```
                    1 *
                TTL   RALBUG V1.4

 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 *           RALBUG V1.4 6809 COLOUR MONITOR       *
 * BOARD ADDRESS $E000. MODIFIED FOR 81 CHARACTERS  *
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

 * THIS IS THE MONITOR FOR A 6809 TANGERINE

 * FINAL RUN ADDRESS IS $F800
 *
 * NOTE DISC DRIVE ROUTINE COME UP TO $DFB0 SO USER STACK MUST
 * NOT EXTEND INTO THIS AREA, IF YOU MUST HAVE A LARGER STACK
 * RELOCATE INTO LOWER RAM.
 *
 * Two points that may not be obvious :-
 *
 * the DISPLA routine is first vectored into RAM at $DFD2 (REDIRCT label)
 * and is returned into the eprom via a JUMP instruction at the above
 * address. This is to allow you to intercept any character or control
 * code going to the display for your own purposes if you so wish.
 * Next the NMI vector at $FFFC also points into RAM (as they all do) at
 * $DFFD before re-entering the eprom. This allows you to intercept any
 * keypress for your own purposes. It is assumed you know what you are
 * doing if you mess with these.
 *
 * Notes on special codes, see CODADD label to see special colour codes
 * needed if sent through the DISPLA routine.
 * this was necessary to allow the original codes to be used for their
 * more normal use, such as $01 for HOME cursor $0C for CLEAR SCREEN etc
 * If you are writting directly to the screen you will have to use the
 * original codes example $01 for RED. Study how the colour codes are
 * set up on ENTRY. By changing colour bytes in RAM at $DFDD-$DFE3
 * you can easily control the colour that the monitor eprom sends out.
 *
 *
 * * START OF VARIABLE EQUATES STATEMENTS
 *
 * COL+1=RED,+2=GREEN,+3=YELLOW,+4=BLUE,+5=MAGENTA,+6=CYAN,+7=WHITE
 *
0000  DIRPAG   EQU   $00       * DIRECT PAGE POINTER
00BF  I_O      EQU   $BF       * I_O AREA
0051  LINOFF   EQU   $51       * LINE LENGTH HEX
DFFD  NMI      EQU   $DFFD     * NON MASKABLE INTERUPT
DFFA  SWI      EQU   $DFFA     * B & C SOFT WARE INERTERUPT
DFF7  IRQ      EQU   $DFF7     * 8 & 9 INTERUPT REQUEST
DFF4  FIRQ     EQU   $DFF4     * 5 & 6 FAST INTERUPT REQUEST
DFF1  SWI2     EQU   $DFF1     * 2 & 3 SOFTWARE INTERUPT 2
DFEE  SWI3     EQU   $DFEE     * F & 0 SOFTWARE INTERUPT 3
DFED  STAT     EQU   $DFED     * RESERVED FOR SYSTEM STATUS
DFEC  TEMP3    EQU   $DFEC     * RESERVED FOR SYSTEM USE
DFEB  TEMP2    EQU   $DFEB     * AS ABOVE
DFEA  TEMP1    EQU   $DFEA     * AS ABOVE
DFE9  ADDLOW   EQU   $DFE9     * AS ABOVE
DFE8  ADDHI    EQU   $DFE8     * TEMPERY STORE USED BY MONITOR
DFE6  USAVE    EQU   $DFE6     * STORE TO SAVE USER STACK POINTER
DFE4  CURSOR   EQU   $DFE4     * STORE FOR CURSOR POSITION
DFDC  COLL     EQU   $DFDC     * NOTE COL USES FROM COL+1 TO +7
```

```
DFDB                    61 KBFLG    EQU   $DFDB     * INC IF KEYBOARD IS TOUCHED
DFD9                    62 TBLEEP   EQU   $DFD9     * TERMINAL BLEEPER TIME DELAY
DFD8                    63 ICHAR    EQU   $DFD8     * TEMP STORE FOR KEYBOARD CHARACTER
DFD6                    64 KBLEEP   EQU   $DFD6     * K/BOARD BLEEPER TIME DELAY
DFD5                    65 KBFLAG   EQU   $DFD5     * K/BOARD BLEEPER FLAG
DFD2                    66 REDIRCT  EQU   $DFD2     * REDIRECT FOR DISPLAY ROUTINE ALLOWS INTERCEPTION
DFD0                    67 SSTACK   EQU   $DFD0     * SYSTEM STACK POINTER
BC00                    68 USTACK   EQU   $BC00     * USER STACK POINTER
BFF3                    69 KBREAD   EQU   $BFF3     * KEYBOARD INPUT PORT
BFF0                    70 KBCLR    EQU   $BFF0     * KEYBOARD CLEAR
BF95                    71 BLEEP    EQU   $BF95     * BLEEPER ADDRESS ON OUR DOS BOARD
E000                    72 SCRTOP   EQU   $E000     * TOP OF SCREEN REAL ADDRESS
E798                    73 SCRBOT   EQU   $E798     * BOTTOM OF SCREEN
E7E8                    74 SCREND   EQU   $E7E8     * END OF SCREEN BOTTOM LINE
F001                    75 EXTENS   EQU   $F001     * DOWNLOADED EPROM START ADDRESS
