APPENDIX

APPENDIX

Notes on text:

1. When using the serial interface software, note that the keyboard input would be coming from the serial port on Tanex, not from the keyboard socket on Microtan.

2. It is worth remembering that if you try to copy a block upwards by a few locations that your program may be corrupted as the copy command is executed, as it over-writes data with new data before it has had a chance to read it. This effect can be of use for clearing down block of memory. A 'Clear-Screen' based on this effect could be:

```
M2ØØ,xx,32
C2ØØ,3FF,2Ø1
```

3. If you are using an expanded Tanex with a 6551 chip onboard and you are <u>not</u> using the serial interface facilities you must use the special header plug to disable the 6551, otherwise Tanbug will sit and wait until you connect up a terminal, which could be a long time! The header plug simply consists of a 10K Resistor taken at one end to +5v and the other end to pins 8, 10 and 11 on socket E1 on Tanex.

4. Note that pressing Control P, V or S will not cause a character to be echoed on the screen.

TANBUG Command Quick Reference Chart

Command	Function
RESET	Initialise system and display TANBUG message
Mxxxx	Modify memory at hexadecimal address xxxx
lf	Step up through memory
esc	Step down through memory
space	Re-open currently displayed memory location
cr	Close currently displayed memory location
Lxxxx,y	List y lines of memory starting at address xxxx
Gxxxx	Go, begin program execution at address xxxx
R	Display pseudo processor register locations
S	Set single step mode
N	Return to normal mode (Clear single step)
P	Proceed, past breakpoint or next instruction (singlestep)
Bxxxx,y	Set breakpoint number y at location xxxx
B	Clear breakpoints
Oxxxx,yyyy	Calculate offset between address xxxx and yyyy for branch
Cxxxx,yyyy,zzzz	Copy block from xxxx to yyyy to location starting zzzz
Cntl P	Switch on/off parallel printer
Cntl V	Switch on/off serial printer
Cntl S	Switch on/off screen
Cntl L	Clear screen (in BASIC)
BAS	Enter BASIC
WAR	BASIC warm start

XBUG Commands

Command	Function
C	Set CUTS standard cassette speed (300 Baud)
F	Set fast cassette speed (2400 Baud)
Dxxxx,yyyy,zzzzzz	Dump to tape from xxxx to yyyy using filename zzzzzz
F,zzzzzz	Fetch file zzzzzz from cassette
E,zzzzzz	Examine (verify) file zzzzzz with tape
Txxxx	Enter translator (assembler) at address xxxx
Ixxxx	Interpret (Disassemble) from address xxxx

```
0000:                ;C
0000:                ;C
0000:                ;C
0000:                        MSB   OFF
0000:                ;SOME DEFS
0000:                PSTAT   EQU   $0
0001:                ICHAR   EQU   1          ;INPUT CH
0002:                OCHAR   EQU   $2
000A:                ICURS   EQU   $A
0003:                VDUIND  EQU   $3
000F:                SIMCOM  EQU   $F          ;ZERO IF ASCII KB
0011:                INTSL2  EQU   $11         ;INT LINK
0000:                ;DEFS FOR SERIAL PRINTER
BFD0:                SERDT   EQU   $BFD0       ;SERIAL DATA
BFD1:                SERST   EQU   $BFD1       ;STATUS
BFD2:                SERCM   EQU   $BFD2       ;COMAND
BFD3:                SERCN   EQU   $BFD3       ;CONTROL
0000:                ;C

F800:                        ORG   $F800       ;A 2K VERSION
F800:                ;C
F800:                ;C
F800:                ;SUBROUTINE JUMP TABLE,LOCATED AT START
F800:                ;OF ROM.THESE JUMP ADDRS WONT CHANGE
F800:                ;WHITH FUTURE TANBUG VERSIONS.
F800:4C 51 F9        PRPUP   JMP   OINT        ;INIT PARALEL PRINTER I/F
F803:4C B2 F9        OUTPAR  JMP   PARPR       ;PARALLEL PRINT OUTPUT
F806:4C 9B F9        OUTSER  JMP   SERPNT      ;SERIAL PRINTER OP
F809:4C 79 FE        OUTSCR  JMP   SCREEN      ;SCREEN OP
F80C:A9 0D           OUTRET  LDA   #$D         ;SAME AS OUTPCR,GIVE CR
F80E:4C 75 FE        OUTALL  JMP   OPCHR       ;SAME AS FE73,OUTPUT CH
F811:4C E9 FA        JMNRW   JMP   MMANG       ;MEM MANAGEMENT,NON-ICMT
F814:4C 0C FB        JMNRWI  JMP   MMINC       ;MEM MANG,AUTOINCMNT
F817:4C 28 FF        JHXPK   JMP   HEXPCK      ;HEXPACK
F81A:4C 0B FF        JHXPN   JMP   HEXPNT      ;HEX PRINTER
F81D:4C FA FD        JPLKB   JMP   POLLKB      ;POLL KB
F820:A2 FF           RETMS   LDX   #$FF        ;RESET STACK
F822:9A                      TXS
F823:4C 37 FC        RETMON  JMP   MONTOR      ;USER RETURN TO MONITOR
F826:4C 16 FB        JCURSN  JMP   CURSON      ;CURSOR ON
F829:4C 23 FB        JCURSF  JMP   CURSOF      ;CURSOR OFF
F82C:4C DD FE        JWASKB  JMP   WASKB       ;USER INPUT TO MONO
F82F:F8              SERNH   DFB   (SERNOX
F830:FC              SERNL   DFB   )SERNOX
F831:                ;C
F831:                ;C
F831:                ;THIS TABLE SETS ZPAGE AFTER
F831:                ;A RESET
F831:4C C3 FE        SETUP   DFB   $4C,$C3,$FE,$4C
F834:4C
F835:64 FF E0                DFB   $64,$FF,$E0,3,1,0,0,0,$40
F838:03 01 00
F83B:00 00 40
F83E:                ;C
F83E:                ;C
F83E:                ;PNSUP SETS UP PRINTER MODE
F83E:                ;AFTER INIT.SCREEN OP ON,BOTH
F83E:                ;PRINTERS OFF
F83E:                ;IT ALSO SAVES BAIC Z PAGE IF YOU
F83E:                ;WERE IN BASIC WHEN RESET HIT
F83E:24 00           PNSUP   BIT   PSTAT       ;BASIC WARM START SET?
F840:10 0A                   BPL   PNSUP1      ;NO;SKIP
F842:A2 4F                   LDX   #$4F        ;ELSE PUT $10-$5F ON STACK
F844:B5 10           PNSUP2  LDA   $10,X
F846:9D 00 01                STA   $100,X
F849:CA                      DEX
F84A:10 F8                   BPL   PNSUP2
F84C:                ;PNSUP1
```

```
F84C:A9 00        PNSUP1  LDA  #0          ;CLEAR STATUS OF PRINTERS
F84E:85 00                STA  PSTAT
F850:             ;SCREEN INH AND PRINT ON
F850:A9 8B                LDA  #$8B        ;SET UP SERIAL OPTION
F852:8D D2 BF             STA  SERCM
F855:A9 FF                LDA  #$FF
F857:8D D3 BF             STA  SERCN
F85A:A9 8B                LDA  #$8B
F85C:CD D2 BF             CMP  SERCM       ;IF CANT READ,NOT THERE.SKIP
F85F:D0 15                BNE  NORSP
F861:A9 93                LDA  #$93        ;ELSE FINISH SETUP
F863:8D D3 BF             STA  SERCN
F866:AD D1 BF             LDA  SERST       ;IS A KEYBOARD CONNECTED?
F869:29 60                AND  #$60
F86B:D0 09                BNE  NORSP       NO,SKIP
F86D:A9 89                LDA  #$89        ;ELSE ENABLE SERIAL INT
F86F:8D D2 BF             STA  SERCM
F872:A9 21                LDA  #$21        ;TURN SER PNR AND INPUT ON
F874:85 00                STA  PSTAT
F876:             ;NORSP
F876:A2 FF        NORSP   LDX  #$FF        ;SET STACK.X MUST BE
F878:9A                   TXS              ;FF ON EXIT
F879:4C 03 FC             JMP  OLSAR1      ;BACK TO OLD TBUG
F87C:             ;C
F87C:             ;C
F87C:             ;C
F87C:             ;IN TANBUG V2,OPCHR JUMPS HERE
F87C:             ;TO DETERMINE WHAT TYPE OF
F87C:             ;OUTPUT.A JUMP IS USED TO
F87C:             ;RETAIN COMPATIBILITY WHITH
F87C:             ;TANBUG V1.OCHAR HOLDS CHAR TO BE OP
F87C:             ;X,Y MUST BE SAVED
F87C:85 02        OPTYPE  STA  OCHAR       ;SAVE CHAR
F87E:8A                   TXA
F87F:48                   PHA
F880:98                   TYA
F881:48                   PHA
F882:             ;THIS PART LOOKS FOR CONTROL CODES
F882:AD 2F F8             LDA  SERNH       ;PUT SERNOF ADDR ON STACK
F885:48                   PHA
F886:AD 30 F8             LDA  SERNL       ;CAN THEN GET BACK VIA RTS
F889:48                   PHA
F88A:A6 02                LDX  OCHAR       ;GET CH IN X
F88C:A4 00                LDY  PSTAT       ;IS DEVISE CONTROL DC1 SET?
F88E:98                   TYA
F88F:29 04                AND  #$4
F891:D0 0A                BNE  LAB1        ;YES GO DEAL
F893:E0 11                CPX  #$11        ;ELE IS THIS DC CODE?
F895:D0 18                BNE  DVCN9       ;NO NORMAL ACN
F897:98                   TYA              ;ELSE SET FLAG
F898:09 04                ORA  #$4
F89A:85 00                STA  PSTAT       ;TO SAY DC
F89C:60                   RTS              ;AND JUMP TO SERNOF FROM STACK
F89D:98           LAB1    TYA
F89E:29 FB                AND  #$FB
F8A0:85 00                STA  PSTAT
F8A2:E0 06                CPX  #$6         ;IS IT A LEGAL DEVICE CODE?
F8A4:B0 09                BCS  DVCN9       ;NO, NORMAL OP
F8A6:BD 05 F9             LDA  HIDC,X      ;ELSE GET SERVICE ADDR
F8A9:48                   PHA
F8AA:BD 0B F9             LDA  LODC,X
F8AD:48                   PHA
F8AE:60                   RTS              ;JMP TO IT VIA RTS
F8AF:             ;GET HERE IF NORMAL PRINT
F8AF:A5 00        DVCN9   LDA  PSTAT       ;GET PRINT STATUS
F8B1:0A                   ASL  A
F8B2:0A                   ASL  A
F8B3:B0 0A                BCS  NOVDU       ;SKIP IF SCREEN DISABLED
```

```
F8B5:48          133          PHA                 ;ELSE SAVE PSTAT SHIFTED
F8B6:8A          134          TXA                 ;GET THE CHAR IN ACC
F8B7:C9 0A       135          CMP  #$A           ;NO ACTION IF LF
F8B9:F0 03       136          BEQ  NOVDU1
F8BB:20 79 FE    137          JSR  SCREEN        ;ELSE PRINT
F8BE:68          138 NOVDU1   PLA                 ;RECOVER STATUS
F8BF:A6 02       139 NOVDU    LDX  OCHAR         ;WAS SYMBOL DEL?
