

# 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 UDRAH *=$+1 ; DATA REG A
0031 A002 UIDRB *=$+1 ; DATA DIR REG B
0032 A003 UIDRA *=$+1 ; DATA DIR REG A
0033 A004 UTIL *=$+1 ; TIMER 1 COUNTER LOW
0034 A005 UTICH *=$+1 ; TIMER 1 COUNTER HIGH
0035 A006 UTILL *=$+1 ; TIMER 1 LATCH LOW
0036 A007 UTILH *=$+1 ; TIMER 1 LATCH HIGH
0037 A008 UT2L *=$+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 UDRA *=$+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 =HISTM+1
TEMPX =HISTM+3
TEMPA =HISTM+5
OPCODE =HISTM+6
CODFLG =HISTM+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 =BK5+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 DFPA7 **+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      ; END OF START BIT ?
0418 E12C 50 FB      RVC RS5      ; NO ,WAIT UNTIL PB6 BACK TO 1
0419 E12E AD 09 AB      LDA T2H      ; STORE TIMING
0420 E131 49 FF      EOR #$FF      ; COMPLEMENT
0421 E133 8D 17 A4      STA CNTH30
0422 E136 AD 08 AB      LDA T2L
0423 E139 49 FF      EOR #$FF
0424 E13B 20 7C FE      JSR PATCH1      ; ADJUST IT
0425 E13E 20 13 EA      RS6 JSR CRLW      ; CLEAR DISPLAY
0426 E141 4C 72 FF      JMP FAT21
0427 E144 A2 13      RS7 LDX #19      ; CLEAR HARWARE CURSORS
0428 E146 8A          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          ; CONTROL TO USER THRU VECTOR
0447 E163          ; IS 'BRK' INSTR ,SHOW PC & DATA
0448 E163 68          ; PC IS OFF BY ONE , SO ADJUST IT
0449 E164 8D 20 A4      IRQ1 PLA
0450 E167 8E 22 A4      STA SAVPS      ; SAVE PROCESSOR STATUS
0451 E16A 8C 23 A4      STX SAVX
0452 E16D D8          STY SAVY
0453 E16E 68          CLD
0454 E16F 38          PLA          ; PROGR CNTR
0455 E170 E9 01      SEC          ; 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	
0507	E209	FE E6	.WOR SHOW, CLRBB, SHIS, REGT, TRACE
0507	E20B	65 E6	
0507	E20D	D9 E6	
0507	E20F	DD E6	
0508	E211	9E F8	
0508	E213	0A E7	.WORD MNEENT, KDISA, TOGTA1, TOGTA2, VECKSM
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 EB	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 EB	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 EB	CHNGG JSR BLANK
0571	E2A3	20 DB E2	JSR WRITAZ ;WRITE ADDR
0572	E2A6	20 3E EB	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 EB	JSR BLANK
0579	E2B5	4C C0 E2	JMP CH4
0580	E2B8		;IS ALTER
0581	E2B8	20 78 EB	CH2 JSR SADDR ;STORE ENTERED VALUE INTO MEMORY
0582	E2B8	F0 03	REQ CH4 ;NO ERROR IN STORE
0583	E2BD	4C 33 EB	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 EB	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)



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 B4 EA      JSR PACK
0767 E435 BD 2F A4      LDA NAME+1,X
0768 E438 4C B4 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 ;CHK 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 B7 E5      JMP DUMPKI ;YES, GO OUTPUT WHOLE FILE
0797 E478 A0 01          LDY #1 ;OUTPUT ONE MORE BYTE
0798 E47A 20 D0 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	DU10 LDY #M5-M1	; ALL DONE
0844	E4DI	20 70 E9	JSR KEPR	; PRINT *MORE ?*
0845	E4E0	C9 59	CMP #'Y	; OUTPUT MSG AND GET AN ANSWER
0846	E4E2	D0 03	RNE *+5	
0847	E4E4	4C 44 E4	JMP DU1	; DUMP MORE DATA
0848	E4E7	68	FLA	; RESTORE FLG
0849	E4EB	8D 10 A4	STA BKFLG	
0850	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 TA0SET	; 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 ;CHKC 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 QM
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 QM ;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     CLRBRK 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	PROMPT1 JSR TTYTST ;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 IIRBUFF,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 WHRO1
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	WHRO1 CMF #*K ;TAPE WITH KIM FORMAT
1339	E887	D0 05	BNE WHRO2
1340	E889	A2 01	LIX #1 ;FOR OUTPUT FILE FLG
1341	E88B	4C A2 E8	JMP FNAM
1342	E88E		;PRINTER
1343	E88E	C9 50	WHRO2 CMF #*P ;PRINTER?
1344	E890	D0 05	BNE WHRO3
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	WHRO3 CMF #*U ;USER RTN?
1349	E899	D0 04	BNE WHRO4
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	WHRO4 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 E8	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          BMI TAP2      ;ONLY 1,2 OK
1366 EBR8  C9 02          CMP #2
1367 EBR8A 30 06          BMI TAP3      ;OK
1368 EBR8C 20 D4 E7        TAP2 JSR RM      ;ERROR
1369 EBRF  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 EBCB  20 83 FE        JSR CUREAD    ;GET ANYTHING
1373 EBCR  20 24 EA        JSR CRCK     ;CCR)
1374 EBCE  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 EBD8  F0 0C          CMP #'
1383 EBD8  C9 20          BEQ NAM02
1384 EBD8  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          BMI RCH3     ;RTN ON NO KEY
418 E912  A2 00          LDX #0
419 E914  20 82 EC        JSR GETK2    ;GET THE KEY
420 E917  C9 1B          CMP #ESCAPE
421 E919  F0 3B          BEQ REA1    ;TO COMMAN & SET I/O TO TERMINAL
422 E91B  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 E92B  50 FB          BVC RCHT2
431 E92D  20 0F EC        JSR DELAY
432 E930  20 0B EB        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 E93B  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 EB        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 PSLS  ;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 TIRYTE         ;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 TDRYTE         ;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          PHF
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 FNTRS 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	68	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	BPL 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 ERBA BA SWSTAK TSX
1814 EBBB A9 02 LDA #2
1815 EBBD 48 SWST1 PHA
1816 ERBE 8D 06 01 LDA #0106,X ;GET PCH OR PCL
1817 ERBC 8C 04 01 LDY #0104,X ;GET Y OR X REGS
1818 ERCA 9D 04 01 STA #0104,X
1819 ERCC 98 TYA
1820 ERCD 9D 06 01 STA #0106,X
1821 ERCE CA DEX
1822 ERCC 68 PLA
1823 ERCD 38 SEC
1824 ERCE E9 01 SBC #1
1825 ERDD 00 EB RNE SWST1
1826 ERDE 8D 08 01 LDA #0108,X ;RESTORE Y & X FROM STACK
1827 ERDE AB TAY
1828 ERDE 8D 07 01 LDA #0107,X
1829 ERD9 AA TAX
1830 ERDA 60 RTS
    
```

```

LINE # LOC CODE LINE
1832 ERDB ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1833 ERDB ;GET A CHAR FROM TTY SUBR INTO ACC ,SAVES X
1834 ERDB BA GETTTY TXA ;SAVE X
1835 ERDC 48 PHA
1836 ERDD A2 07 LDX #*07 ;SET UP FOR 8 BIT CNT
1837 ERDF 8E 2A A4 STX CPIY ;CLEAR MSR
1838 ERE2 2C 00 AB GET1 BIT IRR ;A^M , PB6-DV
1839 ERE5 70 FB BUS GET1 ;WAIT FOR START BIT
1840 ERE7 20 0F EC JSR DELAY ;DELAY 1 BIT
1841 EREA 20 23 EC JSR DEHALF ;DELAY 1/2 BIT TIME
1842 ERED AD 00 AB GET3 LDA BRB ;GET 8 BITS
1843 ERF0 29 40 ANDI #*40 ;MASK OFF OTHER BITS,ONLY PB6
1844 ERF2 4E 2A A4 LSR CPIY ;SHIFT RIGHT CHARACTER
1845 ERF5 0D 2A A4 ORA CPIY
1846 ERF8 8D 2A A4 STA CPIY
1847 ERFB 20 0F EC JSR DELAY ;DELAY 1 BIT TIME
1848 ERFE CA DEX
1849 ERFF 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	8D 80 A4	STA DRA2
1893	EC48	AD 82 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 82 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 82 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-B) & 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          BNE 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 BRA2   ;OUTPUT CLMN STROBE
2004 ED0E 8D 2B A4     STA STKEY    ;SAVE IT
2005 ED11 A0 0B          LDY #B       ;CHECK B 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 TAISRT ;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 ED68 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 ED86 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          PLA
2107 EDDF 8D 4B A4    STA DIRBUFF+19
2108 EDE2 68          PLA
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 TI0SET     ;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    LDIX ##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 F4    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    TI0SET RNE TI0S1 ;BRCH IF TAP1
2140 EE1E A9 14    LDA ##14      ;SET TAPE 2 OFF (PB5=0)
2141 EE20 D0 02    BNE TI0S2
2142 EE22 A9 24    TI0S1 LDA ##24   ;SET TAPE 1 OFF (PB4=0)
2143 EE24 8D 00 AB    TI0S2 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 B0 FB      BCS 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          PLA
171 EE57 38          SEC
172 EE58 ED 96 A4    SBC DIV64     ;(256-T1) - (256-T2) =T2-T1
173 EE5B 48          PHA          ;LOW FREQ TIME-HIGH FREQ TIME
174 EE5C A9 FF    LDA ##FF
175 EE5E 8D 96 A4    STA DIV64     ;SET UP TIMER
176 EE61 68          PLA
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 ?
190 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          PLA
204 EE94 CD 0B A4    CMP TSPEED   ;HIGH OR LOW FREQ
205 EE97 68          PLA          ;C=1 IF HIGH ,C=0 IF LOW
206 EE98 60          RTS
207 EE99 68          CKF3 PLA
208 EE9A CD 0B A4    CMP TSPEED   ;CENTER FREQ
209 EE9D 68          CKF3A PLA
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 FLXY ;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 FLXY ;REST ,SO PRINTR INDEPEN

LINE #	LOC	CODE	LINE
2326	EF79	68	PLA
2327	EF7A	60	RTS
2329	EF7B		; CONVERT X INTO REAL ADDR FOR DISPLAY
2330	EF7B		; AND OUTPUT IT PB=DATA ; PA=W,CE ,A0 A1 (6520)
2331	EF7B	48	OUTDD1 PHA ; SAVE DATA
2332	EF7C	8A	TXA
2333	EF7D	48	PHA ; SAVE X
2334	EF7E	4A	LSR A ; DIVIDE X BY 4
2335	EF7F	4A	LSR A ; TO GET CHIP SELECT
2336	EF80	AA	TAX ; BACK TO X
2337	EF81	A9 04	LDA #4 ; FIRST CHIP SELECT
2338	EF83	E0 00	CFX #0 ; FIRST CHIP ?
2339	EF85	F0 04	BEQ OUTDD3
2340	EF87	0A	OUTDD2 ASL A
2341	EF88	CA	DEX
2342	EF89	D0 FC	BNE OUTDD2 ; BACK TILL RIGH CS
2343	EF8B	8D 28 A4	OUTDD3 STA STIY+1 ; SAVE CS TEMPORARILY
2344	EF8E	68	PLA ; GET X AGAIN FOR CHAR
2345	EF8F	29 03	AND #03 ; IN THAT CHIP
2346	EF91	0D 28 A4	ORA STIY+1 ; OR IN CS AND CHAR
2347	EF94		; STORE ADDR AND DATA INTO DISPL
2348	EF94	49 FF	EDR #FF ; W=1 , CE=0 & A1,A0
2349	EF96	8D 00 AC	STA RA
2350	EF99	AA	TAX ; SAVE A IN X
2351	EF9A	68	PLA ; GET DATA
2352	EF9B	48	PHA
2353	EF9C	8D 02 AC	STA RB
2354	EF9F	8A	TXA
2355	EFA0	49 80	EDR #80 ; SET W=0
2356	EFA2	8D 00 AC	STA RA
2357	EFA5	EA	NOP
2358	EFA6	09 7C	ORA #7C ; SET CE=1
2359	EFA8	8D 00 AC	STA RA
2360	EFA8	A9 FF	LDA #FF ; SET W=1
2361	EFA9	8D 00 AC	STA RA
2362	EFB0	68	PLA ; RETURN DATA
2363	EFB1	60	RTS
2365	EFB2		*=\$EFF9
2366	EFF9	EA	.BYT \$EA
2367	EFFA		*=\$F000
2368	F000		;;
2369	F000		; OUTPUT ACC TO PRINTER SUBROUTINE
2370	F000		; PRINTS ON FIRST CHAR OR WHEN (CCR)
2371	F000		; IT WILL PUT IT ON BUFFER BUT WONT PRINT IF
2372	F000		; PRIFLG=0
2373	F000	48	OUTPRI PHA ; SAVE CHR TO BE OUTPUT
2374	F001	20 9E EB	JSR PHXY ; SAVE X
2375	F004	C9 0D	CMF #0D ; SEE IF CR
2376	F006	F0 07	BEQ OUT01 ; YES SO PRINT THE BUFF
2377	F008	AE 16 A4	LDX CURPOS ; PTR TO NEXT POS IN BUFF
2378	F00B	E0 14	CPX #20 ; SEE IF BUFF FULL

LINE #	LOC	CODE	LINE
2379	F00D	D0 16	BNE OUT04 ; NOT FULL SO RETURN
2380	F00F		; CCR) SO FILL REST OF BUFFER WITH BLANKS
2381	F00F	48	OUT01 PHA
2382	F010	A9 00	LDA #0 ; CURPOS = 0
2383	F012	AE 16 A4	LDX CURPOS ; SEE IF ANYTHING IN BUFFER
2384	F015	8D 16 A4	STA CURPOS
2385	F018	20 38 F0	JSR OUTPR ; CLEAR PRIBUF TO THE RIGHT
2386	F01B		; BUFFER FILLED SO PRINT IT
2387	F01B	20 45 F0	JSR IPST ; START THE PRINT
2388	F01E	A2 00	LDX #0 ; STORE CHR IN BUFF (FIRST LOC)
2389	F020	68	PLA ; GET IT
2390	F021	C9 0D	CMF #CR ; DONT STORE IF (CCR)
2391	F023	F0 0E	BEQ OUT05
2392	F025	9D 60 A4	OUT04 STA IBUFM,X ; STORE CHR IN BUFF
2393	F028	EE 16 A4	INC CURPOS ; INCR BUFF PNTR
2394	F02B	E8	INX
2395	F02C	29 80	AND #80
2396	F02E	D0 03	BNE OUT05 ; DONT CLR IF MSB=1
2397	F030	20 38 F0	JSR OUTPR ; CLEAR PRIBUFF TO THE RIGHT
2398	F033	20 AC EB	OUT05 JSR FLXY ; RESTORE REGS
2399	F036	68	PLA
2400	F037	60	RTS
2401	F038	A9 20	OUTPR LDA #20 ; FILL REST OF BUFF WITH BLANKS
2402	F03A	E0 14	OUTPR1 CPX #20 ; SEE IF END OF BUFF
2403	F03C	F0 06	BEQ OUTPR2
2404	F03E	9D 60 A4	STA IBUFM,X ; NO SO STORE BLANK
2405	F041	EB	INX ; INCR BUFF PNTR
2406	F042	10 F6	BFL OUTPR1
2407	F044	60	OUTPR2 RTS
2409	F045		; SUB TO OUTPUT BUFFER, 70 DOTS (10 DOTS AT
2410	F045		; A TIME BY 7 ROWS) FOR EACH LINE OF PRINTING
2411	F045	2C 11 A4	IPST BIT PRIFLG ; PRINT FLG ON ?
2412	F048	10 2E	BPL IF04
2413	F04A	20 CB F0	IPSO JSR PINT ; INITIALIZE VALUES
2414	F04D	20 E3 F0	JSR IPSU ; SET UP FIRS OUTPUT PATTERN
2415	F050	A9 C1	IP00 LDA #PRST+SP12+MON ; TURN MOTOR ON
2416	F052	8D 0C AB	STA PCR
2417	F055	20 A0 FF	JSR PAT23 ; TIME OUT?
2418	F058	D0 0C	BNE IP02 ; NO, START SIGNAL RECEIVED
2419	F05A	20 A0 FF	JSR PAT23 ; YES, TRY AGAIN
2420	F05D	D0 07	BNE IP02 ; OK
2421	F05F	4C 79 F0	JMP FRIERR ; TWO TIME OUTS - ERROR
2422	F062	EA	NOP
2423	F063	EA	NOP
2424	F064	EA	NOP
2425	F065	EA	NOP
2426	F066	20 87 F0	IP02 JSR PRNDOT ; STRB F1=1 PRINT DOTS (1.7MSEC)
2427	F069	20 87 F0	JSR PRNDOT ; STRB F2=1 PRINT DOTS (1.7MSEC)
2428	F06C		; CHECK FOR 90, WHEN 70 PRNDOT WILL OUTPUT ZEROS
2429	F06C	AD 77 A4	LDA IDDT
2430	F06F	C9 5A	CMF #90
2431	F071	90 F3	BCC IP02 ; L.T. 90 THEN GO STROB F1
2432	F073	A9 E1	IP03 LDA #PRST+SP12+MOFF ; TURN MOTOR OFF
2433	F075	8D 0C AB	STA PCR

```

LINE # LOC      CODE      LINE
2434 F078 60          IP04  RTS
2436 F079 20 44 EB      PRIERR JSR CLR          ;CLEAR PRI PNTR
2437 F07C 20 B1 FE          JSR PATCH5         ;TURN PRI OFF
2438 F07F A0 3B          LDY #M12-M1
2439 F081 20 AF E7          JSR KFP
2440 F084 4C A1 E1          JMP COMIN          ;BACK WHERE SUBR WAS CALLED

2442 F087          ;SUBR TO INCR DOT COUNTER WHEN
2443 F087          ;NEG TRANS OUTPUT CHR FOR 1.7 MSEC
2444 F087          ;CLEAR & SET UP NEXT PATTERN
2445 F087 A9 00          PRNDOT LDA #0          ;CLR INTERRUPTS
2446 F089 8D 01 AB          STA DRAH
2447 F08C AD 0D AB          PRDOTO LDA IFR
2448 F08F 29 02          AND #MSP12         ;ANY STROBES ?
2449 F091 F0 F9          BEQ PRDOTO
2450 F093 AD 0C AB          LDA PCR
2451 F096 49 01          EOR #*01
2452 F098 8D 0C AB          STA PCR
2453 F09B EE 77 A4          INC IDOT
2454 F09E AD 79 A4          LDA IOUTU          ;2 LEFT ELEM
2455 F0A1 0D 00 AB          ORA DRB            ;DO NOT TURN TTY OUTPUT OFF
2456 F0A4 8D 00 AB          STA DRB
2457 F0A7 AD 78 A4          LDA IOUTL          ;7 RIGHT ELEM, CLR CA1 INTER FLB
2458 F0AA 8D 01 AB          STA DRAH
2459 F0AD A9 A4          LDA #CFRIME
2460 F0AF 8D 08 AB          STA T2L
2461 F0B2 A9 06          LDA #DPRIME        ;START T2 FOR 1.7 MSEC
2462 F0B4 8D 09 AB          STA T2H
2463 F0B7 20 E3 F0          JSR IPSU            ;SET NEXT PATTERN WHILE WAITING
2464 F0BA 20 1B EC          JSR DE2            ;WAIT TILL TIME OUT
2465 F0BD A9 00          LDA #0              ;THERMAL ELEM OFF
2466 F0BF 8D 01 AB          STA DRAH
2467 F0C2 AD 00 AB          LDA DRB            ;BUT DONT CHANGE TAPE CONTROLS
2468 F0C5 29 FC          AND #*FC
2469 F0C7 8D 00 AB          STA DRB
2470 F0CA 60          RTS

2472 F0CB          ; SUBROUTINE PINT --- INIT VARS FOR PRINTER
2473 F0CB A9 FF          PINT  LDA #*FF
2474 F0CD 8D 74 A4          STA IDIR           ;DIRECTION (<= -
2475 F0D0 A9 05          LDA #5
2476 F0D2 8D 75 A4          STA ICOL           ;COLUMN (<= LEFTMOST +1
2477 F0D5 A9 01          LDA #1
2478 F0D7 8D 76 A4          STA IOFFST         ;OFFSET (<= LEFT CHARACTER
2479 F0DA 8D 7C A4          STA IMASK
2480 F0DD A9 00          LDA #0
2481 F0DF 8D 77 A4          STA IDOT           ;DOT COUNTER (<= 0
2482 F0E2 60          RTS

2484 F0E3          ;THE VARIABLES FOR THE PRINTER ARE AS FOLLOWS:
2485 F0E3
2486 F0E3          ;IDIR  DIRECT HEAD IS CURRENTLY MOVING (0=+, *FF=-)
2487 F0E3          ;ICOL  CLMN TO BE FRNTD NEXT (LEFTMOST=0, RIGHTMOST=4
2488 F0E3          ;IOFFST OFFSET N PRINT BUFF (0=LEFT CHR, 1=RIGHT CHR)