F000                    76 EPMARK   EQU   $F000     * EPROM NUMBER STORE
0020                    77 EPROM1   EQU   $20       * 1st EPROM ADDRESS = $2000
0028                    78 EPROM2   EQU   $28       * 2nd EPROM ADDRESS = $2800
0030                    79 EPROM3   EQU   $30       * 3rd EPROM ADDRESS = $3000
                        80 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                        81
                        82 * THESE EQU APPLY TO THE DATA V1.4 EPROM
                        83
F004                    84 INVAL    EQU   $F004
F006                    85 HELPAS   EQU   $F006
F008                    86 VECTO    EQU   $F008
F005                    87 GOCOL    EQU   $F005     * EPROM $2800
                        88
                        89 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                        90 *
                        91 * * * * * * USEFULL SUBROUTINES * * * * * * *
                        92 *
                        93 * HEXDIS display characters at address in 'X' as 2 hex characters
                        94 * HEXA   display 'A' as 2 hex characters
                        95 * SPAHEX as hexdis but outputs space first
                        96 * DISX   displays 'X' 4 ASCII characters
                        97 * SPACEX   as disx but display a space first
                        98 * CLRSCR clear screen
                        99 * CURSA  display the system marker @=
                       100 * SCROLL scroll screen
                       101 * HEX    test 'A' for an ASCII hex value
                       102 * GETHEX get ASCII 30-39, 41-46
                       103 * GET2   get 2 hex codes into 'A'
                       104 * GET4   gets 4 hex codes into 'X' & address HI & LO
                       105 * GET8   get start address in 'Y' end address in 'X'
                       106 * VASCII verify if ASCII
                       107 * PD     print D as 2 ASCII
                       108 * INPUT  check if ESC L/F or ascii character
                       109 * DISPLA display 'A' as one ascii character
                       110 * STRING output ascii string pointed to by 'X' until $04 terminater
                       111 * is found.
                       112 * * * * * * * * * * * * * * * * * * * * * * * * * *
                       113 *
                       114 * * * START OF MONITOR PROPER *
FFF0                   115         ORG   $FFF0     * RESET ETC JUMP TABLE
                       116 *
                       117 *
FFF0 FB5A              118         FDB   RESET     THIS GIVES FINAL EPROM ADDRESS
FFF2 DFEE              119         FDB   SWI3      * SWI3
FFF4 DFF1              120         FDB   SWI2      * SWI2
```

```
FFF6 DFF4                121          FDB   FIRQ       % FIRQ
FFF8 DFF7                122          FDB   IRQ        % IRQ
FFFA DFFA                123          FDB   SWI        % SWI
FFFC DFFD                124          FDB   NMI        % NMI USED FOR KEYBOARD
FFFE F85A                125          FDB   RESET      % RESET ENTRY POINT
                         126
F800                     127          ORG   $F800
                         128
F800 F85A                129 URESET   FDB   RESET      % [$F800] ENTRY POINT ON RESET
F802 FBBA                130 RENTR    FDB   WARM       % [$F802] MONITOR RENTRY ADDRESS
F804 FBF8                131 UUPOLL   FDB   UPOLL      % [$F804] POLL THE KEYBOARD NO DISPLAY
F806 FBF3                132 UPDISPL  FDB   PDIS       % [$F806] POL KGBOARD AND DISPLAY
F808 FDF5                133 UCHIN    FDB   CINPUT     % [$F808] CHECK IF KEYBOARD HAS BEEN TOUCHED
F80A DFD2                134 UDIS     FDB   REDIRCT    % [$F80A] DISPLAY USERS 'A' AS ASCII
F80C F855                135 USTRIN   FDB   STRING     % [$F80C] OUTPUT ASCII STRING POINTED TO BY 'X'
                         136 %                          % UNTILL AN $04 IS ENCOUNTERED