F8C1:E0 7F       140          CPX  #$7F
F8C3:D0 04       141          BNE  NPDL          ;NO
F8C5:A2 2F       142          LDX  #$2F          ;ELSE CHANGE TO SLASH
F8C7:86 02       143          STX  OCHAR
F8C9:            144 ;C
F8C9:0A          145 NPDL     ASL  A             ;IS SERIAL ON?
F8CA:90 0D       146          BCC  NOSRN         ;NO,SKIP
F8CC:48          147          PHA                 ;ELSE SAVE PRINT STATUS
F8CD:E0 0D       148          CPX  #$D           ;IS CHAR=CR?
F8CF:D0 04       149          BNE  NOSRN1        ;NO,SKIP
F8D1:29 10       150          AND  #$10          ;LF ONLY ON SERIAL?
F8D3:D0 03       151          BNE  NOSRN2        ;YES SKIP
F8D5:20 9B F9    152 NOSRN1   JSR  SERPNT        ;PRINT IT
F8D8:68          153 NOSRN2   PLA                 ;RECOVER STATUS
F8D9:0A          154 NOSRN    ASL  A             ;PARALLEL PRINT ON?
F8DA:90 0F       155          BCC  NOPAN         ;NO,SKIP
F8DC:48          156          PHA
F8DD:A6 02       157          LDX  OCHAR         ;CHAR=CR?
F8DF:E0 0D       158          CPX  #$D
F8E1:D0 04       159          BNE  NOPAN1        ;NO,SKIP
F8E3:29 20       160          AND  #$20          ;DONT PRINT UNLESS SPECIAL BIT SET
F8E5:F0 03       161          BEQ  NOPAN2
F8E7:20 B2 F9    162 NOPAN1   JSR  PARPR         ;PARALLEL PRINT SUBR
F8EA:68          163 NOPAN2   PLA
F8EB:0A          164 NOPAN    ASL  A             ;EXT PRINTER?
F8EC:90 03       165          BCC  NEXPR         ;NO,SKIP
F8EE:20 02 F9    166          JSR  EXTPNT        ;ELSE PRINT
F8F1:A6 02       167 NEXPR    LDX  OCHAR         ;WAS CHAR=CR?
F8F3:E0 0D       168          CPX  #$D
F8F5:D0 05       169          BNE  SERNOX        ;NO,SKIP
F8F7:A9 0A       170          LDA  #$A           ;ELSE GIVE A LF
F8F9:20 7C F8    171          JSR  OPTYPE        ;BY A RECURSIVE CALL
F8FC:60          172 SERNOX   RTS                 ;RETURN TO SERNOF
F8FD:            173 ;SERNOF MUST FOLLOW THIS LABEL
F8FD:            174 ;C
F8FD:68          175 SERNOF   PLA
F8FE:A8          176          TAY
F8FF:68          177          PLA
F900:AA          178          TAX
F901:60          179          RTS
F902:            180 ;C
F902:            181 ;EXTERNAL USER PRINTER SUBR -GOES VIA
F902:            182 ;INTSL2,3. USER MUST RETURN VIA RTS
F902:6C 11 00    183 EXTPNT   JMP  (INTSL2)
F905:            184 ;C
F905:            185 ;JMP TABLE FOR DC COMMANDS
F905:F9          186 HIDC     DFB  (SERON
F906:F9          187          DFB  (SEROFF
F907:F9          188          DFB  (PARINI
F908:F9          189          DFB  (PAROFF
F909:F9          190          DFB  (SCRON
F90A:F9          191          DFB  (SCROF
F90B:            192 ;C
F90B:            193 ;C
F90B:8D          194 LODC     DFB  )SERON-1
F90C:86          195          DFB  )SEROFF-1
F90D:40          196          DFB  )PARINI-1
F90E:39          197          DFB  )PAROFF-1
F90F:73          198          DFB  )SCRON-1
F910:7A          199          DFB  )SCROF-1
F911:            200 ;C
```

```
F911:                ;C
F911:                ;C
F911:                ;PRINTER VIA DEFS,2ND 6522
BFE1:                PRIAD   EQU  $BFE1      ;PRINTER DATA PORT A
BFE3:                PRIAE   EQU  $BFE3      ;DATA ENABLE
BFE0:                PRIBD   EQU  $BFE0      ;PRINTER STATUS DATA
BFE2:                PRIBE   EQU  $BFE2      ;AND EN
BFEC:                PRPCR   EQU  $BFEC      ;PCR
BFED:                PRIFR   EQU  $BFED
F911:                ;C
F911:                ;C
F911:                ;CONCHK IS ENTERED FROM THE
F911:                ;KB INT RUOT IN OLD TANBUG
F911:                ;IT CHECKS FOR CONTROL KEYS.
F911:                ;ACC PREVIOUSLY SAVED
F911:                ;MUSTNT USE X OR Y WITHOUT SAVING
F911:8D F0 BF        CONCHK  STA  KBINCL     ;CLEAR KB INT
F914:C9 10                   CMP  #$10       ;IS IT CTRLP(CHAR IN ACC)
F916:D0 06                   BNE  CTRLS      ;NO,TRY CTRL S
F918:20 34 F9                JSR  PARCH      ;ELSE SET PARALLEL PNTER
F91B:4C 2F F9                JMP  SPRN
F91E:C9 13           CTRLS   CMP  #$13       ;CTRLS?
F920:D0 06                   BNE  TRSPPP     ;NO
F922:20 6E F9                JSR  SCRCH      ;ELSE CHANGE STATUS OF SCREEN SUPP
F925:4C 2F F9                JMP  SPRN
F928:C9 16           TRSPPP  CMP  #$16       ;CTRLV,V24?
F92A:D0 07                   BNE  NOCTRL     ;NO,SKIP
F92C:20 81 F9                JSR  SERCH      ;ELSE SERIAL CHANGE
F92F:A9 00           SPRN    LDA  #$0        ;IF CONTROL,NO CHAR IP
F931:85 01                   STA  ICHAR
F933:60              NOCTRL  RTS
F934:                ;C
F934:                ;C
F934:                ;PARCH CHANGES STATE OF PARALLEL PRINTER
F934:A5 00           PARCH   LDA  PSTAT
F936:29 10                   AND  #$10       ;PRINTER ON?
F938:F0 07                   BEQ  PARINI     ;NO,TURN IT ON
F93A:                ;ELSE TURN OFF
F93A:A5 00           PAROFF  LDA  PSTAT
F93C:29 EF                   AND  #$EF
F93E:85 00           PARCH1  STA  PSTAT
F940:60                      RTS
F941:                ;PARINI INITS PAR PRINTER AND TURNS ON PARINI
F941:                ;C
F941:A9 00           PARINI  LDA  #0         ;USE DATA ENABLE TO SEE IF PRESENT
F943:8D E3 BF                STA  PRIAE
F946:CD E3 BF                CMP  PRIAE      IF CANT READ,NOT THERE
F949:D0 22                   BNE  OINT1      ;SO  IGNORE
F94B:A5 00                   LDA  PSTAT      ;SET THE ON BIT,PRINT STATUS
F94D:09 10                   ORA  #$10
F94F:85 00                   STA  PSTAT
F951:                ;INITS PRINTER,DOESNT AFFECT STAT
F951:A9 FF           OINT    LDA  #$FF       ;ELSE SET PORTS
F953:8D E3 BF                STA  PRIAE
F956:A9 FE                   LDA  #$FE
F958:8D E0 BF                STA  PRIBD      ;INIT LO,REST HIGH
F95B:A9 09                   LDA  #$9        ;SET OPS
F95D:8D E2 BF                STA  PRIBE
F960:A9 0F                   LDA  #$F        ;CA2 HI,CB1 POS TRIG
F962:8D E1 BF                STA  PRIAD
F965:20 DB F9                JSR  STBPA      ;CAUSE STROBE
F968:A9 FF                   LDA  #$FF       ;REMOVE INIT
F96A:8D E0 BF                STA  PRIBD
F96D:60              OINT1   RTS
F96E:                ;C
F96E:                ;C
F96E:                ;CHANGE STATE OF SCREEN SUPPRESS
F96E:A5 00           SCRCH   LDA  PSTAT      ;IS SCREEN SUPPRESSED?
```

```
F970:29 40      269          AND  #$40
F972:F0 07      270          BEQ  SCROF      ;NO,SUPPRESS
F974:A5 00      271 SCRON    LDA  PSTAT      ;ELSE UNSUPPRESS
F976:29 BF      272          AND  #$BF
F978:85 00      273 SCRON1   STA  PSTAT
F97A:60         274          RTS
F97B:A5 00      275 SCROF    LDA  PSTAT      ;INH IT HERE
F97D:09 40      276          ORA  #$40
F97F:D0 F7      277          BNE  SCRON1
F981:           278 ;C
F981:           279 ;C
F981:           280 ;CHANGE SERIAL PRINTER STATA
F981:A5 00      281 SERCH    LDA  PSTAT      ;STATUS
F983:29 20      282          AND  #$20       ;PRINTER ON?
F985:F0 07      283          BEQ  SERON      ;NO,TURN ON
F987:A5 00      284 SEROFF   LDA  PSTAT      ;TURN OFF
F989:29 DF      285          AND  #$DF
F98B:85 00      286 SEROF1   STA  PSTAT
F98D:60         287          RTS
F98E:AD D2 BF   288 SERON    LDA  SERCM      ;IF READS AS FF,NOT THERE
F991:C9 FF      289          CMP  #$FF       ;SO IGNORE
F993:F0 F2      290          BEQ  SEROFF
F995:A5 00      291          LDA  PSTAT
F997:09 20      292          ORA  #$20
F999:D0 F0      293          BNE  SEROF1     ;OR ON
F99B:           294 ;C
F99B:           295 ;C
F99B:           296 ;SERIAL PRINTER SUBR.R4GS NOT SAVED.
F99B:           297 ;CHAR TO BE OUTPUT IN OPCHR
F99B:A2 00      298 SERPNT   LDX  #$0        ;DELAY COUNT
F99D:AD D1 BF   299 SERPN2   LDA  SERST      ;XMIT BUFF EMPTY?