```

```

LINE # LOC      CODE      LINE
2489 F0E3          ;IDOT  COUNT OF NUMBER OF DOTS PRINTED THUS FAR
2490 F0E3          ;IOUTL SOLENOID PATTERN (8 CHRS ON RIGHT)
2491 F0E3          ;IOUTU SOLENOID PATTERN (2 CHRS ON LEFT)
2492 F0E3          ;IBITL 1 BIT MSK USED IN SETTING NEXT SOLENOID VALUE
2493 F0E3          ;IBITU UPPER PART OF MASK
2494 F0E3          ;IBUFM START OF PRINT BUFFER (LEFTMOST CHR FIRST)
2495 F0E3          ;IMASK MASK FOR CURRENT ROW BEING PRINTED
2496 F0E3          ;JUMP  ADDRESS OF TABLE FOR CURRENT COLUMN
2497 F0E3
2498 F0E3          ; THE DOT PATTERNS FOR THE CHRS ARE STORED SO THAT...
2499 F0E3          ;EACH BYTE CONTAINS THE DOTS FOR ONE COLUMN OF ONE...
2500 F0E3          ;CHR. SINCE EACH COLUMN CONTAINS SEVEN DOTS...
2501 F0E3          ;THIS MEANS THAT ONE BIT PER BYTE IS UNUSED.
2502 F0E3          ; THE PATTERNS ARE ORGANIZED INTO 5 TABLES OF 64...
2503 F0E3          ;BYTES WHERE EACH TABLE CONTAINS ALL THE DOT...
2504 F0E3          ;PATTERNS FOR A PARTICULAR COLUMN. THE BYTES IN EACH...
2505 F0E3          ;TABLE ARE ORDERED ACCORDING TO THE CHR CODE OF...
2506 F0E3          ;THE CHR BEING REFERENCED. THE CHR CODE CAN...
2507 F0E3          ;THUS BE USED TO DIRECTLY INDEX INTO THE TABLE.

2509 F0E3          ;SUBROUTINE IPSU -- SET UP OUTPUT PATTERN FOR PRINTER
2510 F0E3          ; THIS ROUTINE IS CALLED IN ORDER TO
2511 F0E3          ;SET UP THE NEXT GROUP OF SOLENOIDS TO
2512 F0E3          ;BE OUTPUT TO THE PRINTER.
2513 F0E3          ; ON ENTRY THE CONTENTS OF ALL REGISTERS
2514 F0E3          ;ARE ARBITRARY
2515 F0E3          ; ON EXIT THE CONTENTS OF A,X,Y ARE UNDEFINED
2516 F0E3 A2 00          IPSU  LDX #0            ;X POINTS TO VAR BLOCK FOR PNTR
2517 F0E5 20 21 F1          JSR INCP           ;ADVANCE PTRS TO NXT DOT POSITION
2518 F0E8          ;X NOW CONTAINS INDEX INTO PRINT BUFFER
2519 F0E8 8D 60 A4          IPS1  LDA IBUFM,X    ;LOAD NEXT CHAR FROM BUFFER
2520 F0EB 29 3F          AND #*3F
2521 F0ED AB          TAY
2522 F0EE A9 7D          LDA #CJUMP         ;AC= DOT PATTERN FOR CHAR & COL
2523 F0F0 20 58 EB          JSR LDAY
2524 F0F3 2C 7C A4          BIT IMASK          ;SEE IF DOT IS SET
2525 F0F6 F0 16          BEQ IPS2           ;NO SO GO ON TO NEXT CHAR
2526 F0F8 AD 7A A4          LDA IBITL          ;DOT ON SO SET THE CURR SOLENOID
2527 F0FB F0 08          BEQ IPS3           ;LSB OF SOL MASK IS 0, DO MSB
2528 F0FD 0D 78 A4          ORA IOUTL          ;SET THE SOLENOID IN THE PATTERN
2529 F100 8D 78 A4          STA IOUTL
2530 F103 D0 09          BNE IPS2           ;BRANCH ALWAYS
2531 F105 AD 7B A4          LDA IBITU          ;SOLENOID IS ONE OF THE 2 MSD
2532 F108 0D 79 A4          ORA IOUTU          ;SET THE BIT IN THE PATTERN
2533 F10B 8D 79 A4          STA IOUTU
2534 F10E 0E 7A A4          IPS2  ASL IBITL        ;SHIFT MSK TO NXT CHR POSITION
2535 F111 2E 7B A4          ROL IBITU
2536 F114 CA          DEX                ;DECR PTR INTO BUFFER
2537 F115 CA          DEX
2538 F116 10 D0          BPL IPS1           ;NOT END YET
2539 F118          ;SOLENOID PATTERN IS SET UP IN IOUTU, IOUTL
2540 F118 AD 79 A4          LDA IOUTU          ;LEFTMOST 2
2541 F11B 29 03          AND #*03           ;DISABLE FOR SEGMENTS
2542 F11D 8D 79 A4          STA IOUTU
2543 F120 60          RTS

```

LINE #	LOC	CODE	LINE
2545	F121		; SUBROUTINE INCP
2546	F121		; THIS SUBROUTINE IS USED TO UPDATE THE PRINTER VARIABLES
2547	F121		; TO POINT TO THE NEXT DOT POSITION TO BE PRINTED
2548	F121		; X REG IS USED TO POINT TO THE VARIABLE BLOCK OF
2549	F121		; BEING UPDATED
2550	F121		; ON EXIT X CONTAINS THE POINTER TO THE LAST CHARACTER IN
2551	F121		; THE PRINT BUFFER
2552	F121		; CONTENTS OF A,Y ON EXIT ARE ARBITRARY
2553	F121	BD 74 A4	INCP LDA IDIR,X ; EXAMINE DIRECTION(+ OR -)
2554	F124	10 1E	BPL OP03 ; DIRECTION = +
2555	F126		; **DIRECTION = -
2556	F126	BD 75 A4	LDA ICOL,X ; SEE WHAT THE COLUMN IS
2557	F129	F0 05	BEQ OP04 ; COLUMN = 0 SO END OF DIGIT
2558	F12B		; **COLUMN # 0 SO JUST DECREMENT COLUMN
2559	F12B	DE 75 A4	DEC ICOL,X
2560	F12E	10 33	BPL NEWCOL ; BRANCH ALWAYS
2561	F130		; **COLUMN = 0 SO SEE IF EVEN OR ODD DIGIT
2562	F130	BD 76 A4	OP04 LDA IOFFST,X
2563	F133	F0 0A	BEQ OP07 ; OFFSET = 0 SO DIRECTION CHANGE
2564	F135		; ***OFFSET = 1 SO MOVE TO RIGHT DIGIT
2565	F135	DE 76 A4	DEC IOFFST,X ; OFFSET (= 0 (LEFT CHARACTER)
2566	F138	A9 04	LDA #4 ; COLUMN (= 4
2567	F13A	9D 75 A4	STA ICOL,X
2568	F13D	10 24	BPL NEWCOL ; BRANCH ALWAYS
2569	F13F		; ***OFFSET = 0 SO CHANGE DIRECTION TO +
2570	F13F	FE 74 A4	OP07 INC IDIR,X ; DIRECTION (= \$00 (+)
2571	F142	10 1C	BPL NEWROW ; BRANCH ALWAYS
2572	F144		; *DIRECTION = +
2573	F144	BD 75 A4	OP03 LDA ICOL,X ; SEE IF LAST COLUMN IN DIGIT
2574	F147	C9 04	CMP #4
2575	F149	F0 05	BEQ OP05 ; COLUMN = 4 SO GO TO NEXT DIGIT
2576	F14B	FE 75 A4	INC ICOL,X ; JUST INCR COLUMN-NOT END OF DIGIT
2577	F14E	10 13	BPL NEWCOL ; BRANCH ALWAYS
2578	F150		; **AT COLUMN 4 -- SEE IF LEFT OR RIGHT DIGIT
2579	F150	BD 76 A4	OP05 LDA IOFFST,X
2580	F153	D0 08	BNE OP06 ; OFFSET # 0 SO RIGHT DIGIT
2581	F155	9D 75 A4	STA ICOL,X ; COLUMN (= 0
2582	F158	FE 76 A4	INC IOFFST,X ; OFFSET (= 1 (RIGHT CHARACTER)
2583	F15B	10 06	BPL NEWCOL ; BRANCH ALWAYS
2584	F15D		; ***OFFSET = 1 SO DIRECTION CHANGE
2585	F15D	DE 74 A4	OP06 DEC IDIR,X ; DIRECTION (= \$FF (-)
2587	F160		; START OF NEW PRINT ROW
2588	F160	1E 7C A4	NEWROW ASL IMASK,X ; UPDATE ROW MASK FOR DOT PATTERN
2589	F163		; START OF NEW PRINT COLUMN
2590	F163	A9 00	NEWCOL LDA #0 ; CLEAR OUTPUT PATTERN
2591	F165	9D 78 A4	STA IOU1L,X ; PATTERN FOR 8 RIGHT CHRS
2592	F168	9D 79 A4	STA IOU1U,X ; PATTERN FOR 2 LEFT SOLEN
2593	F16B	9D 7B A4	STA IBITU,X ; OUTPUT MSK FOR LEFTMOST SOLEN
2594	F16E	A9 01	LDA #1
2595	F170	9D 7A A4	STA IBITL,X ; OUTPUT MSK FOR RIGHTMOST SOLEN
2596	F173		; GET ADDRESS OF DOT PATTERN TABLE FOR NEXT COLUMN
2597	F173	BD 75 A4	LDA ICOL,X ; GET COLUMN NUMBER (0-4)
2598	F176	0A	ASL A ; *2, INDEX INTO TBL OF TBL ADDR
2599	F177	AB	TAY

LINE #	LOC	CODE	LINE
2600	F17B	B9 D7 F2	LDA MTBL,Y ; LSB OF ADDR OF TABLE
2601	F17B	9D 7D A4	STA JMP,X ; PTR TO TBL WITH DOT PATTERNS
2602	F17E	B9 D8 F2	LDA MTBL+1,Y ; MSB OF TABLE ADDRESS
2603	F181	9D 7E A4	STA JMP+1,X
2604	F184	A9 12	LDA #1B ; COMPUTE INDEX INTO PRNTR BUFFER
2605	F186	1D 76 A4	ORA IOFFST,X ; +1 IF RIGHT CHR
2606	F189	AA	TAX
2607	F18A	60	RTS
2609	F18B		;;
2610	F18B		; OUTPUT ACC TO TAPE BUFFER SUBROUTINE
2611	F18B		; & WHEN FULL OUTPUT BUFF TO TAPE.
2612	F18B		; IF INFLG=OUTFLG= T USE TWO BUFFERS
2613	F18B		; OTHERWISE USE SAME BUFFER FOR INPUT
2614	F18B		; AND OUTPUT (MONIT BUFFER)
2615	F18B	20 9E EB	TORYTE JSR PHXY ; SAVE X
2616	F18E	AE 37 A4	LUX TAPTR2 ; TAPE BUFFER POINTER FOR OUTPUT
2617	F191	20 0F F2	JSR BKCK2 ; STORE IN BUFFER
2618	F194	E8	INX
2619	F195	8E 37 A4	STX TAPTR2 ; FOR NEXT
2620	F198	E0 50	CPX #B0 ; BUFFER FULL?
2621	F19A	D0 32	BNE TABY3 ; NO, GO BACK
2622	F19C		; OUTPUT A BLOCK FROM BUFFER TO TAPE
2623	F19C	20 E7 F1	JSR BKCKSM ; COMPUTE BLOCK CHECKSUM
2624	F19F	20 1D F2	JSR TAOSST ; SET TAPE FOR OUTPUT
2625	F1A2	A9 23	LDA #'# ; CHAR FOR BEGINNING
2626	F1A4	20 4A F2	JSR OUTTAP ; OF BLOCK
2627	F1A7		; OUTPUT CHRS FROM ACTIVE BUFFER
2628	F1A7	20 D2 F1	TABY2 JSR CKBUFF ; LOAD CHR FROM ACTIVE BUFFER
2629	F1AA	20 4A F2	JSR OUTTAP ; FROM BUFFER
2630	F1AD	E8	INX
2631	F1AE	E0 53	CPX #B3 ; 2 BLOCK CKSUM CHR + 1 EXTRA CHR.
2632	F1B0	D0 F5	BNE TABY2 ; OTHERWISE ERROR
2633	F1B2	AD 00 AB	LDA DRB
2634	F1B5	29 CF	AND #*CF ; TURN TAPES OFF PB5,PB4
2635	F1B7	BD 00 AB	STA DRB
2636	F1BA	58	CLI ; ENABLE INTERRUPT
2637	F1BB	A9 00	LDA #0
2638	F1BD	BD 37 A4	STA TAPTR2 ; CLR TAPE BUFF PTR
2639	F1C0	A9 00	LDA #111 ; RESET FREE RUNNING TO 1 SHOT
2640	F1C2	BD 0B AB	STA ACR
2641	F1C5	20 9A FF	JSR PAT22 ; ADD 1 TO BLK COUNT & OUTPUT
2642	F1C8	AD 68 01	LDA BLKO ; PUT BLK CNT IN FIRST LOC (TABUFF)
2643	F1CB	20 8B F1	JSR TORYTE
2644	F1CE	20 AC EB	TABY3 JSR PLXY
2645	F1D1	60	RTS
2647	F1D2		; CHECK ACTIVE BUFFER AND LOAD A CHR
2648	F1D2		; CARRY=0 IF ONLY 1 BUFFER ,C=1 IF 2 BUFFERS
2649	F1D2	AD 12 A4	CKBUFF LDA INFLG
2650	F1D5	CD 13 A4	CMP OUTFLG
2651	F1D8	D0 08	BNE CRUFF1
2652	F1DA	C9 54	CMP #'T ; SEE IF INFLG=OUTFLG = T
2653	F1DC	D0 04	BNE CBUFF1

```

LINE # LOC CODE LINE
2654 F1DE 39 SEC ;USE PAGE 1 FOR OUTPUT BUFFER
2655 F1DF B5 AD LDA TABUF2,X
2656 F1E1 60 RTS
2657 F1E2 18 CRUFF1 CLC ;USE SAME BUFFER FOR I/O
2658 F1E3 BD 16 Q1 LDA TABUFF,X
2659 F1E6 60 RTS

2661 F1E7 ;COMPUTE BLOCK CHECKSUM & PUT IT
2662 F1E7 ;AT THE END OF ACTIVE BUFFER
2663 F1E7 A9 00 BKCKSM LDA #0 ;CLEAR BLK CKSUM LOCAT
2664 F1E9 BD 66 01 STA TABUFF+80
2665 F1EC BD 67 01 STA TABUFF+81
2666 F1EF A2 4F LDX #79
2667 F1F1 20 D2 F1 BKCK1 JSR CKBUFF ;GET CHR FROM EITHER BUFFER
2668 F1F4 18 CLC
2669 F1F5 6D 66 01 ADC TABUFF+80 ;ADD TO CKSUM
2670 F1FB BD 66 01 STA TABUFF+80
2671 F1FB 90 03 BCC #+5
2672 F1FD EE 67 01 INC TABUFF+B1
2673 F200 CA DEX
2674 F201 10 EE BPL BKCK1 ;DO THE WHOLE BUFFER
2675 F203 A2 50 LDX #80
2676 F205 AD 66 01 LDA TABUFF+80 ;PUT CKSUM INTO RIGHT BUFFER
2677 F208 20 0F F2 JSR BKCK2
2678 F20B EB INX
2679 F20C AD 67 01 LDA TABUFF+B1
2680 F20F 48 BKCK2 PHA ;OUTPUT A CHAR TO RIGHT BUFFER
2681 F210 20 D2 F1 JSR CKBUFF ;GET WHICH BUFFER
2682 F213 68 PLA
2683 F214 B0 04 BCS BKCK3 ;BRNCH TO SECOND BUFFER
2684 F216 9D 16 01 STA TABUFF,X
2685 F219 60 RTS
2686 F21A 95 AD BKCK3 STA TABUF2,X ;TO PAG 1
2687 F21C 60 RTS

2689 F21D ;SET TAPE (1 OR 2) FOR OUTPUT
2690 F21D 20 C0 F2 TAOS1 JSR SETSPD ;SET UP SPEED (# OF HALF PULSES)
2691 F220 AD 35 A4 LDA TAPOUT ;OUTPUT FLG (TAPE 1 OR 2)
2692 F223 20 1C EE JSR T10SET ;SET PB4 OR PB5 TO ZERO
2693 F226 A9 EC LDA #DATOUT+MOFF ;SET CA2=0 (DATA OUT)
2694 F228 BD 0C AB STA PCR
2695 F22E A9 C0 LDA #T1FR ;SET TIMER IN FREE RUNNING
2696 F22D BD 0B AB STA ACR
2697 F230 A9 00 LDA #00
2698 F232 BD 05 AB STA T1CH ;START TIMER T1
2699 F235 AE 07 A4 LDX GAP ;OUTPUT 4*GAP SYN BYTES
2700 F238 A9 16 TAOS1 LDA #16 ;SYN CHAR
2701 F23A 20 4A F2 JSR OUTTAP ;TO TAPE
2702 F23D 20 4A F2 JSR OUTTAP
2703 F240 20 4A F2 JSR OUTTAP
2704 F243 20 4A F2 JSR OUTTAP
2705 F246 CA DEX
2706 F247 D0 EF BNE TAOS1
2707 F249 60 RTS
    
```

```

LINE # LOC CODE LINE
2709 F24A ;OUTPUT ACC TO TAPE
2710 F24A BE 2D A4 OUTTAP STX CFIY+3 ;SAVE X
2711 F24D A0 07 LDY #07 ;FOR THE 8 BITS
2712 F24F 8C 27 A4 STY STIY
2713 F252 AE 08 A4 LDX TSPEED
2714 F255 30 39 BMI OUTTA1 ;IF ONE IS SUPER HIPER
2715 F257 48 PHA
2716 F258 A0 02 TRY LDY #2 ;SEND 3 UNITS
2717 F25A BC 28 A4 STY STIY+1 ;STARTING AT 3700 HZ
2718 F25D BE 0A A4 ZON LIX NPUL,Y ;#OF HALF CYCLES
2719 F260 48 PHA
2720 F261 B9 0B A4 ZON1 LDA TIMG,Y ;SET UP LACTH FOR NEXT
2721 F264 BD 06 AB STA T1LL ;PULSE (80 OR CA) (FREC)
2722 F267 A9 00 LDA #0
2723 F269 BD 07 AB STA T1LH
2724 F26C 2C 0D AB ZON2 BIT IFR ;WAIT FOR PREVIOUS
2725 F26F 50 FR RVC ZON2 ;CYCLE (T1 INT FLG)
2726 F271 AD 04 AB LDA T1L ;CLR INTERR FLG
2727 F274 CA DEX
2728 F275 D0 EA BNE ZON1 ;SEND ALL CYCLES
2729 F277 68 PLA
2730 F278 CE 28 A4 DEC STIY+1
2731 F27B F0 05 BEQ SETZ ;BRCH IF LAST ONE
2732 F27D 30 07 BMI ROUT ;BRCH IF NO MORE
2733 F27F 4A LSR A ;TAKE NEXT BIT
2734 F280 90 DB BCC ZON ;... IF IT'S A ONE...
2735 F282 A0 00 SETZ LDY #0 ;SWITCH TO 2400 HZ
2736 F284 F0 D7 BEQ ZON ;UNCONDITIONAL BRCH
2737 F286 CE 27 A4 ROUT DEC STIY ;ONE LESS BIT
2738 F289 10 CD BPL TRY ;ANY MORE? GO BACK
2739 F28B 68 ROUT1 PLA ;RECOVER CHR
2740 F28C AE 2D A4 LDX CFIY+3 ;RESTORE X
2741 F28F 60 RTS

2743 F290 ;OUTPUT HALF PULSE FOR 0 (1200 HZ) $
2744 F290 ;TWO HALF PULSES FOR 1 (2400 HZ) (00 TSPEED)
2745 F290 48 OUTTA1 PHA
2746 F291 BD 28 A4 STA STIY+1 ;STORE ACC
2747 F294 A2 02 OUTTA2 LDX #2 ;# OF HALF PULSES
2748 F296 A9 D0 LDA #D0 ;1/2 PULSE OF 2400
2749 F298 BD 06 AB STA T1LL
2750 F29B A9 00 LDA #00
2751 F29D BD 07 AB STA T1LH
2752 F2A0 20 BC FF JSR PATC25 ;WAIT TILL COMPLETED
2753 F2A3 4E 28 A4 LSR STIY+1 ;GET BITS FROM CHR
2754 F2A6 B0 0A BCS OUTTA3
2755 F2AB A9 00 LDA #A0 ;BIT=0 ,OUTPUT 1200 HZ
2756 F2AA BD 06 AB STA T1LL
2757 F2AD A9 01 LDA #01
2758 F2AF BD 07 AB STA T1LH
2759 F2B2 20 BC FF OUTTA3 JSR PATC25
2760 F2B5 CA DEX
2761 F2B6 10 FA BPL OUTTA3 ;OUTPUT 3 HALF PULSES
2762 F2BB B8 DEY
2763 F2B9 10 D9 BPL OUTTA2 ;ALL BITS ?
    
```

LINE #	LOC	CODE	LINE
2764	F2BB	4C 8B F2	JMP ROUT1 ;RESTORE REGS
2765	F2BE	EA	NOP
2766	F2BF	EA	NOP
2768	F2C0		;SET SPEED FROM NORMAL TO 3 TIMES NORMAL
2769	F2C0	AD 08 A4	SETSPD LDA TSPEED ;SPEED FLG
2770	F2C3	6A	ROR A ;NORMAL OR 3* NORM
2771	F2C4	A9 0C	LDA #12
2772	F2C6	90 02	BCC SETSP1
2773	F2C8	A9 04	LDA #4
2774	F2CA	8D 0A A4	SETSP1 STA NPUL
2775	F2CD	A9 12	LDA #18
2776	F2CF	90 02	BCC SETSP2
2777	F2D1	A9 06	LDA #6
2778	F2D3	8D 0C A4	SETSP2 STA TIMG+1
2779	F2D6	60	RTS
2780	F2D7		.FILE A3/2

LINE #	LOC	CODE	LINE
2782	F2D7		; ADDRESS TABLE FOR EACH PRINT COLUMN.
2783	F2D7		; EACH TBL CONTAINS DOT PATTERNS FOR 1 OF THE 5 COLUMNS.
2784	F2D7		; DATA ARE STORED WITH EACH BYTE DEFINING ONE COLUMN...
2785	F2D7		; OF A CHARACTER, WITH THE TOP DOT CORRESPONDING TO THE...
2786	F2D7		; LSB IN THE BYTE
2787	F2D7	E1 F2	MTBL .WOR COLO,COL1,COL2,COL3,COL4
2787	F2D9	21 F3	
2787	F2DB	61 F3	
2787	F2DD	A1 F3	
2787	F2DF	E1 F3	
2789	F2E1		;DOT PATTERNS FOR COLUMN ZERO (LEFTMOST COLUMN)
2790	F2E1	3E	COLO .BYT \$3E,\$7E,\$7F,\$3E,\$7F,\$7F,\$7F,\$3E @ -- G
2790	F2E2	7E	
2790	F2E3	7F	
2790	F2E4	3E	
2790	F2E5	7F	
2790	F2E6	7F	
2790	F2E7	7F	
2790	F2E8	3E	
2791	F2E9	7F	.BYT \$7F,\$00,\$20,\$7F,\$7F,\$7F,\$7F,\$3E H -- 0
2791	F2EA	00	
2791	F2EB	20	
2791	F2EC	7F	
2791	F2ED	7F	
2791	F2EE	7F	
2791	F2EF	7F	
2791	F2F0	3E	
2792	F2F1	7F	.BYT \$7F,\$3E,\$7F,\$46,\$01,\$3F,\$07,\$7F F -- W
2792	F2F2	3E	
2792	F2F3	7F	
2792	F2F4	46	
2792	F2F5	01	
2792	F2F6	3F	
2792	F2F7	07	
2792	F2F8	7F	
2793	F2F9	63	.BYT \$63,\$07,\$61,\$7F,\$03,\$00,\$02,\$40 X -- C
2793	F2FA	07	
2793	F2FB	61	
2793	F2FC	7F	
2793	F2FD	03	
2793	F2FE	00	
2793	F2FF	02	
2793	F300	40	
2794	F301	00	.BYT \$00,\$00,\$00,\$14,\$24,\$63,\$60,\$00 -- /
2794	F302	00	
2794	F303	00	
2794	F304	14	
2794	F305	24	
2794	F306	63	
2794	F307	60	
2794	F308	00	
2795	F309	00	.BYT \$00,\$00,\$14,\$08,\$40,\$08,\$40,\$60 ( -- /
2795	F30A	00	
2795	F30B	14	

