPOINT |
| 1041 | E658 | B9 01 01 | LDA BKS+1,Y |
| 1042 | E65B | 20 42 EA | JSR WRAX ;SHOW BREAKPOINT ADDRESS |
| 1043 | E65E | C8 | INY |
| 1044 | E65F | C8 | INY |
| 1045 | E660 | C0 08 | CFY #8 |
| 1046 | E662 | D0 EE | BNE SH1 |
| 1047 | E664 | 60 | RTS |
| 1049 | E665 | | ;***** H COMMAND-SHOW TRACE STACK HISTORY ***** |
| 1050 | E665 | | ;LAST FIVE INSTR ADDRS |
| 1051 | E665 | A2 05 | SHIS LDX #5 ;NUMBER OF ENTRIES |
| 1052 | E667 | 8E 29 A4 | STX STIY+2 |
| 1053 | E66A | AC 14 A4 | SH11 LBY HISTP ;POINTER TO LATEST ENTRY |
| 1054 | E66D | 20 13 EA | JSR CRL0W |
| 1055 | E670 | 20 3E EB | JSR BLANK |
| 1056 | E673 | B9 2E A4 | LDA HIST,Y ;OUTPUT ADDRESS OF ENTRY |
| 1057 | E676 | 20 46 EA | JSR NUMA |
| 1058 | E679 | B9 2F A4 | LDA HIST+1,Y |

```

LINE # LOC      CODE      LINE
1059 E67C 20 46 EA      JSR NUMA
1060 E67F 20 88 E6      JSR NHIS      ;UPDATE POINTER
1061 E682 CE 29 A4      DEC STIY+2
1062 E685 D0 E3        BNE SH11
1063 E687 60          RTS

1065 E688      ;UPDATE HISTORY POINTER (PART OF H)
1066 E688 C8          NHIS INY
1067 E689 C8          INY
1068 E68A C0 0A       CPY #10
1069 E68C D0 02       BNE NH1
1070 E68E A0 00       LDY #0      ;WRAPAROUND AT 10
1071 E690 8C 14 A4   NH1 STY HISTP
1072 E693 60          RTS

1074 E694      ;***** 3 COMMAND-VERIFY TAPES *****
1075 E694      ;VERIFY CKSUM OF BLOCKS
1076 E694 20 48 E8   VECKSM JSR WHEREI ;GET THE FILE
1077 E697 20 93 E9   JSR INALL ;CHKC OBJ OR SOURCE
1078 E69A C9 0D     CMP #CR    ;FIRST CHR IS (CR) IF OBJ
1079 E69C D0 0E     BNE VECK2 ;ASSUME SOURCE CODE
1080 E69E 20 93 E9   VECK1 JSR INALL ;OBJECT FILE
1081 E6A1 C9 3B     CMP #' ;
1082 E6A3 D0 F9     BNE VECK1 ;IGNORE ALL CHARS BEFORE ' ;
1083 E6A5 20 93 E9   JSR INALL
1084 E6A8 4C 60 FF   JMP PAT20
1085 E6AA EA        NOP
1086 E6AC 20 93 E9   VECK2 JSR INALL ;IT IS TEXT
1087 E6AF C9 0D     CMP #CR
1088 E6B1 D0 F9     BNE VECK2
1089 E6B3 20 93 E9   JSR INALL ;NEED TWO (CR) TO FINISH
1090 E6B6 C9 0D     CMP #CR
1091 E6B8 D0 F2     BNE VECK2
1092 E6BA 4C 20 E5   JMP DU13 ;CLOSE FILE, IT IS OK

1094 E6BD      ;***** 1 COMMAND-TOGGLE TAPE 1 CONTROL *****
1095 E6BD AD 00 AB   TOGTA1 LDA DRB
1096 E6C0 49 10     EOR #$10 ;INVERT PB4
1097 E6C2 8D 00 AB   STA DRB
1098 E6C5 29 10     AND #$10
1099 E6C7 F0 2B     BEQ BRK3 ;IF 0 TAPE CNTRL IS ON
1100 E6C9 D0 2F     BNE BRK4 ;IF $10 TAPE CNTRL IS OFF

1102 E6CB      ;***** 2 COMMAND-TOGGLE TAPE 2 CONTROL *****
1103 E6CB AD 00 AB   TOGTA2 LDA DRB
1104 E6CE 49 20     EOR #$20 ;INVERT PB5
1105 E6D0 8D 00 AB   STA DRB
1106 E6D3 29 20     AND #$20
1107 E6D5 F0 1A     BEQ BRK3
1108 E6D7 D0 21     BNE BRK4

1110 E6D9      ;***** V COMMAND-TOGGLE REGISTER DISP FLG *****
1111 E6D9      ;DISPLAY REGIST BEFORE EXEC
1112 E6D9 A2 0E     REGT LDX #CREGF
1113 E6DB D0 0A     BNE TOGL
    
```

```

LINE # LOC      CODE      LINE
1115 E6DD      ;***** Z COMMAND-TOGGLE DIS TRACE FLG *****
1116 E6DD      ;DISPL NEXT INSTR BEFORE EXEC
1117 E6DD A2 0F     TRACE LDX #DISFLG
1118 E6DF D0 06     BNE TOGL

1120 E6E1      ;***** \ COMMAND-TOGGLE PRINTER FLAG *****
1121 E6E1 A2 11     PRITR LDX #CFRIFLG
1122 E6E3 D0 02     BNE TOGL

1124 E6E5      ;***** 4 COMMAND-TOGGLE SOFT BRK ENABL FLG *****
1125 E6E5 A2 10     BRK4 LDX #CBKFLG

1127 E6E7 BD 00 A4   TOGL LDA MONRAM,X ;LOAD FLAG
1128 E6EA F0 0A     BEQ TOGL1 ;FLAG IS OFF ,SO TURN ON
1129 E6EC A9 00     LDA #0 ;FLAG IS ON ,SO TURN OFF
1130 E6EE 9D 00 A4   STA MONRAM,X
1131 E6F1 A0 24     LDY #M7-M1 ;PRINT "OFF"
1132 E6F3 4C AF E7   BRK2 JMP KEP
1133 E6F6 3B       TOGL1 SEC ;TURN FLAG ON BY SETTING NON-ZERO
1134 E6F7 7E 00 A4   ROR MONRAM,X ;FLAG IS ON MSB
1135 E6FA A0 21     BRK4 LDY #M6-M1 ;PRINT "ON"
1136 E6FC D0 F5     BNE BRK2

1138 E6FE      ;***** # COMMAND-CLEAR ALL BREAKS *****
1139 E6FE A9 00     CLRBK LDA #0 ;STORE ZEROS INTO BRKPT LIST
1140 E700 A2 07     LDX #7
1141 E702 9D 00 01   RS20 STA BKS,X
1142 E705 CA        DEX
1143 E706 10 FA     BPL RS20
1144 E708 30 E7     BMI BRK3 ;PRINT "OFF"

1146 E70A      ;***** K COMMAND-DISASSEMBLE MEMORY *****
1147 E70A LDA #'* ;GET START ADDRESS
1148 E70C 20 7A E9   JSR OUTPUT
1149 E70F 20 AE EA   JSR ADDIN
1150 E712 80 F6     BCS KDISA ;IF ERROR DO IT AGAIN
1151 E714 20 D7 E5   JSR CGFCO ;GET IT INTO PROG CNTR
1152 E717 20 37 EB   JSR PSL1 ;PRINT '/'
1153 E71A 20 85 E7   JSR GCNT ;GET COUNT
1154 E71D 20 24 EA   JSR CRCK
1155 E720 4C 2B E7   JMP JD2
1156 E723 20 07 E9   JD1 JSR RCHEK ;SEE IF HE WANTS TO INTERRUPT
1157 E726 20 90 E7   JSR DONE
1158 E729 F0 17     BEQ JD4
1159 E72B 20 6C F4   JD2 JSR DISASH ;GO TO DISASSEMBLER
1160 E72E AD 25 A4   LDA SAVFC ;POINT TO NEXT INSTRUC LOCAT
1161 E731 3B       SEC ;ONE MORE TO PROG CNTR
1162 E732 65 EA     ADC LENGTH
1163 E734 8D 25 A4   STA SAVFC
1164 E737 90 03     BCC JD3
1165 E739 EE 26 A4   INC SAVFC+1
1166 E73C 20 24 EA   JD3 JSR CRCK ;(CR)
1167 E73F 4C 23 E7   JMP JD1
1168 E742 60          JD4 RTS
    
```

COMMANDS

PA00-J001A.....PAGE 0026

| LINE # | LOC | CODE | LINE |
|--------|------|----------|-----------------------------------------------------|
| 1170 | E743 | | ;INITIALIZATION TABLE FOR 6522 |
| 1171 | E743 | 34 | INTAB1 .BYT \$34,\$00,\$37,\$FF,\$25,\$FF,\$25,\$FF |
| 1171 | E744 | 00 | |
| 1171 | E745 | 37 | |
| 1171 | E746 | FF | |
| 1171 | E747 | 25 | |
| 1171 | E748 | FF | |
| 1171 | E749 | 25 | |
| 1171 | E74A | FF | |
| 1172 | E74B | FF | .BYT \$FF,\$FF,\$00,T11+T21 |
| 1172 | E74C | FF | |
| 1172 | E74D | 00 | |
| 1172 | E74E | 00 | |
| 1173 | E74F | E1 | .BYT MOFF+PRST+SP12,\$FF,\$7F |
| 1173 | E750 | FF | |
| 1173 | E751 | 7F | |
| 1174 | E752 | | ;INITIALIZATION TABLE FOR 6532 |
| 1175 | E752 | FF | INTAB2 .BYT \$FF,\$FF,\$00,\$00 |
| 1175 | E753 | FF | |
| 1175 | E754 | 00 | |
| 1175 | E755 | 00 | |
| 1176 | E756 | | ;INITIALIZATION TABLE FOR MONTITOR RAM |
| 1177 | E756 | 7B E0 | INTAB3 .WORD NMIV3,IRQV3,OUTDIS |
| 1177 | E75B | 54 E1 | |
| 1177 | E75A | 05 EF | |
| 1178 | E75C | C7 | .BYT \$C7,\$08,\$02,\$CA,\$03,\$80,\$00,\$00 |
| 1178 | E75D | 08 | |
| 1178 | E75E | 02 | |
| 1178 | E75F | CA | |
| 1178 | E760 | 03 | |
| 1178 | E761 | 80 | |
| 1178 | E762 | 00 | |
| 1178 | E763 | 00 | |
| 1179 | E764 | 00 | .BYT \$00,\$80,\$0D,\$0D,\$00,\$00,\$00 |
| 1179 | E765 | 80 | |
| 1179 | E766 | 0D | |
| 1179 | E767 | 0D | |
| 1179 | E768 | 00 | |
| 1179 | E769 | 00 | |
| 1179 | E76A | 00 | |
| 1180 | E76B | | ;SEE IF WE HIT A SOFT BREAKPOINT (PART OF NMV3) |
| 1181 | E76B | A2 07 | CKB LDX #7 ;COMPARE BRKPT LIST TO TRAP ADDR |
| 1182 | E76D | BD 00 01 | CKB2 LDA BKS,X ;GET ADDRESS OF NEXT BREAKPOINT |
| 1183 | E770 | CA | DEX |
| 1184 | E771 | CD 26 A4 | CMP SAVPC+1 ;COMPARE TO SAVED PROGRAM COUNTER |
| 1185 | E774 | DO 0A | BNE CKB1 |
| 1186 | E776 | BD 00 01 | LDA BKS,X |
| 1187 | E779 | CD 25 A4 | CMP SAVPC |
| 1188 | E77C | DO 02 | BNE CKB1 ;NO MATCH SO TRY NEXT BREAKPOINT |
| 1189 | E77E | 38 | SEC ;MATCH-SET MATCH FLAG |
| 1190 | E77F | 60 | RTS |
| 1191 | E780 | CA | CKB1 DEX |
| 1192 | E781 | 10 EA | BPL CKB2 ;MORE TO GO |
| 1193 | E783 | 18 | CLC ;NO MATCH -RESET MATCH FLAG |
| 1194 | E784 | 60 | RTS |