F9A0:29 10      300          AND  #$10       ;YES,GO OP DATA
F9A2:D0 08      301          BNE  OPSER      ;IF OK GO OUTPUT
F9A4:88         302 SERPN1   DEY             ;ELSE DELAY
F9A5:D0 FD      303          BNE  SERPN1
F9A7:CA         304          DEX
F9A8:D0 F3      305          BNE  SERPN2     ;GIVE TIME OUT IF FAILURE
F9AA:F0 3E      306          BEQ  PRERR
F9AC:A5 02      307 OPSER    LDA  OCHAR      ;GET DATA
F9AE:8D D0 BF   308          STA  SERDT      ;WRITE IT
F9B1:60         309          RTS
F9B2:           310 ;C
F9B2:           311 ;C
F9B2:           312 ;C
F9B2:           313 ;PARPR IS THE SERIAL PRINTER SUBR
F9B2:           314 ;ASSUMES PRINTER SET UP
F9B2:           315 ;ENTERED FROM TANBUG AS A RESULT
F9B2:           316 ;OF OPCHR.ASSUMES PRINTER INITTED
F9B2:           317 ;ACC NOT SAVED
F9B2:           318 ;CHAR IN OCHAR
F9B2:AD E0 BF   319 PARPR    LDA  PRIBD      ;ERROR(LOW)?
F9B5:29 04      320          AND  #$4
F9B7:F0 31      321          BEQ  PRERR      ;YES,GO ERROR
F9B9:A2 FF      322          LDX  #$FF       ;ELSE DELAY COUNT
F9BB:98         323 PABSY    TYA             ;PAUSE
F9BC:A0 0F      324          LDY  #$F
F9BE:88         325 DY       DEY
F9BF:10 FD      326          BPL  DY
F9C1:A8         327          TAY
F9C2:88         328          DEY             ;OUTSIDE LOOP
F9C3:D0 03      329          BNE  PABSY1     ;TO GIVE A TIME OUT
F9C5:CA         330          DEX
F9C6:F0 22      331          BEQ  PRERR      ;IF NOT OK IN 10 SEC
F9C8:AD E0 BF   332 PABSY1   LDA  PRIBD      ;PRINT BSY
F9CB:29 02      333          AND  #$2
F9CD:D0 EC      334          BNE  PABSY      ;YES,WAIT
F9CF:AD ED BF   335          LDA  PRIFR      ;ACK,RECEIVED?
```

```
F9D2:29 02      336          AND#$2
F9D4:F0 E5      337          BEQ  PABSY      ;NO,WAIT
F9D6:A5 02      338          LDA  OCHAR      ;ELSE WRITE DATA
F9D8:8D E1 BF   339          STA  PRIAD
F9DB:           340 ;INT IS CLEARED
F9DB:           341 ;STRAIGHT TO STROBE ROUT
F9DB:A9 0D      342 STBPA    LDA  #$D        ;STROBE LO
F9DD:8D EC BF   343          STA  PRPCR
F9E0:E9 01      344 STBPA1   SBC  #1         ;WAIT A FEW MU
F9E2:D0 FC      345          BNE  STBPA1
F9E4:A9 0F      346          LDA  #$F        ;STBE HI
F9E6:8D EC BF   347          STA  PRPCR
F9E9:60         348          RTS
F9EA:           349 ;C
F9EA:           350 ;C
F9EA:           351 ;C
F9EA:           352 ;PRINTER ERROR SUBR
F9EA:A5 00      353 PRERR    LDA  PSTAT      ;PRINTER OFF
F9EC:29 8F      354          AND  #$8F       ;AND SCREEN ON
F9EE:85 00      355          STA  PSTAT
F9F0:A2 0C      356          LDX  #$C        ;ERROR MSG
F9F2:BD FC F9   357 PRERR1   LDA  PREMS,X
F9F5:20 75 FE   358          JSR  OPCHR
F9F8:CA         359          DEX
F9F9:10 F7      360          BPL  PRERR1
F9FB:60         361          RTS
F9FC:0D         362 PREMS    DFB  $D
F9FD:52 4F 52   363          ASC  'RORRE     TNIRP'
FA00:52 45 20
FA03:54 4E 49
FA06:52 50
FA08:0D         364          DFB  $D
FA09:           365 ;CURDL MOVES THE CURSOR BACK ONE ON DEL
FA09:           366 ;ON ENTRY Y HOLDS POSN ON LINE
FA09:           367 ;CURSOR WILL ALSO MOVE UP
FA09:88         368 CURDL    DEY             ;DEC ITS OPOSN
FA0A:10 16      369          BPL  CURDL1     ;RET IF NOT S.O.L
FA0C:A5 0A      370          LDA  ICURS      ;ELSE UP ONE LINE
FA0E:38         371          SEC
FA0F:E9 20      372          SBC  #$20
FA11:85 0A      373          STA  ICURS
FA13:A5 0B      374          LDA  ICURS+1
FA15:E9 00      375          SBC  #0
FA17:85 0B      376          STA  ICURS+1
FA19:A0 1F      377          LDY  #$1F       ;Y TO END OF LINE
FA1B:C9 01      378          CMP  #1         ;OFF TOP OF SCN?
FA1D:D0 03      379          BNE  CURDL1     ;NO,RET
FA1F:20 23 FA   380          JSR  CLSC       ;ELSE CLEAR SCREEN
FA22:60         381 CURDL1   RTS             ;AND RETURN
FA23:           382 ;C
FA23:           383 ;C
FA23:           384 ;C
FA23:           385 ;CLSC CLEARS THE SCREEN AND RESETS
FA23:           386 ;THE POINTERS
FA23:A0 00      387 CLSC     LDY  #0
FA25:A9 20      388          LDA  #$20
FA27:99 00 02   389 CLSC1    STA  $200,Y
FA2A:99 00 03   390          STA  $300,Y
FA2D:C8         391          INY
FA2E:D0 F7      392          BNE  CLSC1
FA30:98         393          TYA
FA31:85 0A      394          STA  ICURS
FA33:A9 02      395          LDA  #2
FA35:85 0B      396          STA  ICURS+1
FA37:60         397          RTS
FA38:           398 ;C
FA38:           399 ;C
FA38:           400 ;C
```

```
FA38:          401 ;SCROLL SCROLLS THE SCREEN UP ONE
FA38:          402 ;AND RESETS PARAMS FOR BOT LINE
FA38:A2 00     403 SCROLL  LDX  #0
FA3A:BD 20 02  404 LOWBLK  LDA  VDUFST,X
FA3D:9D 00 02  405         STA  VDUSTT,X
FA40:E8        406         INX
FA41:D0 F7     407         BNE  LOWBLK
FA43:BD 20 03  408 HIBLK   LDA  VDUTOP,X
FA46:9D 00 03  409         STA  VDUMID,X
FA49:E8        410         INX
FA4A:E0 E0     411         CPX  #$E0
FA4C:D0 F5     412         BNE  HIBLK
FA4E:86 0A     413         STX  ICURS
FA50:A9 03     414         LDA  #3          ;RESETS ICURS
FA52:85 0B     415         STA  ICURS+1
FA54:A9 20     416         LDA  #$20        ;SPACED IN BOTTEM LINE
FA56:A8        417         TAY
FA57:88        418 MORSP   DEY
FA58:91 0A     419         STA  (ICURS),Y
FA5A:D0 FB     420         BNE  MORSP
FA5C:60        421         RTS              ;RET AND OP CIURSOR
FA5D:          422 ;C
FA5D:          423 ;C
FA5D:          424 ;C
FA5D:          425 ;SUBR STNGC COMPARES A COMMAND STRING
FA5D:          426 ;STARTING AT LINE START,WITH A STORED
FA5D:          427 ;STRING.ON EXIT CONDITION CODE EQ
FA5D:          428 ;IS SET IF THERE WAS A COMPARISON
FA5D:          429 ;STORED STRING TERM BY 0
FA5D:          430 ;ON ENTER X MUST BE SET TO OFFSET
FA5D:          431 ;ADDR OF SRING TO BE COMPARED.
FA5D:A0 00     432 STNGC   LDY  #0          ;START OF VDU LINE
FA5F:BD 6D FA  433 STNGC2  LDA  COMST,X
FA62:D0 01     434         BNE  STNGC3      ;IF NON-0,NO COMPARISON
FA64:60        435 STNGC1  RTS              ;ELSE EXIT COMP
FA65:D1 0A     436 STNGC3  CMP  (ICURS),Y ;COMPARE WITH INPUT
FA67:D0 FB     437         BNE  STNGC1      ;IF NON-0,NO COMPARISON
FA69:E8        438         INX
FA6A:C8        439         INY
FA6B:D0 F2     440         BNE  STNGC2
FA6D:          441 ;C
FA6D:          442 ;C
FA6D:          443 ;MESSAGES FOR STRING COMPARISON
FA6D:42 41 53  444 COM1    ASC  'BAS'       ;BASIC COLD START
FA70:00        445         DFB  $0
FA71:57 41 52  446 COM2    ASC  'WAR'       ;BASIC WARM
FA74:00        447         DFB  $0
FA6D:          448 COMST   EQU  COM1
0000:          449 BASIC   EQU  COM1-COMST
0004:          450 WARM    EQU  COM2-COMST
FA75:          451 ;C
FA75:          452 ;C
FA75:          453 ;C
FA75:          454 ;TRCMDS IS ENTERED VIA A JUMP WHEN
FA75:          455 ;TANBUG V1 INSTR PARSE FAILS.
FA75:          456 ;JUMPED TO FROM ERRQ
FA75:          457 ;IT LOOKS FOR 3 LETTER CMDS
FA75:          458 ;NOTE IT IS TRIED BEFORE XBUG CMDS
FA75:          459 ;IF YOU TRY TO ENTER BASIC AND ITS NOT
FA75:          460 ;THERE THEN YOU CRASH!!!
FA75:          461 ;C
E2ED:          462 BASRT   EQU  $E2ED       ;BASIC COLD START
C34B:          463 BSWARM  EQU  $C34B       ;BASIC WARM
C558:          464 STKINI  EQU  $C558       ;BASIC REINIT SUBR.
FA75:A2 00     465 TRCMDS  LDX  #BASIC      ;BAS CMD STRING
FA77:20 5D FA  466         JSR  STNGC       ;EQUALITY?