## DOT PATTERNS

PA00-J001A.....PAGE 0058

LINE #	LOC	CODE	LINE
2795	F30C	08	
2795	F30D	40	
2795	F30E	08	
2795	F30F	40	
2795	F310	60	
2796	F311	3E	.BYT \$3E,\$44,\$62,\$41,\$18,\$27,\$3C,\$01 0 -- 7
2796	F312	44	
2796	F313	62	
2796	F314	41	
2796	F315	18	
2796	F316	27	
2796	F317	3C	
2796	F318	01	
2797	F319	36	.BYT \$36,\$46,\$00,\$40,\$08,\$14,\$41,\$02 8 -- 7
2797	F31A	46	
2797	F31B	00	
2797	F31C	40	
2797	F31D	08	
2797	F31E	14	
2797	F31F	41	
2797	F320	02	
2799	F321		.DOT PATTERNS FOR COLUMN 1
2800	F321	41	COL1 .BYT \$41,\$09,\$49,\$41,\$41,\$49,\$09,\$41 e -- G
2800	F322	09	
2800	F323	49	
2800	F324	41	
2800	F325	41	
2800	F326	49	
2800	F327	09	
2800	F328	41	
2801	F329	08	.BYT \$08,\$41,\$40,\$08,\$40,\$02,\$06,\$41 H -- D
2801	F32A	41	
2801	F32B	40	
2801	F32C	08	
2801	F32D	40	
2801	F32E	02	
2801	F32F	06	
2801	F330	41	
2802	F331	09	.BYT \$09,\$41,\$09,\$49,\$01,\$40,\$18,\$20 P -- W
2802	F332	41	
2802	F333	09	
2802	F334	49	
2802	F335	01	
2802	F336	40	
2802	F337	18	
2802	F338	20	
2803	F339	14	.BYT \$14,\$08,\$51,\$41,\$04,\$00,\$01,\$40 X -- C
2803	F33A	08	
2803	F33B	51	
2803	F33C	41	
2803	F33D	04	
2803	F33E	00	
2803	F33F	01	
2803	F340	40	

## DOT PATTERNS

PA00-J001A.....PAGE 0059

LINE #	LOC	CODE	LINE
2804	F341	00	.BYT \$00,\$00,\$07,\$7F,\$2A,\$13,\$4E,\$04 -- /
2804	F342	00	
2804	F343	07	
2804	F344	7F	
2804	F345	2A	
2804	F346	13	
2804	F347	4E	
2804	F348	04	
2805	F349	1C	.BYT \$1C,\$41,\$08,\$08,\$30,\$08,\$00,\$10 ( -- /
2805	F34A	41	
2805	F34B	08	
2805	F34C	08	
2805	F34D	30	
2805	F34E	08	
2805	F34F	00	
2805	F350	10	
2806	F351	51	.BYT \$51,\$42,\$51,\$41,\$14,\$45,\$4A,\$71 0 -- 7
2806	F352	42	
2806	F353	51	
2806	F354	41	
2806	F355	14	
2806	F356	45	
2806	F357	4A	
2806	F358	71	
2807	F359	49	.BYT \$49,\$49,\$00,\$34,\$14,\$14,\$41,\$01 8 -- 7
2807	F35A	49	
2807	F35B	00	
2807	F35C	34	
2807	F35D	14	
2807	F35E	14	
2807	F35F	41	
2807	F360	01	
2809	F361		.DOT PATTERNS FOR COLUMN 2
2810	F361	5D	COL2 .BYT \$5D,\$09,\$49,\$41,\$41,\$49,\$09,\$41 e -- G
2810	F362	09	
2810	F363	49	
2810	F364	41	
2810	F365	41	
2810	F366	49	
2810	F367	09	
2810	F368	41	
2811	F369	08	.BYT \$08,\$7F,\$41,\$14,\$40,\$0C,\$08,\$41 H -- D
2811	F36A	7F	
2811	F36B	41	
2811	F36C	14	
2811	F36D	40	
2811	F36E	0C	
2811	F36F	08	
2811	F370	41	
2812	F371	09	.BYT \$09,\$51,\$19,\$49,\$7F,\$40,\$60,\$18 P -- W
2812	F372	51	
2812	F373	19	
2812	F374	49	
2812	F375	7F	

LINE #	LOC	CODE	LINE
2812	F376	40	
2812	F377	60	
2812	F378	18	
2813	F379	08	.BYT \$0B,\$7B,\$49,\$41,\$0B,\$41,\$01,\$40 X -- C
2813	F37A	78	
2813	F37B	49	
2813	F37C	41	
2813	F37D	0B	
2813	F37E	41	
2813	F37F	01	
2813	F380	40	
2814	F381	00	.BYT \$00,\$4F,\$00,\$14,\$7F,\$0B,\$59,\$02 --- /
2814	F382	4F	
2814	F383	00	
2814	F384	14	
2814	F385	7F	
2814	F386	0B	
2814	F387	59	
2814	F388	02	
2815	F389	22	.BYT \$22,\$22,\$3E,\$3E,\$00,\$0B,\$00,\$0B ( -- /
2815	F38A	22	
2815	F38B	3E	
2815	F38C	3E	
2815	F38D	00	
2815	F38E	0B	
2815	F38F	00	
2815	F390	0B	
2816	F391	49	.BYT \$49,\$7F,\$51,\$49,\$12,\$45,\$49,\$09 0 -- 7
2816	F392	7F	
2816	F393	51	
2816	F394	49	
2816	F395	12	
2816	F396	45	
2816	F397	49	
2816	F398	09	
2817	F399	49	.BYT \$49,\$49,\$44,\$00,\$22,\$14,\$22,\$51 8 -- ?
2817	F39A	49	
2817	F39B	44	
2817	F39C	00	
2817	F39D	22	
2817	F39E	14	
2817	F39F	22	
2817	F3A0	51	
; DOT PATTERNS FOR COLUMN 3			
2819	F3A1		
2820	F3A1	55	COL3 .BYT \$55,\$09,\$49,\$41,\$22,\$49,\$09,\$49 @ -- 6
2820	F3A2	09	
2820	F3A3	49	
2820	F3A4	41	
2820	F3A5	22	
2820	F3A6	49	
2820	F3A7	09	
2820	F3A8	49	
2821	F3A9	0B	.BYT \$0B,\$41,\$3F,\$22,\$40,\$02,\$30,\$41 H -- 0
2821	F3AA	41	

LINE #	LOC	CODE	LINE
2821	F3AB	3F	
2821	F3AC	22	
2821	F3AD	40	
2821	F3AE	02	
2821	F3AF	30	
2821	F3B0	41	
2822	F3B1	09	.BYT \$09,\$21,\$29,\$49,\$01,\$40,\$18,\$20 P -- W
2822	F3B2	21	
2822	F3B3	29	
2822	F3B4	49	
2822	F3B5	01	
2822	F3B6	40	
2822	F3B7	18	
2822	F3B8	20	
2823	F3B9	14	.BYT \$14,\$0B,\$45,\$00,\$10,\$41,\$01,\$40 X -- C
2823	F3BA	0B	
2823	F3BB	45	
2823	F3BC	00	
2823	F3BD	10	
2823	F3BE	41	
2823	F3BF	01	
2823	F3C0	40	
2824	F3C1	00	.BYT \$00,\$00,\$07,\$7F,\$2A,\$64,\$26,\$01 --- /
2824	F3C2	00	
2824	F3C3	07	
2824	F3C4	7F	
2824	F3C5	2A	
2824	F3C6	64	
2824	F3C7	26	
2824	F3C8	01	
2825	F3C9	41	.BYT \$41,\$1C,\$0B,\$0B,\$00,\$0B,\$00,\$04 ( -- /
2825	F3CA	1C	
2825	F3CB	0B	
2825	F3CC	0B	
2825	F3CD	00	
2825	F3CE	0B	
2825	F3CF	00	
2825	F3D0	04	
2826	F3D1	45	.BYT \$45,\$40,\$49,\$55,\$7F,\$45,\$49,\$05 0 -- 7
2826	F3D2	40	
2826	F3D3	49	
2826	F3D4	55	
2826	F3D5	7F	
2826	F3D6	45	
2826	F3D7	49	
2826	F3D8	05	
2827	F3D9	49	.BYT \$49,\$29,\$00,\$00,\$41,\$14,\$14,\$09 8 -- ?
2827	F3DA	29	
2827	F3DB	00	
2827	F3DC	00	
2827	F3DD	41	
2827	F3DE	14	
2827	F3DF	14	
2827	F3E0	09	
2828	F3E1		
; DOT PATTERNS FOR COLUMN 4			



DOT PATTERNS

PA00-J001A.....PAGE 0062

LINE #	LOC	CODE	LINE
2829	F3E1	1E	COL4 .BYT \$1E,\$7E,\$36,\$22,\$1C,\$41,\$01,\$7A @ -- G
2829	F3E2	7E	
2829	F3E3	36	
2829	F3E4	22	
2829	F3E5	1C	
2829	F3E6	41	
2829	F3E7	01	
2829	F3E8	7A	
2830	F3E9	7F	.BYT \$7F,\$00,\$01,\$41,\$40,\$7F,\$7F,\$3E H -- 0
2830	F3EA	00	
2830	F3EB	01	
2830	F3EC	41	
2830	F3ED	40	
2830	F3EE	7F	
2830	F3EF	7F	
2830	F3F0	3E	
2831	F3F1	06	.BYT \$06,\$5E,\$46,\$31,\$01,\$3F,\$07,\$7F P -- W
2831	F3F2	5E	
2831	F3F3	46	
2831	F3F4	31	
2831	F3F5	01	
2831	F3F6	3F	
2831	F3F7	07	
2831	F3F8	7F	
2832	F3F9	63	.BYT \$63,\$07,\$43,\$00,\$60,\$7F,\$02,\$40 X -- C
2832	F3FA	07	
2832	F3FB	43	
2832	F3FC	00	
2832	F3FD	60	
2832	F3FE	7F	
2832	F3FF	02	
2832	F400	40	
2833	F401	00	.BYT \$00,\$00,\$00,\$14,\$12,\$63,\$50,\$00 -- /
2833	F402	00	
2833	F403	00	
2833	F404	14	
2833	F405	12	
2833	F406	63	
2833	F407	50	
2833	F408	00	
2834	F409	00	.BYT \$00,\$00,\$14,\$08,\$00,\$08,\$00,\$03 < -- /
2834	F40A	00	
2834	F40B	14	
2834	F40C	08	
2834	F40D	00	
2834	F40E	08	
2834	F40F	00	
2834	F410	03	
2835	F411	3E	.BYT \$3E,\$40,\$46,\$22,\$10,\$39,\$31,\$03 0 -- 7
2835	F412	40	
2835	F413	46	
2835	F414	22	
2835	F415	10	
2835	F416	39	
2835	F417	31	

DOT PATTERNS

PA00-J001A.....PAGE 0063

LINE #	LOC	CODE	LINE
2835	F418	03	
2836	F419	36	
2836	F41A	1E	.BYT \$36,\$1E,\$00,\$00,\$41,\$14,\$08,\$06 B -- ?
2836	F41B	00	
2836	F41C	00	
2836	F41D	41	
2836	F41E	14	
2836	F41F	08	
2836	F420	06	
2838	F421		ASCII CHARACTERS FOR KB
2839	F421	20	ROW1 .BYT \$20,\$0B,\$00,\$0D,\$00,\$00,\$00,\$00
2839	F422	08	
2839	F423	00	
2839	F424	0D	
2839	F425	00	
2839	F426	00	
2839	F427	00	
2839	F428	00	
2840	F429	00	ROW2 .BYT \$00,\$60,'\'', \$00,\$00,\$00,\$7F,\$00
2840	F42A	60	
2840	F42B	5C	
2840	F42C	00	
2840	F42D	00	
2840	F42E	00	
2840	F42F	7F	
2840	F430	00	
2841	F431	2E 4C	ROW3 .BYT 'LP=0,/'
2842	F439	4D 4A	ROW4 .BYT 'MJ109BK,'
2843	F441	42 47	ROW5 .BYT 'BGYU76HN'
2844	F449	43 44	ROW6 .BYT 'CDRT54FV'
2845	F451	5A 41	ROW7 .BYT 'ZANE32SX'
2846	F459	00	ROW8 .BYT \$00,\$00,\$1B,'01',\$5E,'JL'
2846	F45A	00	
2846	F45B	1B	
2846	F45C	51 31	
2846	F45E	5E	
2846	F45F	5D 5B	

LINE #	LOC	CODE	LINE
2848	F461		;DISASSEMBLE INSTRUCTION AND SHOW REGS IS REGF SET
2849	F461	AD 0E A4	REGQ LDA REGF ;GET FLAG
2850	F464	F0 06	BEQ DISASM
2851	F466	20 32 E2	JSR REG1 ;SHOW THE SIX REGS
2852	F469	20 24 EA	JSR CRCK ;CCR)
2854	F46C	20 45 F5	DISASM JSR FRBL2
2855	F46F	20 3C F5	JSR PRFC ;OUTPUT PRG COUNTR
2856	F472	A0 00	LDY #0
2857	F474	20 56 EB	JSR PCLLD
2858	F477	A8	TAY
2859	F478	4A	LSR A
2860	F479	90 08	BCC IEVEN
2861	F47B	4A	LSR A
2862	F47C	B0 17	BCS ERR
2863	F47E	C9 22	CHP ##22
2864	F480	F0 13	BEQ ERR
2865	F482	29 07	AND #7
2866	F484	09 80	ORA ##80
2867	F486	4A	IEVEN LSR A
2868	F487	AA	TAX
2869	F48B	B0 5B F5	LDA MODE,X
2870	F48B	B0 04	BCC RTMODE
2871	F48D	4A	LSR A
2872	F48E	4A	LSR A
2873	F48F	4A	LSR A
2874	F490	4A	LSR A
2875	F491	29 0F	RTMODE AND ##F
2876	F493	D0 04	BNE GETFMT
2877	F495	A0 80	ERR LDY ##80
2878	F497	A9 00	LDA #0
2879	F499	AA	GETFMT TAX
2880	F49A	B0 9F F5	LDA MODE2,X
2881	F49D	B0 16 01	STA FORMA
2882	F4A0	29 03	AND #3
2883	F4A2	85 EA	STA LENGTH
2884	F4A4	98	TYA ;OPCODE
2885	F4A5	29 8F	AND ##8F
2886	F4A7	AA	TAX
2887	F4A8	98	TYA ;OPCODE IN A AGAIN
2888	F4A9	A0 03	LDY #3
2889	F4AB	E0 8A	CPX ##8A
2890	F4AD	F0 0B	BEQ MNNIX3
2891	F4AF	4A	MNNIX1 LSR A
2892	F4B0	90 0B	BCC MNNIX3
2893	F4B2	4A	LSR A
2894	F4B3	4A	MNNIX2 LSR A
2895	F4B4	09 20	ORA ##20
2896	F4B6	B8	DEY
2897	F4B7	D0 FA	BNE MNNIX2
2898	F4B9	CB	INY
2899	F4BA	B8	MNNIX3 DEY
2900	F4BB	D0 F2	BNE MNNIX1
2901	F4BD	48	PHA ;SAVE MNEMONIC TABLE INDEX
2902	F4BE	20 56 EB	JSR PCLLD

LINE #	LOC	CODE	LINE
2903	F4C1	20 46 EA	JSR NUMA
2904	F4C4	20 45 F5	JSR FRBL2 ;PRINT LAST BLANK
2905	F4C7	68	FLA
2906	F4C8	AB	TAY
2907	F4C9	B9 B9 F5	LDA MNEML,Y
2908	F4CC	B0 17 01	STA LMNEM
2909	F4CF	B9 F9 F5	LDA MNEMR,Y
2910	F4D2	B0 18 01	STA RMNEM
2911	F4D5	A2 03	LDX #3 ;MUST BE
2912	F4D7	A9 00	FRMN1 LDA #0
2913	F4D9	A0 05	LDY #5
2914	F4DB	0E 18 01	FRMN2 ASL RMNEM
2915	F4DE	2E 17 01	ROL LMNEM
2916	F4E1	2A	ROL A
2917	F4E2	88	DEY
2918	F4E3	D0 F6	BNE FRMN2
2919	F4E5	69 BF	ADC ##BF ;ADD '?' OFFSET
2920	F4E7	20 BC E9	JSR OUTALL
2921	F4EA	CA	DEX
2922	F4EB	D0 EA	BNE FRMN1
2923	F4ED	20 45 F5	JSR FRBL2
2924	F4F0	A2 06	LDX #6
2925	F4F2	A9 00	LDA #0
2926	F4F4	B0 29 A4	STA STIY+2 ;FLAG
2927	F4F7	E0 03	PRADR1 CPX #3
2928	F4F9	D0 1E	BNE PRADR3 ;IF X=3 PRINT ADDR VALUE
2929	F4FB	A4 EA	LDY LENGTH
2930	F4FD	F0 1A	BEQ PRADR3 ;1 BYTE INSTR
2931	F4FF	AD 16 01	PRADR2 LDA FORMA
2932	F502	C9 E8	CMF ##E8 ;RELATIVE ADDRESSING
2933	F504	20 56 EB	JSR PCLLD
2934	F507	B0 27	BCC RELADR
2935	F509		;SEE IF SYMBOL
2936	F509	48	PHA
2937	F50A	AD 29 A4	LDA STIY+2
2938	F50D	D0 03	BNE MR11A
2939	F50F	EE 29 A4	INC STIY+2 ;SHOW WE WERE HERE
2941	F512	68	MR11A FLA
2942	F513	20 46 EA	JSR NUMA
2943	F516	88	DEY
2944	F517	D0 E6	BNE PRADR2
2945	F519	0E 16 01	PRADR3 ASL FORMA
2946	F51C	90 0E	BCC PRADR4
2947	F51E	B0 AC F5	LDA CHAR1-1,X
2948	F521	20 BC E9	JSR OUTALL
2949	F524	B0 B2 F5	LDA CHAR2-1,X
2950	F527	F0 03	BEQ PRADR4
2951	F529	20 BC E9	JSR OUTALL
2952	F52C	CA	PRADR4 DEX
2953	F52D	D0 C8	BNE PRADR1
2954	F52F	60	RTS
2955	F530	20 4D F5	RELADR JSR PCADJ3
2956	F533	AA	TAX
2957	F534	EB	INX

LINE #	LOC	CODE	LINE
2958	F535	D0 01	BNE PRNTYX
2959	F537	C8	INY
2960	F538	98	PRNTYX TYA
2961	F539	4C 42 EA	JMP WRAX ;PRINT A &X
2962	F53C	AD 26 A4	PRPC LDA SAVPC+1 ;PRINT PC
2963	F53F	AE 25 A4	LIX SAVPC
2964	F542	20 42 EA	JSR WRAX
2965	F545	A9 20	PRBL2 LDA #*20
2966	F547	4C 8C E9	JMP OUTALL
2967	F54A	A5 EA	LDA LENGTH
2968	F54C	38	SEC
2969	F54D	AC 26 A4	PCAIJ3 LDY SAVPC+1 ;PRG CNTR HIGH
2970	F550	AA	TAX
2971	F551	10 01	RPL PCAIJ4
2972	F553	88	DEY
2973	F554	6D 25 A4	PCAIJ4 ADC SAVPC ;PRG CNTR LOW
2974	F557	90 01	BCC RTS1
2975	F559	C8	INY
2976	F55A	60	RTS1 RTS
2978	F55B	40	MODE .BYT \$40,2,\$45,3,\$D0,8,\$40,9
2978	F55C	02	
2978	F55D	45	
2978	F55E	03	
2978	F55F	D0	
2978	F560	08	
2978	F561	40	
2978	F562	09	
2979	F563	30	.BYT \$30,\$22,\$45,\$33,\$D0,8,\$40,9
2979	F564	22	
2979	F565	45	
2979	F566	33	
2979	F567	D0	
2979	F568	08	
2979	F569	40	
2979	F56A	09	
2980	F56B	40	.BYT \$40,2,\$45,\$33,\$D0,8,\$40,9
2980	F56C	02	
2980	F56D	45	
2980	F56E	33	
2980	F56F	D0	
2980	F570	08	
2980	F571	40	
2980	F572	09	
2981	F573	40	.BYT \$40,2,\$45,\$B3,\$D0,8,\$40,9
2981	F574	02	
2981	F575	45	
2981	F576	B3	
2981	F577	D0	
2981	F578	08	
2981	F579	40	
2981	F57A	09	
2982	F57B	00	.BYT 0,\$22,\$44,\$33,\$D0,\$BC,\$44,0
2982	F57C	22	

LINE #	LOC	CODE	LINE
2982	F57D	44	
2982	F57E	33	
2982	F57F	D0	
2982	F580	8C	
2982	F581	44	
2982	F582	00	
2983	F583	11	.BYT \$11,\$22,\$44,\$33,\$D0,\$BC,\$44,\$9A
2983	F584	22	
2983	F585	44	
2983	F586	33	
2983	F587	D0	
2983	F588	8C	
2983	F589	44	
2983	F58A	9A	
2984	F58B	10	.BYT \$10,\$22,\$44,\$33
2984	F58C	22	
2984	F58D	44	
2984	F58E	33	
2985	F58F	D0	.BYT \$D0,8,\$40,9
2985	F590	08	
2985	F591	40	
2985	F592	09	
2986	F593	10	.BYT \$10,\$22,\$44,\$33,\$D0,8,\$40,9
2986	F594	22	
2986	F595	44	
2986	F596	33	
2986	F597	D0	
2986	F598	08	
2986	F599	40	
2986	F59A	09	
2987	F59B	62	.BYT \$62,\$13,\$78,\$A9
2987	F59C	13	
2987	F59D	78	
2987	F59E	A9	
2989	F59F	00	MODE2 .BYT 0,\$21,1,2,0,\$80,\$59,\$4D
2989	F5A0	21	
2989	F5A1	01	
2989	F5A2	02	
2989	F5A3	00	
2989	F5A4	80	
2989	F5A5	59	
2989	F5A6	4D	
2990	F5A7	11	.BYT \$11,\$12,6,\$4A,5,\$1D
2990	F5A8	12	
2990	F5A9	06	
2990	F5AA	4A	
2990	F5AB	05	
2990	F5AC	1D	
2992	F5AD	2C	CHAR1 .BYT ',',\$29,',#(',',','
2992	F5AE	29	
2992	F5AF	2C 23 28	
2992	F5B2	2E	
2993	F5B3	59	CHAR2 .BYT 'Y',0,'X',0,0,'A'

LINE #	LOC	CODE	LINE
2993	F5B4	00	
2993	F5B5	5B	
2993	F5B6	00	
2993	F5B7	00	
2993	F5B8	41	
2995	F5B9	1C	MNEML .BYT \$1C, \$8A, \$1C, \$23, \$5D, \$8B, \$1B
2995	F5BA	8A	
2995	F5BB	1C	
2995	F5BC	23	
2995	F5BD	5D	
2995	F5BE	8B	
2995	F5BF	1B	
2996	F5C0	A1	.BYT \$A1
2997	F5C1	9D	.BYT \$9D, \$8A, \$1D, \$23, \$9D, \$8B, \$1D, \$A1
2997	F5C2	8A	
2997	F5C3	1D	
2997	F5C4	23	
2997	F5C5	9D	
2997	F5C6	8B	
2997	F5C7	1D	
2997	F5C8	A1	
2998	F5C9	00	.BYT 0, \$29, \$19, \$AE, \$69, \$AB, \$19, \$23
2998	F5CA	29	
2998	F5CB	19	
2998	F5CC	AE	
2998	F5CD	69	
2998	F5CE	AB	
2998	F5CF	19	
2998	F5D0	23	
2999	F5D1	24	.BYT \$24, \$53, \$1B, \$23, \$24, \$53, \$19, \$A1
2999	F5D2	53	
2999	F5D3	1B	
2999	F5D4	23	
2999	F5D5	24	
2999	F5D6	53	
2999	F5D7	19	
2999	F5D8	A1	
3000	F5D9	00	.BYT 0, \$1A, \$5B, \$5B, \$A5, \$69, \$24, \$24
3000	F5DA	1A	
3000	F5DB	5B	
3000	F5DC	5B	
3000	F5DD	A5	
3000	F5DE	69	
3000	F5DF	24	
3000	F5E0	24	
3001	F5E1	AE	.BYT \$AE, \$AE, \$AB, \$AD, \$29, 0, \$7C, 0
3001	F5E2	AE	
3001	F5E3	AB	
3001	F5E4	AD	
3001	F5E5	29	
3001	F5E6	00	
3001	F5E7	7C	
3001	F5E8	00	
3002	F5E9	15	.BYT \$15, \$9C, \$6D, \$9C, \$A5, \$69, \$29, \$53