COMMANDS

PA00-J001A.....PAGE 0027

| LINE # | LOC | CODE | LINE |
|--------|------|----------|------------------------------------------------------|
| 1196 | E785 | | ;GET # OF LINES COUNT FOR GO-COMMAND,LIST-COMM |
| 1197 | E785 | 20 5D EA | GCNT JSR RD2 |
| 1198 | E788 | 90 02 | RCC GCN1 |
| 1199 | E78A | 49 0C | EOR #\$0C ;(SPACE)----> \$2C ,(CCR)----> \$01 |
| 1200 | E78C | 8D 19 A4 | GCN1 STA COUNT |
| 1201 | E78F | 60 | RTS |
| 1203 | E790 | | ;CHECK IF COUNT HAS REACHED ZERO |
| 1204 | E790 | | ;COUNT=\$2C MEANS FOREVER |
| 1205 | E790 | AD 19 A4 | DONE LDA COUNT ;IF COUNT=0 WE ARE DONE |
| 1206 | E793 | C9 2C | CMP #\$2C ;THIS MEANS FOR EVER |
| 1207 | E795 | F0 09 | BEQ DON1 ;SET ACC DIFF FROM ZERO |
| 1208 | E797 | FB | SEC ;DECREMENT COUNT IN DECIMAL |
| 1209 | E798 | 38 | SEC |
| 1210 | E799 | E9 01 | SBC #1 |
| 1211 | E79B | D8 | CLD |
| 1212 | E79C | 8D 19 A4 | STA COUNT |
| 1213 | E79F | 60 | RTS |
| 1214 | E7A0 | A9 2C | DON1 LDA #\$2C |
| 1215 | E7A2 | 60 | RTS |
| 1217 | E7A3 | A0 00 | FROM LDY #0 ;PRINT "FR=" |
| 1218 | E7A5 | F0 02 | BEQ T01 |
| 1220 | E7A7 | A0 05 | T0 LDY #M3-M1 ;PRINT "TO=" |
| 1221 | E7A9 | 20 AF E7 | T01 JSR KEP |
| 1222 | E7AC | 4C B1 EA | JMP ADDNE ;GET ADDRESS |
| 1224 | E7AF | | ;PRINT MSG POINTED TO BY Y REG |
| 1225 | E7AF | B9 00 E0 | KEP LDA M1,Y |
| 1226 | E7B2 | 48 | FHA |
| 1227 | E7B3 | 29 7F | ANDI #\$7F ;STRIP OFF MSB |
| 1228 | E7B5 | 20 7A E9 | JSR OUTPUT |
| 1229 | E7B8 | C8 | INY |
| 1230 | E7B9 | 68 | PLA |
| 1231 | E7BA | 10 F3 | BPL KEP ;MSB =1 ? |
| 1232 | E7BC | 60 | RTS |
| 1234 | E7BD | | ;PRINT "*" ,BUT NOT TO TAPE RECORDER, NOR LOADING... |
| 1235 | E7BD | | ;PAPER TAPE OR TO DISPLAY |
| 1236 | E7BD | AD 12 A4 | PROMPT LDA INFLG ;WHICH DEV (FOR EDITOR) |
| 1237 | E7C0 | C9 54 | CMP #'T ;NO PROMPT IF 'T' OR 'L' |
| 1238 | E7C2 | 4C EF FE | JMP FATC11 |
| 1239 | E7C5 | 20 42 E8 | PROMP1 JSR TTYST ;PROMPT ONLY TO TTY |
| 1240 | E7C8 | D0 05 | BNE PR2 ;BRANCH ON KB |
| 1241 | E7CA | A9 2A | LDA #'* |
| 1242 | E7CC | 4C 7A E9 | PR1 JMP OUTPUT ;ONLY TO TERMIN |
| 1243 | E7CF | A9 0D | PR2 LDA #CR ;CLR DISP |
| 1244 | E7D1 | 4C 05 EF | JMP OUTDIS |
| 1246 | E7D4 | A9 3F | QM LDA #'? ;PRINT "?" |
| 1247 | E7D6 | D0 F4 | BNE PR1 |
| 1249 | E7D8 | A9 3D | EQUAL LDA #'= ;PRINT "=" |
| 1250 | E7DA | D0 F0 | BNE PR1 |

COMMANDS

PA00-J001A.....PAGE 0028

| LINE # | LOC | CODE | LINE |
|--------|------|----------|--------------------------------------------|
| 1252 | E7DC | ? | ;ON DELETE KEY OUTPUT SLASH IF TTY & |
| 1253 | E7DC | | ;BACK UP CURSOR IF KB (MAY NEED SCROLLING) |
| 1254 | E7DC | 20 42 E8 | PSLS JSR TTYTST ;TTY OR KB ? |
| 1255 | E7DF | F0 56 | BEQ PSL1 ;BRANCH ON TTY |
| 1256 | E7E1 | 20 9E EB | JSR PHXY ;SAVE X,Y |
| 1257 | E7E4 | CE 15 A4 | DEC CURP02 ;DECR DISP PNTR |
| 1258 | E7E7 | AE 15 A4 | LIX CURP02 |
| 1259 | E7EA | E0 14 | CPX #20 ;IF MORE THAN 20 JUST SCROLL THEM |
| 1260 | E7EC | B0 0D | BCL PSL0 |
| 1261 | E7EE | A9 20 | LDA #' ;C 20 ,SO CLR CUR |
| 1262 | E7F0 | 20 02 EF | JSR OUTDP1 |
| 1263 | E7F3 | CE 15 A4 | DEC CURP02 |
| 1264 | E7F6 | 4C 02 E8 | JMP PSL00 |
| 1265 | E7F9 | EA | NOF |
| 1266 | E7FA | EA | NOF |
| 1267 | E7FB | 20 FB FE | PSL0 JSR PATC12 ;CLR PRIFLG |
| 1268 | E7FE | CA | DEX ;ONE CHR LESS |
| 1269 | E7FF | 20 2F EF | JSR OUTD2A ;SCROLL THEM |
| 1270 | E802 | AD 15 A4 | PSL00 LDA CURP02 ;DISBUFF----> PRIBUFF |
| 1271 | E805 | C9 15 | CMF #21 |
| 1272 | E807 | 90 13 | BCC PSL0R |
| 1273 | E809 | C9 29 | CMF #41 |
| 1274 | E80B | 90 07 | BCC PSL0A |
| 1275 | E80D | A0 2B | LIX #40 ;CHR 40-59 |
| 1276 | E80F | E9 2B | SBC #40 |
| 1277 | E811 | 4C 1E EB | JMP PSL0C |
| 1278 | E814 | A0 14 | PSL0A LIX #20 ;CHR 20-39 |
| 1279 | E816 | 38 | SEC |
| 1280 | E817 | E9 14 | SBC #20 |
| 1281 | E819 | 4C 1E EB | JMP PSL0C |
| 1282 | E81C | A0 00 | PSL0B LIX #0 ;CHR 00-19 |
| 1283 | E81E | BD 16 A4 | PSL0C STA CURPOS |
| 1284 | E821 | A2 00 | LIX #0 |
| 1285 | E823 | B9 38 A4 | PSL0D LDA IIBUFF,Y ;TRANSFER THEM |
| 1286 | E826 | 9D 60 A4 | STA IIBUFM,X |
| 1287 | E829 | E8 | INX |
| 1288 | E82A | C8 | INX |
| 1289 | E82B | EC 16 A4 | CPX CURPOS ;PRI PNTR |
| 1290 | E82E | 90 F3 | BCC PSL0D |
| 1291 | E830 | 20 38 F0 | JSR OUTPR ;CLR PRI BUFF TO THE RIGHT |
| 1292 | E833 | 20 AC EB | JSR PLXY ;RESTORE X,Y |
| 1293 | E836 | 60 | RTS |
| 1294 | E837 | A9 2F | PSL1 LDA #'/ ;PRINT '/' |
| 1295 | E839 | D0 91 | BNE PR1 |
| 1297 | E83B | 20 3E EB | BLANK2 JSR BLANK ;TWO SPACES |
| 1298 | E83E | A9 20 | BLANK LDA #' |
| 1299 | E840 | D0 8A | BNE PR1 |
| 1301 | E842 | | ;CHECK TTY/KRD SWITCH (Z=1 FOR TTY) |
| 1302 | E842 | A9 08 | TTYTST LDA #*0B ;CHECK IF TTY OR KB |
| 1303 | E844 | 2C 00 AB | BIT IIRB ;TTY OR KB SWICTH =PB3 |
| 1304 | E847 | 60 | RTS |
| 1306 | E84B | | ;WHERE IS INPUT COMING FROM? |