FA7A:D0 08     467         BNE  TRCMD1      ;NO,SKIP
```

```
FA7C:06 00      468          ASL  PSTAT      ;ELSE SET IN BASIC FLAG
FA7E:38         469          SEC
FA7F:66 00      470          ROR  PSTAT
FA81:4C ED E2   471          JMP  BASRT      ;BASIC START
FA84:A2 04      472 TRCMD1   LDX  #WARM      ;TRY WAR CMD
FA86:20 5D FA   473          JSR  STNGC
FA89:D0 15      474          BNE  TRCMD2     ;NO,SKIP
FA8B:A2 4F      475          LDX  #$4F       ;ELSE RECOVER BASIC Z PAGE
FA8D:BD 00 01   476 TRCMD3   LDA  $100,X
FA90:95 10      477          STA  $10,X
FA92:CA         478          DEX
FA93:10 F8      479          BPL  TRCMD3
FA95:06 00      480          ASL  PSTAT      ;SET BASIC WARM START BIT
FA97:38         481          SEC
FA98:66 00      482          ROR  PSTAT
FA9A:20 58 C5   483          JSR  STKINI     ;INIT BASIC PARAMS
FA9D:4C 4B C3   484          JMP  BSWARM     ;GO DO WARM START
FAA0:4C F7 F7   485 TRCMD2   JMP  INPERR
FAA3:           486 ;--------------------------------
FAA3:           487 ;ENDS UP AT RETERR(NO XBUG) OR F7F7(XBUG)
FAA3:           488 ;C
FAA3:           489 ;SUBR TRYSR1 IS EXECUTED FROM THE INT
FAA3:           490 ;HANDLER KBINT.IN ADDITION TO LOOKING
FAA3:           491 ;AT THE MICROTAN KB PORT IT LOOKS AT THE
FAA3:           492 ;SER INPUT PORT IF ON,AND OBTAINS DATA
FAA3:           493 ;THE NEG BIT IN PSW IS SET AND CHAR IN
FAA3:           494 ;ACC IF KB OR TTY.USING THE MICRON KB
FAA3:           495 ;TURNS OFF SER TTY INPUT
FAA3:A5 00      496 TRYSRI   LDA  PSTAT      ;ALLOWING SERIAL I/P?
FAA5:29 01      497          AND  #$1
FAA7:F0 1A      498          BEQ  TRYSR1     ;NO,GO TRY THE KB
FAA9:AD D1 BF   499          LDA  SERST      ;SERIAL INTERRUPT?
FAAC:10 15      500          BPL  TRYSR1     ;NO,GO TRY KB
FAAE:08         501          PHP             ;ELSE SAVE STATUS
FAAF:AD D0 BF   502          LDA  SERDT      ;AND DATA
FAB2:48         503          PHA
FAB3:AD D2 BF   504          LDA  SERCM      ;CLEAR INTERUPT
FAB6:48         505          PHA
FAB7:09 02      506          ORA  #$2
FAB9:8D D2 BF   507          STA  SERCM
FABC:68         508          PLA
FABD:8D D2 BF   509          STA  SERCM      ;REENABLE INT
FAC0:68         510 TRYSR3   PLA             ;RECOVER CHARAC
FAC1:28         511          PLP             ;AND NEG BIT SET
FAC2:60         512 TRYSR2   RTS             ;RETURN
FAC3:AD F3 BF   513 TRYSR1   LDA  KBREAD     ;TRY MICROTAN KB
FAC6:10 FA      514          BPL  TRYSR2     ;WASNT,EXIT PSW POS
FAC8:08         515          PHP             ;ELSE SAVE VALUES
FAC9:48         516          PHA
FACA:A5 00      517          LDA  PSTAT      ;IS SERIAL INPUT ENABLED?
FACC:29 01      518          AND  #$1
FACE:F0 F0      519          BEQ  TRYSR3     ;NO,RETURN
FAD0:A9 00      520          LDA  #0         ;ELSE DISABLE
FAD2:85 00      521          STA  PSTAT
FAD4:A9 8B      522          LDA  #$8B       ;AND DISABLE INTERRUPTS
FAD6:8D D2 BF   523          STA  SERCM
FAD9:D0 E5      524          BNE  TRYSR3     ;UNCONDITIONAL TO RETURN
FADB:           525 ;C
FADB:           526 ;C
FADB:           527 ;PADINI IS CALLED FROM STARTUP SEQUENCE
FADB:           528 ;AND DISABLES THE KEYPAD IF A TTY
FADB:           529 ;INPUT IS ENABLED
FADB:8D F0 BF   530 PADINI   STA  KBINCL     ;CLEAR KB INTERUPT
FADE:A5 00      531          LDA  PSTAT      ;IS SERIAL INT ENABLED?
FAE0:29 01      532          AND  #$1
FAE2:F0 04      533          BEQ  PADIN1     ;NO,SKIP
FAE4:A9 00      534          LDA  #0         ;ELSE DISABLE KEYPAD
FAE6:85 0F      535          STA  SIMCOM
```

```
FAE8:60          536 PADIN1   RTS
FAE9:            537 ;C
FAE9:            538 ;C
FAE9:            539 ;C
FAE9:            540 ;MEM MANAGEMENT SUBR
0040:            541 MEMSEG   EQU  $40        ;READ AND WRITE SEGMENTS
0041:            542 MEMDAW   EQU  $41        ;DATA TO BE WRITTEN
0042:            543 MEMDAR   EQU  $42        ;DATA READ
0043:            544 MEMLO    EQU  $43        ;LO BYTE ADDR
0044:            545 MEMHI    EQU  $44        ;HI BYTE ADDR
FFFF:            546 MNGST    EQU  $FFFF
FAE9:            547 ;NONE OF THIS ARE CORRUPTED EXCEPT BY
FAE9:            548 ;USE OF INPUT IN BASIC OR YOUR OWN CODE
FAE9:48          549 MMANG    PHA              ;SAVE ACC AND Y
FAEA:98          550          TYA
FAEB:48          551          PHA
FAEC:A5 40       552          LDA  MEMSEG      ;GET MAPPING INFO
FAEE:8D FF FF    553          STA  MNGST       ;SET STATUS
FAF1:A0 00       554          LDY  #0
FAF3:48          555          PHA              ;SAVE MANAGEMENT SETTING
FAF4:29 F0       556          AND  #$F0
FAF6:F0 04       557          BEQ  MMANG1      ;NO,SKIP
FAF8:B1 43       558          LDA  (MEMLO),Y   ;GET DATA
FAFA:85 42       559          STA  MEMDAR      ;AND STORE
FAFC:68          560 MMANG1   PLA
FAFD:29 0F       561          AND  #$F         ;IS A WRITE REQD?
FAFF:F0 04       562          BEQ  MMANG2      ;NO,SKIP
FB01:A5 41       563          LDA  MEMDAW      ;ELSE WRITE
FB03:91 43       564          STA  (MEMLO),Y
FB05:8C FF FF    565 MMANG2   STY  MNGST       ;RESET PAGE 0
FB08:68          566          PLA
FB09:A8          567          TAY
FB0A:68          568          PLA
FB0B:60          569          RTS
FB0C:            570 ;C
FB0C:            571 ;C
FB0C:            572 ;MMINC CALLS THE ABOVE,THEN INCS MEM ADDR
FB0C:20 E9 FA    573 MMINC    JSR  MMANG
FB0F:E6 43       574          INC  MEMLO
FB11:D0 02       575          BNE  MMINC1
FB13:E6 44       576          INC  MEMHI
FB15:60          577 MMINC1   RTS
FB16:            578 ;SUBROUTINES TO TURN CURSUR ON/OFF
FB16:            579 ;TURNS OFF AT CURRENT POSN
FB16:48          580 CURSON   PHA              ;TURN ON
FB17:98          581          TYA
FB18:48          582          PHA
FB19:A9 FF       583          LDA  #$FF
FB1B:A4 03       584 CURSN1   LDY  VDUIND
FB1D:91 0A       585          STA(ICURS),Y
FB1F:68          586          PLA
FB20:A8          587          TAY
FB21:68          588          PLA
FB22:60          589          RTS
FB23:48          590 CURSOF   PHA
FB24:98          591          TYA
FB25:48          592          PHA
FB26:A9 20       593          LDA  #$20
FB28:D0 F1       594          BNE  CURSN1
FB2A:            595 ;C
FB2A:            596 ;C
FB2A:            597 ;C
FB2A:            598 ;C
FB2A:            599 ;C
FB2A:            600          CHN  OTBUG
FB2A:              1 ;OTBUG-OLD TANBUG V1
FB2A:              2 ;TANBUG PROGRAM
FB2A:              3 ;ALL LOCATION ADDRESSES AFTER THIS POINT
FB2A:              4 ;MUST BE PRESERVED IN FUTURE VERSIONS
```

```
FB2A:                ;C
FB2A:                ;ZERO PAGE DEFS
FB2A:                ;C
FB2A:                ;C
0004:                INTFS1   EQU  $4          ;FAST INTERRUPT LINK
0005:                INTFS2   EQU  $5
0006:                INTFS3   EQU  $6
0007:                NMIJP    EQU  $7          ;NON MASK INT LINK
0008:                NMIJP1   EQU  $8
0009:                NMIJP2   EQU  $9
000B:                ICURSH   EQU  $B
000C:                RUNIND   EQU  $C          ;0 IF RUNNING USER PROG
000D:                SINGLE   EQU  $D          ;NON-0 IF SINGLE INSTR
000E:                PROCED   EQU  $E          ;USED FOR PN TIMES
0010:                INTSL1   EQU  $10         ;SLOW INT LINK
0012:                INTSL3   EQU  $12
0013:                HXPKL    EQU  $13         ;RESULT OF HEXPCK HOLDS NEX
0014:                HXPKH    EQU  $14         ;NO DOUBLE PEC
0015:                PCLBCK   EQU  $15         ;PSEUDO REG STORE.PCH
0016:                PCHBCK   EQU  $16         ;PCH
0017:                PSWBCK   EQU  $17         ;PSW
0018:                SPBCK    EQU  $18         ;SP
0019:                XBCK     EQU  $19         ;X
001A:                YBCK     EQU  $1A         ;Y
001B:                ABCK     EQU  $1B         ;A
001C:                MODADL   EQU  $1C         ;TEMP STORE
001D:                MODADH   EQU  $1D
001E:                COPL     EQU  $1E         ;TEMP USED IN COPY,AS IN MODODL
001F:                COPH     EQU  $1F
0020:                BPTLO    EQU  $20         ;BPT STATUS TABLE
0021:                BPTHI    EQU  $21
0030:                BPTCOD   EQU  $30         ;BPT CODE STORE
FB2A:                ;C
FB2A:                ;DEFINE NAMES OF SPECIAL ADDRESSES
0100:                STKBSE   EQU  $100        ;STACK BASE
0200:                VDUSTT   EQU  $200        ;VDU LOWEST ADDRESS
0220:                VDUFST   EQU  $220
0300:                VDUMID   EQU  $300
0320:                VDUTOP   EQU  $320
03E0:                LINBOT   EQU  $3E0        ;BOTTEM LINE
BFF0:                SGRAPH   EQU  $BFF0       ;GRAPHICS SET CLEAR/ADDRESS
BFF0:                KBINCL   EQU  $BFF0
BFF1:                SNMI     EQU  $BFF1       ;SET UP TO GIVE NMI FOR SINGLE