LINE #	LOC	CODE	LINE
3002	F5EA	9C	
3002	F5EB	6D	
3002	F5EC	9C	
3002	F5ED	A5	
3002	F5EE	69	
3002	F5EF	29	
3002	F5F0	53	
3003	F5F1	84	.BYT \$84, \$13, \$34, \$11, \$A5, \$69, \$23, \$A0
3003	F5F2	13	
3003	F5F3	34	
3003	F5F4	11	
3003	F5F5	A5	
3003	F5F6	69	
3003	F5F7	23	
3003	F5F8	A0	
3005	F5F9	D8	MNEMR .BYT \$D8, \$62, \$5A, \$4B, \$26, \$62, \$94
3005	F5FA	62	
3005	F5FB	5A	
3005	F5FC	4B	
3005	F5FD	26	
3005	F5FE	62	
3005	F5FF	94	
3006	F600	8B	.BYT \$8B
3007	F601	54	.BYT \$54, \$44, \$CB, \$54, \$68, \$44, \$EB, \$94
3007	F602	44	
3007	F603	CB	
3007	F604	54	
3007	F605	68	
3007	F606	44	
3007	F607	EB	
3007	F608	94	
3008	F609	00	.BYT 0, \$B4, B, \$B4, \$74, \$B4, \$2B, \$6E
3008	F60A	B4	
3008	F60B	0B	
3008	F60C	84	
3008	F60D	74	
3008	F60E	B4	
3008	F60F	2B	
3008	F610	6E	
3009	F611	74	.BYT \$74, \$F4, \$CC, \$4A, \$72, \$F2, \$A4, \$8A
3009	F612	F4	
3009	F613	CC	
3009	F614	4A	
3009	F615	72	
3009	F616	F2	
3009	F617	A4	
3009	F618	8A	
3010	F619	00	.BYT 0, \$AA, \$A2, \$A2, \$74, \$74, \$74, \$72
3010	F61A	AA	
3010	F61B	A2	
3010	F61C	A2	
3010	F61D	74	
3010	F61E	74	
3010	F61F	74	

LINE #	LOC	CODE	LINE
3010	F620	72	
3011	F621	44	
3011	F622	68	.BYT \$44,\$68,\$B2,\$32,\$B2,0,\$22,0
3011	F623	B2	
3011	F624	32	
3011	F625	B2	
3011	F626	00	
3011	F627	22	
3011	F628	00	
3012	F629	1A	.BYT \$1A,\$1A,\$26,\$26,\$72,\$72,\$88,\$C8
3012	F62A	1A	
3012	F62B	26	
3012	F62C	26	
3012	F62D	72	
3012	F62E	72	
3012	F62F	88	
3012	F630	C8	
3013	F631	C4	.BYT \$C4,\$CA,\$26,\$4B,\$44,\$44,\$A2,\$C8
3013	F632	CA	
3013	F633	26	
3013	F634	4B	
3013	F635	44	
3013	F636	44	
3013	F637	A2	
3013	F638	C8	

LINE #	LOC	CODE	LINE
3015	F639		;*****
3016	F639		;*** AIM TEXT EDITOR ***
3017	F639		;*** 05/01/78 ***
3018	F639		;*****
3020	F639		; R=READ FROM ANY INPUT DEVICE
3021	F639		; I=INSERT A LINE FROM INPUT DEV
3022	F639		; K=DELETE A LINE
3023	F639		; U=GO UP ONE LINE
3024	F639		; D=GO DOWN ONE LINE
3025	F639		; L=LIST LINES TO OUTPUT DEV
3026	F639		; T=GO TO TOP OF TEXT
3027	F639		; B=GO TO BOTTOM OF TEXT
3028	F639		; F=FIND STRING
3029	F639		; C=CHANGE STRING TO NEW STRING
3030	F639		; Q=QUIT EDITOR
3031	F639		; (SPACE)=DISPLAY CURRENT LINE
3033	F639		;***** E COMMAND-EDITOR ENTRY (FROM MONITOR) *****
3034	F639	20 13 EA	EDIT JSR CRLOW
3035	F63C	A0 6C	LDY #EMSG1-M1
3036	F63E	20 AF E7	JSR KEP ;START UP MSG
3037	F641	20 13 EA	JSR CRLOW
3038	F644	20 A3 E7	EDIO JSR FROM
3039	F647	B0 FB	BCS EDIO
3040	F649	AD 1E A4	LDA CKSUM ;IS CLR IF ADDR WAS INPUTTED
3041	F64C	F0 03	BEQ *+5
3042	F64E	20 DB E2	JSR WRITAZ ;OUTPUT DEFAULT ADDR (0200)
3043	F651	A2 01	LDX #1
3044	F653	BD 1C A4	EDI1 LDA ADDR,X
3045	F656	95 E3	STA TEXT,X
3046	F658	95 E1	STA BOTLN,X
3047	F65A	9D 1A A4	STA S1,X ;FOR MEMORY TEST
3048	F65D	CA	DEX
3049	F65E	10 F3	BPL EDI1
3050	F660	20 3B E8	JSR BLANK2
3051	F663	20 A7 E7	EDI2 JSR TO ;END
3052	F666	B0 FB	BCS EDI2
3053	F668	20 BC FB	JSR TOPNO ;TRANSF TEXT TO ADDR FOR RAM CHECK
3054	F66B	AD 1E A4	LDA CKSUM ;IS CLR IF ADDR WAS INPUTTED
3055	F66E	F0 10	BEQ EDI4 ;BRNCH IF NOT DEFAULT VALUE
3056	F670	20 34 F9	JSR SAVNOW
3057	F673	20 B6 F6	EDI3 JSR EDI ;CARRY IS SET IF NO RAM THERE
3058	F676	90 FB	BCC EDI3
3059	F678	A9 00	LDA #0 ;SET UPPER LIMIT TO BEGINNING...
3060	F67A	BD 1C A4	STA ADDR ;OF PAGE
3061	F67D	20 DB E2	JSR WRITAZ ;OUTPUT DEFAULT VALUE ,UPPER LIMIT
3062	F680	AD 1C A4	EDI4 LDA ADDR
3063	F683	85 E5	STA END
3064	F685	AD 1D A4	LDA ADDR+1
3065	F688	85 E6	STA END+1
3066	F68A	20 34 F9	JSR SAVNDW
3067	F68D		;NOW SEE IF MEMORY IS THERE

LINE #	LOC	CODE	LINE
3068	F68D	20 B6 F6	EDI5 JSR EDI
3069	F690	90 FB	BCC EDI5
3070	F692	A5 E6	LDA ENDR+1 ;CMP WITH END
3071	F694	CD 1D A4	CMP ADDR+1
3072	F697	F0 11	BEQ EDI7
3073	F699	80 13	BCS EDI8
3074	F69B	20 BC FB	EDI6 JSR TOPND ;RESTORE NOWLN
3075	F69E	A9 00	LDA #0
3076	F6A0	91 DF	STA (NOWLN)Y ;END OF TEXT MARKER
3077	F6A2	20 13 EA	JSR CRLOW
3078	F6A5	A9 52	LDA #'R ;FORCE READ COMMAND
3079	F6A7	4C 8D FA	JMP ENTRY
3080	F6AA	A5 E5	EDI7 LDA END ;IF ZERO MEM IS OK
3081	F6AC	F0 ED	BEQ EDI6
3082	F6AE	A9 00	EDI8 LDA #0
3083	F6B0	8D 1C A4	STA ADDR
3084	F6B3	4C 33 EB	JMP MEMERR ;NO MEMORY FOR THOSE LIMITS
3086	F6B6	A0 00	EDI LDY #0 ;CHK IF MEMORY WRITES
3087	F6B8	20 B7 FE	JSR PATCH6 ;GET BYTE ADDR BY ADDR,ADDR+1
3088	F6BB	4B	PHA ;SAVE IT
3089	F6BC	A9 AA	LDA #'AA ;SET THIS PATTERN
3090	F6BE	20 78 EB	JSR SADDR ;CHK IT
3091	F6C1	D0 09	BNE EDI2B
3092	F6C3	68	PLA
3093	F6C4	20 78 EB	JSR SADDR ;RESTORE CHR
3094	F6C7	EE 1D A4	INC ADDR+1 ;NEXT PAG
3095	F6CA	18	CLC ;IT WROTE
3096	F6CB	60	RTS
3097	F6CC	38	EDI2B SEC ;DIDNT WRITE
3098	F6CD	68	PLA
3099	F6CE	60	RTS
3101	F6CF		***** T COMMAND-REENTRY EDITOR *****
3102	F6CF		;RE-ENTRY POINT, TEXT ALREADY THERE
3103	F6CF	20 24 EA	REENTR JSR CRCK ;(CR) IF PRI ON
3104	F6D2	20 BC FB	TF JSR TOPND ;GO TO TOP
3105	F6D5	4C B9 F7	JMP IN03A ;DISPLAY LINE
3107	F6D8		***** U COMMAND-UP LINE *****
3108	F6D8		;GO UP ONE LINE BUT...
3109	F6D8		;DOWN IN ADDRESSING MEMORY
3110	F6D8	20 DB F8	DNNO JSR ATTOP ;THIS RTN DOESNT PRINT
3111	F6DB	90 06	BCC DOW1 ;NOT TOP
3112	F6DD	20 27 F7	JSR PLNE ;ARE AT TOP
3113	F6E0	4C 78 FA	JMP ERRO
3114	F6E3	A0 00	DOW1 LDY #0
3115	F6E5	20 1D F9	JSR SUB ;DECREMENT NOWLN PAST (CR)
3116	F6E8	20 1D F9	JSR SUB
3117	F6EB	20 DB F8	JSR ATTOP
3118	F6EE	80 30	BCS UP4
3119	F6F0	B1 DF	LDA (NOWLN)Y
3120	F6F2	C9 0D	CMP #CR
3121	F6F4	D0 F2	BKE DOW2
3122	F6F6	4C 28 F9	JMP AD1

LINE #	LOC	CODE	LINE
3124	F6F9		***** D COMMAND-DOWN LINE *****
3125	F6F9		;GO DOWN ONE LINE BUT...
3126	F6F9		;UP IN ADDRESSING MEMORY
3127	F6F9	20 09 F7	UP JSR UPND
3128	F6FC	20 27 F7	JSR PLNE ;DISPLAY LINE & CHCK BOTTOM
3129	F6FF	20 E9 FB	JSR ATBOT
3130	F702	90 1C	BCC UP4
3131	F704	AQ 72	LDY #EMSG2-M1 ;PRINT 'END'
3132	F706	4C AF E7	JMP KEP
3133	F709	A0 00	UPND LDY #0
3134	F70B	20 E9 FB	JSR ATBOT
3135	F70E	90 03	BCC UP1
3136	F710	4C 5C FA	JMP ENDERR
3137	F713	B1 DF	UP1 LDA (NOWLN)Y
3138	F715	F0 09	BEQ UP4
3139	F717	C8	INY
3140	F718	C9 0D	CMP #CR
3141	F71A	D0 F7	BNE UP1
3142	F71C	9B	TYA
3143	F71D	20 2A F9	JSR ADDA ;ADD LENGTH TO CURRENT LINE
3144	F720	60	UP4 RTS
3146	F721		***** B COMMAND-GO TO BOTTOM *****
3147	F721	20 C5 F8	BT JSR SETBOT
3148	F724		;START U-COMMAND HERE
3149	F724	20 DB F6	DOWN JSR DNNO ;U COMMAND
3151	F727		***** (SPACE) COMMAND-DISPLAY CURRENT LINE *****
3152	F727	A0 00	PLNE LDY #0 ;PRINT CURRENT LINE
3153	F729	B1 DF	F02 LDA (NOWLN)Y
3154	F72B	F0 0E	BEQ F01 ;FAST END ?
3155	F72D	C9 0D	CMP #CR ;DONE?
3156	F72F	F0 0A	BEQ F01
3157	F731	20 BC E9	JSR OUTALL ;PUT IT SOMEWHERE
3158	F734	99 3B A4	STA DIRUFF,Y
3159	F737	C8	INY
3160	F738	4C 29 F7	JMP F02
3161	F73B	84 EA	P01 STY LENGTH
3162	F73D	84 E9	STY OLDLEN
3163	F73F	AC 13 A4	P03 LDY OUTFLG ;ONE MORE (CR) FOR TAPE
3164	F742	C0 0D	CPY #CR
3165	F744	F0 03	BEQ F00
3166	F746	4C F0 E9	JMP CRLF ;TO OUTPUT DEV
3167	F749	4C 24 EA	P00 JMP CRCK ;(CR), & DONT CLR DISPL
3169	F74C		***** K COMMAND-KILL LINE *****
3170	F74C		;DELETE CURRENT LINE
3171	F74C	20 B6 FB	DLNE JSR KIFLG ;CLR K OR I COMM FLG
3172	F74F	EA	NOP
3173	F750	EA	NOP
3174	F751	EA	NOP
3175	F752	20 27 F7	JSR PLNE
3176	F755	20 E9 FB	JSR ATBOT
3177	F758	80 CD	BCS PLNE ;AT END OF TEXT
3178	F75A	A0 00	LDY #0

LINE #	LOC	CODE	LINE
3179	F75C	B4 EA	STY LENGTH
3180	F75E	20 3F F9	JSR REPLAC ; KILL LINE
3181	F761	4C 27 F7	JMP PLNE
!			
3183	F764		;***** I COMMAND-INSERT LINE *****
3184	F764	20 6D F7	IN JSR INL
3185	F767	20 F9 F6	JSR UP ; DISPLAY NEXT LINE DOWN
3186	F76A	4C 78 FA	JMP ERRO ; IF AT BOTTOM PRINT "END"
3187	F76D	20 B6 F8	INL JSR KIFLG ; CLR K OR I COMM FLG
3188	F770	A0 00	LDY #0 ; GET LINE INTO DIRUFF
3189	F772	B4 E9	STY OLDLEN
3190	F774	20 BD E7	JSR FROMPT
3191	F777	20 44 EB	JSR CLR
3192	F77A	20 93 E9	IN02 JSR INALL
3193	F77D	20 FB FE	JSR PATC12 ; CLR, SO WE CAN OUTPUT TO PRI
3194	F780	C9 7F	CMF #7F ; RUB
3195	F782	4C 2A FF	JMP PATC17 ; NO ZEROS IN CASE OF PAPER TAPE
3196	F785	C9 0A	IN02A CMF #1F
3197	F787	F0 F1	BEQ IN02
3198	F789	C9 0D	CMF #CR
3199	F78B	F0 1B	BEQ IN03
3200	F78D	C0 3C	CPY #60 ; DO NOT INCR Y IF 60
3201	F78F	B0 08	BCC IN03B
3202	F791	99 38 A4	STA DIRUFF,Y
3203	F794	C8	INY
3204	F795	C0 3C	CPY #60
3205	F797	D0 E1	BNE IN02 ; CONTIN, DISP WONT ALLOW > 60 CH
3206	F799	A0 3C	IN03B LDY #60 ; SET Y TO MAX OF 60
3207	F79B	A9 01	LDA #*01
3208	F79D	0D 11 A4	ORA PRIFLG ; DO NOT OUTPUT TO PRI ANY MORE
3209	F7A0	8D 11 A4	STA PRIFLG ; OTHERWISE CLOBBERS THE BUFFER
3210	F7A3	8C 15 A4	STY CURP02
3211	F7A6	D0 D2	BNE IN02 ; GO BACK
3212	F7A8	84 EA	IN03 STY LENGTH
3213	F7AA	C0 00	CPY #0 ; FIRST CHAR?
3214	F7AC	D0 17	BNE IN05
3215	F7AE	AD 19 A4	LDA COUNT ; K OR I COMM FLG ?
3216	F7B1	D0 12	BNE IN05 ; BRANCH IF C COMMAND
3217	F7B3	20 24 EA	JSR CRCK ; (CR) IF PRI PNTR DIFF FROM 0
3218	F7B6	20 03 FF	JSR PATC13 ; TURN ON TAPES & SET DEFAULT DEV
3219	F7B9	20 27 F7	IN03A JSR PLNE ; DISPLAY NEXT LINE DOWN
3220	F7BC	20 09 F7	JSR UPNO ; PRINT "END" IF BOTTOM
3221	F7B8	20 D8 F6	JSR DNNO
3222	F7C2	4C 78 FA	JMP ERRO
3223	F7C5	20 3F F9	IN05 JSR REPLAC ; INSERT THE LINE
3224	F7C8	4C 24 EA	JMP CRCK ; (CR) IF PRI PTR NOT 0
3226	F7CB		;***** R COMMAND-READ LINE *****
3227	F7CB		; READ TEXT FROM ANY INPUT DEVICE UNTIL
3228	F7CB		; TWO CONSECUTIVE (CR) ARE ENCOUNTER.
3229	F7CB	20 48 EB	INPU JSR WHEREI
3230	F7CE	AC 12 A4	LDY INFLG ; IF TAPE DO NOT ERRASE BUFFER
3231	F7D1	C0 54	CPY #T
3232	F7D3	F0 03	BEQ INPU1
3233	F7D5	20 13 EA	JSR CRLOW

LINE #	LOC	CODE	LINE
3234	F7D8	20 6D F7	INPU1 JSR INL
3235	F7DB	20 09 F7	JSR UPNO ; NEXT LINE
3236	F7DE	4C DB F7	JMP INPU1
3238	F7E1		;***** L COMMAND-LIST LINES *****
3239	F7E1		; PRINT FROM HERE N LINES TO ACTIVE OUTPUT DEV
3240	F7E1	20 37 E8	LST JSR PSL1 ; PRINT "/"
3241	F7E4	20 85 E7	JSR GCNT ; GET LINES COUNT
3242	F7E7	20 13 EA	JSR CRLOW
3243	F7EA	20 71 E8	JSR WHEREO ; WHERE TO
3244	F7ED	4C F8 F7	JMP LST02 ; ONE MORE LINE
3245	F7F0	20 07 E9	LST01 JSR RCHEK
3246	F7F3	20 90 E7	JSR DONE
3247	F7F6	F0 0B	BEQ LST3
3248	F7F8	20 27 F7	LST02 JSR PLNE
3249	F7FB	20 09 F7	JSR UPNO ; NEXT LINE
3250	F7FE	20 E9 F8	JSR ATBOT
3251	F801	90 ED	BCC LST01 ; NO
3252	F803	20 3F F7	LST3 JSR P03 ; ONE MORE CRLF FOR TAPE
3253	F806	20 0D FF	JSR PATC14 ; CLOSE TAPE IF NEEDED
3254	F809	4C 5C FA	JMP ENDERR
3256	F80C		;***** F COMMAND-FIND STRING *****
3257	F80C		; FIND STRING AND PRINT LINE TO TERMINAL
3258	F80C	20 1E F8	FCHAR JSR FCH
3259	F80F	AD 15 A4	FCHA1 LDA CURP02 ; SAVE BUFFER PNTR
3260	F812	4B	PHA
3261	F813	20 44 EB	JSR CLR ; CLEAR DISP PNTR
3262	F816	20 27 F7	JSR PLNE
3263	F819	68	PLA
3264	F81A	8D 15 A4	STA CURP02
3265	F81D	60	RTS
3266	F81E		; FIND A CHARACTER STRING
3267	F81E	A0 00	FCH LDY #0
3268	F820	20 BD E7	JSR PROMPT
3269	F823	20 5F E9	FC1 JSR RDRUB ; GET THE CHARACTER
3270	F826	C9 0D	CMF #*D ; REUSE OLD ARGUMENT??
3271	F828	D0 0A	BNE FC3
3272	F82A	C0 00	CPY #0 ; FIRST CHAR?
3273	F82C	D0 06	BNE FC3
3274	F82E	20 09 F7	FC2 JSR UPNO ; NEXT LINE DOWN
3275	F831	4C 49 F8	JMP FC5
3276	F834	C9 0B	FC3 CMF #CR ; DONE
3277	F836	F0 0B	BEQ FC4
3278	F838	99 EB 00	STA STRING,Y
3279	F83B	C8	INY
3280	F83C	D0 14	CPY #20 ; MAX LENGTH
3281	F83E	D0 E3	BNE FC1
3282	F840	4C 72 FA	JMP ERROR
3283	F843	20 24 EA	FC4 JSR CRCK ; CLEAR DISPLAY
3284	F846	8C 29 A4	STY STIY+2 ; COUNT OF CHARACTERS
3285	F849	A0 00	FC5 LDY #0
3286	F84B	8C 15 A4	STY CURP02 ; START AT BEGINNING OF LINENTR IS
3287	F84E	AC 15 A4	FC6 LDY CURP02 ; CLOBBER
3288	F851	A2 00	LDX #0

LINE #	LOC	CODE	LINE	LINE
3289	F853	B1 DF	FC7	LDA (NOWLN)Y ;GET THE CHARACTER
3290	F855	D0 03		BNE FC8 ;NOT AT END
3291	F857	4C 5C FA		JMP ENDERR
3292	F85A	C9 0D	FC8	CMP #CR ;END OF LINE
3293	F85C	F0 10		BEQ FC2
3294	F85E	D5 E8		CMP STRING.X
3295	F860	F0 06		BEQ FC9
3296	F862	EE 15 A4		INC CURPO2
3297	F865	4C 4E F8		JMP FC6
3298	F868	C8	FC9	INY
3299	F869	E8		INX
3300	F86A	EC 29 A4		CPX STIY+2 ;DONE?
3301	F86D	D0 E4		BNE FC7
3302	F86F	60		RTS
3304	F870			***** Q COMMAND-EXIT EDITOR *****
3305	F870			; EXIT THE TEXT EDITOR NEATLY
3306	F870	20 13 EA		STOP JSR CROW
3307	F873	4C A1 E1		JMP COMIN
3309	F876			***** C COMMAND-CHANGE STRING *****
3310	F876			;CHANGE STRING TO ANOTHER STRING IN A LINE
3311	F876	20 B2 F8	CHNG	JSR CFLG ;SET C COMMAND FLG
3312	F879	20 0C F8		JSR FCHAR ;FIND CORRECT LINE
3313	F87C	20 3C E9	CHN1	JSR READ ;IS (CR) IF OK
3314	F87F	C9 0D		CMP #CR
3315	F881	F0 09		BEQ CHN2
3316	F883	20 2E F8		JSR FC2 ;TRY NEXT ONE
3317	F886	20 0F F8		JSR FCHA1 ; SHOW LINE
3318	F889	4C 7C F8		JMP CHN1
3319	F88C	AD 29 A4	CHN2	LDA STIY+2 ;GET CHAR COUNT
3320	F88F	85 E9		STA OLDLEN ;GET READY FOR REPLAC
3321	F891	AD 15 A4		LDX CURPO2 ;PNTR TO BEGINNING OF STRING
3322	F894	48		PHA ;SAVE IT
3323	F895	20 2A F9		JSR ADDA ;ADD TO NOWLN (LINE PNTR)
3324	F898	20 44 EB		JSR CLR ;CLEAR DISP
3325	F89B	A0 05		LDY #M3-M1 ;PRINT 'D'
3326	F89D	20 AF E7		JSR KEP
3327	F8A0	A0 00		LDY #0
3328	F8A2	20 7A F7		JSR INO2 ;GET NEW STRING & REPLAC
3329	F8A5	68		FLA
3330	F8A6	AA		TAX
3331	F8A7	F0 06		BEQ CHN4
3332	F8A9	20 1D F9	CHN3	JSR SUB ;RESTORE NOWLN WHERE IT WAS
3333	F8AC	CA		DEX
3334	F8AD	D0 FA		BNE CHN3
3335	F8AF	4C 27 F7	CHN4	JMP PLNE ;DISPLAY THE CHANGED LINE
3337	F8B2			;THE FOLLOWING ARE SUBROUTINES USED BY COMMANDS
3338	F8B2	A9 01	CFLG	LDA #1 ;SET FLG FOR C COMMAND
3339	F8B4	D0 02		BNE KI2
3340	F8B6	A9 00	KIFLG	LDA #0 ;CLR K OR I COMMAND FLG