COMMANDS

PA00-J001A.....PAGE 0029

| LINE # | LOC | CODE | LINE |
|--------|------|-----------|---------------------------------------|
| 1307 | E84B | | ;SET UP FOR INPUT ACTIVE DEVICE |
| 1308 | E84B | A0 2A | WHEREI LDY #M9-M1 ;PRINT 'IN' |
| 1309 | E84A | 20 70 E9 | JSR KEPR ;OUTPUT MSG AND INPUT CHR |
| 1310 | E84D | BD 12 A4 | STA INFLG |
| 1311 | E850 | C9 54 | CMF #*T |
| 1312 | E852 | D0 08 | BNE WHE1 |
| 1313 | E854 | A2 00 | LIX #0 ;FOR INPUT FILE FLG |
| 1314 | E856 | 20 A2 E8 | JSR FNAM ;OPEN FILE FOR TAPE (1 OR 2) |
| 1315 | E859 | 4C 2F E3 | JMP LOADITA ;GET FILE |
| 1316 | E85C | C9 4B | WHE1 CMF #*K ;TAPE WITH KIM FORMAT |
| 1317 | E85E | D0 08 | BNE WHE2 |
| 1318 | E860 | A2 00 | LIX #0 ;FOR INPUT FILE FLG |
| 1319 | E862 | 20 A2 E8 | JSR FNAM ;OPEN FILE FOR TAP (1 OR 2) |
| 1320 | E865 | 4C A4 E3 | JMP LOADKI ;THE WHOLE FILE |
| 1321 | E868 | C9 55 | WHE2 CMF #*U ;USER RTN? |
| 1322 | E86A | D0 04 | BNE WHE3 |
| 1323 | E86C | 18 | CLC ;SET FLG FOR INITIALIZATION |
| 1324 | E86D | 6C 08 01 | JMP (UIN) ;USER INPUT SETUP |
| 1325 | E870 | 60 | WHE3 RTS |
| 1327 | E871 | | ;WHERE IS OUTPUT GOING TO? |
| 1328 | E871 | | ;SET UP FOR OUTPUT ACTIVE DEVICE |
| 1329 | E871 | A0 2D | WHEREO LDY #M10-M1 ;PRINT 'OUT' |
| 1330 | E873 | 20 70 E9 | JSR KEPR ;OUTPUT MSG & INPUT CHR |
| 1331 | E876 | BD 13 A4 | STA OUTFLG ;DEVICE FLG |
| 1332 | E879 | | ;TAPES |
| 1333 | E879 | C9 54 | CMF #*T |
| 1334 | E87B | D0 08 | BNE WHR01 |
| 1335 | E87D | A2 01 | LIX #1 ;FOR OUTPUT FILE FLG |
| 1336 | E87F | 20 A2 'EB | JSR FNAM ;FILENAME & TAPE (1 OR 2) |
| 1337 | E882 | 4C 6F E5 | JMP DUMFTA ;INITIALIZE FILE |
| 1338 | E885 | C9 4B | WHR01 CMF #*K ;TAPE WITH KIM FORMAT |
| 1339 | E887 | D0 05 | BNE WHR02 |
| 1340 | E889 | A2 01 | LIX #1 ;FOR OUTPUT FILE FLG |
| 1341 | E88B | 4C A2 E8 | JMP FNAM |
| 1342 | E88E | | ;PRINTER |
| 1343 | E88E | C9 50 | WHR02 CMF #*P ;PRINTER? |
| 1344 | E890 | D0 05 | BNE WHR03 |
| 1345 | E892 | A9 0D | LDA #CR ;OUTPUT LAST LINE IF ON |
| 1346 | E894 | 4C 00 F0 | JMP OUTPRI ;& CLEAR PRINTER PTR |
| 1347 | E897 | | ;USER SET UP |
| 1348 | E897 | C9 55 | WHR03 CMF #*U ;USER RTN? |
| 1349 | E899 | D0 04 | BNE WHR04 |
| 1350 | E89B | 18 | CLC ;CLR FLG FOR INITIALIZATION |
| 1351 | E89C | 6C 0A 01 | JMP (UOUT) ;USER OUTPUT SETUP |
| 1352 | E89F | | ;ANY OTHER |
| 1353 | E89F | 4C 13 EA | WHR04 JMP CRLOW |
| 1355 | E8A2 | | ;GET FILE NAME & TAPE UNIT |
| 1356 | E8A2 | 20 9E EB | FNAM JSR PHXY ;SAVE IN/OUT FLG (X) |
| 1357 | E8A5 | 20 CF EB | JSR NAMO ;GET NAME |
| 1358 | E8A8 | A0 50 | WHICHT LDY #*MSG2-M1 ;PRINT '*T=' |
| 1359 | E8AA | 20 70 E9 | JSR KEPR ;OUTPUT MSG & INPUT CHR |
| 1360 | E8AD | C9 0D | CMF #*CR |
| 1361 | E8AF | D0 02 | BNE TAP1 |


```

LINE # LOC      CODE      LINE
1362 EBR1  A9 31          LDA ##31      ;CCR) ==> TAPE 1
1363 EBR3  38          TAP1 SEC
1364 EBR4  E9 31          SRC ##31      ;SUBSTRACT 31
1365 EBR6  30 04          BHI TAP2      ;ONLY 1,2 OK
1366 EBR8  C9 02          CMP #2
1367 EBR8A 30 06          BHI TAP3      ;OK
1368 EBR8C 20 D4 E7        TAP2 JSR RM      ;ERROR
1369 EBR8F 4C A8 EB        JMP WHICHT
1370 EBC2  20 AC EB        TAP3 JSR FLXY      ;IN/OUT FLG
1371 EBC5  9D 34 A4        STA TAPIN,X   ;IF X=0 --->TAPIN (TAPE 1 OR 2)
1372 EBC8  20 83 FE        JSR CUREAD    ;GET ANYTHING
1373 EBC8  20 24 EA        JSR CRCK      ;CCR)
1374 EBC8  60          RTS           ;IF X=1 --->TAPOUT (TAPE 1 OR 2)

;GET FILE NAME
1376 EBCF          NAM0 LDY #TMSG1-M1 ;PRINT "F="
1377 EBCF  A0 4D          JSR KEP       ;NO CRLF
1378 EBD1  20 AF E7        LDY #0
1379 EBD4  A0 00          NAM01 JSR RDRUB    ;GET CHAR
1380 EBD6  20 5F E9        CMP #CR       ;DONE?
1381 EBD9  C9 0D          BEQ NAM02
1382 EBD9  F0 0C          CMP #'
1383 EBD9  C9 20          BEQ NAM02
1384 EBD9  F0 08          STA NAME,Y    ;STORE
1385 EBE1  99 2E A4        INY
1386 EBE4  C8          CPY #5
1387 EBE5  C0 05          BNE NAM01
1388 EBE7  D0 ED          ;BLANK REST OF NAME
1389 EBE9          NAM02 LDA #'
1390 EBE9  A9 20          NAM03 CPY #5
1391 EBE8  C0 05          BEQ NAM04
1392 EBE8  F0 06          STA NAME,Y
1393 EBEF  99 2E A4        INY
1394 EBF2  C8          BNE NAM03
1395 EBF3  D0 F6          NAM04 JMP BLANK
1396 EBF5  4C 3E E8

;SET INPUT FROM TERMINAL (KB OR TTY)
1398 EBF8          INLOW LDA #CR
1399 EBF8  A9 0D          STA INFLG
1400 EBF8  8D 12 A4        RTS
1401 EBF8  60

;SET I/O TO TERMINAL (KB & D/P ,OR TTY)
1403 EBF8          LL JSR INLOW
1404 EBF8  20 FB EB

;SET OUTPUT TO TERMINAL (D/P OR TTY)
1406 E901          OUTLOW LDA #CR
1407 E901  A9 0D          STA OUTFLG
1408 E903  8D 13 A4        OUTL1 RTS
1409 E906  60

;ON (ESCAPE) STOPS EXECUTION & BACK TO MONITOR
1411 E907          ;ON (SPACE) STOPS EXECUTION & CONTINUE ON ANY OTHER KEY
1412 E907          RCHK JSR TTYTST ;TTY OR KB ?
1413 E907  20 42 EB        BEQ RCHTTY
1414 E90A  F0 1A          JSR ROONEK    ;CLR MSK & GET A KEY
1415 E90C  20 EF EC        DEY
1416 E90F  8B

```

```

LINE # LOC      CODE      LINE
417 E910  30 13          BHI RCH3      ;RTN ON NO KEY
418 E912  A2 00          LDX #0
419 E914  20 82 EC        JSR GETK2     ;GET THE KEY
420 E917  C9 18          CMP #ESCAPE
421 E919  F0 3B          BEQ REA1     ;TO COMMAN & SET I/O TO TERMINAL
422 E918  C9 20          CMP #'
423 E91D  D0 06          BNE RCH3     ;RTN, IGNORE OTHER KEYS
424 E91F  20 EF EC        RCH2 JSR ROONEK ;WAIT TILL HE RELEASE IT &
425 E922  88          DEY         ;QUIT WAITING ON NEXT KEY
426 E923  30 FA          RMI RCH2
427 E925  60          RCH3 RTS
428 E926  70 13          RCHTTY BVS RCHT1 ;TTI=PB6 ---> V (OVERFL FLG)
429 E928  2C 00 AB        RCHT2 BIT DRB   ;WAIT TILL HE RELEASE IT
430 E928  50 FB          BVC RCHT2
431 E92D  20 0F EC        JSR DELAY
432 E930  20 DB ER        JSR GETTTY   ;GET A CHAR
433 E933  C9 1B          CMP #ESCAPE
434 E935  F0 1F          BEQ REA1     ;TO COMMAN
435 E937  C9 20          CMP #'
436 E939  D0 ED          BNE RCHT2
437 E938  60          RCHT1 RTS    ;QUIT WAITING ON ANY KEY

439 E93C          ;READ ONE CHAR FROM KB/TTY & PRESERVE X,Y
440 E93C  20 9E EB        READ JSR PHXY ;PUSH X & Y
441 E93F  20 42 EB        JSR TTYTST   ;TTY OR KB ?
442 E942  D0 06          BNE READ1
443 E944  20 DB ER        JSR GETTTY
444 E947  4C 4D E9        JMP READ2
445 E94A  20 40 EC        READ1 JSR GETKEY
446 E94D  20 AC EB        READ2 JSR PLXY  ;PULL X & Y
447 E950  29 7F          AND #7F     ;STRIP PARITY
448 E952  C9 1B          CMP #ESCAPE
449 E954  D0 E5          BNE RCHT1   ;RTN
450 E956  20 3D FF        REA1 JSR PATC1B ;CCR) & CLR BUFFERS
451 E959  4C A1 E1        JMP COMIN   ;BOTH I/O TO TERMINAL

453 E95C          ;READ WITH RUBOUT OR DELETE POSSIBLE
454 E95C  20 DC E7        RB2 JSR PSL5 ;SLASH OR BACK SPACE
455 E95F  20 83 FE        RDRUB JSR CUREAD
456 E962  C9 08          CMP #RUB    ;RUBOUT
457 E964  F0 04          BEQ RDR1
458 E966  C9 7F          CMP #7F     ;ALSO DELETE
459 E968  D0 0C          BNE RED2    ;ECHO IF NOT (CR)
460 E96A          ;RUBOUT TO DELETE CHAR
461 E96A  88          RDR1 DEY
462 E96B  10 EF          BPL RB2
463 E96D  C8          INY
464 E96E  F0 EF          BEQ RDRUB

466 E970          ;OUTPUT MESSAGE THEN INPUT CHR
467 E970  20 AF E7        KEPR JSR KEP

469 E973          ;READ AND ECHO A CHAR FROM KB OR TTY
470 E973  20 83 FE        REDOUT JSR CUREAD
471 E976  C9 0D          RED2 CMP #CR

```

```

LINE # LOC      CODE      LINE
1472 E97B F0 C1          BEQ RCHT1          ;DO NOT ECHO (CR)
1474 E97A          ;OUTPUTS A CHAR TO EITHER TTY OR D/P
1475 E97A -4B          OUTPUT PHA          ;SAVE IT
1476 E97B AD 11 A4      OUT1 LDA PRIFLG     ;IF LSR=1 OUTPUT ONLY TO DISP
1477 E97E 29 01          AND **01
1478 E980 F0 04          BEQ OUT1A
1479 E982 68          PLA
1480 E983 4C 02 EF      JMP OUTDP1         ;ONLY TO DISPL
1481 E984 20 42 EB      OUT1A JSR TTYTST    ;TTY OR KB ?
1482 E989 D0 04          BNE OUT2
1483 E98B 68          PLA
1484 E98C 4C AB EE      JMP OUTTTY         ;TO TTY
1485 E98F 68          OUT2 PLA
1486 E990 4C FC EE      JMP OUTDP          ;TO DISP & PRINTR

1488 E993          ;GET A CHR FROM CURRENT INPUT DEVICE (SET ON INFLG)
1489 E993 AD 12 A4      INALL LDA INFLG
1490 E996 C9 54          CMP #'T
1491 E998 D0 03          BNE *+5
1492 E99A 4C 3B ED      JMP TIBYTE         ;CHAR FROM BUFFER
1493 E99D C9 4B          CMP #'K            ;WITH KIM FORMAT
1494 E99F D0 03          BNE *+5
1495 E9A1 4C 29 EE      JMP GETTAP         ;DIRECTLY FROM TAPE
1496 E9A4 C9 4D          CMP #'M            ;MEMORY FOR ASH?
1497 E9A6 D0 03          BNE *+5
1498 E9A8 4C D0 FA      JMP MREAD
1499 E9AB C9 55          CMP #'U            ;USER ROUTINE?
1500 E9AD D0 04          BNE *+6
1501 E9AF 3B          SEC                ;SET FLG FOR NORMAL INPUT
1502 E9B0 6C 0B 01      JMP (UIN)
1503 E9B3 C9 4C          CMP #'L            ;TO LOAD PPR TAPE
1504 E9B5 D0 AB          BNE RDRUB
1505 E9B7 4C DB EB      JMP GETTTY         ; FROM TTY

1507 E9BA          .FILE A2
1508 E9BA A9 3B          SEMI LDA #' ;      ;OUTPUT A ' ;'
1509 E9BC          ;WRITE A CHR TO OUTPUT DEVICE (SET ON OUTFLG)
1510 E9BC 4B          OUTALL PHA
1511 E9BD AD 13 A4      LDA OUTFLG
1512 E9C0          ;TAPE BY BLOCKS
1513 E9C0 C9 54          CMP #'T            ;TAPES ?
1514 E9C2 D0 04          BNE OUTA1
1515 E9C4 68          PLA
1516 E9C5 4C 8B F1      JMP TORYTE         ;OUTPUT ONE CHAR TO TAPE BUFFER
1517 E9C8          ;TAPE KIM FORMAT
1518 E9C8 C9 4B          OUTA1 CMP #'K         ;KIM-1 ?
1519 E9CA D0 04          BNE OUTA2
1520 E9CC 68          PLA
1521 E9CD 4C 4A F2      JMP OUTTAP
1522 E9D0          ;PRINTER
1523 E9D0 C9 50          OUTA2 CMP #'P         ;PRINTER ?
1524 E9D2 D0 0E          BNE OUTA3
1525 E9D4 3B          SEC                ;TURN PRINTR ON
1526 E9D5 6E 11 A4      ROR PRIFLG
    
```

```

LINE # LOC      CODE      LINE
27 E9D8 68          PLA
28 E9D9 0B          PHP
29 E9DA 20 00 F0      JSR OUTPRI
30 E9DD 2B          PLP
31 E9DE 2E 11 A4      ROL PRIFLG         ;RESTORE FLG
32 E9E1 60          RTS
33 E9E2          ;USER DEFINED
34 E9E2 C9 55          OUTA3 CMP #'U         ;USER ROUTINE?
35 E9E4 D0 04          BNE OUTA4
36 E9E6 3B          SEC                ;SET FLG FOR NORMAL OUTPUT
37 E9E7 6C 0A 01      JMP (UOUT)         ;YES
38 E9EA          ;NOWHERE OR TO TTY ,D/P
39 E9EA C9 5B          OUTA4 CMP #'X         ;EAT IT?
40 E9EC D0 8D          BNE OUT1           ;OUTPUT TO TTY OR D/P
41 E9EE 68          PLA
42 E9EF 60          RTS

44 E9F0          ;THIS ROUTINE OUTPUTS A CRLF TO ANY OUTPUT DEV
45 E9F0          ;LF AND NULL IS SENT ONLY TO TTY
46 E9F0 A9 0D          CRLF LDA #CR
47 E9F2 20 BC E9      JSR OUTALL
48 E9F5 20 42 EB      JSR TTYTST         ;TTY OR KB ?
49 E9F8 D0 29          BNE CR2J
50 E9FA AD 13 A4      LDA OUTFLG         ;LF ONLY TO TTY
51 E9FD C9 54          CMP #'T
52 E9FF F0 22          BEQ CR2J
53 EA01 C9 4B          CMP #'K
54 EA03 F0 1E          BEQ CR2J
55 EA05 C9 50          CMP #'P
56 EA07 F0 1A          BEQ CR2J
57 EA09 A9 0A          LDA #LF
58 EA0B 20 BC E9      JSR OUTALL
59 EA0E A9 FF          LDA #NULLC
60 EA10 4C BC E9      JMP OUTALL

62 EA13          ;CRLF TO TERMINAL (TTY OR D/P) ONLY
63 EA13 4B          CRLOW PHA           ;SAVE A
64 EA14 AD 13 A4      LDA OUTFLG
65 EA17 4B          PHA
66 EA18 20 01 E9      JSR OUTLOW
67 EA1B 20 F0 E9      JSR CRLF
68 EA1E 68          PLA
69 EA1F 8D 13 A4      STA OUTFLG
70 EA22 68          PLA
71 EA23 60          CR2J RTS

73 EA24          ;OUTPUT (CR) TO TTY IF SWITCH ON TTY & INFLG NOT L
74 EA24          ;DONT CLR DISPLAY BUT CLEARS PNTRS FOR NEXT LINE
75 EA24          ;IF PRNTR HAS PRINTED ON 21ST CHR DONT OUTPUT (CR)
76 EA24 AD 12 A4      CRCK LDA INFLG     ;NO (CR) IF 'L'
77 EA27 C9 4C          CMP #'L
78 EA29 D0 01          BNE CRCK1
79 EA2B 60          RTS
80 EA2C 20 42 EB      CRCK1 JSR TTYTST       ;CHECK IF TTY OR KB
81 EA2F F0 E2          BEQ CRLOW         ;BRNCH IF TTY
    
```