BFF2:                KBWRIT   EQU  $BFF2       ;WRITE TO KB
BFF3:                KBREAD   EQU  $BFF3       ;READ FROM KB
BFF3:                STEXT    EQU  $BFF3
F7F7:                INPERR   EQU  $F7F7       ;ERROR EXIT FOR XBUG
FB2A:                ;C
FB2A:                ;C
FB2A:                ;C
FB2A:                ;START OF MAIN PROG,JUMPED TO
FB2A:                ;FROM RESET VECTOR.THIS CODE
FB2A:                ;LOOKS WIERD BECAUSE IT IS
FB2A:                ;PATCHED SO THAT ADDRESSES
FB2A:                ;REMAIN COMPATIBLE FOR PROGRAMS
FB2A:                ;WRITTEN WITH TANBUG V1
FB2A:                ;THE RESET VECTOR
FB2A:                ;LOCATE AT $FC00

FC00:                         ORG  $FC00
FC00:                ;ALSO CHECK FOR/SAVE DTAT FROM WARM START
FC00:4C 3E F8        OLSAR    JMP  PNSUP       ;SET PRINTER MODE,STACK
FC03:E8              OLSAR1   INX
FC04:86 17                    STX  PSWBCK      ;INIT THE STORED PSW
FC06:20 B7 FF                 JSR  BPTCLR      ;CLEAR THE BREAKPOINTS
FC09:8D F3 BF                 STA  STEXT       ;SET TEXT CLEAR GRAPHICS
FC0C:A2 0C                    LDX  #$C         ;SET UP AN INDEX
FC0E:BD 31 F8        SETUP1   LDA  SETUP,X     ;AND SET UP ZERO PAGE
FC11:95 04                    STA  INTFS1,X    ;STORE BY COPYING FROM
```

```
FC13:CA          73          DEX              ;THE TABLE SET UP
FC14:10 F8       74          BPL  SETUP1
FC16:            75 ;NOW DETERMINE KB TYPE AND SET SIMCOM FLG
FC16:E8          76          INX              ;MAKE IX EQ 0
FC17:8E F2 BF    77 TSFIV    STX  KBWRIT      ;CLEAR KB WRITE LATCH
FC1A:8D F0 BF    78          STA  KBINCL      ;CLEAR KB INT FLAG
FC1D:CA          79          DEX
FC1E:8E F2 BF    80          STX  KBWRIT      ;TURN ON ALL KB LINES
FC21:E8          81          INX              ;IX EQ 0
FC22:AD F3 BF    82          LDA  KBREAD      ;READ THE KB.IF JUMPER NOT
FC25:10 02       83          BPL  KPCPLX      ;IN THEN ASCII
FC27:E6 0F       84          INC  SIMCOM      ;ELSE KB SO SET FLAG
FC29:20 DB FA    85 KPCPLX   JSR  PADINI      ;DISABLE KP IF SER I/P
FC2C:BD EB FF    86 TBMS     LDA  HDR,X       ;PRINT TANBUG
FC2F:F0 06       87          BEQ  MONTOR
FC31:20 75 FE    88          JSR  OPCHR
FC34:E8          89          INX
FC35:D0 F5       90          BNE  TBMS
FC37:            91 ;C
FC37:            92 ;THE MONITOR LOOP
FC37:D8          93 MONTOR   CLD              ;SET BINARY MODE
FC38:58          94          CLI              ;ENABLE INTERRUPTS
FC39:20 FA FD    95          JSR  POLLKB      ;WAIT FOR A KEY TO BE HIT
FC3C:A5 01       96          LDA  ICHAR       ;GET THE CHAR
FC3E:C9 21       97          CMP  #$21        ;IF <=SPACE,ITS A TERMINATOR
FC40:30 06       98          BMI  MONCH1      ;SO GO DO SOMETHING
FC42:20 75 FE    99 ISTERM   JSR  OPCHR       ;ELSE PRINT THE CHAR
FC45:4C 37 FC   100          JMP  MONTOR      ;THEN LOOK FOR NEXT
FC48:20 4F FC   101 MONCH1   JSR  MONEN2      ;CALL STRING PROCESSOR
FC4B:A9 0D      102 RC1      LDA  #$D         ;SET UP ACC WITH CAR RET
FC4D:D0 F3      103          BNE  ISTERM      ;AND OUTPUT CR AFTER THE SUBR
FC4F:           104 ;C
FC4F:           105 ;STRING PROCESSING SUBROUTINE
FC4F:A0 00      106 MONEN2   LDY  #0
FC51:B1 0A      107          LDA  (ICURS),Y   ;SET CHAR AT START OF BOT LINE
FC53:AA         108          TAX
FC54:C8         109          INY              ;LOOK AT NEXT CHAR
FC55:B1 0A      110          LDA  (ICURS),Y
FC57:10 36      111          BPL  MULTI       ;IF PLUS A STRING
FC59:A9 00      112          LDA  #0          ;ELSE A STRING LETTER,ZERO ACC
FC5B:E0 53      113          CPX  #'S         ;WAS IT SINGLE?
FC5D:D0 03      114          BNE  TRYN        ;NO,SKIP
FC5F:86 0D      115          STX  SINGLE      ;ELSE SET SINGLE MODE
FC61:60         116          RTS
FC62:E0 4E      117 TRYN     CPX  #'N         ;NORMAL?
FC64:D0 03      118          BNE  TRYP
FC66:85 0D      119          STA  SINGLE
FC68:60         120          RTS
FC69:           121          CHN  TBUGA
FC69:             1 ;TBUGA
FC69:E0 50        2 TRYP     CPX  #'P         ;P WITH NO ARG?
FC6B:D0 04        3          BNE  TRYR
FC6D:85 0E        4          STA  PROCED      ;YES,CLEAR PROCED COUNT
FC6F:F0 56        5          BEQ  PROC1       ;UNCONDITIONAL BR TO PROC LBL
FC71:E0 52        6 TRYR     CPX  #'R         ;REGISTER EXAM/MOD?
FC73:D0 09        7          BNE  TRYB
FC75:85 1D        8          STA  MODADH      ;IF SO SET ADDRESS 15 HEX
FC77:A9 15        9          LDA  #$15
FC79:85 1C       10          STA  MODADL
FC7B:4C E5 FD    11          JMP  REOPEN      ;GO OFF TO MODIFY
FC7E:E0 42       12 TRYB     CPX  #'B         ;B TO REMOVE BPTS?
FC80:D0 04       13          BNE  ERRQ        ;NO,ERROR
FC82:20 B7 FF    14          JSR  BPTCLR      ;ELSE CLEAR THEM
FC85:60          15          RTS
FC86:            16 ;C
FC86:4C 75 FA    17 ERRQ     JMP  TRCMDS      ;GO TRY CMD STRING
FC89:A9 3F       18 RETERR   LDA  #'?         ;AND AN ERROR RETURN IS HERE
FC8B:20 75 FE    19          JSR  OPCHR       ;PRINT Q MARK
```

```
FC8E:60          20 PRERC   RTS
FC8F:            21 ;C
FC8F:            22 ;WE GET HERE IF THE COMMAND WAS A DATA BEARER
FC8F:88          23 MULTI   DEY              ;ADJ IY
FC90:8A          24         TXA              ;SAVE COMMAND ON STACK
FC91:48          25         PHA
FC92:20 28 FF    26         JSR  HEXPCK      ;PACK THE ARGUMENT
FC95:D0 54       27         BNE  MOREY       ;IF ANOTHER ARG,SKIP
FC97:68          28         PLA              ;ELSE PULL CMD
FC98:50 EC       29         BVC  ERRQ        ;ERROR IF ILLEGAL ARG
FC9A:C9 47       30         CMP  #'G         ;G FOR GO?
FC9C:D0 21       31         BNE  TRYP1
FC9E:A2 00       32         LDX  #0          ;IF SO CLEAR PROCED COUNT
FCA0:86 0E       33         STX  PROCED
FCA2:CA          34         DEX              ;X-FF
FCA3:9A          35 GOEND   TXS              ;RESET STACK
FCA4:A5 14       36         LDA  HXPKH
FCA6:48          37         PHA              ;SET UP STACK TO START OF PORDG A
FCA7:A5 13       38         LDA  HXPKL
FCA9:48          39         PHA
FCAA:A5 17       40         LDA  PSWBCK      ;STACK PSW
FCAC:48          41         PHA
FCAD:C6 0C       42         DEC  RUNIND      ;SET RUN FLAG
FCAF:A9 20       43         LDA  #$20        ;OBLITERATE CURSOR
FCB1:A4 03       44         LDY  VDUIND
FCB3:91 0A       45         STA  (ICURS),Y
FCB5:A6 19       46         LDX  XBCK        ;SET UP REGISTERS
FCB7:A4 1A       47         LDY  YBCK
FCB9:A5 1B       48 SRET    LDA  ABCK
FCBB:8D F1 BF    49         STA  SNMI        ;SET NMI TO BRK ON GO
FCBE:40          50         RTI              ;USE RTI TO INIT THE PROGRAM
FCBF:C9 50       51 TRYP1   CMP  #'P         ;IS IT P WITH ARG?
FCC1:D0 11       52         BNE  TRYM
FCC3:A5 13       53         LDA  HXPKL       ;IF SO SET PROCEED COUNT
FCC5:85 0E       54         STA  PROCED
FCC7:A5 16       55 PROC1   LDA  PCHBCK      ;RESTORE USER PC
FCC9:85 14       56         STA  HXPKH
FCCB:A5 15       57         LDA  PCLBCK
FCCD:85 13       58         STA  HXPKL
FCCF:A6 18       59         LDX  SPBCK       ;SET IX TO USER SP
FCD1:4C A3 FC    60         JMP  GOEND       ;THEN AS FOR GO
FCD4:C9 4D       61 TRYM    CMP  #'M         ;IS IT MOD MEM
FCD6:D0 AE       62         BNE  ERRQ        ;NO,ERROR
FCD8:A9 2C       63 EQPT    LDA  #',         ;OUTPUT A COMMA
FCDA:20 75 FE    64         JSR  OPCHR
FCDD:A0 00       65         LDY  #0          ;PICK UP WHATS IN THE ADRESS
FCDF:B1 13       66         LDA  (HXPKL),Y
FCE1:20 0B FF    67         JSR  HEXPNT      ;PRINT IT
FCE4:68          68         PLA
FCE5:68          69         PLA
FCE6:A9 2C       70         LDA  #',         ;PRINT ANOTHER COMMA
FCE8:4C 42 FC    71         JMP  ISTERM      ;VIA TERMINATING CODE
FCEB:            72 ;C
FCEB:            73 ;WE GET HERE IF MORE THAN ONE ARG
FCEB:E0 2C       74 MOREY   CPX  #',         ;WAS A COMMA BETWEEN ARGS?
FCED:F0 04       75         BEQ  GETPT2      ;YES CONTINUE
FCEF:68          76 LINKPH  PLA              ;ELSE PULL CMD
FCF0:4C 86 FC    77 LINKR   JMP  ERRQ        ;AND GIVE ERROR
FCF3:A5 13       78 GETPT2  LDA  HXPKL       ;STORE FIRST PARAM
FCF5:85 1C       79         STA  MODADL
FCF7:A5 14       80         LDA  HXPKH
FCF9:85 1D       81         STA  MODADH
FCFB:20 28 FF    82         JSR  HEXPCK      ;PACK SECOND PARAM
FCFE:D0 7C       83         BNE  MOREY1      ;NOT CURSOR,SO THERES A 3RD PARAM
FD00:68          84         PLA               ELSE PULL CMD
FD01:50 ED       85         BVC  LINKR       ;NO ARG AFTER COMMA ERROR
FD03:C9 4C       86         CMP  #'L         ;L FOR LIST
FD05:F0 42       87         BEQ  LISTIT      ;YES BRANCH
```

```
FD07:C9 4F        88          CMP  #'O        ;O FOR OFFSET?