LINE #	LOC	CODE	LINE	LINE
3341	F8B8	8D 19 A4	KI2	STA COUNT
3342	F8BB	60		RTS
3344	F8BC	A5 E3	TOPND	LDA TEXT ;SET CURRENT LINE TO TOP
3345	F8BE	A6 E4		LDX TEXT+1
3346	F8C0	85 DF	TP01	STA NOWLN
3347	F8C2	86 E0		STX NOWLN+1
3348	F8C4	60		RTS
3350	F8C5	A5 E1	SETBOT	LDA BOTLN ;SET CURRENT LINE TO BOTTOM
3351	F8C7	A6 E2		LDX BOTLN+1
3352	F8C9	85 E7		STA SAVE
3353	F8CB	86 E8		STX SAVE+1
3354	F8CD	4C C0 F8		JMP TP01
3356	F8D0	AD 1C A4	RESNOW	LDA ADDR ;RESTORE CURRENT LINE ADDRESS
3357	F8D3	85 DF		STA NOWLN
3358	F8D5	AD 1D A4		LDA ADDR+1
3359	F8D8	85 E0		STA NOWLN+1
3360	F8DA	60		RTS
3362	F8DB			; SEE IF CURRENT LINE AT TOP (C SET IF SO)
3363	F8DB	A5 DF	ATTOP	LDA NOWLN
3364	F8DD	C5 E3		CMP TEXT
3365	F8DF	D0 16		BNE AT01
3366	F8E1	A5 E0		LDA NOWLN+1
3367	F8E3	C5 E4		CMP TEXT+1
3368	F8E5	D0 10		BNE AT01
3369	F8E7	38		SEC
3370	F8E8	60		RTS
3372	F8E9			; SEE IF CURRENT LINE AT BOTTOM (C SET IF SO)
3373	F8E9	A5 DF	ATBOT	LDA NOWLN
3374	F8EB	A6 E0		LDX NOWLN+1
3375	F8ED	C5 E1		CMP BOTLN
3376	F8EF	D0 06		BNE AT01
3377	F8F1	E4 E2		CPX BOTLN+1
3378	F8F3	D0 02		BNE AT01
3379	F8F5	38	AT02	SEC
3380	F8F6	60		RTS
3381	F8F7	18	AT01	CLC
3382	F8F8	60		RTS
3384	F8F9			;SEE IF WE RAN PAST END OF BUFFER LIMIT
3385	F8F9	A5 E1	ATEND	LDA BOTLN
3386	F8FB	A6 E2		LDX BOTLN+1
3387	F8FD	E4 E6		CPX END+1 ;HIGH BYTE > OR = ?
3388	F8FF	90 F6		BCC AT01
3389	F901	D0 F2		BNE AT02
3390	F903	C5 E5		CMP END ;LOW BYTE > OR = ?
3391	F905	90 F0		BCC AT01
3392	F907	80 EC		BCS AT02
3394	F909			; SAVE CURRENT LINE (NOWLN) IN S1
3395	F909	A5 DF	NOWS1	LDA NOWLN



LINE #	LOC	CODE	LINE
3396	F90B	A6 E0	LDX NOWLN+1
3397	F90D	4C 16 F9	JMP ADDS1A
3399	F910		; MOVE ADDR INTO S1
3400	F910	AD 1C A4	ADDRS1 LDA ADDR
3401	F913	AE 1D A4	LDX ADDR+1
3402	F916	8D 1A A4	ADDS1A STA S1
3403	F919	8E 1B A4	STX S1+1
3404	F91C	60	RTS
3406	F91D		; SUBTRACT ONE FROM CURRENT LINE (NOWLN)
3407	F91D	C6 DF	SUB DEC NOWLN
3408	F91F	A5 DF	LDA NOWLN
3409	F921	C9 FF	CMF ##FF
3410	F923	D0 02	BNE SUB1
3411	F925	C6 E0	DEC NOWLN+1
3412	F927	60	SUB1 RTS
3414	F928		; ADD ACC TO CURRENT LINE (NOWLN)
3415	F928	A9 01	AD1 LDA #1
3416	F92A	18	ADDA CLC
3417	F92B	65 DF	ADC NOWLN
3418	F92D	85 DF	STA NOWLN
3419	F92F	90 02	BCC ADDA1
3420	F931	E6 E0	INC NOWLN+1
3421	F933	60	ADDA1 RTS
3423	F934	A5 DF	SAVNOW LDA NOWLN ;SAVE CURRENT LINE INTO ADDR
3424	F936	8D 1C A4	STA ADDR
3425	F939	A5 E0	LDA NOWLN+1
3426	F93B	8D 1D A4	STA ADDR+1
3427	F93E	60	REF2 RTS

LINE #	LOC	CODE	LINE
3429	F93F		; MOVE CURRENT TEXT AROUND TO HAVE
3430	F93F		;SPACE TO PUT IN THE NEW BUFFER
3431	F93F	A4 EA	REPLAC LDY LENGTH
3432	F941	C4 E9	CPY OLDLEN ;COMPARE OLD AND NEW LENGTHS
3433	F943	D0 1A	BNE R2W ;BRANCH IF DIFF
3434	F945	F0 07	BEQ R87 ;LENGTHS ARE EQUAL. JUST REPLACE
3435	F947	A9 0D	R8 LDA #CR
3436	F949	91 DF	STA (NOWLN)Y
3437	F94B	20 4A FA	JSR GOGO
3439	F94E		;LENGTH = OLDLEN
3440	F94E	88	R87 DEY
3441	F94F	C0 FF	CPY ##FF
3442	F951	F0 E8	BEQ REP2
3443	F953	B9 38 A4	R88 LDA DIRUFF,Y
3444	F956	91 DF	STA (NOWLN)Y
3445	F958	20 4A FA	JSR GOGO
3446	F95B	88	DEY
3447	F95C	10 F5	BPL R88
3448	F95E	60	RTS
3449	F95F	B0 6E	R2W BCS R100 ;LENGTH > OLDLEN
3451	F961		;LENGTH < OLDLEN
3452	F961	20 34 F9	JSR SAVNOW ;PUT NOWLN INTO ADDR
3453	F964	20 10 F9	JSR ADDR1 ;PUT IT IN S1 ALSO
3454	F967	A5 E9	LDA OLDLEN
3455	F969	38	SEC
3456	F96A	E5 EA	SBC LENGTH ;GET DIFFERENCE IN LENGTHS
3457	F96C	A4 EA	LDY LENGTH
3458	F96E	D0 07	BNE RQP
3459	F970	AE 19 A4	LDX COUNT ;C-COMM ?
3460	F973	D0 02	BNE RQP ;YES, JUMP
3461	F975	69 00	ADC #0 ;INCLUDE (CR)
3462	F977	48	RQP PHA
3463	F978	18	CLC
3464	F979	6D 1A A4	ADC S1
3465	F97C	8D 1A A4	STA S1
3466	F97F	90 03	BCC R6
3467	F981	EE 1B A4	INC S1+1
3468	F984	A9 1A	R6 LDA #(S1
3469	F986	20 58 EB	JSR LDAY
3470	F989	91 DF	STA (NOWLN)Y ;... AND MOVE IT UP (DOWN IN ADDR)
3471	F98B	20 4A FA	JSR GOGO
3472	F98E	AA	TAX
3473	F98F	AD 1A A4	LDA S1
3474	F992	C5 E1	CMR BOTLN ;DONE ??
3475	F994	D0 07	BNE R5
3476	F996	AD 1B A4	LDA S1+1
3477	F999	C5 E2	CMR BOTLN+1
3478	F99B	F0 0E	BEQ R7
3479	F99D	20 2B F9	R5 JSR AD1
3480	F9A0	EE 1A A4	INC S1
3481	F9A3	D0 03	BNE R55
3482	F9A5	EE 1B A4	INC S1+1
3483	F9A8	4C B4 F9	R55 JMP R6

LINE #	LOC	CODE	LINE	LINE
3484	F9AE	20 D0 F8	R7	JSR RESNOW ;RESTORE NOWLN
3485	F9AE	68		PLA ;RESTORE DIFFERENCE
3486	F9AF	8D 2A A4		STA CFIY ;SAVE IT
3487	F9B2	A5 E1		LDA BOTLN
3488	F9B4	38		SEC
3489	F9B5	E1 2A A4		SBC CFIY ;AND SUBTRACT IT FROM BOTTOM
3490	F9B8	85 E1		STA BOTLN
3491	F9BA	B0 02		BCS R9
3492	F9BC	C6 E2		DEC BOTLN+1
3493	F9BE	AD 19 A4	R9	LDA COUNT ;C COMM OR K , I COMM ?
3494	F9C1	D0 04		BNE R10
3495	F9C3	A4 EA		LDY LENGTH
3496	F9C5	D0 05		BNE R11
3497	F9C7	A4 EA	R10	LDY LENGTH
3498	F9C9	D0 83		BNE R87
3499	F9CB	60		RTS
3500	F9CC	4C 47 F9	R11	JMP R8
3502	F9CF			;LENGTH > OLDLEN
3503	F9CF	A5 EA	R100	LDA LENGTH ;NEW LINE IS LONGER
3504	F9D1	38		SEC
3505	F9D2	E5 E9		SBC OLDLEN
3506	F9D4	A4 E9		LDY OLDLEN
3507	F9D6	D0 02		BNE R101 ;ALREADY HAVE ROOM FOR CR
3508	F9D8	69 00		ADC #0 ;ADD ONE TO DIFFERENCE
3509	F9DA	48	R101	PHA
3510	F9DB	20 34 F9		JSR SAVNOW ;NOWLN INTO S1
3511	F9DE	20 C5 F8		JSR SETBOT
3512	F9E1	A0 00		LDY #0
3513	F9E3	B1 DF	R102	LDA (NOWLN)Y
3514	F9E5	C9 00		CMP #0
3515	F9E7	F0 06		BEQ R108
3516	F9E9	20 28 F9		JSR AD1
3517	F9EC	4C E3 F9		JMP R102
3518	F9EF	68	R108	PLA
3519	F9F0	48		PHA
3520	F9F1	18		CLC
3521	F9F2	65 E1		ADC BOTLN ;ADD DIFFERENCE TO END
3522	F9F4	85 E1		STA BOTLN ;STORE NEW END
3523	F9F6	90 02		BCC R103
3524	F9F8	E6 E7		INC BOTLN+1
3525	F9FA	20 F9 FB	R103	JSR ATEND
3526	F9FD	90 08		BCC R107
3527	F9FF	A5 E7		LDA SAVE ;RESTORE OLD BOTTOM
3528	FA01	85 E1		STA BOTLN
3529	FA03	A5 EB		LDA SAVE+1
3530	FA05	85 E2		STA BOTLN+1
3531	FA07	4C 5C FA		JMP ENDERR ;RAN PAST BUFFER END
3532	FA0A	20 09 F9	R107	JSR NOWS1 ;SAVE CURRENT END
3533	FA0D	68		PLA
3534	FA0E	18		CLC
3535	FA0F	65 DF		ADC NOWLN
3536	FA11	85 DF		STA NOWLN
3537	FA13	90 02		BCC R104
3538	FA15	E6 E0		INC NOWLN+1

LINE #	LOC	CODE	LINE	LINE
339	FA17	A9 1A	R104	LDA #CS1
340	FA19	20 58 EB		JSR LIAY
341	FA1C	91 DF		STA (NOWLN)Y
342	FA1E	20 4A FA		JSR GOGO
343	FA21	AD 1A A4		LDA S1
344	FA24	CD 1C A4		CMP ADDR
345	FA27	D0 08		BNE R105
346	FA29	AD 1B A4		LDA S1+1
347	FA2C	CD 1D A4		CMP ADDR+1
348	FA2F	F0 13		BEQ R106 ;BACK WHERE WE STARTED ??
349	FA31	20 1D F9	R105	JSR SUB ;BRANCH IF DONE
350	FA34	CE 1A A4		DEC S1
351	FA37	AD 1A A4		LDA S1
352	FA3A	C9 FF		CMP #FF
353	FA3C	D0 03		BNE R1051
354	FA3E	CE 1B A4		DEC S1+1
355	FA41	4C 17 FA	R1051	JMP R104
356	FA44	20 D0 F8	R106	JSR RESNOW
357	FA47	4C BE F9		JMP R9
359	FA4A			;SEE IF IT WROTE INTO MEMORY
360	FA4A	D1 DF	GOGO	CMP (NOWLN)Y
361	FA4C	F0 0D		BEQ GOGO1
362	FA4E			;MOVE ADDRESS
363	FA4E	A5 DF		LDA NOWLN
364	FA50	8D 1C A4		STA ADDR
365	FA53	A5 E0		LDA NOWLN+1
366	FA55	8D 1D A4		STA ADDR+1
367	FA58	4C 33 EB		JMP MEMERR
368	FA5B	60	GOGO1	RTS ;OK

## ERROR HANDLERS

PA00-J001A.....PAGE 0082

LINE #	LOC	CODE	LINE
3570	FASC	20 44 EB	ENDERR JSR CLR ;CLEAR PNTR
3571	FA5F	A0 72	LDY #MSG2-M1 ;PRINT "END"
3572	FA61	20 AF E7	JSR KEF
3573	FA64	20 B8 F6	JSR DNNO ;BACK UP TO LAST LINE
3574	FA67	20 42 E8	JSR TTY1ST ;IF TTY (CR)
3575	FA6A	D0 03	BNE ENDE2
3576	FA6C	20 13 EA	JSR CRL0W
3577	FA6F	4C 78 FA	ENDE2 JMP ERRO
3578	FA72	20 FE E8	ERROR JSR LL
3579	FA75	20 D4 E7	JSR QM
3580	FA78	20 44 EB	ERRO JSR CLR
3581	FA7B	A2 FF	LDX ##FF
3582	FA7D		COM=ERRO
3583	FA7D	9A	TXS
3584	FA7E	20 FE E8	JSR LL ;I/O TO TERMINAL (KB,D/P OR TTY)
3585	FA81	B8	CLD
3586	FA82	20 88 FA	JSR COMM
3587	FA85	4C 78 FA	JMP COM
3589	FA88		;GET EDITOR COMMANDS & DECODE
3590	FA88	A2 00	COMM LDX #0
3591	FA8A	20 BC FE	JSR PATCH8 ;READ A CHAR WITH "=C >"
3592	FA8D	A2 0B	ENTRY LDX #COMCN1
3593	FA8F	D0 AC FA	CD02 CMP COMTRL,X ;COMPARE WITH ALLOWABLE COMMANDS
3594	FA92	F0 0C	BEQ CFND1 ;MATCH ,SO PROCESS COMMAND
3595	FA94	CA	DEX
3596	FA95	10 FB	BPL CD02
3597	FA97	20 D4 E7	JSR QM ;NOT IN LIST ,SO NOT LEGAL COMMAND
3598	FA9A	20 24 EA	JSR CRCK
3599	FA9D	4C 78 FA	JMP ERRO
3600	FAA0	20 17 FF	CFND1 JSR PATC15 ;(CR) & START DECODING COMMAND
3601	FAA3	8D B9 FA	LDA JTBL+1,X
3602	FAA6	8D 1B A4	STA S1+1
3603	FAA9	6C 1A A4	JMP (S1)
3605	FAAC		COMCN1=11
3606	FAAC		;COMMAND TABLE
3607	FAAC	4B 20	COMTRL .BYT 'K RIUULTBFQC'
3608	FA8B	4C F7	JTBL .WOR DLNE,PLNE,INPU,IN,DOWN,UP
3608	FA8A	27 F7	
3608	FA8C	CB F7	
3608	FA8E	64 F7	
3608	FAC0	24 F7	
3608	FAC2	F9 F6	
3609	FAC4	E1 F7	.WOR LST,TP,BT,FCHAR,STOP,CHNG
3609	FAC6	D2 F6	
3609	FACB	21 F7	
3609	FACA	0C F8	
3609	FACC	70 F8	
3609	FACE	76 F8	
3611	FAD0		;READ FROM MEMORY FOR ASSEMBLER
3612	FAD0	98	MREAD TYA
3613	FAD1	4B	PHA

## ERROR HANDLERS

PA00-J001A.....PAGE 0083

LINE #	LOC	CODE	LINE
3614	FAD2	A0 00	LDY #0
3615	FAD4	B1 DF	LDA (NDWLN)Y
3616	FAD6	8D 2A A4	STA CPIY
3617	FAD9	20 28 F9	JSR AD1
3618	FADC	68	PLA
3619	FADD	AB	TAY
3620	FADE	AD 2A A4	LDA CPIY
3621	FAE1	60	RTS

LINE #	LOC	CODE	LINE
3623	FAE2		; THIS PROGRAM CONVERTS MNEMONIC INSTRUCTIONS INTO MACHIN
3624	FAE2		; CODE AND STORES IT IN THE DESIGNATED MEMORY AREA
3626	FAE2		; FROM TABLE LOCATIONS:
3627	FAE2	00	TYPTR1 .BYT 00,02,00,0B,\$F2,\$FF,\$B0,01
3627	FAE3	02	
3627	FAE4	00	
3627	FAE5	0B	
3627	FAE6	F2	
3627	FAE7	FF	
3627	FAE8	80	
3627	FAE9	01	
3628	FAEA	C0	.BYT \$C0,\$E2,\$C0,\$C0,\$FF,00,00
3628	FAEB	E2	
3628	FAEC	C0	
3628	FAED	C0	
3628	FAEE	FF	
3628	FAEF	00	
3628	FAF0	00	
3629	FAF1	0B	TYPTR2 .BYT 0B,00,\$10,\$B0,\$40,\$C0,00,\$C0
3629	FAF2	00	
3629	FAF3	10	
3629	FAF4	80	
3629	FAF5	40	
3629	FAF6	C0	
3629	FAF7	00	
3629	FAF8	C0	
3630	FAF9	00	.BYT \$00,\$40,00,00,\$E4,\$20,\$B0
3630	FAFA	40	
3630	FAFB	00	
3630	FAFC	00	
3630	FAFD	E4	
3630	FAFE	20	
3630	FAFF	80	
3631	FB00	00	CORR .BYT 00,\$FC,00,0B,0B,\$F8,\$FC,\$F4
3631	FB01	FC	
3631	FB02	00	
3631	FB03	0B	
3631	FB04	0B	
3631	FB05	FB	
3631	FB06	FC	
3631	FB07	F4	
3632	FB08	0C	.BYT \$0C,\$10,04,\$F4,00,\$20,\$10
3632	FB09	10	
3632	FB0A	04	
3632	FB0B	F4	
3632	FB0C	00	
3632	FB0D	20	
3632	FB0E	10	
3633	FB0F	00	! SIZEH .BYT 00,00,\$0F,01,01,01,\$11,\$11
3633	FB10	00	
3633	FB11	0F	
3633	FB12	01	
3633	FB13	01	
3633	FB14	01	

LINE #	LOC	CODE	LINE
433	FB15	11	
433	FB16	11	
434	FB17	02	.BYT 02,02,\$11,\$11,02,\$12,00
434	FB18	02	
434	FB19	11	
434	FB1A	11	
434	FB1B	02	
434	FB1C	12	
434	FB1D	00	
436	FB1E	00	STCODE .BYT \$00,\$0B,\$10,\$1B,\$20,\$2B,\$30,\$3B
436	FB1F	0B	
436	FB20	10	
436	FB21	1B	
436	FB22	20	
436	FB23	2B	
436	FB24	30	
436	FB25	3B	
437	FB26	40	.BYT \$40,\$4B,\$50,\$5B,\$60,\$6B,\$70,\$7B
437	FB27	4B	
437	FB28	50	
437	FB29	5B	
437	FB2A	60	
437	FB2B	6B	
437	FB2C	70	
437	FB2D	7B	
438	FB2E	80	.BYT \$80,\$8B,\$90,\$9B,\$AC,\$AB,\$B0,\$BB
438	FB2F	8B	
438	FB30	90	
438	FB31	9B	
438	FB32	AC	
438	FB33	AB	
438	FB34	BO	
438	FB35	BB	
439	FB36	CC	.BYT \$CC,\$CB,\$D0,\$DB,\$EC,\$EB,\$F0,\$FB
439	FB37	CB	
439	FB38	DO	
439	FB39	DB	
439	FB3A	EC	
439	FB3B	EB	
439	FB3C	FO	
439	FB3D	FB	
440	FB3E	0C	.BYT \$0C,\$2C,\$4C,\$4C,\$8C,\$AC,\$CC,\$EC
440	FB3F	2C	
440	FB40	4C	
440	FB41	4C	
440	FB42	8C	
440	FB43	AC	
440	FB44	CC	
440	FB45	EC	
441	FB46	8A	.BYT \$8A,\$9A,\$AA,\$BA,\$CA,\$DA,\$EA,\$FA
441	FB47	9A	
441	FB48	AA	
441	FB49	BA	
441	FB4A	CA	

LINE #	LOC	CODE	LINE
3641	FB4B	DA	
3641	FB4C	EA	
3641	FB4D	FA	
3642	FB4E	OE	.BYT \$0E,\$2E,\$4E,\$6E,\$8E,\$AE,\$CE,\$EE
3642	FB4F	2E	
3642	FB50	4E	
3642	FB51	6E	
3642	FB52	8E	
3642	FB53	AE	
3642	FB54	CE	
3642	FB55	EE	
3643	FB56	0D	.BYT \$0D,\$2D,\$4D,\$6D,\$8D,\$AD,\$CD,\$ED
3643	FB57	2D	
3643	FB58	4D	
3643	FB59	6D	
3643	FB5A	8D	
3643	FB5B	AD	
3643	FB5C	CD	
3643	FB5D	ED	
3644	FB5E	0D	TYPTB .BYT 13,13,12,13,14,13,12,13
3644	FB5F	0D	
3644	FB60	0C	
3644	FB61	0D	
3644	FB62	0E	
3644	FB63	0D	
3644	FB64	0C	
3644	FB65	0D	
3645	FB66	0D	.BYT 13,13,12,13,13,13,12,13
3645	FB67	0D	
3645	FB68	0C	
3645	FB69	0D	
3645	FB6A	0D	
3645	FB6B	0D	
3645	FB6C	0C	
3645	FB6D	0D	
3646	FB6E	0F	.BYT 15,13,12,13,9,13,12,13
3646	FB6F	0D	
3646	FB70	0C	
3646	FB71	0D	
3646	FB72	09	
3646	FB73	0D	
3646	FB74	0C	
3646	FB75	0D	
3647	FB76	0B	.BYT 8,13,12,13,8,13,12,13
3647	FB77	0D	
3647	FB78	0C	
3647	FB79	0D	
3647	FB7A	0B	
3647	FB7B	0D	
3647	FB7C	0C	
3647	FB7D	0D	
3648	FB7E	0F	.BYT 15,6,11,11,4,10,8,8
3648	FB7F	06	
3648	FB80	0B	
3648	FB81	0B	