COMMANDS

PA00-J001A.....PAGE 0034

| LINE # | LOC | CODE | LINE |
|--------|------|----------|----------------------------------------------------|
| 1582 | EA31 | | ; IF PRINTR PTR=0 ,DO NOT CLR PRI |
| 1583 | EA31 | AD 16 A4 | LDA CURPOS |
| 1584 | EA34 | F0 05 | BEQ CRCK2 ;IF PTR=0 ,NO (CR) |
| 1585 | EA36 | A9 0D | LDA #CR |
| 1586 | EA38 | 20 00 F0 | JSR OUTPRI |
| 1587 | EA3B | A9 8D | CRCK2 LDA ##BD ;(CR) ONLY FOR TV |
| 1588 | EA3D | 4C 02 EF | JMP OUTDP1 |
| 1589 | EA40 | EA | NOP |
| 1590 | EA41 | EA | NOP |
| 1592 | EA42 | | ;WRITE A THEN X IN ASCII TO THE OUTPUT DEV |
| 1593 | EA42 | 20 46 EA | WRAX JSR NUMA |
| 1594 | EA45 | 8A | TXA |
| 1596 | EA46 | | ;PRINT ONE BYTE=TWO ASCII CHARS TO OUTPUT DEVICE |
| 1597 | EA46 | 4B | NUMA PHA |
| 1598 | EA47 | 4A | LSR A |
| 1599 | EA48 | 4A | LSR A |
| 1600 | EA49 | 4A | LSR A |
| 1601 | EA4A | 4A | LSR A |
| 1602 | EA4B | 20 51 EA | JSR NOUT |
| 1603 | EA4E | 68 | FLA |
| 1604 | EA4F | 29 0F | ANDI ##F |
| 1605 | EA51 | 18 | NOUT CLC |
| 1606 | EA52 | 69 30 | ADC ##30 |
| 1607 | EA54 | C9 3A | CMP ##3A |
| 1608 | EA56 | 90 02 | BCC LT10 |
| 1609 | EA58 | 69 06 | ADC #6 ;CARRY IS SET |
| 1610 | EA5A | 4C BC E9 | LT10 JMP OUTALL |
| 1612 | EA5D | | ;READ TWO CHR & PACK THEM INTO ONE BYTE |
| 1613 | EA5D | | ;PART OF ALTER MEMORY , / COMM |
| 1614 | EA5D | 20 73 E9 | RI2 JSR REIDOUT |
| 1615 | EA60 | C9 0D | CMP ##D ;(CR)? |
| 1616 | EA62 | F0 17 | BEQ RSPAC |
| 1617 | EA64 | C9 20 | CMP #' ;FOR MEMORY ALTER |
| 1618 | EA66 | F0 13 | BEQ RSPAC |
| 1619 | EA68 | C9 2E | CMP #' ;TREAT * . AS (SPACE) |
| 1620 | EA6A | D0 04 | BNE RD1 |
| 1621 | EA6C | A9 20 | LDA ##20 |
| 1622 | EA6E | D0 0B | BNE RSPAC |
| 1623 | EA70 | 20 84 EA | RI1 JSR PACK |
| 1624 | EA73 | B0 06 | BCC RSPAC |
| 1625 | EA75 | 20 73 E9 | JSR REIDOUT |
| 1626 | EA78 | 4C B4 EA | JMP PACK |
| 1627 | EA7B | | ;WAS SPACE OR (CR) |
| 1628 | EA7B | 3B | RSPAC SEC |
| 1629 | EA7C | 60 | RTS |
| 1631 | EA7D | | ;CONVERT ACC IN ASCII TO ACC IN HEX (4 MSB=0) |
| 1632 | EA7D | 4B | HEX PHA ;SAVE A |
| 1633 | EA7E | A9 00 | LDA #0 ;CLEAR STIY IF HEX |
| 1634 | EA80 | B0 29 A4 | STA STIY+2 ;BECAUSE ONLY ONCE |
| 1635 | EA83 | 6B | FLA |
| 1636 | EA84 | | ;PACK TWO ASCII INTO ONE HEX (CALL SUBR TWO TIMES) |

COMMANDS

PA00-J001A.....PAGE 0035

| LINE # | LOC | CODE | LINE |
|--------|------|----------|---------------------------------------------------|
| 637 | EAB4 | | ;RESULT IS GIVEN ON ACC WITH FIRST CHR INTO 4 MSB |
| 638 | EAB4 | C9 30 | PACK CMP ##30 ;C 30 ? |
| 639 | EAB6 | 90 F3 | BCC RSPAC |
| 640 | EAB8 | C9 47 | CMP ##47 ;C 47 ? |
| 641 | EABA | B0 EF | BCC RSPAC |
| 642 | EABC | C9 3A | CMP ##3A ;C #10 |
| 643 | EABE | 90 06 | BCC FAK1 |
| 644 | EA90 | C9 40 | CMP ##40 ;C #10 ? |
| 645 | EA92 | 90 E7 | BCC RSPAC |
| 646 | EA94 | 69 0B | ADC #B ;ADD 9 IF LETTER (C IS SET) |
| 647 | EA96 | 2A | PAK1 ROL A ;SHIFT A 4 TIMES |
| 648 | EA97 | 2A | ROL A |
| 649 | EA98 | 2A | ROL A |
| 650 | EA99 | 2A | ROL A |
| 651 | EA9A | BE 2D A4 | STX CPIY+3 ;SAVE X |
| 652 | EA9D | A2 04 | LDX #4 |
| 653 | EA9F | 2A | PAK2 ROL A ;TRANSFER A TO STIY |
| 654 | EAA0 | 2E 29 A4 | ROL STIY+2 ;THRU CARRY |
| 655 | EAA3 | CA | DEX |
| 656 | EAA4 | D0 F9 | BNE FAK2 |
| 657 | EAA6 | AE 2D A4 | LDX CPIY+3 ;REST X |
| 658 | EAA9 | AD 29 A4 | LDA STIY+2 |
| 659 | EAAC | 18 | CLC |
| 660 | EAAE | 60 | RTS |
| 662 | EAAE | | ;GET FOUR BYTE ADDR ,TAKE LAST FOUR CHR TO... |
| 663 | EAAE | | ;CALULATE ADDR ,ALLOW DELETE ALSO |
| 664 | EAAE | 20 D8 E7 | ADDIN JSR EQUAL |
| 665 | EAB1 | AD 15 A4 | ADDNE LDA CURFO2 ;SAVE POSITION |
| 666 | EAB4 | 4B | PHA |
| 667 | EAB5 | A0 00 | LDY #0 |
| 668 | EAB7 | 20 5F E9 | ADDN1 JSR RDRUB |
| 669 | EABA | C9 0D | CMP #CR |
| 670 | EABC | F0 09 | BEQ ADDN2 |
| 671 | EABE | C9 20 | CMP #' |
| 672 | EAC0 | F0 05 | BEQ ADDN2 |
| 673 | EAC2 | C8 | INY |
| 674 | EAC3 | C0 0B | CPY #11 ;ALLOW 10 |
| 675 | EAC5 | 90 F0 | BCC ADDN1 |
| 676 | EAC7 | 68 | ADDN2 PLA |
| 677 | EAC8 | 8D 2D A4 | STA CPIY+3 ;SAVE |
| 678 | EACB | C0 00 | CPY #0 ;IF FIRST CHR PUT DEFAULT VALUES |
| 679 | EACD | D0 0D | BNE ADDN3 |
| 680 | EACF | A9 02 | LDA ##02 |
| 681 | EAD1 | 8D 1D A4 | STA ADDR+1 ;DEFAULT OF 0200 |
| 682 | EAD4 | 8D 1E A4 | STA CKSUM ;DEFAULT |
| 683 | EAD7 | 8C 1C A4 | STY ADDR |
| 684 | EADA | 18 | CLC |
| 685 | EADB | 60 | RTS |
| 686 | EADC | A2 00 | ADDN3 LDX #0 |
| 687 | EAD6 | 8B | DEY ;Y--4 |
| 688 | EADF | 8B | DEY |
| 689 | EAE0 | 8B | DEY |
| 690 | EAE1 | 8B | DEY |
| 691 | EAE2 | 10 13 | BFL ADDN5 ;BRNCH IF > 4 CHR |