FD09:D0 27        89          BNE  TRYBPT
FD0B:A5 13        90          LDA  HXPKL      ;ELSE GET BRANCH OFFSET
FD0D:38           91          SEC
FD0E:E9 02        92          SBC  #2         ;ADJUST FOR THE HARDWARE
FD10:B0 02        93          BCS  NOTOPO
FD12:C6 14        94          DEC  HXPKH      ;DOUBLE PRECISION
FD14:38           95 NOTOPO   SEC             ;NOW SUB SOURCE FROM DESD
FD15:E5 1C        96          SBC  MODADL
FD17:AA           97          TAX
FD18:A5 14        98          LDA  HXPKH      ;DP
FD1A:E5 1D        99          SBC  MODADH
FD1C:A8          100          TAY
FD1D:8A          101          TXA
FD1E:30 05       102          BMI  RNGNG
FD20:98          103          TYA
FD21:D0 CD       104          BNE  LINKR      ;WE NEED TO TEST FOR OVERANGE
FD23:F0 03       105          BEQ  PNTITO     ;IF OK PRINT IT
FD25:C8          106 RNGNG    INY
FD26:D0 C8       107          BNE  LINKR
FD28:A9 3D       108 PNTITO   LDA  #'=        ;PRINT =
FD2A:20 75 FE    109          JSR  OPCHR
FD2D:8A          110          TXA
FD2E:20 0B FF    111          JSR  HEXPNT     ;AND THE NO
FD31:60          112          RTS
FD32:C9 42       113 TRYBPT   CMP  #'B        ;B FOR SET BRKPT?
FD34:D0 BA       114 LINK1    BNE  LINKR      ;NO ILL CMD
FD36:A5 13       115          LDA  HXPKL      ;CHECK LEGAL BPT NO
FD38:30 B6       116          BMI  LINKR
FD3A:C9 08       117          CMP  #8
FD3C:10 B2       118          BPL  LINKR
FD3E:0A          119          ASL  A          ;DOUBLE NO AND USE FOR INDEX
FD3F:AA          120          TAX
FD40:A5 1C       121          LDA  MODADL     ;STORE BPT ADRESS
FD42:95 20       122          STA  BPTLO,X
FD44:A5 1D       123          LDA  MODADH
FD46:95 21       124          STA  BPTHI,X
FD48:60          125 JPRTRN   RTS
FD49:            126 ;NXLI
FD49:            127          CHN  TBUGB
FD49:              1 ;TBUGB
FD49:20 73 FE      2 LISTIT   JSR  OUTPCR     ;LIST CMD..OP CR
FD4C:A0 00         3          LDY  #0
FD4E:A5 1D         4          LDA  MODADH     ;PRINT ADRESS
FD50:20 0B FF      5          JSR  HEXPNT
FD53:A5 1C         6          LDA  MODADL
FD55:20 0B FF      7 NXLIST   JSR  HEXPNT
FD58:A9 20         8          LDA  #$20       ;OUTPUT A SPACE
FD5A:20 75 FE      9          JSR  OPCHR
FD5D:B1 1C        10          LDA  (MODADL),Y ;GET CONTENTS OF LOC
FD5F:C8           11          INY             ;INC INDEX
FD60:C0 09        12          CPY  #9         ;8 WORDS PER LINE
FD62:30 F1        13          BMI  NXLIST
FD64:C6 13        14          DEC  HXPKL      ;DEC REQUESTED LINE COUNT
FD66:F0 E0        15          BEQ  JPRTRN     ;UNTIL 0,WHEN 0 END
FD68:88           16 DELX1    DEY             ;ELSE A DELAY BETWEEN EACH LINE
FD69:D0 FD        17          BNE  DELX1
FD6B:CA           18          DEX
FD6C:D0 FA        19          BNE  DELX1
FD6E:A5 1C        20          LDA  MODADL     ;NOW ADD 8 TO ACCESS ADDRESS
FD70:18           21          CLC
FD71:69 08        22          ADC  #8
FD73:85 1C        23          STA  MODADL
FD75:90 D2        24          BCC  LISTIT     ;DP ADDRESS
FD77:E6 1D        25          INC  MODADH
FD79:4C 49 FD     26          JMP  LISTIT     ;THEN NEXT LINE
FD7C:             27 ;C
```

```
FD7C:                ;IF PROG GETS HERE CMD HAS 3 PARAMS
FD7C:E0 2C           MOREY1  CPX  #',        ;COMMA SEPERATES?
FD7E:F0 03                   BEQ  TERMOK
FD80:4C EF FC        ERJUM2  JMP  LINKPH     ;ERROR IF NOT
FD83:A5 13           TERMOK  LDA  HXPKL      ;STORE PARAM IN COPL AND H
FD85:85 1E                   STA  COPL
FD87:A5 14                   LDA  HXPKH
FD89:85 1F                   STA  COPH
FD8B:20 28 FF                JSR  HEXPCK     ;PACK NEW PARAM
FD8E:D0 F0                   BNE  ERJUM2     ;TOUGH-CANT HAVE >3 PARAMS
FD90:68                      PLA             ;GET CMD
FD91:C9 4D                   CMP  #'M        ;M FOR MODEM?
FD93:F0 27                   BEQ  MEM100     ;YES SKIP
FD95:C9 43                   CMP  #'C        ;C FOR COPY
FD97:D0 9B                   BNE  LINK1      ;NO CRASH
FD99:50 99                   BVC  LINK1      ;CHECK PARAM LEGALITY
FD9B:A0 00                   LDY  #0         ;IF SO EXECUTE COPY CMD
FD9D:B1 1C           NXCOP   LDA  (MODADL),Y ;PICK UP THE BYTE
FD9F:91 13                   STA  (HXPKL),Y  ;SAVE IT TO ITS DEST.
FDA1:A5 1F                   LDA  COPH       ;DO A COMPARE TO SEE IF DONE
FDA3:C5 1D                   CMP  MODADH
FDA5:D0 06                   BNE  ICMCOP     ;DP
FDA7:A5 1E                   LDA  COPL
FDA9:C5 1C                   CMP  MODADL
FDAB:F0 5B                   BEQ  ENDLS      ;IF DONE END
FDAD:E6 1C           ICMCOP  INC  MODADL     ;ELSE INC SOURCE
FDAF:D0 02                   BNE  NOHIH1     ;DP
FDB1:E6 1D                   INC  MODADH
FDB3:E6 13           NOHIH1  INC  HXPKL      ;AND DEST
FDB5:D0 E6                   BNE  NXCOP
FDB7:E6 14                   INC  HXPKH
FDB9:D0 E2                   BNE  NXCOP      ;STOPS IF YOU ROLL OVER TOP
FDBB:60                      RTS
FDBC:                ;C
FDBC:                ;MOD MEM CODE
FDBC:A6 01           MEM100  LDX  ICHAR      ;CHAR LAST TYPED IN X
FDBE:E0 20                   CPX  #$20       ;WAS IT A SPACE?
FDC0:F0 23                   BEQ  REOPEN     ;YES, REOPEN THIS LOCATION
FDC2:50 06                   BVC  NOENT      ;IF NOT TYPED GOTO ERROR
FDC4:A5 13                   LDA  HXPKL      ;ELSE GET TYPED DATA
FDC6:A0 00                   LDY  #0
FDC8:91 1C                   STA  (MODADL),Y ;STUFF IT IN THE LOC
FDCA:E0 0A           NOENT   CPX  #$A        ;LINE FEED TYPED?
FDCC:F0 11                   BEQ  WASLF      ;YES,SKIP
FDCE:E0 1B                   CPX  #$1B       ;WAS IT ESC?
FDD0:D0 36                   BNE  ENDLS      ;NO,RETURN
FDD2:                ;ELSE DEC MOD MEM ADDRESS
FDD2:C6 1C           ESCIT   DEC  MODADL
FDD4:A5 1C                   LDA  MODADL     ;DOUBLE PREC
FDD6:C9 FF                   CMP  #$FF
FDD8:D0 0B                   BNE  REOPEN
FDDA:C6 1D                   DEC  MODADH
FDDC:4C E5 FD                JMP  REOPEN
FDDF:E6 1C           WASLF   INC  MODADL     ;IF WAS LF INC MODEM ADR
FDE1:D0 02                   BNE  REOPEN
FDE3:E6 1D                   INC  MODADH
FDE5:20 73 FE        REOPEN  JSR  OUTPCR     ;GIVE CR
FDE8:A9 4D                   LDA  #'M        ;PRINT NEW ADDRESS
FDEA:20 75 FE                JSR  OPCHR
FDED:A5 1D                   LDA  MODADH
FDEF:20 0B FF                JSR  HEXPNT
FDF2:A5 1C                   LDA  MODADL
FDF4:20 0B FF                JSR  HEXPNT
FDF7:4C 4F FC                JMP  MONEN2     ;GO WAIT COMMAND,MISS CR
FDFA:                ;C
FDFA:                ;C
FDFA:                ;C
```

```
FDFA:                 CHN  TSUB
FDFA:                ;TANBUG SUBROUTINES FILE TSUB
FDFA:                ;C
FDFA:                ;C
FDFA:                ;POLLKB SUBROUTINES.USES SIMCOM TO SEE
FDFA:                ;WHICH TYPE OF KB.WAITS UNTIL A KEY HIT
FDFA:                ;THEN EXITS WITH ITS ASCII CODE IN ICHAR
FDFA:                ;ACC AND REGISTERS ARE CORRUPTED
FDFA:A9 00           POLLKB  LDA  #$0        ;PUSH A SHIFT INDICATOR
FDFC:48                      PHA
FDFD:85 01           PLKB1   STA  ICHAR      ;SET ICHAR TO 0
FDFF:C5 0F                   CMP  SIMCOM    ;SKIP IF KEYPAD
FE01:D0 06                   BNE  SIMPLE
FE03:C5 01           WAIT1   CMP  ICHAR      ;ASCII-WAIT FOR INTERRUPT
FE05:F0 FC                   BEQ  WAIT1     ;TO MODIFY ICHAR
FE07:68              PLKEND  PLA             ;POP
FE08:60              ENDLS   RTS
FE09:                ;KEYPAD CODING
FE09:A9 0F           SIMPLE  LDA  #$F        ;ENABLE KB DRIVE LINES
FE0B:8D F2 BF                STA  KBWRIT
FE0E:AD F3 BF                LDA  KBREAD    ;EXAM INPUTS
FE11:D0 F6                   BNE  SIMPLE    ;IF A KEY DOWN WAIT FOR IT TO GO
FE13:A2 40                   LDX  #$40      ;DEBOUNCE KEY GOING UP
FE15:88              DEBOUN  DEY
FE16:D0 FD                   BNE  DEBOUN
FE18:CA                      DEX
FE19:D0 FA                   BNE  DEBOUN
FE1B:                ;NOW POLL THE KB PROPERLY
FE1B:A0 FF           PLK1    LDY  #$FF       ;SET Y TO INDEX ASCII TBL
FE1D:68                      PLA
FE1E:48                      PHA             ;SHIFT ASSERTED
FE1F:F0 02                   BEQ  NOSHIF
FE21:A0 13                   LDY  #$13       ;IF SO MOD THE INDEX
FE23:A2 08           NOSHIF  LDX  #8         ;SET X TO DRIVE WTE LINES
FE25:8E F2 BF        PLK2    STX  KBWRIT     ;BLAT
FE28:AD F3 BF                LDA  KBREAD    ;KEY DOWN THIS LINE?