LINE #	LOC	CODE	LINE
3648	FB82	04	
3648	FB83	0A	
3648	FB84	0B	
3648	FB85	0B	
3649	FB86	0D	.BYT 13,13,13,13,13,15,13,15
3649	FB87	0D	
3649	FB88	0D	
3649	FB89	0D	
3649	FB8A	0D	
3649	FB8B	0F	
3649	FB8C	0D	
3649	FB8D	0F	
3650	FB8E	07	.BYT 7,7,7,7,5,9,3,3
3650	FB8F	07	
3650	FB90	07	
3650	FB91	07	
3650	FB92	05	
3650	FB93	09	
3650	FB94	03	
3650	FB95	03	
3651	FB96	01	.BYT 1,1,1,1,2,1,1,1
3651	FB97	01	
3651	FB98	01	
3651	FB99	01	
3651	FB9A	02	
3651	FB9B	01	
3651	FB9C	01	
3651	FB9D	01	
3653	FB9E		;PROGRAM STARTS HERE:
3654	FB9E	AD 25 A4	MNEENT LDA SAVFC ;TRANSF PC TO ADDR
3655	FBA1	8D 1C A4	STA ADDR
3656	FBA4	AD 26 A4	LDA SAVFC+1
3657	FBA7	8D 1D A4	STA ADDR+1
3658	FBA8	20 24 EA	STARTM JSR CRCK ;(CR) IF PRI PTR DIFF FROM 0
3659	FBA9	A9 00	LDA #0
3660	FBAF	8D 37 A4	STA CODFLG
3661	FB82	20 3E EB	JSR BLANK
3662	FB85	20 DB E2	JSR WRITAZ ;WRITE ADDRESS
3663	FB88	20 3B EB	JSR BLANK2
3664	FB8B	20 3B EB	JSR BLANK2
3665	FB8E	4C 06 FE	JMP MNEM ;JUMP TO INPUT MNEMONIC OPCODE
3666	FBC1	A9 00	LDA #00 ;SET UP TO FORM MODE MATCH
3667	FBC3	8D 26 01	STA TMASK1
3668	FBC6	8D 27 01	STA TMASK2
3669	FBC9	20 3E EB	JSR BLANK
3670	FBCA	AC 2E 01	LDY TYPE
3671	FBCF	38	SEC
3672	FBD0	6E 26 01	PNTLUP ROR TMASK1 ;SHIFT POINTER TO INSTRUCTION TYPE
3673	FBD3	6E 27 01	ROR TMASK2
3674	FBD6	8B	DEY
3675	FBD7	10 F7	BNE PNTLUP
3677	FBD9		;TEST FOR ONE BYTE INSTRUCTION
3678	FBD9	AC 2E 01	LDY TYPE

LINE #	LOC	CODE	LINE
3679	FBIC	C0 0D	CPY ##0D
3680	FBIE	D0 05	BNE RDADDR
3681	FBE0	A2 00	LIX #00
3682	FBE2	4C CB FC	JMP OFCOMP
3684	FBES		;INPUT ADDRESS FIELD
3685	FBES	A0 06	RDADDR LDY #06 ;CLEAR ADDRESS FIELD (NON HEX)
3686	FBE7	A9 51	LDA #'Q'
3687	FBE9	99 32 01	CLRLUP STA ADFLD-1, Y
3688	FBEC	88	DEY
3689	FBED	D0 FA	BNE CLRLUP ;(LEAVES Y = 0 FOR NEXT PHASE)
3690	FBF7	20 5F E9	JSR RDRUB ;WITH RUBOUT
3691	FBF2	C9 20	CMF ##20 ;IGNORE SPACE CHARACTERS
3692	FBF4	F0 EF	BEQ RDADDR
3693	FBF6	99 33 01	STORCH STA ADFLD, Y ;STORE ADDRESS CHARACTER
3694	FBF9	C8	INY
3695	FBFA	C0 07	CPY #07
3696	FBFC	R0 5C	BCC TRY56
3697	FBFE	20 5F E9	JSR RDRUB ;READ REMAINDER OF ADDRESS CHARS
3698	FC01	C9 20	CMF ##20 ;THRU WHEN (SPACE) OR (CR)
3699	FC03	D0 05	BNE STOR1
3700	FC05	EE 37 A4	INC CDBFLG ;SET CODE FLG
3701	FC08	D0 04	BNE EVAL
3702	FC0A	C9 0D	STOR1 CMP ##0D ;CHECK FOR (CR)
3703	FC0C	D0 EB	BNE STORCH
3705	FC0E		;SEPARATE ADDRESSING MODE FROM ADDRESS FIELD
3706	FC0E	8C 31 A4	EVAL STY TEMPX ;TEMPX NOW HAS NUMBER OF CHAR
3707	FC11	AD 33 01	LDA ADFLD ;CHECK FIRST CHAR FOR # OR (
3708	FC14	C9 23	CMF #'#'
3709	FC16	F0 25	BEQ HATCJ
3710	FC18	C9 28	CMF #'('
3711	FC1A	F0 5A	BEQ PAREN
3712	FC1C	AD 31 A4	LDA TEMPX ;CHECK FOR ACCUMULATOR MODE
3713	FC1F	C9 01	CMF #01
3714	FC21	D0 05	BNE TRYZP
3715	FC23	A2 01	ACCUM LDX #01
3716	FC25	4C CB FC	JMP OFCOMP
3717	FC28	C9 02	TRYZP CMF #02 ;CHECK FOR ZERO PAGE MODE
3718	FC2A	D0 14	BNE TRY34
3719	FC2C	AD 2E 01	LDA TYPE ;CHK FOR BRNCH WITH RELATIVE ADDR
3720	FC2F	C9 0C	CMF ##0C
3721	FC31	D0 05	BNE ZPAGE
3722	FC33	A2 02	LIX #02
3723	FC35	4C CB FC	JMP OFCOMP
3724	FC38	A2 05	ZPAGE LDX #05
3725	FC3A	4C CB FC	JMP OFCOMP
3726	FC3B	4C B6 FC	HATCJ JMP HATCH
3727	FC40	A9 04	TRY34 LDA #04 ;CHECK FOR ABSOLUTE OR ZP, X OR ZP,
3728	FC42	D0 31 A4	CMF TEMPX
3729	FC45	90 15	BCC ABSIND
3730	FC47	A2 02	LIX #02
3731	FC49	20 F1 FD	JSR XORYZ ;CC = X, CS = Y, NE = ABSOLUTE
3732	FC4C	D0 58	BNE ABSOL
3733	FC4E	90 05	BCC ZPX

LINE #	LOC	CODE	LINE
3734	FC50	A2 03	ZPY LIX #03 ;CARRY SET SO ZP, Y MODE
3735	FC52	4C CB FC	JMP OFCOMP
3736	FC55	A2 04	ZPX LIX #04 ;CARRY CLEAR SO ZP, X MODE
3737	FC57	4C CB FC	JMP OFCOMP
3738	FC5A	B0 69	TRY56 BCC ERRORM
3739	FC5C	20 EF FD	ABSIND JSR XORY ;CC=ABS, X CS=ABS, Y NE=ERROR
3740	FC5F	D0 64	BNE ERRORM
3741	FC61	90 0F	BCC ABSX
3742	FC63	A9 09	ABSY LDA #09
3743	FC65	CD 2E 01	CMF TYPE
3744	FC68	D0 04	BNE ABSY1
3745	FC6A	A2 0E	LIX ##0E
3746	FC6C	D0 5D	BNE OFCOMP
3747	FC6E	A2 08	ABSY1 LDX ##08
3748	FC70	D0 59	BNE OFCOMP
3749	FC72	A2 09	ABSX LIX #09 ;CARRY CLEAR SO ABS, X MODE
3750	FC74	D0 55	BNE OFCOMP
3751	FC76	AD 36 01	PAREN LDA ADFLD+3 ;SEE IF (HH, X), (HH)Y OR (HHHH)
3752	FC79	C9 2C	CMF #'/' ;(HHX) (HH), Y ARE OK TOO
3753	FC7B	F0 04	BEQ INDX ;COMMA IN 4TH POSITION = (HH, X)
3754	FC7D	C9 58	CMF #'X'
3755	FC7F	D0 04	BNE TRYINY ;X IN 4TH POSITION = (HHX)
3756	FC81	A2 0B	INDX LIX ##0B
3757	FC83	D0 46	BNE OFCOMP
3758	FC85	C9 29	TRYINY CMF #'/' ;'' IN 4TH POS = (HH)Y OR (HH), Y
3759	FC87	D0 0B	BNE TRYJMP
3760	FC89	20 EF FD	JSR XORY ;CHCK TO SEE IF Y INDEX REG DESIRE
3761	FC8C	D0 37	BNE ERRORM
3762	FC8E	90 35	BCC ERRORM
3763	FC90	A2 0A	LIX ##0A
3764	FC92	D0 37	BNE OFCOMP
3765	FC94	AD 38 01	TRYJMP LDA ADFLD+5 ;CHECK FOR FINAL PAREN
3766	FC97	C9 29	CMF #'/'
3767	FC99	D0 2A	BNE ERRORM
3768	FC9B	AD 2E 01	LDA TYPE ;CONFIRM CORRECT ADDRESS TYPE
3769	FC9E	C9 0B	CMF ##0B
3770	FC9A	D0 23	BNE ERRORM
3771	FCA2	A2 0D	LIX ##0D ;OK, FORM IS JMP (HHHH)
3772	FCA4	D0 25	BNE OFCOMP
3773	FCA6	AD 2E 01	ABSOL LDA TYPE ;CHECK FOR BRANCH TO ABSOLUTE LOC
3774	FCA9	C9 0C	CMF ##0C
3775	FCAB	D0 05	BNE ABSOL1
3776	FCAD	A2 02	LIX #02
3777	FCAF	4C CB FC	JMP OFCOMP
3778	FCB2	A2 0C	ABSOL1 LDX ##0C
3779	FCB4	D0 15	BNE OFCOMP
3780	FCB6		;SELECT IMMEDIATE ADDRESSING TYPE
3781	FCB6	AD 2E 01	HATCH LDA TYPE
3782	FCB9	C9 01	CMF #01
3783	FCBB	F0 04	BEQ IMMEDI
3784	FCBD	A2 07	LIX #07
3785	FCBF	D0 0A	BNE OFCOMP
3786	FCC1	A2 06	IMMEDI LIX #06
3787	FCC3	D0 06	BNE OFCOMP
3788	FCC5	20 94 E3	ERRORM JSR CKEROO ;OUTPUT ERROR MESSAGE

```

LINE # LOC      CODE      LINE
3789 FCCB 4C AA FB          JMP STARTM

3791 FCCB          ; COMPUTE FINAL OP CODE FOR DEFINED ADDRESSING MODE
3792 FCCB B0 E2 FA      OPCOMP LDA TYPTR1,X ;MATCH TYPE MASK WITH VALID MODE
3793 FCCB F0 05          BEQ OPCMP1 ; PATTERNS & SKIP 1ST WORD TEST IF
3794 FCD0 2D 26 01      AND TMASK1 ;ALREADY ZERO
3795 FCD3 D0 08          BNE VALID
3796 FCD5 B0 F1 FA      OPCMP1 LDA TYPTR2,X ; TEST 2ND PART
3797 FCD8 2D 27 01      AND TMASK2
3798 FCDB F0 EB          BEQ ERRORM ; INST DOES NOT HAVE SPECIFIED MOD
3799 FCDD 18          VALID CLC ;FORM FINAL OP CODE
3800 FCDE B0 00 FB      LDA CORR,X
3801 FCE1 6D 34 A4      ADC OPCODE
3802 FCE4 8D 34 A4      STA OPCODE

3804 FCE7          ; PROCESS ADDRESSES TO FINAL FORMAT
3805 FCE7 B0 0F FB      LDA SIZEM,X ; OBTAIN ADDRESS FORMAT FROM TABLE
3806 FCEA C9 00          CMP #00
3807 FCEC F0 50          BEQ ONEBYT
3808 FCEE C9 0F          CMP #0F ; NEED BRANCH COMPUTATION?
3809 FCF0 F0 1D          BEQ BRNCHC
3810 FCF2 8D 33 A4      STA TEMPA ; SAVE START POINT & CHAR COUNT
3811 FCF5 29 0F          AND #0F
3812 FCF7 A8          TAY ; SEPARATE CHARACTER COUNT
3813 FCF8 8D 2F A4      STA BYTESM ; LOAD ADDR BYTES INTO Y (0,1,OR 2)
3814 FCFB EE 2F A4      INC BYTESM ; SAVE IN BYTES
3815 FCFE AD 33 A4      LDA TEMPA ; TO INSTR LENGTH (1,2,OR 3 BYTES)
3816 FD01 29 F0          AND #0F ; SEPARATE STARTING POINT
3817 FD03 4A          LSR A
3818 FD04 4A          LSR A
3819 FD05 4A          LSR A
3820 FD06 4A          LSR A
3821 FD07 AA          TAX ; AND PUT IT IN X
3822 FD08 20 12 FD      JSR CONVRT ; CONVERT ASCII ADDRESS TO HEX
3823 FD0B B0 B8          BCS ERRORM ; SKIP OUT IF ERROR IN INPUT
3824 FD0D 90 1D          BCC STASH
3825 FD0F 4C 86 FD      BRNCHC JMP BRCOMP

3827 FD12          ; ***** SUBROUTINE *****
3828 FD12          ; CONVERT FORMATTED ADDRESS INTO PROPER HEX ADDRESS
3829 FD12 B0 33 01      CONVRT LDA ADFLD,X ; PICK UP 1ST ADDR CHARACTER
3830 FD15 20 7D EA      JSR HEX ; CONVERT TO MOST SIG HEX
3831 FD18 B0 11          BCS ERRFLG
3832 FD1A EB          INX ; GET NEXT ASCII CHARACTER
3833 FD1B B0 33 01      LDA ADFLD,X
3834 FD1E EB          INX ; POINT TO NEXT CHARACTER, IF ANY
3835 FD1F 20 84 EA      JSR PACK
3836 FD22 B0 07          BCS ERRFLG
3837 FD24 99 34 A4      STA OPCODE,Y ; SAVE IN MOST SIG. BYTE LOCATION
3838 FD27 B8          DEY ; SET UP FOR NEXT ADDR BYTE, IF ANY
3839 FD28 B0 EB          BNE CONVRT ; IF NECESSARY, FORM NEXT ADDR BYTE
3840 FD2A 18          CLC
3841 FD2B 60          ERRFLG RTS ; NON HEX CLEARED CARRY
3842 FD2C          ; *****

```

```

LINE # LOC      CODE      LINE
3844 FD2C AC 2F A4      STASH LDY BYTESM ; SET UP TO STORE COMMAND
3845 FD2F B8          DEY
3846 FD30 B9 34 A4      STSHLP LDA OPCODE,Y
3847 FD33 20 78 EB      JSR SADDR ; STORE ONE BYTE OF COMMAND
3848 FD36 C0 00          CFY #00
3849 FD38 F0 0B          BEQ FORMD5
3850 FD3A B8          DEY
3851 FD3B B8          CLV
3852 FD3C 50 F2          BVC STSHLP ; REPEAT TILL THRU

3854 FD3E A9 01          ONEBYT LDA #01 ; SET BYTES = 1
3855 FD40 8D 2F A4      STA BYTESM
3856 FD43 D0 E7          BNE STASH

3858 FD45          ; FORMAT FOR SYSTEM 65 DISPLAY (REFORMAT FOR AIM)
3859 FD45          FORMD5 JSR CLR
3860 FD48 20 D0 E5      JSR CGPC1 ; ADDR TO SAVPC FOR DISASSEMBLY
3861 FD4B 20 42 E8      JSR TTYTST ; IF TTY DO NOT GO TO DISASS
3862 FD4E D0 08          BNE FORMD1
3863 FD50 20 3B E8      JSR BLANK2 ; IT IS TTY
3864 FD53 20 3B E8      JSR BLANK2
3865 FD56 D0 11          BNE FORMD2 ; OUTPUT OPCODE
3866 FD58 20 6C F4      FORMD1 JSR DISASM
3867 FD5B 20 24 EA      JSR CRCK ; <CR> IF PRI PTR DIFF FROM 0
3868 FD5E AD 37 A4      LDA CODFLG ; SEE IF HE WANTS CODE ALSO
3869 FD61 F0 1A          BEQ FORM1
3870 FD63 20 3E E8      JSR BLANK
3871 FD66 20 3C F5      JSR PRFC ; PROG CNTR
3872 FD69          ; OUTPUT OPCODE
3873 FD69 AE 2F A4      FORMD2 LDY BYTESM
3874 FD6C A0 00          LDY #00
3875 FD6E A9 1C          DISPLY LDA #(ADDR ; DO LDA (ADDR),Y ,WHITOUT PAG 0
3876 FD70 20 58 EB      JSR LIAY
3877 FD73 20 46 EA      JSR NUMA
3878 FD76 20 3E E8      JSR BLANK
3879 FD79 C8          INY
3880 FD7A CA          DEX
3881 FD7B D0 F1          BNE DISPLY

3883 FD7D          ; POINT TO NEXT INSTRUCTION LOCATION
3884 FD7D AC 2F A4      FORM1 LDY BYTESM ; ADD BYTESM TO ADDR
3885 FD80 20 CD E2      JSR NXTADD
3886 FD83 4C 24 FF      JMP PATC16 ; UPDATE PC

3888 FD86          ; RELATIVE BRANCH ADDRESS COMPUTATION
3889 FD86 AD 31 A4      BRCOMP LDA TEMPX
3890 FD89 C9 02          CMP #02 ; IF REL BRANCH INPUT, USE IT
3891 FD8B D0 11          BNE COMPFER
3892 FD8D A2 00          LDX #00
3893 FD8F A0 01          LDY #01
3894 FD91 20 12 FD      JSR CONVRT
3895 FD94 B0 40          BCS ERRJMP
3896 FD96 A9 02          LDA #02
3897 FD98 B0 2F A4      STA BYTESM ; SET PROPER BYTES
3898 FD9B 4C 2C FD      JMP STASH

```

LINE #	LOC	CODE	LINE
3899	F09E	A2 00	COMPBR LDX #00
3900	FDA0	A0 02	LDY #02
3901	FDA2	20 12 FD	JSR CONVRT
3902	FDA5	B0 2F	BCS ERRJMP
3903	FDA7	AD 1D A4	LDA ADDR+1 ;ADD BRANCH OFFSET
3904	FDA8	BD 27 01	STA MOVAD+1
3905	FDA1	AD 1C A4	LDA ADDR
3906	FDB0	18	CLC
3907	FDB1	69 02	ADC #02
3908	FDB3	8D 26 01	STA MOVAD
3909	FDB6	90 03	BCC CMPBR1
3910	FDB8	EE 27 01	INC MOVAD+1
3911	FDBB	38	CMPBR1 SEC ;COMPUTE BRANCH RELATIVE ADDRESS
3912	FDBC	AD 35 A4	LDA OFCODE+1
3913	FDBF	ED 26 01	SRC MOVAD
3914	FDC2	8D 35 A4	STA OFCODE+1
3915	FDC5	AD 36 A4	LDA OFCODE+2
3916	FDC8	ED 27 01	SRC MOVAD+1
3917	FDCB	8D 36 A4	STA OFCODE+2
3918	FDC E	C9 00	CMP #00
3919	FDD0	F0 0E	BEQ FORWRD
3920	FDD2	C9 FF	CMP #FF
3921	FDD4	F0 03	BEQ BACKWD
3922	FDD6	4C C5 FC	ERRJMP JMP ERRORM
3923	FDD9	AD 35 A4	BACKWD LDA OFCODE+1 ;CHECK IN RANGE
3924	FDDC	30 09	BMI OK
3925	FDD E	10 F6	BPL ERRJMP
3926	FDE0	AD 35 A4	FORWRD LDA OFCODE+1
3927	FDE3	10 02	BPL OK
3928	FDE5	30 EF	BMI ERRJMP
3929	FDE7	A9 02	OK LDA #02 ;SET UP FOR STASH
3930	FDE9	8D 2F A4	STA BYTESM
3931	FDEC	4C 2C FD	JMP STASH
3933	FDEF		;***** SUBROUTINE *****
3934	FDEF		;SUBROUTINE FOR DETERMINING X OR Y OR NEITHER
3935	FDEF	A2 04	XORY LDX #04
3936	FDF1	BD 33 01	XORYZ LDA ADFLD,X
3937	FDF4	C9 2C	CMP #','
3938	FDF6	D0 04	BNE XORY1
3939	FDF8	E8	INX
3940	FDF9	BD 33 01	LDA ADFLD,X
3941	FDFC	C9 58	XORY1 CMP #'X'
3942	FDFE	F0 03	BEQ ISX
3943	FE00	C9 59	CMP #'Y'
3944	FE02		XORYRT
3945	FE02	60	RTS; NOT ZERO IS NOT X OR NOT Y
3946	FE03	18	CLC ;CARRY SET IS Y
3947	FE04	90 FC	BCC XORYRT ; CARRY CLEAR IS X
3948	FE06		;***** END OF SUB *****
3950	FE06		; INPUT FOR MNEMONIC CODE
3951	FE06	A0 00	MNEM LDY #00
3952	FE08	8C 34 A4	STY OFCODE
3953	FE0B	8C 35 A4	STY OFCODE+1

LINE #	LOC	CODE	LINE
3954	FE0E	8C 36 A4	STY OFCODE+2 ;CLEARS OP CODE FOR NEW INPUT
3955	FE11	8C 26 01	STY MOVAD ;CLEARS UNUSED BIT IN FINAL FORMAT
3956	FE14	20 5F E9	JSR RDLUP
3957	FE17	C9 2A	CMP #'*'
3958	FE19	F0 58	BEQ STLOAD ;GO TO SET CURRENT ADDRESS POINTER
3959	FE1B	C9 20	CMP #*20 ;IGNORE SPACE BAR INPUT
3960	FE1D	F0 F5	BEQ RDLUP
3961	FE1F	29 1F	AND #*1F ;MASK OFF UPPER 3 BITS
3962	FE21	99 30 01	STA CH,Y
3963	FE24	98	TYA
3964	FE25	AA	TAX ;Y----> X
3965	FE26	FE 30 01	INC CH,X ;FORMAT TO MATCH DISASSEMBLER TAB
3966	FE29	CB	INY
3967	FE2A	C0 03	CPY #03 ;REPEAT FOR EACH OF 3 CHARACTERS
3968	FE2C	D0 E6	BNE RDLUP
3970	FE2E		;COMPRESS 3 FORMATTED CHARACTERS TO MOVAD & MOVAD+1
3971	FE2E	A0 03	LDY #03 ;SET UP OUTER LOOP
3972	FE30	B9 2F 01	OUTLUP LDA CH-1,Y ;COMPRESS 3 CHARACTERS
3973	FE33	A2 05	LDX #05 ;SET UP INNER LOOP
3974	FE35	4A	INLUP LSK A ;SHIFT 5 BITS ACC TO MOVAD,MOVAD+1
3975	FE36	6E 26 01	ROR MOVAD
3976	FE39	6E 27 01	ROR MOVAD+1
3977	FE3C	CA	DEX
3978	FE3D	D0 F6	BNE INLUP
3979	FE3F	88	DEY
3980	FE40	D0 EE	BNE OUTLUP
3982	FE42		;SEARCH FOR MATCHING COMPRESSED CODE
3983	FE42	A2 40	LDX #*40
3984	FE44	AD 26 01	SRCHLP LDA MOVAD
3985	FE47	DD B8 F5	SRCHM CMP MNEML-1,X ;MATCH LEFT HALF
3986	FE4A	F0 05	BEQ MATCH
3987	FE4C	CA	DEX
3988	FE4D	D0 FB	BNE SRCHM ;IF NO - TRY AGAIN
3989	FE4F	F0 0B	BEQ MATCH1
3990	FE51	AD 27 01	MATCH LDA MOVAD+1 ;ALSO MATCH RIGHT HALF
3991	FE54	DD FB F5	CMP MNEMR-1,X
3992	FE57	F0 06	BEQ GOTIT
3993	FE59	CA	DEX
3994	FE5A	D0 EB	BNE SRCHLP
3995	FE5C	4C C5 FC	MATCH1 JMP ERRORM
3997	FE5F		;GET INSTRUCTION TYPE FROM TYPE TABLE
3998	FE5F	BD 5D FB	GOTIT LDA TYPTB-1,X
3999	FE62	BD 2E 01	STA TYPE
4001	FE65		;GET OP CODE FROM OP CODE UE
4002	FE65	BD 1D FB	LDA STCODE-1,X
4003	FE68	BD 34 A4	STA OFCODE
4004	FE6B	4C C1 FB	JMP MODEM
4006	FE6E		;THIS SECTION SETS THE CURRENT ADDRESS POINTER
4007	FE6E	A9 2A	STLO LDA #'*