COMMANDS

PA00-J001A.....PAGE 0036

| LINE # | LOC | CODE | LINE |
|--------|------|----------|----------------------------------------------|
| 1692 | EAE4 | 98 | TYA |
| 1693 | EAE5 | 49 FF | EOR **\$FF |
| 1694 | EAE7 | A8 | TAY |
| 1695 | EAE8 | A9 30 | ADDN4 LDA **30 ;# OF LEADING 0 |
| 1696 | EAEA | 9D 1C A4 | STA ADDR, X |
| 1697 | EAE0 | E8 | INX |
| 1698 | EAE6 | 88 | DEY |
| 1699 | EAEF | 10 F7 | BPL ADDN4 |
| 1700 | EAF1 | AC 2D A4 | LDY CFIY+3 ;NOW THE CHR |
| 1701 | EAF4 | 4C FD EA | JMP ADDN6 |
| 1702 | EAF7 | 98 | ADDN5 TYA ;PUT CHR |
| 1703 | EAF8 | 18 | CLC |
| 1704 | EAF9 | 6D 2D A4 | ADC CFIY+3 |
| 1705 | EAF0 | A8 | TAY |
| 1706 | EAFD | B9 38 A4 | ADDN6 LDA DIBUFF, Y ;FROM DISP BUFF |
| 1707 | EB00 | 9D 1C A4 | STA ADDR, X |
| 1708 | EB03 | C8 | INX |
| 1709 | EB04 | EB | INX |
| 1710 | EB05 | E0 04 | CPX #4 |
| 1711 | EB07 | D0 F4 | BNE ADDN6 |
| 1712 | EB09 | A2 01 | LDX #1 |
| 1713 | EB0B | A0 00 | LDY #0 ;CNVRT CHR TO HEX |
| 1714 | EB0D | B9 1C A4 | ADDN7 LDA ADDR, Y |
| 1715 | EB10 | 20 7D EA | JSR HEX |
| 1716 | EB13 | 80 16 | BCC ADDN8 |
| 1717 | EB15 | C8 | INX |
| 1718 | EB16 | B9 1C A4 | LDA ADDR, Y |
| 1719 | EB19 | C8 | INX |
| 1720 | EB1A | 20 84 EA | JSR PACK ;PACK TWO CHRS INTO 1 BYTE |
| 1721 | EB1D | 80 0C | BCC ADDN8 ;BRCNH IF ERROR |
| 1722 | EB1F | 9D 1C A4 | STA ADDR, X |
| 1723 | EB22 | CA | DEX |
| 1724 | EB23 | 10 EB | BPL ADDN7 |
| 1725 | EB25 | EB | INX ;X=0 |
| 1726 | EB26 | BE 1E A4 | STX CKSUM ;TO INDICATE WE GOT AN ADDR |
| 1727 | EB29 | 18 | CLC ;NO INVALID CHARS |
| 1728 | EB2A | 60 | RTS |
| 1729 | EB2B | 20 94 E3 | ADDN8 JSR CKERR0 ;OUTPUT ERROR MSG |
| 1730 | EB2E | 20 24 EA | JSR CRCK ;(CR) |
| 1731 | EB31 | 38 | SEC ;SET CARRY FOR INVALID CHR |
| 1732 | EB32 | 60 | RTS |
| 1734 | ER33 | | ;MEMORY FAIL TO WRITE MSG & SPECIFIC ADDRESS |
| 1735 | ER33 | 20 24 EA | MEMERR JSR CRCK |
| 1736 | ER36 | 20 CD E2 | JSR NXTADD ;ADD Y TO ADDR+1, ADDR |
| 1737 | ER39 | A0 31 | LDY #M11-M1 ;PRINT "MEM FAIL" |
| 1738 | ER3B | 20 AF E7 | JSR KEP ;FAIL MSG |
| 1739 | ER3E | 20 DB E2 | JSR WRITAZ ;PRINT ADDR+1, ADDR |
| 1740 | ER41 | 4C A1 E1 | JMP COMIN |
| 1742 | ER44 | | ;CLEAR DISPLAY & PRINTER POINTERS |
| 1743 | ER44 | A9 00 | CLR LDA #0 |
| 1744 | ER46 | BD 15 A4 | STA CURPO2 ;DISP PNTR |
| 1745 | ER49 | BD 16 A4 | STA CURPOS ;PRINTR PNTR |
| 1746 | ER4C | 60 | RTS |

COMMANDS

PA00-J001A.....PAGE 0037

| LINE # | LOC | CODE | LINE |
|--------|------|----------|-------------------------------------------------------|
| 748 | EB4D | | ;CLEAR CKSUM |
| 749 | EB4D | A9 00 | CLRCK LDA #0 |
| 750 | EB4F | BD 1F A4 | STA CKSUM+1 |
| 751 | EB52 | 8D 1E A4 | STA CKSUM |
| 752 | EB55 | 60 | RTS |
| 754 | EB56 | | ;CODE FOR PAGE ZERO SIMULATION |
| 755 | EB56 | | ;SUBR LDAY--SIMULATES LDA (N),Y INSTR WITHOUT PAG 0 |
| 756 | EB56 | | ;BY PUTTING INDIR ADDR INTO RAM & THEN EXEC LDA NM, Y |
| 757 | ER56 | A9 25 | PCLLD LDA *CSAVFC ;FOR DISASSEMBLER |
| 758 | EB58 | 8C 2D A4 | LDAY STY CFIY+3 ;SAVE Y |
| 759 | EB58 | AB | TAY |
| 760 | EB5C | B9 00 A4 | LDA MONRAM, Y ;MONRAM=MONITOR RAM |
| 761 | EB5F | 8D 2B A4 | STA LDY+1 |
| 762 | EB62 | B9 01 A4 | LDA MONRAM+1, Y |
| 763 | EB65 | 8D 2C A4 | STA LDY+2 |
| 764 | EB68 | AC 2D A4 | LDY CFIY+3 ;REST Y |
| 765 | EB6B | A9 B9 | LDA **B9 ;INST FOR LDA NM, Y |
| 766 | EB6D | 8D 2A A4 | STA LDY |
| 767 | EB70 | A9 60 | LDA **60 ;RTS |
| 768 | EB72 | 8D 2D A4 | STA LDY+3 |
| 769 | EB75 | 4C 2A A4 | JMP LDY ;START EXECUTING LDA (),Y |
| 771 | EB78 | | ;SUBR STORE AT ADDR & CMP WITHOUT PAG 0 |
| 772 | EB78 | | ;REPLACES STA (ADDR),Y & CMP (ADDR),Y |
| 773 | EB78 | | ;LOOK THAT ADDR & ADDR+1 ARE NOT ON PAG 0 |
| 774 | EB78 | 48 | SADDR PHA |
| 775 | EB79 | AD 1C A4 | LDA ADDR |
| 776 | EB7C | 8D 28 A4 | STA STIY+1 |
| 777 | EB7F | 8D 2B A4 | STA CFIY+1 |
| 778 | EB82 | AD 1D A4 | LDA ADDR+1 |
| 779 | EB85 | 8D 29 A4 | STA STIY+2 |
| 780 | EB88 | 8D 2C A4 | STA CFIY+2 |
| 781 | EB8B | A9 99 | LDA **99 ;STA INSTR |
| 782 | EB8D | 8D 27 A4 | STA STIY |
| 783 | EB90 | A9 D9 | LDA **D9 ;CMP INSTR |
| 784 | EB92 | 8D 2A A4 | STA CFIY |
| 785 | EB95 | A9 60 | LDA **60 ;RTS |
| 786 | EB97 | 8D 2D A4 | STA CFIY+3 |
| 787 | EB9A | 68 | PLA |
| 788 | EB9B | 4C 27 A4 | JMP STIY ;START EXECUTING STA (),Y |
| 790 | EB9E | | ;PUSH X & Y WITHOUT CHANGING THE REGS |
| 791 | EB9E | 8D 2D A4 | PHXY STA CFIY+3 ;SAVE ACC |
| 792 | EBA1 | 98 | TYA |
| 793 | EBA2 | 48 | PHA ;PUSH Y |
| 794 | EBA3 | 8A | TXA |
| 795 | EBA4 | 48 | PHA ;PUSH X |
| 796 | EBAS | 20 BA EB | JSR SWSTAK ;SWAP X, Y WITH RTN ADDR FROM SL |
| 797 | EBAB | AD 2D A4 | LDA CFIY+3 |
| 798 | EBAB | 60 | RTS |
| 800 | EBAC | | ;PULL X & Y WITHOUT CHANGING ACC |
| 801 | EBAC | | ;IT HAS TO BE CALLED BY JSR & NOT BY JMP INSTR |
| 802 | EBAC | | ;SINCE IT SWAPS THE STACK |

```

LINE # LOC      CODE      LINE
1803 ERAC 8D 2D A4    PLXY STA CPIY+3
1804 ERAF 20 BA EB    JSR SWSTAK ;SWAP X , Y WITH RTRN ADDR FROM
1805 EBB2 68          PLA
1806 EBB3 AA          TAX ;PULL X
1807 EBB4 68          PLA
1808 EBB5 AB          TAY ;PULL Y
1809 EBB6 AD 2D A4    LDA CPIY+3
1810 EBB9 60          RTS

1812 EBBA          ;SWAP STACK
1813 EBBA BA          SWSTAK TSX
1814 EBBB A9 02      LDA #2
1815 EBBD 48          SWST1 PHA
1816 EBBE 8D 06 01  LDA #0106,X ;GET PCH OR PCL
1817 EBC1 8C 04 01  LDY #0104,X ;GET Y OR X REGS
1818 EBC4 9D 04 01  STA #0104,X
1819 EBC7 98          TYA
1820 EBC8 9D 06 01  STA #0106,X
1821 EBCB CA          DEX
1822 EBCC 68          PLA
1823 EBCE 38          SEC
1824 EBCE E9 01      SBC #1
1825 EBD0 D0 EB      RNE SWST1
1826 EBD2 8D 08 01  LDA #0108,X ;RESTORE Y & X FROM STACK
1827 EBD5 AB          TAY
1828 EBD6 8D 07 01  LDA #0107,X
1829 EBD9 AA          TAX
1830 EBDA 60          RTS
    
```

```

LINE # LOC      CODE      LINE
1832 EBD8          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1833 EBD8          ;GET A CHAR FROM TTY SUBR INTO ACC ,SAVES X
1834 EBD8 BA          GETTY TXA ;SAVE X
1835 EBD8 48          PHA
1836 EBD8 A2 07      LDX #*07 ;SET UP FOR 8 BIT CNT
1837 EBD8 8E 2A A4   STX CPIY ;CLEAR MSR
1838 EBE2 2C 00 AB   GET1 BIT IRR ;A^M , PB6-DV
1839 EBE5 70 FB      BUS GET1 ;WAIT FOR START BIT
1840 EBE7 20 0F EC   JSR DELAY ;DELAY 1 BIT
1841 EBEA 20 23 EC   JSR DEHALF ;DELAY 1/2 BIT TIME
1842 EBED AD 00 AB   GET3 LDA BRB ;GET 8 BITS
1843 EBF0 29 40      ANDI #*40 ;MASK OFF OTHER BITS,ONLY PB6
1844 EBF2 4E 2A A4   LSR CPIY ;SHIFT RIGHT CHARACTER
1845 EBF5 0B 2A A4   ORA CPIY
1846 EBF8 8D 2A A4   STA CPIY
1847 EBF8 20 0F EC   JSR DELAY ;DELAY 1 BIT TIME
1848 EBF8 CA          DEX
1849 EBF8 D0 EC      RNE GET3 ;GET NEXT BIT
1850 EC01 20 0F EC   JSR DELAY ;DO NOT CARE FOR PARITY BIT
1851 EC04 20 23 EC   JSR DEHALF ;UNTIL WE GET BACK TO ONE AGAIN
1852 EC07 68          PLA ;RESTORE X
1853 EC08 AA          TAX
1854 EC09 AD 2A A4   LDA CPIY
1855 EC0C 29 7F      ANDI #*7F ;CLEAR PARITY BIT
1856 EC0E 60          RTS

1858 EC0F          ;DELAY 1 BIT TIME AS GIVEN BY BAUD RATE
1859 EC0F AD 18 A4   DELAY LDA CNL30 ;START TIMER T2
1860 EC12 8D 08 AB   STA T2L
1861 EC15 AD 17 A4   LDA CNTH30
1862 EC18 8D 09 AB   DE1 STA T2H
1863 EC1B AD 0D AB   DE2 LDA IFR ;GET INT FLG FOR T2
1864 EC1E 29 20      ANDI #MT2
1865 EC20 F0 F9      BEQ DE2 ;TIME OUT ?
1866 EC22 60          RTS

1868 EC23          ;DELAY HALF BIT TIME
1869 EC23          ;TOTAL TIME DIVIDED BY 2
1870 EC23 AD 17 A4   DEHALF LDA CNTH30
1871 EC26 4A          LSR A ;LSB TO CARRY
1872 EC27 AD 18 A4   LDA CNL30
1873 EC2A 6A          ROR A ;SHIFT WITH CARRY
1874 EC2B 8D 08 AB   STA T2L
1875 EC2E AD 17 A4   LDA CNTH30
1876 EC31 4A          LSR A
1877 EC32 8D 09 AB   STA T2H
1878 EC35 4C 1B EC   JMP DE2

1880 EC38          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1881 EC38 A9 00      GETKD0 LDA #0
1882 EC3A 8D 77 A4   STA IDOT ;GO ANOTHER 90 DOTS
1883 EC3D 20 50 F0   JSR IP00 ;OUTPUT 90 DOTS TO PRI (ZEROS)

1885 EC40          ;GET A CHAR FROM KB SUBROUTINE
1886 EC40          ;FROM KB Y=ROW , STKEY=COLUMNS (STROBE)
    
```