FE2B:D0 0C                   BNE  ACHAR     ;YES GO DEAL
FE2D:C8                      INY
FE2E:C8                      INY
FE2F:C8                      INY
FE30:C8                      INY
FE31:C8                      INY            ;ELSE INC Y BY 5
FE32:8A                      TXA            ;AND SHIFT X RIGHT
FE33:4A                      LSR  A
FE34:AA                      TAX
FE35:F0 E4                   BEQ  PLK1      ;IF SCAN DONE RESET,AGAIN
FE37:D0 EC                   BNE  PLK2      ;ELSE CONTINUE SCAN
FE39:                ;IF WE GET HERE A KEY WAS PRESSED
FE39:C8              ACHAR   INY
FE3A:4A                      LSR  A         ;DETERMINE WHICH OF 5 KEYS THIS IS
FE3B:90 FC                   BCC  ACHAR     ;KEEP ROTATING TILL FOUND
FE3D:B9 4B FE                LDA  CHRTBL,Y  ;GET ASCII EQUIV
FE40:85 01                   STA  ICHAR
FE42:D0 C3                   BNE  PLKEND    ;IF WAS 0 WAS SHIFT KEY
FE44:68                      PLA            ;SO PULL AND CHANGE STATE
FE45:49 FF                   EOR  #$FF
FE47:48                      PHA            ;SHUV AGAIN
FE48:4C 09 FE                JMP  SIMPLE    ;AND REPEAT POLL
FE4B:                ;C
FE4B:                ;CHARACTER LOOKUP TBL
FE4B:33 37 42        CHRTBL  DFB  $33,$37,$42,$46,$3F,$32
FE4E:46 3F 32
FE51:36 41 45                DFB  $36,$41,$45,$D,$31,$35,$39,$44,$A
FE54:0D 31 35
FE57:39 44 0A
FE5A:30 34 38                DFB  $30,$34,$38,$43,$0,$2C,$52,$4C
FE5D:43 00 2C
FE60:52 4C
```

```
FE62:4E 3F 32    63          DFB   $4E,$3F,$32,$43,$49,$53,$20,$31,$4F
FE65:43 49 53
FE68:20 31 4F
FE6B:1B 47 7F    64          DFB   $1B,$47,$7F,$30,$34,$50,$4D,$0
FE6E:30 34 50
FE71:4D 00
FE73:            65 ;C
FE73:            66 ;C
FE73:            67 ;OUTPUT CHARACTER SUBROUTINE.X AND Y SAVED
FE73:            68 ;CHAR TO BE PRINTED IN A ON ENTRY
FE73:            69 ;THE OUTPCR GIVES A CR
FE73:A9 0D       70 OUTPCR  LDA   #$D        ;SET UP A FOR CR
FE75:4C 7C F8    71 OPCHR   JMP   OPTYPE     ;GO TO OUTPUT ROUTINE
FE78:EA          72         NOP              ;ADDRESS FIXER
FE79:            73 ;SCREEN IS NOW THE SUBROUTINE WHICH OPS
FE79:            74 ;TO THE SCREEN.NO REGISTERS SAVED
FE79:            75 ;CHAR IN OCHAR ON ENTRY
FE79:            76 ;C
FE79:85 02       77 SCREEN  STA   OCHAR
FE7B:A4 03       78         LDY   VDUIND     ;GET CURSOR POSN
FE7D:A9 20       79         LDA   #$20
FE7F:91 0A       80         STA   (ICURS),Y  ;OBLIT CURS
FE81:A6 02       81         LDX   OCHAR
FE83:E0 0C       82         CPX   #$C
FE85:D0 0A       83         BNE   TRYDEL
FE87:20 23 FA    84 SCREN3  JSR   CLSC       ;CLEAR SCREEN
FE8A:A9 FF       85 DICURS  LDA   #$FF       ;RESTORE CURSOR AND DISPLAY
FE8C:91 0A       86         STA   (ICURS),Y
FE8E:84 03       87         STY   VDUIND     ;SAVE VDU INDEX
FE90:60          88         RTS              ;AND RETURN
FE91:E0 7F       89 TRYDEL  CPX   #$7F       ;DELETE?
FE93:D0 05       90         BNE   TRYCR      ;NO GO TRY CR
FE95:20 09 FA    91         JSR   CURDL      ;ELSE DELETE
FE98:10 F0       92         BPL   DICURS     ;UNCOND TO DIS CURSOR
FE9A:E0 0D       93 TRYCR   CPX   #$D        ;CR?
FE9C:F0 08       94         BEQ   SCTY       ;YES SKIP
FE9E:8A          95         TXA              ;ELSE OP CHAR
FE9F:91 0A       96         STA   (ICURS),Y
FEA1:C8          97         INY              ;INC PTR
FEA2:C0 20       98         CPY   #$20       ;CHECK LINE ROLLOVER
FEA4:D0 E4       99         BNE   DICURS     ;NOT, SO OP CURSOR
FEA6:A9 20      100 SCTY    LDA   #$20       ;ELSE ADD 20 TO LINE PTR
FEA8:18         101         CLC
FEA9:65 0A      102         ADC   ICURS
FEAB:85 0A      103         STA   ICURS
FEAD:A5 0B      104         LDA   ICURS+1
FEAF:69 00      105         ADC   #0
FEB1:85 0B      106         STA   ICURS+1
FEB3:A0 00      107         LDY   #0         ;RESET Y
FEB5:C9 04      108         CMP   #4         ;OFF END OF SCREEN
FEB7:D0 D1      109         BNE   DICURS     ;NO,CURS OP
FEB9:20 38 FA   110         JSR   SCROLL     ;ELSE SCROLL
FEBC:F0 CC      111         BEQ   DICURS     ;UNCOND
FEBE:           112 ;C
FEBE:           113 ;C

FEC3:           114         ORG   $FEC3      ;COMPAT TANBUG V1
FEC3:           115 ;INTERRUPT SUBROUTINE
FEC3:           116 ;THIS IS ENTERED FROM A INTERRUPT
FEC3:           117 ;VIA THE JUMP STORED IN LOC INTFS1
FEC3:           118 ;SO CAN BE MODIFIED
FEC3:           119 ;A RESET ALWAYS POINTS INTFS1 HERE
FEC3:           120 ;INTSL1 CAN BE USED TO SERVICE
FEC3:           121 ;INTS AFTER KB DEALT WITH
FEC3:           122         CHN   TSUBA
FEC3:             1 ;TSUBA
FEC3:             2 ;SAVE ACC
```

```
FEC3:48          KBINT   PHA                 ;SAVE ACC
FEC4:D8                  CLD                 ;SET BIN MODE
FEC5:8A                  TXA                 ;SAVE X
FEC6:48                  PHA
FEC7:BA                  TSX
FEC8:E8                  INX                 ;GET SP IN X,USE TO ADDRESS OLD PS
FEC9:E8                  INX
FECA:E8                  INX
FECB:BD 00 01            LDA  STKBSE,X       ;GET PSW
FECE:29 10               AND  #$10           ;SEE IF IT WAS BRK INSTR
FED0:D0 1E               BNE  BRKP           ;IF SO GO PROCESS
FED2:68                  PLA                 ;ELSE PULL AND RESET IX
FED3:AA                  TAX
FED4:20 A3 FA            JSR  TRYSRI         ;SKIP TO TRY SERIAL INPUT ETS
FED7:30 04               BMI  WASKB          ;YEAH,GO PROCESS
FED9:            ;ELSE ANOTHER BACK VIA INTSL1
FED9:            ;NOTE INTSL1 HOLDS RTI UNLESS USER MODS
FED9:68          USER    PLA
FEDA:4C 10 00            JMP  INTSL1
FEDD:29 7F       WASKB   AND  #$7F           ;KB,MASK OFF TOP BIT
FEDF:85 01               STA  ICHAR          ;PUT IN ICHAR
FEE1:20 11 F9            JSR  CONCHK         ;TBUG V2 CHECK PRINT CONT
FEE4:C9 1B               CMP  #$1B           ;WAS IT ESC KEY
FEE6:D0 F1               BNE  USER
FEE8:A5 0C               LDA  RUNIND         ;IF SO,IS A USER
FEEA:D0 ED               BNE  USER           ;RUNNING HIS PROG?