LINE #	LOC	CODE	LINE	
4008	FE70	20 7A E9		JSR OUTPUT
4009	FE73	20 AE EA	STLOAD	JSR ADDIN ;GET ADDR
4010	FE76	B0 F6		BCS STLO ;IN CASE OF ERROR
4011	FE78	4C 24 FF		JMP PATC16 ;ADDR TO PC THEN TO STARTM
4013	FE7B			;PATCHES TO CORRECT PROBLEMS WITHOUT
4014	FE7B			;CHANGING ENTRY POINTS TO THE ROUTINES
4015	FE7B	41		BYT 'A'
4016	FE7C	38	PATCH1	SEC ;ADJUST BAUD
4017	FE7D	E9 2C		SBC #44
4018	FE7F	8D 18 A4		STA CNTL30
4019	FEB2	60		RTS
4021	FEB3	8A	CUREAD	TXA ;SAVE X , OUTPUT CUR
4022	FEB4	48		PHA
4023	FEB5	AE 15 A4		LDX CURP02
4024	FEB8	E0 14		CPX #20 ;ONLY IF C 20
4025	FEB8	B0 05		BCS PAT2A
4026	FEB8	A9 DE		LDA #*DE
4027	FEBE	20 7B EF		JSR OUTD1
4028	FE91	68	PAT2A	PLA
4029	FE92	AA		TAX
4030	FE93	4C 3C E9		JMP READ ;CONTINUE
4032	FE96	20 3C E9	RED1	JSR READ ;READ & ECHO WITHOUT CURSOR
4033	FE99	4C 76 E9		JMP RED2
4035	FE9C	AE 15 A4	PATCH4	LDX CURP02 ;DONT DO ANYTHING IF *BD*
4036	FE9F	C9 8D		CMP #*BD ;SO (CR) FOR TV & NOT FOR DISP
4037	FEA1	D0 0B		BNE PAT4A
4038	FEA3	A9 A0		LDA #*A0 ;CLR CURSOR
4039	FEA5	20 7B EF		JSR OUTD1
4040	FEA8	20 44 EB		JSR CLR ;CLR PNTRS
4041	FEAB	4C 76 EF		JMP OUTD7 ;EXIT
4042	FEAE	4C 17 EF	PAT4A	JMP OUTD1A ;CONTINUE
4044	FER1	8D 11 A4	PATCH5	STA PRIFLG ;TURN PRI OFF
4045	FEB4	4C 73 F0		JMP IPO3
4047	FER7	A9 1C	PATCH6	LDA #*ADDR ;SIMULATE LDA (ADDR),Y
4048	FER9	4C 58 EB		JMP LDAY
4050	FERC	20 3C E9	PATCH8	JSR READ ;READ & ECHO WITH CARROTS
4051	FERF	48		PHA
4052	FEC0	20 D8 E7		JSR EQUAL
4053	FEC3	A9 3C		LDA #'C
4054	FEC5	20 7A E9		JSR OUTPUT
4055	FEC8	68		PLA
4056	FEC9	48		PHA
4057	FECA	C9 0D		CMP #CR
4058	FECB	F0 03		BEQ PATCBC
4059	FECE	20 7A E9		JSR OUTPUT
4060	FED1	A9 3E	PATCBC	LDA #'>

LINE #	LOC	CODE	LINE	
4061	FED3	20 7A E9		JSR OUTPUT
4062	FED6	68		PLA
4063	FED7	60		RTS
4065	FED8	C9 F7	PATCH9	CMP #*F7 ;CHCK LOWER TRANSITION OF TIMER
4066	FEDA	B0 06		BCS PAT9A
4067	FEDC	CD 0B A4		CMP TSPEED
4068	FEDF	4C 9D EE		JMP CKF3A
4069	FEE2	CD 0B A4	PAT9A	CMP TSPEED
4070	FEE5	68		PLA
4071	FEE6	C9 FF		CMP #*FF
4072	FEE8	60	PAT9B	RTS
4074	FEE9	20 F0 E9	PATC10	JSR CRLF ;CLR DISP (ONLY 1 (CR))
4075	FEEC	4C 85 E1		JMP STA1
4077	FEF7	F0 F7	PATC11	BEQ PAT9B ;GO OUTPUT PROMPT
4078	FEF1	C9 4C		CMP #'L ;NO PROMPT FOR 'T' OR 'L'
4079	FEF3	F0 F3		BEQ PAT9B
4080	FEF5	4C C5 E7		JMP PROMP1
4082	FEF8	48	PATC12	PHA ;CLEAR PRIFLG SO WE CAN OUTPUT
4083	FEF9	AD 11 A4		LDA PRIFLG ;TO PRINTER IF FLG WAS ON (MSB)
4084	FEFC	29 F0		AND #*F0
4085	FEFE	8D 11 A4		STA PRIFLG
4086	FF01	68		PLA
4087	FF02	60		RTS
4089	FF03	AD 12 A4	PATC13	LDA INFLG ;TURN TAPES ON ONLY IF TAPES
4090	FF06	C9 54		CMP #'T
4091	FF0B	D0 DE		BNE PAT9B
4092	FF0A	4C 29 E5		JMP DU14 ;TURN ON TAPES & SET DEF DEV
4094	FF0D	AD 13 A4	PATC14	LDA OUTFLG ;TURN ON TAPES ONLY IF TAPES
4095	FF10	C9 54		CMP #'T
4096	FF12	D0 D4		BNE PAT9B
4097	FF14	4C 0A E5		JMP DU11
4099	FF17	20 F0 E9	PATC15	JSR CRLF ;DECODE COMMAND
4100	FF1A	8A		TXA ;SAVE INDEX
4101	FF1B	0A		ASL A
4102	FF1C	AA		TAX
4103	FF1D	BD B8 FA		LDA JTRL,X ;PART OF ENTRY
4104	FF20	8D 1A A4		STA S1
4105	FF23	60		RTS
4107	FF24	20 D8 E5	PATC16	JSR CGPC1 ;ADDR TO PC
4108	FF27	4C AA FB		JMP STARTM ;BACK TO MNEMONIC START
4110	FF2A	F0 0E	PATC17	BEQ PAT17B ;RUB ,SO READ ANOTHER
4111	FF2C	C9 00		CMP #0
4112	FF2E	F0 03		BEQ PAT17A
4113	FF30	4C B5 F7		JMP INO2A
4114	FF33	20 93 E9	PAT17A	JSR INALL ;NEITHER ,CONTINUE
4115	FF36	C9 7F		CMP #*7F ;SKIP ON ZEROS
				;UNTILL RUB

LINE #	LOC	CODE	LINE	
4116	FF38	D0 F9	RNE PAT17A	
4117	FF3A	4C 7A F7	PAT17B JMP IN02	;GO BACK
4119	FF3D	20 F8 FE	PATC18 JSR PATC12	;RESET PRIFLG
4120	FF40	48	PHA	
4121	FF41	20 42 EB	JSR TTYTST	;IF TTY JUST RTN
4122	FF44	D0 02	RNE PAT18A	
4123	FF46	68	PLA	
4124	FF47	60	RTS	
4125	FF48	20 FE EB	PAT18A JSR LL	;SET TO LOW SPEED
4126	FF4B	20 45 F0	JSR IPST	;PRINT WHAT IS IN BUFFER
4127	FF4E	20 44 EB	JSR CLR	;CLR PRINTER BUFFER BY OUTPUTTING
4128	FF51	20 3E EB	JSR BLANK	;AN SPACE
4129	FF54	20 44 EB	JSR CLR	
4130	FF57	68	PLA	;RTN ACC
4131	FF58	60	RTS	
4133	FF59	D8	PAT19 CLD	
4134	FF5A	20 24 EA	JSR CRCK	
4135	FF5D	4C 85 E1	JMP STA1	
4137	FF60	F0 0D	PAT20 BEQ VECK4	;END (DATA BYTES=0)
4138	FF62	18	CLC	
4139	FF63	69 04	ADC #4	
4140	FF65	AA	TAX	
4141	FF66	20 93 E9	VECK5 JSR INALL	;SKIP OVER DATA
4142	FF69	CA	DEX	
4143	FF6A	D0 FA	RNE VECK5	
4144	FF6C	4C 9E E6	JMP VECK1	;PROCESS NEXT RCD
4145	FF6F	4C 20 E5	VECK4 JMP DU13	
4147	FF72	A0 00	PAT21 LDY #0	
4148	FF74	B9 88 FF	PAT21A LDA POMSG,Y	;RESET MSG
4149	FF77	F0 06	BEQ PAT21B	
4150	FF79	20 7A E9	JSR OUTPUT	
4151	FF7C	C8	INY	
4152	FF7D	D0 F5	RNE PAT21A	
4153	FF7F	20 F0 E9	PAT21B JSR CRLF	
4154	FF82	20 F0 E9	JSR CRLF	
4155	FF85	4C 82 E1	JMP START	
4157	FF88	20 20	POMSG .BYT ' ROCKWELL AIM 65'	
4158	FF99	00	.BYT 0	
4160	FF9A	EE 68 01	PAT22 INC BLKD	
4161	FF9D	4C BD ED	JMP ADDBK1	
4162	FFA0			
4163	FFA0	A9 FF	PAT23 LDA #FF	;START TIMER
4164	FFA2	BD 97 A4	STA DI1024	
4165	FFA5	AD 85 A4	PAT23A LDA RINT	;TIME OUT?
4166	FFAB	30 08	BMI PAT23B	;YES
4167	FFAA	AD 0D AB	LDA IFR	;START SIGNAL?
4168	FFAD	29 10	AND #MPRST	
4169	FFAF	F0 F4	BEQ PAT23A	;NO
4170	FFB1	60	RTS	;YES

LINE #	LOC	CODE	LINE	
4171	FFB2	A9 00	PAT23B LDA #0	;TIME OUT RETURN
4172	FFB4	60	RTS	
4174	FFB5	20 75 EE	PATC24 JSR CKFREQ	;READ BIT FROM FOURTH HALF PULSE
4175	FFB8	6A	ROR A	
4176	FFB9	29 80	AND #80	
4177	FFBB	60	RTS	
4179	FFBC	2C 0D AB	PATC25 BIT IFR	;WAIT TILL TIMES OUT
4180	FFBF	50 FB	BVC PATC25	
4181	FFC1	AD 04 AB	LDA T1L	;CLR INTERRUPT FLG
4182	FFC4	60	RTS	
4184	FFC5		*=\$FFF9	
4185	FFF9		;INTERRUPT VECTORS	
4186	FFF9	FA	.BYT \$FA	
4187	FFFA	75 E0	.WORD NMIV1,RSET,IRQV1	;SET UP VECTORS
4187	FFFC	BF E0		
4187	FFFE	7B E0		
4188	0000		.END A0/1	

ERRORS = 0000 (0000)

SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE
ABSIND	FC5C	ABSOL	FCA6	ABSOL1	FCB2	ABSX	FC72
ABSY	FC63	ABSY1	FC6E	ACCUM	FC23	ACR	A80B
AD1	F92B	ADD1	E565	ADDA	F92A	ADDA1	F933
ADDBK1	EDBD	ADDBLK	EDBA	ADDIN	EAAE	ADDN1	EAB7
ADDN2	EAC7	ADDIN3	EADC	ADDN4	EAE8	ADDN5	EAF7
ADDN6	EAFD	ADDN7	EB0D	ADDN8	EB2B	ADDNE	EAB1
ADDR	A41C	ADDRS1	F910	ADDS1	E55D	ADDS1A	F913
ADFLD	0133	ASSEM	D000	AT01	F8F7	AT02	F8F5
ATBOT	F8E9	ATEND	F8F9	ATTOP	F8DB	BACKWD	FDD9
BASLEN	B000	BASIRE	B003	BKCK1	F1F1	BKCK2	F20F
BKCK3	F21A	BKCKSM	F1E7	BKERR	E62F	BKFLG	A410
BKQ2	E64C	BKOK	E634	BKS	0100	BLANK	E83E
BLANK2	E83B	BLK	0115	BLKO	016B	BOTLN	00E1
BRCOMP	FD86	BRK1	E620	BRK2	E6F3	BRK3	E6F1
BRK4	E6FA	BRKA	E61B	BRKK	E6E5	BRNCHC	FD0F
BT	F721	BYTESM	A42F	CBUFF1	F1E2	CD02	FABF
CFLG	F8B2	CFND1	FAA0	CGA	E5EE	CGALL	E5FC
CGPC	E5D4	CGPC0	E5D7	CGPC1	E5DD	CGPS	E5EA
CGS	E5FA	CGX	E5F2	CGY	E5F6	CH	0130
CH2	E2B8	CH3	E2C5	CH4	E2C0	CHAR1	F5AD
CHAR2	F5B3	CHEKA	E54E	CHEKAR	E54B	CHN1	F87C
CHN2	F88C	CHN3	F8A9	CHN4	F8AF	CHNG	F876
CHNG1	E2A6	CHNGG	E2A0	CKR	E76B	CKR1	E780
CKB2	F76D	CKRUFF	F1D2	CKERO	E38E	CKEROO	E394
CKER1	E396	CKER2	E3A3	CKERR	E385	CKF1	EE7A
CKF2	EEB1	CKF3	EE99	CKF3A	EE9D	CKF4	EEA1
CKFREQ	EE75	CKSUM	A41E	CLR	EB44	CLRRK	E6FE
CLRCK	EB4D	CLRLUP	FBE9	CMFBR1	FD8B	CNTH30	A417
CNTL30	A418	CODFLG	A437	COLO	F2E1	COL1	F321
COL2	F361	COL3	F3A1	COL4	F3E1	COM	FA78
COMB	E1C4	COMCN1	000B	COMIN	E1A1	COMM	FA8B
COMPR	FD9E	COMTBL	FAAC	CONVRT	FD12	CORR	FB00
COUNT	A419	CFIY	A42A	CR	000D	CR2J	EA23
CRA	AC01	CRB	AC03	CRCK	EA24	CRCK1	EA2C
CRCK2	EA3B	CRLF	E9F0	CRIOW	EA13	CUREAD	FE83
CURP02	A415	CURPOS	A416	DATIN	000E	DATOUT	000C
DDRA	A803	DDRA2	A481	DDRB	A802	DDRB2	A483
DE1	E018	DE2	EC1B	DEBK1	ED2C	DEBKEY	ED2A
DEBTIM	1388	DEHALF	EC23	DELAY	EC0F	DI1024	A497
DIBUFF	A438	DILINK	A406	DISASM	F46C	DISFLG	A40F
DISPLY	FD6E	DIV1	A494	DIV64	A496	DIV8	A495
DLNE	F74C	DNNO	F6D8	DNPA7	A484	DON1	E7A0
DONE	E790	DOW1	F6E3	DOW2	F6E8	DOWN	F724
DPPA7	A485	DR	A80F	DRA2	A480	DRAH	A801
DRB	A800	DRB2	A482	DUO	E447	DU1	E444
DU10	E4DB	DU10A	E4F8	DU11	E50A	DU12	E511
DU13	E520	DU14	E529	DU1A	E46D	DU1B	E452
DU2	E47D	DU6	E49F	DU7	E4A0	DU8	E4A2
DU9	E4B9	DUK2	E5A4	DUMP	E43B	DUMPK1	E587
DUMPT1	E57B	DUMPTA	E56F	EDI	F6B6	EDI0	F644
EDI1	F653	EDI2	F663	EDI2B	F6CC	EDI3	F673
EDI4	F680	EDI5	F68D	EDI6	F69B	EDI7	F6AA
EDIB	F6AE	EDIT	F639	EMSG1	E06C	EMSG2	E072
END	00E5	ENDE2	FA6F	ENDERR	FA5C	ENPA7	A486
ENTRY	FA8D	EPPA7	A487	EQS	00BD	EQUAL	E7D8
ERR	F495	ERRO	FA7B	ERRFLG	FD2B	ERRJMP	FDD6

SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE
ERROR	FA72	ERRORM	FCC5	ESCAPE	001B	EVAL	FC0E
FC1	F823	FC2	F82E	FC3	F834	FC4	F843
FC5	F849	FC6	F84E	FC7	F853	FC8	F85A
FC9	F868	FCH	F81E	FCHA1	F80F	FCHAR	F80C
FNAM	E8A2	FORM1	FD7D	FORMA	0116	FORMD1	FD58
FORMD2	FD69	FORMDS	FD45	FORWRD	FDE0	FROM	E7A3
GAP	A409	GCN1	E78C	GCNT	E785	GET1	EBE2
GET3	EBED	GETA1	EE2B	GETFMT	F499	GETID	E425
GETK0	EC55	GETK00	EC67	GETK1	EC71	GETK10	ECEC
GETK11	ECC9	GETK12	ECD2	GETK13	ECE1	GETK14	ECEB
GETK1B	EC80	GETK2	EC82	GETK3	EC8D	GETK4	EC93
GETK5	ECA4	GETK6	ECB9	GETK7	ECRE	GETK8	ECBF
GETKDO	EC38	GETKEY	EC40	GETKY	EC43	GETTAP	EE29
GETTTY	ERDB	GID1	E427	GO	E261	GOBK	E26D
GOBK0	E27B	GOBK1	E286	GOERR	E608	GOGO	FA4A
GOG01	FA5B	GOTIT	FE5F	HATCH	FCB6	HATCJ	FC3D
HEX	EA7D	HIST	A42E	HISTM	A42E	HISTP	A414
IBITL	A47A	IBITU	A47B	IBUFM	A460	ICOL	A475
IDIR	A474	IDOT	A477	IER	A80E	IEVEN	F486
IFR	A80D	IMASK	A47C	IMMED1	FCC1	IN	F764
IN02	F77A	IN02A	F785	IN03	F7A8	IN03A	F7B9
IN03B	F799	IN05	F7C5	INALL	E993	INCP	F121
INCS2	E566	INDX	FC81	INFLG	A412	INL	F76D
INLOW	E9F8	INLUP	FE35	INPU	F7CB	INPU1	F7D8
INTAB1	E743	INTAB2	E752	INTAB3	E756	IOFFST	A476
IOUTL	A478	IOUTU	A479	IP00	F050	IP02	F066
IP03	F073	IP04	F078	IPSO	F04A	IPS1	F0E8
IPS2	F10E	IPS3	F105	IPST	F045	IPSU	F0E3
IRQ1	E163	IRQ2	E17F	IRQV1	E078	IRQV2	A404
IRQV3	E154	IRQV4	A400	ISX	FE03	JD1	E723
JD2	E72B	JD3	E73C	JD4	E742	JMPR	E1C1
JTBL	FA8B	JUMP	A47D	KIISA	E70A	KEP	E7AF
KEPR	E970	KEYF1	010C	KEYF2	010F	KEYF3	0112
KI2	F8B8	KIFLG	F8B6	KMASK	A42A	LDAY	EB58
LDIY	A42A	LENGTH	00EA	LF	000A	LL	EBFE
LMNEM	0117	LOAD	E2E6	LOAD1	E2E9	LOAD1A	E349
LOAD2	E306	LOAD4	E321	LOAD5	E323	LOADK1	E3A7
LOADK2	E3AA	LOADK3	E3B7	LOADK5	E3D1	LOADK6	E3D3
LOADK7	E3E8	LOADK1	E3A4	LOADT2	E364	LOADTA	E32F
LST	F7E1	LST01	F7F0	LST02	F7FB	LST3	F803
LT10	EA5A	M1	E000	M10	E02D	M11	E031
M12	E03B	M3	E005	M4	E008	M5	E01C
M6	E021	M7	E024	M8	E027	M9	E02A
MATCH	FE51	MATCH1	FE5C	MCM2	E196	MCM3	E1AC
MCNT	0020	MEIN	E24D	MEM	E248	MEM1	E24F
MEM2	E251	MEM3	E260	MEMERR	EB33	MNEENT	FB9E
MNEM	FE06	MNEML	F5B9	MNEMR	F5F9	MNNDX1	F4AF
MNNDX2	F4B3	MNNDX3	F4BA	MODE	F55B	MODE2	F59F
MODEM	FBC1	MOFF	00E0	MON	00C0	MONCOM	E1E5
MONRAM	A400	MOVAD	0126	MFRST	0010	MR11A	F512
MREAD	FAD0	MSP12	0002	MT2	0020	MTBL	F2D7
NAME	A42E	NAM0	EBCF	NAM01	EBD6	NAM02	EBE9
NAM03	EBEB	NAM04	EBF5	NEWCOL	F163	NEWROW	F160
NH1	E690	NHIS	E688	NM14	E0B1	NM15	E0B4
NMIV1	E075	NMIV2	A402	NMIV3	E07B	NOUT	EA51
NOWLN	00DF	NOWS1	F909	NPUL	A40A	NULLC	00FF