| LINE # | LOC | CODE | LINE |
|--------|------|----------|------------------------------------------------|
| 1887 | EC40 | | ;X=CTRL OR SHIFT ,OTHERWISE X=0 |
| 1888 | EC40 | 20 EF EC | GETKEY JSR ROONEK ;WAIT IF LAST KEY STILL DOWN |
| 1889 | EC43 | 20 2A ED | GETKY JSR DEBKEY ;DEBOUNCE KEY (5 MSEC) |
| 1890 | EC46 | | ;CTRL OR SHIFT ? |
| 1891 | EC46 | A9 BF | LDA #5BF ;CHCK CLMN 5,6,7 |
| 1892 | EC48 | BD 80 A4 | STA DRA2 |
| 1893 | EC48 | AD B2 A4 | LDA DRB2 ;CHCK ROW 1 |
| 1894 | EC4E | 4A | LSR A |
| 1895 | EC4F | B0 20 | BCC GETK1 ;IF=1 ,NO CTRL OR SHIFT |
| 1896 | EC51 | A2 03 | LDX #3 ;CLMN 5,6,7 (CNTRL,SHIFTL,SHIFTR) |
| 1897 | EC53 | A9 7F | LDA #57F ;CTRL OR SHIFT ,SO WHICH ONE ? |
| 1898 | EC55 | 38 | GETK0 SEC |
| 1899 | EC56 | 6A | ROR A |
| 1900 | EC57 | 4B | PHA |
| 1901 | EC58 | 20 0B ED | JSR ONEK2 ;LETS GET CTRL OR SHIFT INTO X |
| 1902 | EC5B | AD B2 A4 | LDA DRB2 |
| 1903 | EC5E | 4A | LSR A ;ONLY ROW 1 |
| 1904 | EC5F | 90 06 | BCC GETK00 ;GOT YOU |
| 1905 | EC61 | 68 | PLA |
| 1906 | EC62 | CA | DEX |
| 1907 | EC63 | D0 F0 | BNE GETK0 |
| 1908 | EC65 | F0 DC | BEQ GETKY ;THERE IS A MISTAKE CHECK AGAIN |
| 1909 | EC67 | 68 | GETK00 PLA ;NOW GET STRKEY INTO X |
| 1910 | EC68 | AD 2B A4 | LDA STBKEY ;CLMN INTO X |
| 1911 | EC6B | 49 FF | EDR #5FF ;COMPLEMENT BECAUSE STRBS ARE 0 |
| 1912 | EC6D | AA | TAX ;CTRL OR SHIFT TO X |
| 1913 | EC6E | EE 2A A4 | INC KMASK ;SET MSK=#01 |
| 1914 | EC71 | | ;NOW GET ANY KEY |
| 1915 | EC71 | 20 05 ED | GETK1 JSR ONEKEY ;GET A KEY |
| 1916 | EC74 | 88 | DEY ;CHK THE ROW (1-8) |
| 1917 | EC75 | D0 09 | BNE GETK1B ;CHCK IF CTRL OR SHIFT |
| 1918 | EC77 | AD 2B A4 | LDA STBKEY ;WERE ENTERED AT THE LAST MOMENT |
| 1919 | EC7A | C9 F7 | CMF #5F7 ;IF CLMN 5,6,7,8 DO IT AGAIN |
| 1920 | EC7C | B0 04 | BCC GETK2 |
| 1921 | EC7E | 90 C3 | BCC GETKY ;SEND IT TO GET CTRL OR SHIFT |
| 1922 | EC80 | 30 C1 | GETK1B BMI GETKY ;NO KEY ,CLEAR MSK |
| 1923 | EC82 | | ;WE HAVE A KEY ,DECODE IT |
| 1924 | EC82 | 20 2C ED | GETK2 JSR DEBK1 ;DEBOUNCE KEY (5 MSEC) |
| 1925 | EC85 | 98 | TYA ;MULT BY 8 |
| 1926 | EC86 | 0A | ASL A |
| 1927 | EC87 | 0A | ASL A |
| 1928 | EC88 | 0A | ASL A |
| 1929 | EC89 | A8 | TAY ;NOW Y HAS ROW ADDR FROM ROW 1 |
| 1930 | EC8A | AD 2B A4 | LDA STBKEY ;ADD COLUMN TO Y |
| 1931 | EC8D | 4A | GETK3 LSR A |
| 1932 | EC8E | 90 03 | BCC GETK4 |
| 1933 | EC90 | C8 | INY |
| 1934 | EC91 | D0 FA | BNE GETK3 |
| 1935 | EC93 | B9 21 F4 | GETK4 LDA ROW1.Y ;GET THE CHR |
| 1936 | EC96 | 48 | PHA |
| 1937 | EC97 | 8A | TXA ;SEE IF CTRL OR SHIFT WAS USED |
| 1938 | EC98 | F0 24 | BEQ GETK7 ;BRCH IF NO CTRL OR SHIFT |
| 1939 | EC9A | 29 10 | AND #510 ;CTRL ? |
| 1940 | EC9C | F0 06 | BEQ GETK5 ;NO ,GO GETK5 |
| 1941 | EC9E | 68 | PLA |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|------------------------------------------------------|
| 1942 | EC9F | 29 3F | AND #53F ;MSK OFF 2 MSB FOR CONTROL |
| 1943 | ECA1 | 4C BF EC | JMP GETK8 ;EXIT |
| 1944 | ECA4 | 68 | GETK5 FLA |
| 1945 | ECA5 | 48 | PHA ;SAVE IT |
| 1946 | ECA6 | 29 40 | AND #540 ;IF ALPHA CHARS DO NOT SHIFT |
| 1947 | ECA8 | D0 14 | BNE GETK7 |
| 1948 | ECAA | 68 | FLA |
| 1949 | ECAB | 48 | PHA |
| 1950 | ECAC | 29 0F | AND #50F ;ONLY LSB |
| 1951 | ECAE | F0 0E | BEQ GETK7 ;DO NOT INTERCHANGE (SPACE) OR 0 |
| 1952 | ECB0 | C9 0C | CMF #50C ;ACC=#0C ? |
| 1953 | ECB2 | B0 05 | BCC GETK6 ;YES ACC=#0C |
| 1954 | ECB4 | 68 | FLA ;NO, ACC#0C |
| 1955 | ECB5 | 29 EF | AND #5EF ;STRIP OFF BIT 4 |
| 1956 | ECB7 | D0 06 | BNE GETK8 ;EXIT |
| 1957 | ECB9 | 68 | GETK6 FLA ;ACC=#0C |
| 1958 | ECBA | 90 10 | ORA #510 ;BIT 4= 1 |
| 1959 | ECBC | D0 01 | BNE GETK8 ;EXIT |
| 1960 | ECBE | 68 | GETK7 FLA |
| 1961 | ECBF | | ;CHECK FOR 'ADV PAPER', 'PRI LINE', OR 'TOGL PRIFLG' |
| 1962 | ECBF | | ;IN THIS WAY WE DONT HAVE TO CHCK FOR THIS COMM |
| 1963 | ECBF | C9 60 | GETK8 CMF #560 ;ADV PAPER COMM |
| 1964 | ECC1 | D0 06 | BNE GETK11 |
| 1965 | ECC3 | E0 00 | CFX #0 ;IF SHIFT IS NOT ADV PAPER |
| 1966 | ECC5 | F0 25 | BEQ GETK10 ;NO SHIFT ,SO ADVPAPER |
| 1967 | ECC7 | 29 4F | AND #54F ;CONVRT TO '@' |
| 1968 | ECC9 | C9 1C | GETK11 CMF #51C ;SEE IF TOGGL PRIFLG (CONTRL PRI) |
| 1969 | ECCB | D0 14 | BNE GETK13 |
| 1970 | ECCD | 20 E1 E6 | JSR PRITR ;GO TOGGLE FLG |
| 1971 | ECD0 | A0 01 | LDY #1 ;GET THE PTRS BACK 3 SPACES |
| 1972 | ECD2 | B9 15 A4 | GETK12 LDA CURP02,Y |
| 1973 | ECD5 | 38 | SEC |
| 1974 | ECD6 | E9 03 | SBC #3 ;BECAUSE 'ON ,OFF' MSGS |
| 1975 | ECD8 | 99 15 A4 | STA CURP02,Y |
| 1976 | ECD8 | 88 | DEY |
| 1977 | ECD8 | 10 F4 | BPL GETK12 |
| 1978 | ECD8 | 4C 40 EC | JMP GETKEY |
| 1979 | ECE1 | C9 5C | GETK13 CMF #5C ;PRINT LINE COMMAND |
| 1980 | ECE3 | D0 06 | BNE GETK14 |
| 1981 | ECE5 | 20 4A F0 | JSR IPS0 ;PRINT WHATEVER IS IN BUFFER |
| 1982 | ECE8 | 4C 40 EC | JMP GETKEY |
| 1983 | ECEB | 60 | GETK14 RTS |
| 1984 | ECEC | 4C 38 EC | GETK10 JMP GETK10 |
| 1986 | ECEF | | ;WAIT IF LAST KEY STILL DOWN (ROLLOVER) |
| 1987 | ECEF | AD B2 A4 | ROONEK LDA DRB2 ;SEE IF KEY STILL DOWN |
| 1988 | ECF2 | C9 FF | CMF #5FF |
| 1989 | ECF4 | F0 0A | BEQ ROO1 ;NO KEY AT ALL, CLR ROLLFL |
| 1990 | ECF6 | 0D 7F A4 | ORA ROLLFL ;ACCEPT ONLY LAST KEY |
| 1991 | ECF9 | 49 FF | EDR #5FF ;STRBS ARE ZEROS SO INVER |
| 1992 | ECFB | D0 F2 | BNE ROONEK |
| 1993 | ECFD | 20 2A ED | JSR DEBKEY ;CLR KMASK & DEBOUNCE RELEASE |
| 1994 | ED00 | A9 00 | ROO1 LDA #0 ;CLR KMASK |
| 1995 | ED02 | BD 2A A4 | STA KMASK |
| 1996 | ED05 | | ;GO THRU KB ONCE AND RTN ,IF ANY |

```

LINE # LOC      CODE      LINE
1997 ED05          ;KEY Y=ROW (1-8) & STKEY=CLMN
1998 ED05          ;IF NO KEY Y=0 ,STKEY=FFF
1999 ED05 A9 7F      ONEKEY LDA #7F      ;FIRST STROBE TO MSB
2000 ED07 D0 02          RNE ONEK2      ;START AT ONEK2
2001 ED09 38          ONEK1 SEC          ;ONLY ONE PULSE (ZERO)
2002 ED0A 6A          ROR A          ;SHIFT TO RIGHT
2003 ED0B 8D 80 A4     ONEK2 STA DRA2      ;OUTPUT CLMN STROBE
2004 ED0E 8D 2B A4     STA STKEY      ;SAVE IT
2005 ED11 A0 0B          LDY #B         ;CHECK 8 ROWS
2006 ED13 AD 82 A4     LDA DRB2       ;ANY KEY ?
2007 ED16 0D 2A A4     ORA KMASK      ;DISABLE ROW 1 IF CTRL OR SHIFT
2008 ED19 8D 7F A4     STA ROLLFL     ;SAVE WHICH KEY IT WAS
2009 ED1C 0A          ONEK3 ASL A
2010 ED1D 90 0A          RCC ONEK4     ;JUMP IF KEY (ZERO)
2011 ED1F 88          DEY
2012 ED20 D0 FA          BNE ONEK3
2013 ED22 AD 2B A4     LDA STKEY
2014 ED25 C9 FF          CMP #FFF      ;LAST CLMN ?
2015 ED27 D0 E0          BNE ONEK1     ;NO ,DO NEXT CLMN
2016 ED29 60          ONEK4 RTS

2018 ED2A A2 00      DERKEY LDX #0     ;CLEAR CNTRL OR SHIFT
2019 ED2C A9 00      DEBK1 LDA #0     ;CLR KMASK
2020 ED2E 8D 2A A4     STA KMASK
2021 ED31 A9 88          LDA #DEBTIM   ;DEBOUNCE TIME FOR KEYBOARD
2022 ED33 8D 08 AB     STA T2L
2023 ED36 A9 13          LDA #DEBTIM
2024 ED38 4C 18 EC     JMP DE1       ;WAIT FOR 5 MSEC

2026 ED3B          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2027 ED3B          ;GET A CHAR FROM TAPE SUBROUTINE
2028 ED3B          ;A BUFFER IS USED TO GET BLOCKS OF DATA
2029 ED3B          ;FROM TAPE ,EXCEPT WHEN FORMAT EQUAL TO
2030 ED3B          ;KIM-1 (THE WHOLE FILE IS LOADED AT ONE TIME)
2031 ED3B 20 9E EB      TIBYTE JSR PHXY    ;PUSH X
2032 ED3E AE 36 A4      LDX TAPTR     ;POINTER FOR BUFFER
2033 ED41 E0 50          CPX #B0      ;IS BUFFER EMPTY ?
2034 ED43 D0 03          BNE TIB1
2035 ED45 20 53 ED      JSR TIBY1     ;LOAD ANOTHER BLOCK
2036 ED48 8D 16 01     TIB1 LDA TABUFF,X
2037 ED4B EB          INX
2038 ED4C 8E 36 A4     STX TAPTR
2039 ED4F 20 AC EB      JSR PLYX     ;PULL X
2040 ED52 60          RTS
2041 ED53          ;LOAD A BLOCK FROM TAPE INTO BUFFER
2042 ED53 20 EA ED      TIBY1 JSR TAISSET ;SET TAPE FOR INPUT
2043 ED56 20 29 EE      TIBY3 JSR GETTAP  ;GET A CHAR FROM TAPE
2044 ED59 C9 23          CMP #*#      ;CHECK FIRST CHR FOR
2045 ED5B F0 06          BEQ TIBY4    ;START OF BLOCK
2046 ED5D C9 16          CMP #*16     ;IF NOT * SHOULD BE SYN
2047 ED5F D0 F2          BNE TIBY1
2048 ED61 F0 F3          BEQ TIBY3
2049 ED63 A2 00          TIBY4 LDX #0
2050 ED65 20 29 EE      TIBY5 JSR GETTAP  ;NOW LOAD INTO BUFFER
    
```

```

LINE # LOC      CODE      LINE
2051 ED6B 9D 16 01     STA TABUFF,X
2052 ED6B EB          INX
2053 ED6C E0 52          CPX #B2
2054 ED6E D0 F5          BNE TIBY5
2055 ED70 AD 00 AB      LDA DRB
2056 ED73 29 CF          AND #*CF
2057 ED75 8D 00 AB      STA DRB      ;TURN OFF TAPES
2058 ED78 58          CLI          ;ENABL INTERR
2059 ED79 20 8D ED      JSR ADDBK1   ;DISPLAY BLK COUNT
2060 ED7C A2 00          LDX #0       ;TO CLEAR PTR IN TIBYTE
2061 ED7E AD 15 01     LDA BLK      ;CHECK THE BLOCK COUNT
2062 ED81 F0 05          BEQ TIBY5A   ;IF FIRST BLK ,DO NOT CMP
2063 ED83 D0 16 01     CMP TABUFF,X
2064 ED84 D0 28          BNE TIBY7    ;BRANCH IF WE MISSED ONE BLOCK
2065 ED88 EB          TIBY5A INX
2066 ED89 8E 36 A4     STX TAPTR
2067 ED8C EE 15 01     INC BLK      ;INCR BLK CONT
2068 ED8F AD 67 01     LDA TABUFF+81 ;STORE THIS BLK CKSUM
2069 ED92 48          PHA
2070 ED93 AD 66 01     LDA TABUFF+80
2071 ED96 48          PHA
2072 ED97 CE 12 A4     DEC INFLG   ;SET INFLG DIFF FROM OUTFLG
2073 ED9A 20 E7 F1     JSR BKCKSM  ;COMPUT BLK CKSUM FOR THIS BLK
2074 ED9D 68          PLA
2075 ED9E CD 66 01     CMP TABUFF+80 ;DO THEY AGREE ?
2076 EDA1 D0 0C          BNE TIBY6
2077 EDA3 68          PLA
2078 EDA4 CD 67 01     CMP TABUFF+81
2079 EDA7 D0 07          BNE TIBY7
2080 EDA9 EE 12 A4     INC INFLG   ;RESTORE INPUT DEVICE
2081 EDAC A2 01          LDX #1
2082 EDAD 60          RTS
2083 EDAF 68          TIBY6 PLA     ;RESTORE STACK PTR
2084 EDB0 68          TIBY7 PLA
2085 EDB1 68          PLA
2086 EDB2 68          PLA
2087 EDB3 68          PLA
2088 EDB4 20 BE E3     JSR CKERO
2089 EDB7 4C A1 E1     JMP COMIN

2091 EDBA          ;ADD 1 TO BLK COUNT AND OUTPUT IT
2092 EDBA EE 15 01     ADDBLK INC BLK ;INCR BLK CNT
2093 EBD0 EE 11 A4     ADDBK1 INC PRIFLG ;SO DONT OUTPUT TO PRINTR
2094 EDC0 A9 12          LDA #18     ;ONLY OUTPUT IN THIS POSITION
2095 EDC2 8D 15 A4     STA CURFO2
2096 EDC5 AD 4A A4     LDA DIRUFF+18 ;SAVE DIRUFF (FOR EDIT)
2097 EDC8 48          PHA
2098 EDC9 AD 4B A4     LDA DIRUFF+19
2099 EDCC 48          PHA
2100 EDCD AE 13 A4     LDX OUTFLG  ;SAVE OUTFLG
2101 EDD0 A9 0D          LDA #CR
2102 EDD2 8D 13 A4     STA OUTFLG  ;TO OUTPUT TO TERMINAL
2103 EDD5 AD 16 01     LDA BLK+1   ;BLK CNT COMING FROM TAPE
2104 EDD8 20 46 EA     JSR NUMA    ;OUTPUT IN ASCII
2105 EDD8 BE 13 A4     STX OUTFLG  ;RESTORE OUTFLG
    
```