FEEC:85 0E               STA  PROCED         ;YES,CLEAR PROCEED COUNT
FEEE:F0 12               BEQ  ACTBP          ;AND CAUSE A BREAKPOINT
FEF0:            ;BREAKPOINT PROCES CODE
FEF0:            ;CAUTION IF USING THIS WITH USER INTS
FEF0:            ;DO NOT SET A BP AT ONE OF YOUR OWN
FEF0:            ;BREAK INSTRS
FEF0:E8          BRKP    INX                 ;ADJUST PROG COUNTER TO
FEF1:38                  SEC                 ;RETURN TO THE RIGHT PLACE
FEF2:BD 00 01            LDA  STKBSE,X       ;AFTER BREAK PROCEED INSTR
FEF5:E9 02               SBC  #2             ;TO DO THIS SUBTRACT 2 FROM THE
FEF7:9D 00 01            STA  STKBSE,X       ;STACKED PC VALUE
FEFA:B0 04               BCS  NOROL1         ;DOUBLE PRECISION
FEFC:E8                  INX
FEFD:DE 00 01            DEC  STKBSE,X
FF00:68          NOROL1  PLA                 ;NOW PULL YOUR SAVED X VALUE
FF01:AA                  TAX
FF02:68          ACTBP   PLA
FF03:85 1B               STA  ABCK           ;GIVE IT TO USER
FF05:20 C1 FF            JSR  BPTREM         ;REMOVE BPTS
FF08:4C 75 FF    NOROL   JMP  NMNT1          ;SERVICE AS FOR SING INST
FF0B:            ;C
FF0B:            ;HEXPNT TAKES A HEX VALUE IN ACC AND
FF0B:            ;PRINTS AS 2 CHARS.A AND X DESTROYED
FF0B:48          HEXPNT  PHA                 ;SAVE CHAR VALUE
FF0C:A2 01               LDX  #1             ;SET X
FF0E:4A                  LSR  A
FF0F:4A                  LSR  A
FF10:4A                  LSR  A
FF11:4A                  LSR  A              ;GET 1ST CHAR
FF12:18          PNT2    CLC
FF13:69 30               ADC  #$30           ;ADD 30 FOR ASC11 FIGS CONV
FF15:C9 3A               CMP  #$3A           ;SKIP IF NOT A FIG
FF17:30 03               BMI  PNT1
FF19:18                  CLC
FF1A:69 07               ADC  #7             ;IF A LETTER ADD MORE FOR ASCII
FF1C:20 75 FE    PNT1    JSR  OPCHR          ;OP IT
FF1F:CA                  DEX
FF20:10 01               BPL  MOR1           ;DO THE SECOND
FF22:60                  RTS                 ;UNLESS DONE,THEN RETURN
FF23:68          MOR1    PLA                 ;SEC PART,REC CHAR
FF24:29 0F               AND  #$F            ;CLEAR OUT 1ST CHAR BITS
FF26:10 EA               BPL  PNT2           ;UNCOND
```

```
FF28:              ;C
FF28:              ;HEXPCK STORES HEX VALUES OF ENTERED NO
FF28:              ;UP TO 4 DIGITS IN HXPKL,H
FF28:              ;IY POINTS TO ONE BEFORE CHAR AT START
FF28:              ;REG CORRUPT.EXIT CAUSED BY NONHEX
FF28:              ;CHAR.ON EXIT IY POINTS TO LAST CHAR
FF28:              ;ON BOT VDU LINE,ZERO FLAG CLEAR IF
FF28:              ;CURSOR TERMINATED,V FLAG SET IF
FF28:              ;HEX DATA WAS ENCOUNTERED
FF28:A9 00         HEXPCK  LDA  #0
FF2A:48                    PHA              ;PSEUDO PSW
FF2B:85 13                 STA  HXPKL       ;CLEAR RESULT
FF2D:85 14                 STA  HXPKH
FF2F:C8            NXHX    INY              ;INC INDEX,GET CHAR
FF30:B1 0A                 LDA  (ICURS),Y
FF32:AA                    TAX              ;STORE IN X
FF33:38                    SEC              ;SUB HEX 30,ASCII FIGS TO HEX
FF34:E9 30                 SBC  #$30
FF36:30 10                 BMI  ENDTS       ;NONHEX END
FF38:C9 0A         HX1     CMP  #$A         ;SEE IF 0 TO 9
FF3A:30 10                 BMI  HX2         ;YES PACK
FF3C:38                    SEC              ;ELSE TRY HEX LETTERS
FF3D:E9 11                 SBC  #$11
FF3F:30 07                 BMI  ENDTS       ;ILLEGAL CHAR
FF41:18            HX3     CLC              ;ADD 9
FF42:69 0A                 ADC  #$A
FF44:C9 10                 CMP  #$10        ;AND CHECK ABOVE ASCII F
FF46:30 04                 BMI  HX2         ;OK
FF48:28            ENDTS   PLP              ;ELSE END,PULL PSEUDOPSW
FF49:E0 FF                 CPX  #$FF        ;SET Z FOR CURSOR
FF4B:60                    RTS
FF4C:              ;C
FF4C:              ;C
FF4C:              ;CHAR IN IX ON EXIT
FF4C:A2 04         HX2     LDX  #4          ;IT IS HEX,SHIFT OLD LEFT
FF4E:06 14         HX5     ASL  HXPKH       ;4 TIMES DP
FF50:06 13                 ASL  HXPKL
FF52:90 02                 BCC  HX4
FF54:E6 14                 INC  HXPKH
FF56:CA            HX4     DEX
FF57:D0 F5                 BNE  HX5
FF59:18                    CLC              ;ADD IN THE NEW CHAR
FF5A:65 13                 ADC  HXPKL
FF5C:85 13                 STA  HXPKL
FF5E:68                    PLA              ;GET PSEUD PSW
FF5F:A9 40                 LDA  #$40        ;SET V BIT
FF61:48                    PHA
FF62:D0 CB                 BNE  NXHX        ;UNCOND
FF64:              ;C
FF64:              ;C
FF64:              ;NONMASKABLE INTERRUPT ROUTINE
FF64:                      CHN  TSUBB
FF64:              ;TSUBB
FF64:              ;DEALS WITH SINGLE INSTR MODE
FF64:85 1B         NMNT    STA  ABCK        ;SAVE ACC FOR OPERATOR
FF66:D8                    CLD              ;BIN MODE
FF67:A5 0D                 LDA  SINGLE      ;IN SINGLE INST MODE
FF69:D0 0A                 BNE  NMNT1       ;YES GO DO
FF6B:              ;IF NOT COULD BE AFTER A GO OR
FF6B:              ;PROCEED,SO SET UP ANY USER BKPTS
FF6B:              ;IN THE PROGRAM,THE N RETURN WITHOUT
FF6B:              ;ENABLING THE NMI
FF6B:8A                    TXA              ;SAVE X
FF6C:48                    PHA
```

```
FF6D:20 D6 FF   14          JSR  BPSET      ;SET UP BPS
FF70:68         15          PLA
FF71:AA         16          TAX             ;REST X
FF72:A5 1B      17          LDA  ABCK       ;RESTORE ACC
FF74:40         18          RTI             ;RETURN
FF75:A5 0E      19 NMNT1    LDA  PROCED     ;ELSE IS SINGLE GET PROCE
FF77:F0 05      20          BEQ  ZERBCK     ;COUNT,BREAK IF 0
FF79:C6 0E      21          DEC  PROCED     ;ELSE DEC IT
FF7B:4C B9 FC   22          JMP  SRET       ;AND CONTINUE,SETTING NMI
FF7E:E6 0C      23 ZERBCK   INC  RUNIND     ;CLEAR USER RUN
FF80:68         24          PLA             ;POP PSW
FF81:85 17      25          STA  PSWBCK     ;UPDATE USER PSEUDO REG
FF83:68         26          PLA
FF84:85 15      27          STA  PCLBCK     ;DO FOR ALL REGS
FF86:68         28          PLA
FF87:85 16      29          STA  PCHBCK
FF89:86 19      30          STX  XBCK
FF8B:84 1A      31          STY  YBCK
FF8D:BA         32          TSX
FF8E:86 18      33          STX  SPBCK
FF90:           34 ;PRINT ALL THESE REGS
FF90:20 73 FE   35 PSEUDO   JSR  OUTPCR
FF93:A5 16      36          LDA  PCHBCK
FF95:20 0B FF   37          JSR  HEXPNT
FF98:A5 15      38          LDA  PCLBCK
FF9A:20 0B FF   39          JSR  HEXPNT
FF9D:A0 00      40          LDY  #0
FF9F:A9 20      41 PSNX     LDA  #$20       ;STEP BY SPACES
FFA1:48         42          PHA
FFA2:20 75 FE   43          JSR  OPCHR
FFA5:68         44          PLA
FFA6:20 75 FE   45          JSR  OPCHR
FFA9:B9 17 00   46          LDA  PSWBCK,Y
FFAC:20 0B FF   47          JSR  HEXPNT
FFAF:C8         48          INY
FFB0:C0 05      49          CPY  #5
FFB2:30 EB      50          BMI  PSNX
FFB4:4C 4B FC   51          JMP  RC1        ;THEN GO TO MONITOR,USER INPUT
FFB7:           52 ;C
FFB7:           53 ;BREAKPOINT CLEARING ROUTINE USES
FFB7:           54 ;IX AND A
FFB7:A2 1F      55 BPTCLR   LDX  #$1F       ;SET A COUNT
FFB9:A9 00      56          LDA  #0
FFBB:           57 ;WRITE IN ALL ZERO'S
FFBB:95 20      58 BPTCL1   STA  BPTLO,X
FFBD:CA         59          DEX
FFBE:10 FB      60          BPL  BPTCL1
FFC0:60         61          RTS
FFC1:           62 ;C
FFC1:           63 ;THIS SUBR REMOVES BPS BY WRITING
FFC1:           64 ;BACK USER'S CODE,IX,XI USED
FFC1:8A         65 BPTREM   TXA
FFC2:48         66          PHA
FFC3:A2 0E      67          LDX  #$E        ;SET COUNT
FFC5:B5 20      68 BPTRM1   LDA  BPTLO,X    ;IF 0 NO ACTION
FFC7:15 21      69          ORA  BPTLO+1,X
FFC9:F0 04      70          BEQ  BPTRM3
FFCB:B5 30      71 BPTRM2   LDA  BPTCOD,X   ;ELSE WRITE IT TO PROG
FFCD:81 20      72          STA  (BPTLO,X)
FFCF:           73 ;C
FFCF:CA         74 BPTRM3   DEX
FFD0:CA         75          DEX
FFD1:10 F2      76          BPL  BPTRM1
FFD3:68         77          PLA
FFD4:AA         78          TAX
```

```
FFD5:60          79          RTS
FFD6:            80 ;BPSET LOOKS AT STORED BPT ADDRESSES
FFD6:            81 ;AND STORES PRESENT INSTR,THEN 0(BRK) TO
FFD6:            82 ;PROG MEM
FFD6:A2 0E       83 BPSET    LDX   #$E        ;SET INDEX FOR 8
FFD8:B5 21       84 BPS1     LDA   BPTLO+1,X  ;NO ACN IF
FFDA:15 20       85          ORA   BPTLO,X    ;BPT ZERO
FFDC:F0 08       86          BEQ   BPS3
FFDE:A1 20       87          LDA   (BPTLO,X)  ;ELSE GET USERS INST
FFE0:95 30       88          STA   BPTCOD,X   ;SAVE IT
FFE2:A9 00       89          LDA   #0         ;SET X FOR BRK
FFE4:81 20       90          STA   (BPTLO,X)  ;SHUV IN PROG CODE
FFE6:CA          91 BPS3     DEX
FFE7:CA          92          DEX              ;INDEX ADJ
FFE8:10 EE       93          BPL   BPS1       ;DO FOR ALL
FFEA:60          94          RTS              ;NOTE THIS CORRUPTS LOC 0
FFEB:            95 ;FOR UNUSED BPT NOS.
FFEB:            96 ;C
FFEB:            97 ;TANBUG MG
FFEB:0D          98 HDR      DFB   $D
FFEC:54 41 4E    99          ASC   'TANBUG'
FFEF:42 55 47
FFF2:0D 00      100          DFB   $D,0
FFF4:           101 ;C
FFF4:           102 ;ERROR LINKIN
----- NEXT OBJECT FILE NAME IS TV2AB.OBJ3
FFF7:           103          ORG   $FFF7
FFF7:4C 89 FC   104          JMP   RETERR
FFFA:           105 ;C
FFFA:           106 ;VECTORS FOR NMI,INT,RESET
FFFA:07 00 00   107          DFB   7,0,0,$FC,4,0
FFFD:FC 04 00
0000:           108 ;C
0000:           109 ;.END TBUG

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```