SYMBOL	VALUE						
NUMA	EA46	NXT5	E60D	NXTA1	E2DA	NXTADD	E2CD
OK	FDE7	OLDLEN	00E9	ONEBYT	FD3E	ONEK1	ED09
ONEK2	ED0B	ONEK3	ED1C	ONEK4	ED29	ONEKEY	ED05
OP03	F144	OP04	F130	OP05	F150	OP06	F15D
OP07	F13F	OPCMP1	FCB5	OPCODE	A434	OPCOMP	FCCB
OUT01	F00F	OUT04	F025	OUT05	F033	OUT1	E97B
OUT1A	E986	OUT2	E98F	OUTA1	E9C8	OUTA2	E9D0
OUTA3	E9E2	OUTA4	E9EA	OUTALL	E9BC	OUTCK	E538
OUTCK1	E53B	OUTCK2	E547	OUTCK5	E531	OUTD1	EF14
OUTD1A	EF17	OUTD2	EF20	OUTD2A	EF2F	OUTD3	EF33
OUTD4	EF48	OUTD5	EF56	OUTD7	EF76	OUTDI1	EF7B
OUTDI2	EF87	OUTDI3	EF8B	OUTDIS	EF05	OUTDP	EEFC
OUTDP1	EF02	OUTFLG	A413	OUTL1	E906	OUTLOW	E901
OUTLUP	FE30	OUTPR	F038	OUTPR1	F03A	OUTPR2	F044
OUTPRI	F000	OUTPUT	E97A	OUTT1	EECB	OUTT2	EEFB
OUTTA1	F290	OUTTA2	F294	OUTTA3	F2B2	OUTTAP	F24A
OUTTTY	EEA8	P00	F749	P01	F73B	P02	F729
P03	F73F	PACK	EA8A	PAK1	EA96	PAK2	EA9F
PAREN	FC76	PAT17A	FF33	PAT17B	FF3A	PAT18A	FF48
PAT19	FF59	PAT20	FF60	PAT21	FF72	PAT21A	FF74
PAT21B	FF7F	PAT22	FF9A	PAT23	FFA0	PAT23A	FFA5
PAT23B	FFB2	PAT2A	FE91	PAT4A	FEAE	PAT9A	FE2E
PAT9B	FE08	PATC10	FE09	PATC11	FE0F	PATC12	FE08
PATC13	FE03	PATC14	FE0D	PATC15	FE17	PATC16	FE24
PATC17	FE2A	PATC18	FE3D	PATC24	FFB5	PATC25	FFBC
PATC8C	FE01	PATCH1	FE7C	PATCH4	FE9C	PATCH5	FEB1
PATCH6	FE07	PATCH8	FEBC	PATCH9	FE08	PCADJ3	FE5D
PCADJ4	F554	PCLLD	EB56	PCR	A80C	PHXY	EB9E
PINT	F0CB	PLNE	F727	PLXY	EBAC	PNTLUP	FBD0
PMSG	FF88	PR1	E7CC	PR2	E7CF	PRADR1	F4F7
PRADR2	F4FF	PRADR3	F519	PRADR4	F52C	PRBL2	F545
PRDOT0	F08C	PRIERR	F079	PRIFLG	A411	PRITR	E6E1
PRMN1	F4D7	PRMN2	F4DB	PRNDOT	F087	PRNTYX	F538
PROMP1	E7C5	PROMPT	E7BD	PRPC	F53C	PRST	0000
PRTIME	06A4	PSLO	E7FB	PSL00	E802	PSLOA	EB14
PSLOB	EB1C	PSLOC	EB1E	PSL0D	E823	PSL1	E837
PSLS	E7DC	QM	E7D4	R10	F9C7	R100	F9CF
R101	F9DA	R102	F9E3	R103	F9FA	R104	FA17
R105	FA31	R1051	FA41	R106	FA44	R107	FA0A
R108	F9EF	R11	F9CC	R2W	F95F	R5	F99D
R55	F9A8	R6	F984	R7	F9AB	R8	F947
R87	F94E	R88	F953	R9	F9BE	RA	AC00
RB	AC02	RB2	E95C	RBYT1	E407	RBYTE	E3FD
RCH2	E91F	RCH3	E925	RCHEK	E907	RCHT1	E93B
RCHT2	E928	RCHTTY	E926	RD1	EA70	RD2	EA5D
RDADDR	FBE5	RDBIT	EE3B	RDBIT1	EE43	RDBIT2	EE51
RDBIT4	EE67	RDLUP	FE14	RDR1	E96A	RDRUB	E95F
REA1	E956	READ	E93C	READ1	E94A	READ2	E94D
RED1	FE96	RED2	E976	REOUT	E973	REENTR	F6CF
REG	E227	REG1	E232	REGF	A40E	REGQ	F461
REGT	E6D9	RELADR	F530	REP2	F93E	REPLAC	F93F
RESNOW	F8D0	RINT	A485	RMNEM	0118	ROLLFL	A47F
RO01	ED00	ROONEK	ECEF	ROUT	F286	ROUT1	F288
ROW1	F421	ROW2	F429	ROW3	F431	ROW4	F439
ROW5	F441	ROW6	F449	ROW7	F451	ROW8	F459
RQP	F977	RS1	E0C9	RS2	E0D4	RS20	E702

SYMBOL	VALUE						
RS3	E0F3	RS3A	E0F1	RS3B	E11A	RS4	E11D
RS5	E129	RS6	E13E	RS7	E144	RS8	E146
RSET	E0BF	RSPAC	EA7B	RTMODE	F491	RTS1	F55A
RUB	0008	S1	A41A	S2	0106	SADDR	EB78
SAVA	A421	SAVE	00E7	SAVNOW	F934	SAVPC	A425
SAVPS	A420	SAVS	A424	SAVX	A422	SAVY	A423
SEMI	E9BA	SETBOT	F8C5	SETREG	E113	SETSP1	F2CA
SETSP2	F2B3	SETSPD	F2C0	SETZ	F282	SH1	E652
SH11	E66A	SHIS	E665	SHOW	E64D	SIZEM	F80F
SP12	0001	SR	A80A	SRCHLP	FE44	SRCHM	FE47
STA1	E185	START	E182	STARTM	FBA4	STASH	FD2C
STRKEY	A42B	STBYTE	E413	STCODE	F81E	STIY	A427
STLO	FE6E	STLOAD	FE73	STOP	F870	STOR1	FC0A
STORCH	F8F6	STRING	00EB	STSHLP	FD30	SUB	F91D
SUB1	F927	SWST1	EBBD	SWSTAK	EBBA	SYNC	EDFF
SYNC1	EE11	T1CH	A805	T1FR	00C0	T1I	0000
T1L	A804	T1LH	A807	T1LL	A806	T2H	A809
T2I	0000	T2L	A808	TARUF2	00AD	TABUFF	0116
TABY2	F1A7	TABY3	F1CE	TAISET	EDEA	TAOS1	F238
TAOSET	F21D	TAP1	E8B3	TAP2	E8BC	TAP3	E8C2
TAPIN	A434	TAPOUT	A435	TAPTR	A436	TAPTR2	A437
TEMPA	A433	TEMPX	A431	TEXT	00E3	TIB1	ED48
TIBY1	ED53	TIBY3	ED56	TIBY4	ED63	TIBY5	ED65
TIBY5A	ED88	TIBY6	EDAF	TIBY7	EDB0	TIBYTE	ED3E
TIMG	A40B	TIOS1	EE22	TIOS2	EE24	TIOSSET	EE1C
TMASK1	0126	TMASK2	0127	TMSG0	E048	TMSG1	E04D
TMSG2	E050	TMSG3	E052	TMSG5	E05F	TMSG6	E061
TMSG7	E066	TO	E7A7	TO1	E7A9	TORYTE	F18B
TOGL	E6E7	TOGL1	E6F6	TOGTA1	E6BD	TOGTA2	E6CB
TOPNO	F8BC	TP	F6D2	TP01	F8C0	TRACE	E6DD
TRY	F258	TRY34	FC40	TRY56	FC5A	TRYINY	FC85
TRYJMP	FC94	TRYZF	FC28	TSPEED	A408	TYTST	E842
TYPE	012E	TYPTB	F85E	TYPTR1	FAE2	TYPTR2	FAF1
UACR	A00B	UDDRA	A003	UDDRB	A002	UDRA	A00F
UDRAH	A001	UDRB	A000	UIER	A00E	UIFR	A00D
UIN	0108	UOUT	010A	UP	F6F9	UP1	F713
UP4	F720	UPCR	A00C	UPNO	F709	USR	A00A
UT1CH	A005	UT1L	A004	UT1LH	A007	UT1LL	A006
UT2H	A009	UT2L	A008	VALID	FCDD	VECK1	E69E
VECK2	E6AC	VECK4	FF6F	VECK5	FF66	VECKSM	E694
WHE1	E85C	WHE2	E868	WHE3	E870	WHEREI	E848
WHEREO	E871	WHICHT	E8A8	WHRO1	E885	WHRO2	E88E
WHRO3	E897	WHRO4	E89F	WRAX	EA42	WRITAD	E2DD
WRITAZ	E2DB	XORY	FDEF	XORY1	FDFC	XORYRT	FE02
XORYZ	FD11	ZON	F25D	ZON1	F261	ZON2	F26C
ZPAGE	FC38	ZPX	FC55	ZPY	FC50		

END OF ASSEMBLY





## SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0105

## SYMBOL DEFINED REFERENCES

DDRB	0207	2115
DDRB2	0179	
DE1	1862	2024
DE2	1863	2464 1878 1865
DEBK1	2019	1924
DEBK2	2018	1993 1889
DEBTIM	0236	2023 2021
DEHALF	1870	1851 1841
DELAY	1859	2245 2238 2227 2222 1850 1847 1840 1431
DI1024	0199	4164
DIBUFF	0154	3443 3202 3158 2291 2280 2109 2107 2098 2096 1706 1285
DILINK	0105	2262
DISASM	2854	3866 2850 1159 0554
DISFLG	0112	1117 0552
DISFLY	3875	3881
DIV1	0196	
DIV64	0198	2175 2172 2167 2164
DIV8	0197	
DLNE	3171	3608
DNNO	3110	3573 3221 3149
DNFA7	0183	
DON1	1214	1207
DONE	1205	3246 1157 0356
DOW1	3114	3111
DOW2	3116	3121
DOWN	3149	3608
DPFA7	0184	
DRA	0220	
DRA2	0176	2003 1892 0376
DRAH	0206	2466 2458 2446
DRE	0205	2635 2633 2469 2467 2456 2455 2244 2236 2225 2223 2212 2192 2190 2143 2057 2055 1842 1838 1429 1303 1105 1103 1097 1095 0878 0417 0412 0370
DRE2	0178	2006 1987 1902 1893
DU0	0777	0778
DU1	0776	0847
DU10	0843	0812
DU10A	0856	0861
DU11	0864	4097
DU12	0867	0872
DU13	0873	4145 1092 0966 0869 0866 0737 0641
DU14	0877	4092
DU1A	0792	0785
DU1B	0781	0782
DU2	0799	0841
DU6	0816	0809
DU7	0817	0815
DUB	0819	0814
DU9	0829	0834
DUK2	0946	0952
DUMP	0772	0505
DUMPKI	0933	0796
DUMPT1	0925	0929
DUMPTA	0920	1337
EDI	3086	3068 3057

## SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0106

## SYMBOL DEFINED REFERENCES

EDI0	3038	3039
EDI1	3044	3049
EDI2	3051	3052
EDI2B	3097	3091
EDI3	3057	3058
EDI4	3062	3055
EDI5	3068	3069
EDI6	3074	3081
EDI7	3080	3072
EDI8	3082	3073
EDIT	3034	0504
EMSG1	0324	3035
EMSG2	0325	3571 3131
END	0057	3390 3387 3080 3070 3065 3063
ENDE2	3577	3575
ENDERR	3570	3531 3291 3254 3136
ENFA7	0185	
ENTRY	3592	3079
EPFA7	0186	
EQS	0262	0321 0318 0317 0313 0312 0306 0305
EQUAL	1249	4052 1664 0996
ERR	2877	2864 2862
ERR0	3580	3599 3582 3577 3222 3186 3113
ERRFLG	3841	3836 3831
ERRJMP	3922	3928 3925 3902 3895
ERROR	3578	3282
ERRORM	3788	3995 3922 3823 3798 3770 3767 3762 3761 3740 3738
ESCAPE	0260	1448 1433 1420
EVAL	3706	3701
FC1	3269	3281
FC2	3274	3316 3293
FC3	3276	3273 3271
FC4	3283	3277
FC5	3285	3275
FC6	3287	3297
FC7	3289	3301
FC8	3292	3290
FC9	3298	3295
FCH	3267	3258
FCHA1	3259	3317
FCHAR	3258	3609 3312
FNAM	1356	1341 1336 1319 1314
FORM1	3884	3869
FORMA	0076	2945 2931 2881
FORMD1	3866	3862
FORMD2	3873	3865
FORMDS	3859	3849
FORWRD	3926	3919
FROM	1217	3038 0777
GAP	0107	2699
GCN1	1200	1198
GCNT	1197	3241 1153 0544
GET1	1838	1839
GET3	1842	1849
GETA1	2150	2155







## SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0111

## SYMBOL DEFINED REFERENCES

OPCOMP	3792	3787 3785 3779 3777 3772 3764 3757 3750 3748 3746 3737 3735 3725 3723 3716 3682
OUT01	2381	2376
OUT04	2392	2379
OUT05	2398	2396 2391
OUT1	1476	1540
OUT1A	1481	1478
OUT2	1485	1482
OUTA1	1518	1514
OUTA2	1523	1519
OUTA3	1534	1524
OUTA4	1539	1535
OUTALL	1510	3157 2966 2951 2948 2920 1610 1560 1558 1547
OUTCK	0887	0944 0942 0827 0825 0823
OUTCK1	0888	0938 0859 0857 0855 0839 0837
OUTCK2	0894	0891
OUTCKS	0882	0946 0829
OUTD1	2274	2270
OUTD1A	2275	4042
OUTD2	2280	2276
OUTD2A	2287	2284 1269
OUTD3	2290	2297
OUTD4	2299	2283
OUTD5	2307	2314 2273
OUTD7	2325	4041 2315 2308 2304 2285
OUTDD1	2331	4039 4027 2311 2301 2293 0431
OUTDD2	2340	2342
OUTDD3	2343	2339
OUTDIS	2267	2252 1244 1177
OUTDF	2258	1486
OUTDF1	2262	1588 1480 1262
OUTFLG	0116	4094 3163 2650 2105 2102 2100 1569 1564 1550 1511 1408 1331 0889 0864 0792
OUTL1	1409	
OUTLOW	1407	1566
OUTLUF	3972	3980
OUTPR	2401	2397 2385 1291
OUTPR1	2402	2406
OUTPR2	2407	2403
OUTPRI	2373	2258 1586 1529 1346
OUTPUT	1475	4150 4061 4059 4054 4008 1242 1228 1148 0694 0658 0473 0469
OUTT1	2232	2241
OUTT2	2253	2251 2249
OUTTA1	2745	2714
OUTTA2	2747	2763
OUTTA3	2759	2761 2754
OUTTAP	2710	2704 2703 2702 2701 2629 2626 1521 0964 0963 0955 0935
OUTTTY	2219	1484
P00	3167	3165
P01	3161	3156 3154
P02	3153	3160
P03	3163	3252
PACK	1638	3835 1720 1626 1623 0768 0766 0748 0746 0723
PAK1	1647	1643
PAK2	1653	1656

## SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0112

## SYMBOL DEFINED REFERENCES

PAREN	3751	3711
PAT17A	4114	4116 4112
PAT17B	4117	4110
PAT18A	4125	4122
PAT19	4133	0467
PAT20	4137	1084
PAT21	4147	0426
PAT21A	4148	4152
PAT21B	4153	4149
PAT22	4160	2641
PAT23	4163	2419 2417
PAT23A	4165	4169
PAT23B	4171	4166
PAT2A	4028	4025
PAT4A	4042	4037
PAT9A	4069	4066
PAT9B	4072	4096 4091 4079 4077
PATC10	4074	0590
PATC11	4077	1238
PATC12	4082	4119 3193 1267
PATC13	4089	3218
PATC14	4094	3253
PATC15	4099	3600
PATC16	4107	4011 3886
PATC17	4110	3195
PATC18	4119	1450 1028
PATC24	4174	2188
PATC25	4179	4180 2759 2752
PATC8C	4060	4058
PATCH1	4016	0424
PATCH4	4035	2274
PATCH5	4044	2437
PATCH6	4047	3087
PATCH8	4050	3591
PATCH9	4065	
PCADJ3	2969	2955
PCADJ4	2973	2971
PCLLD	1757	2933 2902 2857
PCR	0217	2694 2452 2450 2433 2416 2119
PHXY	1791	2615 2374 2268 2220 2031 1440 1356 1256
PINT	2473	2413
PLNE	3152	3608 3335 3262 3248 3219 3181 3177 3175 3128 3112
PLXY	1803	2644 2398 2325 2277 2246 2039 1446 1370 1292
PNTLUF	3672	3675
POMSG	4157	4148
PR1	1242	1299 1295 1250 1247
PR2	1243	1240
PRADR1	2927	2953
PRADR2	2931	2944
PRADR3	2945	2930 2928
PRADR4	2952	2950 2946
PREL2	2965	2923 2904 2854
PRDOT0	2447	2449
PRIERR	2436	2421
PRIFLG	0114	4085 4083 4044 3209 3208 2411 2110 2093 1531 1526 1476 1121

## SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0113

## SYMBOL DEFINED REFERENCES

		0681 0676 0665 0654
PRTR	1121	1970
PRMN1	2912	2922
PRMN2	2914	2918
PRNDOT	2445	2427 2426
PRNTYX	2960	2958
PROMPF1	1239	4080
PROMPT	1236	3268 3190
PRFC	2962	3871 2855
PRST	0225	2432 2415
PRTIME	0235	2461 2459
PSL0	1267	1260
PSL00	1270	1264
PSL0A	1278	1274
PSL0B	1282	1272
PSL0C	1283	1281 1277
PSL0D	1285	1290
PSL1	1294	3240 1255 1152 1014 0543
PSLS	1254	1454
QM	1246	3597 3579 1368 1021 1001 0481
R10	3497	3494
R100	3503	3449
R101	3509	3507
R102	3513	3517
R103	3525	3523
R104	3539	3555 3537
R105	3549	3545
R1051	3555	3553
R106	3556	3548
R107	3532	3526
R108	3518	3515
R11	3500	3496
R2W	3449	3433
R5	3479	3475
R55	3483	3481
R6	3468	3483 3466
R7	3484	3478
R8	3435	3500
R87	3440	3498 3434
R88	3443	3447
R9	3493	3557 3491
RA	0246	2361 2359 2356 2349 0405
RB	0248	2353 0406
RB2	1454	1462
REBT1	0745	0743
REYTE	0741	0897 0732 0729 0708 0637 0632 0629 0624
RCH2	1424	1426
RCH3	1427	1423 1417
RCHK	1413	3245 1156 0800 0551 0358
RCHT1	1437	1472 1449 1428
RCHT2	1429	1436 1430
RCHTY	1428	1414
RD1	1623	1620
RD2	1614	1197 0997 0573
RDADDR	3685	3692 3680

## SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0114

## SYMBOL DEFINED REFERENCES

RDBIT	2159	2150 2123
RDBIT1	2162	2163
RDBIT2	2168	2169
RDBIT4	2184	2185 2160
RDLUP	3956	3968 3960
RDR1	1461	1457
RDRUE	1455	3956 3697 3690 3269 1668 1504 1464 1380
REA1	1450	1434 1421
READ	1440	4050 4032 4030 3313
READ1	1445	1442
READ2	1446	1444
RED1	4032	0470
RED2	1471	4033 1459
REDOUT	1470	1625 1614 1015
REENTR	3103	0504
REG	0514	0504
REG1	0518	2851 0549
REGF	0111	2849 1112 0547
REGQ	2849	0463
REGT	1112	0507
RELADR	2955	2934
REP2	3427	3442
REFLAC	3431	3223 3180
RESNOW	3356	3556 3484
RINT	0190	4165
RMNEM	0078	2914 2910
ROLLFL	0170	2008 1990
ROO1	1994	1989
ROONEK	1987	1992 1888 1424 1415
ROUT	2737	2732
ROUT1	2739	2764
ROW1	2839	1935
ROW2	2840	
ROW3	2841	
ROW4	2842	
ROW5	2843	
ROW6	2844	
ROW7	2845	
ROW8	2846	
RQF	3462	3460 3458
RS1	0369	0372
RS2	0375	0378
RS20	1141	1143
RS3	0389	0393 0387
RS3A	0388	0385 0382
RS3B	0408	0404
RS4	0412	0414
RS5	0417	0418
RS6	0425	0436
RS7	0427	0413
RS8	0428	0435
RSET	0362	4187
RSPAC	1628	1645 1641 1639 1624 1622 1618 1616
RTMODE	2875	2870
RTS1	2976	2974



SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0117

SYMBOL DEFINED REFERENCES

TOGTAZ	1103	0508
TOPNO	3344	3104 3074 3053
TP	3104	3609
TP01	3346	3354
TRACE	1117	0507
TRY	2716	2738
TRY34	3727	3718
TRY56	3738	3696
TRYINY	3758	3755
TRYJMP	3765	3759
TRYZF	3717	3714
TSPEED	0106	4069 4067 2769 2713 2208 2204 2201 2159
TTYTST	1302	4121 3861 3574 1580 1548 1481 1441 1413 1254 1239
TYPE	0082	3999 3781 3773 3768 3743 3719 3678 3670
TYPEB	3644	3998
TYPETR1	3627	3792
TYPETR2	3629	3796
UACR	0040	
UDDRA	0032	
UDDRE	0031	
UDRA	0044	
UDRAH	0030	
UDRB	0029	
UIER	0043	
UIFR	0042	
UIN	0066	1502 1324
UOUT	0067	1537 1351
UP	3127	3608 3185
UP1	3137	3141 3135
UP4	3144	3138 3130 3118
UPCR	0041	
UPNO	3133	3274 3249 3235 3220 3127
USR	0039	
UT1CH	0034	
UT1L	0033	
UT1LH	0036	
UT1LL	0035	
UT2H	0038	
UT2L	0037	
VALID	3799	3795
VECK1	1080	4144 1082
VECK2	1086	1091 1088 1079
VECK4	4145	4137
VECK5	4141	4143
VECKSM	1076	0508
WHE1	1316	1312
WHE2	1321	1317
WHE3	1325	1322
WHEREI	1308	3229 1076 0609
WHEREO	1329	3243 0786
WHICHT	1358	1369
WHRO1	1338	1334
WHRO2	1343	1339
WHRO3	1348	1344
WHRO4	1353	1349

SYMBOL CROSS REFERENCE FOR PA00-J001A.....PAGE 0118

SYMBOL DEFINED REFERENCES

WRAX	1593	2964 2961 1042 0605
WRITAD	0603	0520
WRITAZ	0602	3662 3061 3042 1739 1008 0686 0571
X	0000	4103 4002 3998 3991 3985 3965 3940 3936 3833 3829 3805 3800
		3796 3792 3601 3593 3294 3047 3046 3045 3044 2949 2947 2880
		2869 2686 2684 2658 2655 2605 2603 2601 2597 2595 2593 2592
		2591 2588 2585 2582 2581 2579 2576 2573 2570 2567 2565 2562
		2559 2556 2553 2519 2404 2392 2280 2063 2051 2036 1828 1826
		1820 1818 1817 1816 1722 1707 1696 1371 1286 1186 1182 1141
		1134 1130 1127 1033 1031 0999 0925 0767 0765 0761 0670 0669
		0657 0650 0493 0491 0476 0406 0405 0390 0389 0376 0375 0370
		0369
XORY	3935	3760 3739
XORY1	3941	3938
XORYRT	3944	3947
XORYZ	3936	3731
Y	0000	4148 3972 3962 3846 3837 3693 3687 3615 3560 3541 3513 3470
		3444 3443 3436 3289 3278 3202 3158 3153 3137 3119 3076 2909
		2907 2720 2718 2602 2600 2291 1975 1972 1935 1762 1760 1718
		1714 1706 1393 1385 1285 1225 1058 1056 1041 1040 0691 0604
		0603 0349 0347
ZON	2718	2736 2734
ZON1	2720	2728
ZON2	2724	2725
ZPAGE	3724	3721
ZFX	3736	3733
ZFY	3734	