| LINE # | LOC | CODE | LINE |
|--------|------|----------|--------------------------------------------------|
| 2106 | EDDE | 68 | FLA |
| 2107 | EDDF | 8D 4B A4 | STA DIRBUFF+19 |
| 2108 | EDE2 | 68 | FLA |
| 2109 | EDE3 | 8D 4A A4 | STA DIRBUFF+18 |
| 2110 | EDE6 | CE 11 A4 | DEC PRIFLG ;RESTORE PRI FLG |
| 2111 | EDE9 | 60 | RTS |
| | | | |
| 2113 | EDEA | | ;SET TAPE (1 OR 2) FOR INPUT |
| 2114 | EDEA | A9 37 | TAISET LDA ##37 ;SET PB7 FOR INPUT |
| 2115 | EDEC | 8D 02 A8 | STA DDRB |
| 2116 | EDEF | AD 34 A4 | LDA TAPIN ;INPUT FLG (TAP 1=0 OR TAP 2=1) |
| 2117 | EDF2 | 20 1C EE | JSR TIOSSET ;RESET PB4 OR PB5 |
| 2118 | EDF5 | A9 EE | LDA #MOFF+DATIN ;SET CA2=1 (DATA IN) |
| 2119 | EDF7 | 8D 0C AB | STA PCR |
| 2120 | EDFA | A9 FF | LDA ##FF ;PREPARE T2 |
| 2121 | EDFC | 8D 08 AB | STA T2L ;LACTH |
| 2122 | EDFF | | ;CHCK BIT BY BIT UNTIL \$16 |
| 2123 | EDFF | 20 3B EE | SYNC JSR RDBIT ;GET A BIT IN MSR |
| 2124 | EE02 | 4E 2A A4 | LSR CPIY ;MAKE ROOM FOR BIT |
| 2125 | EE05 | 0D 2A A4 | ORA CPIY ;PUT BIT INTO MSR |
| 2126 | EE08 | 8D 2A A4 | STA CPIY |
| 2127 | EE0B | C9 16 | CMP ##16 ;SYN CHAR ? |
| 2128 | EE0D | 10 F0 | BNE SYNC |
| 2129 | EE0F | A2 05 | LIX ##05 ;TEST FOR 5 SYN CHARS |
| 2130 | EE11 | 20 29 EE | SYNC1 JSR GETTAP |
| 2131 | EE14 | C9 16 | CMP ##16 |
| 2132 | EE16 | D0 E7 | BNE SYNC ;IF NOT 2 CHAR RE-SYNC |
| 2133 | EE18 | CA | DEX |
| 2134 | EE19 | D0 F6 | BNE SYNC1 |
| 2135 | EE1B | 60 | RTS |
| | | | |
| 2137 | EE1C | | ;SET PB4 OR PB5 OFF |
| 2138 | EE1C | | ;USED BY IN/OUT SET UPS |
| 2139 | EE1C | D0 04 | TIOSSET BNE TIOS1 ;BRCH IF TAP1 |
| 2140 | EE1E | A9 14 | LDA ##14 ;SET TAPE 2 OFF (PB5=0) |
| 2141 | EE20 | D0 02 | BNE TIOS2 |
| 2142 | EE22 | A9 24 | TIOS1 LDA ##24 ;SET TAPE 1 OFF (PB4=0) |
| 2143 | EE24 | 8D 00 AB | TIOS2 STA DRB |
| 2144 | EE27 | 78 | SEI ;DISABLE INTERR WHILE TAP |
| 2145 | EE28 | 60 | RTS |
| | | | |
| 2147 | EE29 | | ;GET 1 CHAR FROM TAPE AND RETURN |
| 2148 | EE29 | | ;WITH CHR IN ACC, USE CPIY TO ASM CHR, USES Y |
| 2149 | EE29 | A0 08 | GETTAP LDY ##08 ;READ 8 BITS |
| 2150 | EE2B | 20 3B EE | GETA1 JSR RDBIT ;GET NEXT DATA BIT |
| 2151 | EE2E | 4E 2A A4 | LSR CPIY ;MAKE ROOM FOR MSB |
| 2152 | EE31 | 0D 2A A4 | ORA CPIY ;OR IN SIGN BIT |
| 2153 | EE34 | 8D 2A A4 | STA CPIY ;REPLACE CHAR |
| 2154 | EE37 | 88 | DEY |
| 2155 | EE38 | D0 F1 | BNE GETA1 |
| 2156 | EE3A | 60 | RTS |
| 2157 | EE3B | | ;GET ONE BIT FROM TAPE AND |
| 2158 | EE3B | | ;RETURN IT IN SIGN OF A (MSB) |
| 2159 | EE3B | AD 08 A4 | RDBIT LDA TSPEED ;ARE WE IN C7 OR 5B,5A FREQUENC |
| 2160 | EE3E | 30 27 | BMI RDBIT4 ;JUMP TO C7 FREQ FORMAT |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|---------------------------------------------------------|
| 161 | EE40 | 20 75 EE | JSR CKFREQ ;START BIT IN HIGH FREQ |
| 162 | EE43 | 20 75 EE | RDBIT1 JSR CKFREQ ;HIGH TO LOW FREQ TRANS |
| 163 | EE46 | 80 FB | BCC RDBIT1 |
| 164 | EE48 | AD 96 A4 | LDA DIV64 ;GET HIGH FREQ TIMING |
| 165 | EE4B | 48 | PHA |
| 166 | EE4C | A9 FF | LDA ##FF ;SET UP TIMER |
| 167 | EE4E | 8D 96 A4 | STA DIV64 |
| 168 | EE51 | 20 75 EE | RDBIT2 JSR CKFREQ ;LOW TO HIGH FREQ TRANS |
| 169 | EE54 | 90 FB | BCC RDBIT2 ;WAIT TILL FREQ IS HIGH |
| 170 | EE56 | 68 | FLA |
| 171 | EE57 | 38 | SEC |
| 172 | EE58 | ED 96 A4 | SBC DIV64 ;(256-T1) - (256-T2) =T2-T1 |
| 173 | EE5B | 48 | PHA |
| 174 | EE5C | A9 FF | LDA ##FF ;LOW FREQ TIME-HIGH FREQ TIME |
| 175 | EE5E | 8D 96 A4 | STA DIV64 ;SET UP TIMER |
| 176 | EE61 | 68 | FLA |
| 177 | EE62 | 49 FF | EOR ##FF |
| 178 | EE64 | 29 80 | AND ##80 |
| 179 | EE66 | 60 | RTS |
| 180 | EE67 | | ;EACH BIT STARTS WITH HALF PULSE OF 2400 & THEN |
| 181 | EE67 | | ;3 HALF PULSES OF 1200 HZ FOR 0, 3 PULSES OF 2400 FOR 1 |
| 182 | EE67 | | ;THE READING IS MADE ON THE FOURTH 1/2 PULSE, WHERE |
| 183 | EE67 | | ;THE SIGNAL HAS STABILIZED |
| 184 | EE67 | 20 75 EE | RDBIT4 JSR CKFREQ ;SEE WHICH FREQ |
| 185 | EE6A | 90 FB | BCC RDBIT4 |
| 186 | EE6C | 20 75 EE | JSR CKFREQ |
| 187 | EE6F | 20 75 EE | JSR CKFREQ |
| 188 | EE72 | 4C B5 FF | JMP PATC24 ;NOW READ THE BIT |
| 189 | EE75 | 2C 00 AB | CKFREQ BIT DRB ;ARE WE HIGH OR LOW ? |
| 191 | EE7B | 30 27 | BMI CKF4 |
| 192 | EE7A | 2C 00 AB | CKF1 BIT DRB ;WAIT TILL HIGH |
| 193 | EE7D | 10 FB | RPL CKF1 |
| 194 | EE7F | 65 00 | ADC \$00 ;EQUALIZER |
| 195 | EE81 | AD 09 AB | CKF2 LDA T2H ;SAVE CNTR |
| 196 | EE84 | 48 | PHA |
| 197 | EE85 | AD 08 AB | LDA T2L |
| 198 | EE88 | 48 | PHA |
| 199 | EE89 | A9 FF | LDA ##FF |
| 200 | EE8B | 8D 09 AB | STA T2H ;START CNTR |
| 201 | EE8E | AD 08 A4 | LDA TSPEED |
| 202 | EE91 | 30 06 | BMI CKF3 ;SUPER SPEED ? |
| 203 | EE93 | 68 | FLA |
| 204 | EE94 | CD 08 A4 | CMP TSPEED ;HIGH OR LOW FREQ |
| 205 | EE97 | 68 | FLA ;C=1 IF HIGH, C=0 IF LOW |
| 206 | EE98 | 60 | RTS |
| 207 | EE99 | 68 | CKF3 FLA |
| 208 | EE9A | CD 08 A4 | CMP TSPEED ;CENTER FREQ |
| 209 | EE9D | 68 | CKF3A FLA |
| 210 | EE9E | E9 FE | SBC ##FE |
| 211 | EEA0 | 60 | RTS |
| 212 | EEA1 | 2C 00 AB | CKF4 BIT DRB ;WAIT TILL LOW |
| 213 | EEA4 | 30 FB | BMI CKF4 |
| 214 | EEA6 | 10 D9 | RPL CKF2 ;GO GET TIMING |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|---------------------------------------------------------|
| 2216 | EEA8 | | ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; |
| 2217 | EEA8 | | ; OUTPUT ACC TO TTY SUBROUTINE |
| 2218 | EEA8 | | ; X,Y ARE PRESERVED |
| 2219 | EEA8 | 48 | OUTTTY PHA ;SAVE A |
| 2220 | EEA9 | 20 9E EB | JSR PHXY ;PUSH X |
| 2221 | EEAC | 8D 27 A4 | STA STIY ;PUT CHAR HERE |
| 2222 | EEAF | 20 0F EC | JSR DELAY ;STOP BIT FROM LAST CHAR |
| 2223 | EEB2 | AD 00 AB | LDA DRB |
| 2224 | EEB5 | 29 FB | AND *\$FB ;START BIT PB2=0 |
| 2225 | EEB7 | 8D 00 AB | STA DRB ;ITTO=PB2 |
| 2226 | EEBA | 8D 28 A4 | STA STIY+1 ;SAVE THIS PATTERN |
| 2227 | EEBD | 20 0F EC | JSR DELAY |
| 2228 | EEC0 | A2 08 | LIX *\$08 ;8 BITS |
| 2229 | EEC2 | 2E 27 A4 | ROL STIY ;GET FIRST LSB INTO BIT 2 |
| 2230 | EEC5 | 2E 27 A4 | ROL STIY |
| 2231 | EEC8 | 2E 27 A4 | ROL STIY |
| 2232 | EECB | 6E 27 A4 | OUTT1 ROR STIY |
| 2233 | EECE | AD 27 A4 | LDA STIY |
| 2234 | EED1 | 29 04 | AND *\$04 ;GET ONLY BIT 2 FOR PB2 |
| 2235 | EED3 | 0D 28 A4 | ORA STIY+1 ;PUT BIT INTO PATTERN |
| 2236 | EED6 | 8D 00 AB | STA DRB ;NOW TO TTY |
| 2237 | EED9 | 08 | PHP ;PRESERVE CARRY FOR ROTATE |
| 2238 | EEDA | 20 0F EC | JSR DELAY |
| 2239 | EEDD | 28 | PLP |
| 2240 | EEDF | CA | DEX |
| 2241 | EEDF | D0 EA | BNE OUTT1 |
| 2242 | EEE1 | A9 04 | LDA *\$04 ;STOP BIT |
| 2243 | EEE3 | 0D 28 A4 | ORA STIY+1 |
| 2244 | EEE6 | 8D 00 AB | STA DRB |
| 2245 | EEE9 | 20 0F EC | JSR DELAY ;STOP BIT |
| 2246 | EEEC | 20 AC EB | JSR PLXY ;FULL X |
| 2247 | EEEF | 68 | PLA |
| 2248 | EEF0 | C9 0A | CMF *LF |
| 2249 | EEF2 | F0 07 | BEQ OUTT2 |
| 2250 | EEF4 | C9 FF | CMF *NULLC |
| 2251 | EEF6 | F0 03 | BEQ OUTT2 |
| 2252 | EEF8 | 4C 05 EF | JMP OUTDIS ;USE THAT BUFF |
| 2253 | EEFB | 60 | OUTT2 RTS |
| 2255 | EEFC | | ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; |
| 2256 | EEFC | | ; OUTPUT A CHR TO D/P SUBR (SINGLE ENTRY FOR BOTH SUBR) |
| 2257 | EEFC | | ; IF CHAR=CCR) CLEAR DISPLAY & PRINTER |
| 2258 | EEFC | 20 00 F0 | OUTDP JSR OUTPRI ;FIRST TO PRI THEN TO DISP |
| 2259 | EEFF | EA | NOP |
| 2260 | EF00 | EA | NOP |
| 2261 | EF01 | EA | NOP |
| 2262 | EF02 | 6C 06 A4 | OUTDP1 JMP (DILINK) ;HERE HE COULD ECHO SOMEWHERE ELSE |
| 2264 | EF05 | | ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; |
| 2265 | EF05 | | ; OUTPUT ACC TO DISPLAY SUBROUTINE |
| 2266 | EF05 | | ; IF SIGN BIT (MSB)=1 DISP DO NOT CLR TO THE RIGHT |
| 2267 | EF05 | 48 | OUTDIS PHA ;SAVE A |
| 2268 | EF06 | 20 9E EB | JSR PHXY ;PUSH X |
| 2269 | EF09 | C9 0D | CMF *CR ;CCR) ? |
| 2270 | EF0B | D0 07 | BNE OUTD1 |

| LINE # | LOC | CODE | LINE |
|--------|------|----------|-------------------------------------------|
| 2271 | EF0D | A2 00 | LIX #0 ;YES |
| 2272 | EF0F | 8E 15 A4 | STX CURP02 ;CLEAR DISP POINTER |
| 2273 | EF12 | F0 42 | BEQ OUTD5 ;GO CLEAR DISP |
| 2274 | EF14 | 4C 9C FE | OUTD1 JMP PATCH4 |
| 2275 | EF17 | E0 3C | OUTD1A CPX #60 ;LAST CHAR FOR DISP? |
| 2276 | EF19 | 90 05 | BCC OUTD2 |
| 2277 | EF1B | 20 AC EB | JSR PLXY ;GO BACK |
| 2278 | EF1E | 68 | PLA ;DO NOT STORE |
| 2279 | EF1F | 60 | RTS |
| 2280 | EF20 | 9D 38 A4 | OUTD2 STA DIBUFF,X ;PUT CHAR IN BUFF |
| 2281 | EF23 | EE 15 A4 | INC CURP02 ;INC POINTER |
| 2282 | EF26 | E0 14 | CPX #20 ;DISPLAY FULL? |
| 2283 | EF28 | 90 1E | BCC OUTD4 |
| 2284 | EF2A | 20 2F EF | JSR OUTD2A ;THIS WAY SCROLL IS A SUBR |
| 2285 | EF2D | 30 47 | RMI OUTD7 ;EXIT DISP |
| 2286 | EF2F | | ;YES, SCROLL CHARS TO THE LEFT |
| 2287 | EF2F | 8A | OUTD2A TXA ;X--> Y |
| 2288 | EF30 | A8 | TAY |
| 2289 | EF31 | A2 13 | LIX #19 ;ADDR FOR DISP DO NOT |
| 2290 | EF33 | 8E 27 A4 | OUTD3 STX STIY ;DECREM IN BINARY |
| 2291 | EF36 | B9 38 A4 | LDA DIRUFF,Y ;FROM BUFFER TO DISP |
| 2292 | EF39 | 09 80 | ORA *\$80 ;NO CURSOR |
| 2293 | EF3B | 20 7B EF | JSR OUTDD1 ;CONVERT X INTO REAL ADDR |
| 2294 | EF3E | 88 | DEY |
| 2295 | EF3F | CE 27 A4 | DEC STIY |
| 2296 | EF42 | AE 27 A4 | LIX STIY |
| 2297 | EF45 | 10 EC | BPL OUTD3 ;AGAIN UNTIL WHOLE DISP |
| 2298 | EF47 | 60 | RTS |
| 2299 | EF48 | 48 | OUTD4 PHA |
| 2300 | EF49 | 09 80 | ORA *\$80 ;NO CURSOR |
| 2301 | EF4B | 20 7B EF | JSR OUTDD1 ;X=<\$19 ,CONVRT TO REAL ADDR |
| 2302 | EF4E | 68 | PLA |
| 2303 | EF4F | 29 80 | AND *\$80 ;IF MSB=0 CLEAR REST OF DISPLAY |
| 2304 | EF51 | D0 23 | BNE OUTD7 |
| 2305 | EF53 | AE 15 A4 | LIX CURP02 |
| 2306 | EF56 | | ;CLEAR DISP TO THE RIGHT |
| 2307 | EF56 | E0 14 | OUTD5 CPX #20 |
| 2308 | EF58 | R0 1C | BCS OUTD7 |
| 2309 | EF5A | 8E 27 A4 | STX STIY |
| 2310 | EF5D | A9 A0 | LDA *\$A0 ;(SPACE) |
| 2311 | EF5F | 20 7B EF | JSR OUTDD1 ;CONVRT TO REAL ADDR |
| 2312 | EF62 | EE 27 A4 | INC STIY |
| 2313 | EF65 | AE 27 A4 | LIX STIY |
| 2314 | EF68 | D0 EC | BNE OUTD5 ;GO NEXT |
| 2315 | EF6A | 4C 76 EF | JMP OUTD7 |
| 2316 | EF6D | EA | NOP |
| 2317 | EF6E | EA | NOP |
| 2318 | EF6F | EA | NOP |
| 2319 | EF70 | EA | NOP |
| 2320 | EF71 | EA | NOP |
| 2321 | EF72 | EA | NOP |
| 2322 | EF73 | EA | NOP |
| 2323 | EF74 | EA | NOP |
| 2324 | EF75 | EA | NOP |
| 2325 | EF76 | 20 AC EB | OUTD7 JSR PLXY ;REST ,SO PRINTR INDEPEN |