F80E F8C7                137 UCR      FDB   CR         % [$F80E] OUTPUT A C/R FOR A NEW LINE
F810 FDEF                138 UPSTR    FDB   NEWSTR     % [$F810] C/R AND PRINT STING AS ABOVE
F812 FE85                139 UPULL    FDB   PULL       % [$F812] CHECK $F000 AGAINST 'A' DOWNLOAD ()
F814 FA36                140 UOUT2H   FDB   HEXDIS     % [$F814] OUTPUT 2 HEX CHAR POINTED TO BY 'X'
F816 F8B2                141 UOUT4H   FDB   DISX       % [F816] OUTPUT 4 HEX CHAR FROM 'X'
F818 FA7F                142 UIN2HE   FDB   GET2       % [F818] INPUT 2HEX CHAR INTO 'A'
F81A FA8D                143 UIN4HE   FDB   GET4       % [F81A] INPUT 4 HEX CHAR INTO 'X'
F81C FE08                144 UIN8HE   FDB   GET8       % [F81C] INPUT 8 HEX CHAR INTO 'Y'&'X'
F81E F863                145 UCLRSC   FDB   CLRSCR     % CLEAR SCREEN
F820 F879                146 USCROL   FDB   SCROLL     % SCROLL SCREEN
F822 FA4E                147 UGET     FDB   GETHEX     % GET 1 HEX CHARACTER
F824 FC11                148 UPD      FDB   PD         % PRINT 'D' AS 2 ASCII CHARACTERS
F826 FE2F                149 UXPLAC   FDB   XPLACE     % CURSOR TO ADDRESS IN 'X'
F828 FCF2                150 UCOL     FDB   COLOUR     % SET COLOUR
F82A F870                151 UUHOME   FDB   UHOME      % HOME CURSOR TOP LEFT
F82C FF94                152 USET     FDB   SETCOL     % SET COLOURS DOWN LEFT HAND SIDE OF SCREEN
F82E FABC                153 UINPUT   FDB   INPUT      % SEE MANUAL
F830 FE32                154 UPLACE   FDB   PLACE      % PLACE CURSOR TO CURSOR STORE
                         155 %
                         156 % % % GET KEY THIS IS THE KEYBOARD INTERUPT ROUTINE % % %
                         157 %
F832   B6 BFF3           158 GETKEY   LDA   KBREAD
F835   B7 DFD8           159          STA   ICHAR
F838   7F BFF0           160          CLR   KBCLR
F83B   7C DFDB           161          INC   KBFLG
F83E   7D DFD5           162          TST   KBFLAG     % BLEEP ON KEY PRESS ?
F841   27 11      (F854) 163          BEQ   GETOUT
F843   34 10           164          PSHS  X
F845   BE DFD6           165          LDX   KBLEEP     % BLEEP TIME DELAY
F848   B7 BF95           166          STA   BLEEP      % START BLEEP
F84B 301F                167 GETB     DEX
F84D   26 FC      (F84B) 168          BNE   GETB
F84F   35 10           169          PULS  X
F851   B7 BF95           170          STA   BLEEP      % STOP BLEEP
F854   3B               171 GETOUT   RTI
                         172 %
                         173 % % % STRING % % %
                         174 %
F855   34 02           175 STRING   PSHS  A
F857   A6 80           176 STR1     LDA   ,X+        % NOTE STRING START MUST BE IN 'X' ON ENTRY
F859   81 04           177          CMPA  #$04       % STRING TERMINATOR
F85B   27 04      (F861) 178          BEQ   STR2
F85D   8D 6E      (F8CD) 179          BSR   DISPLA
F85F   20 F6      (F857) 180          BRA   STR1
```

```
F861  35 82            181 STR2    PULS  A,PC
                       182 *
                       183 * * * CLEAR SCREEN HOME CURSER * * *
                       184 *
F863  8E E000          185 CLRSCR  LDX   #SCRTOP
F866  CC 2020          186 SCR2    LDD   #$2020
F869  ED 81            187 SCR1    STD   0,X++
F86B  8C E7EA          188         CMPX  #SCREND+2
F86E  23 F9     (F869) 189         BLS   SCR1
F870  8E E001          190 UHOME   LDX   #SCRTOP+1
F873  17 05B9   (FE2F) 191         LBSR  XPLACE
F876  16 071B   (FF94) 192         LBRA  SETCOL    GO SET COLOUR CODES
                       193 *
                       194 * * * SCROLL SCREEN * * *
                       195 *
F879  34 16            196 SCROLL  PSHS  A,B,X
F87B  8E E000          197         LDX   #SCRTOP
F87E  EC 88 51         198 SCROL1  LDD   LINOFF,X  * $51 IS CURRENT SCREEN LINE LENGTH
F881  ED 81            199         STD   0,X++
F883  8C E797          200         CMPX  #SCREND-LINOFF * BOTTOM ROW
F886  25 F6     (F87E) 201         BCS   SCROL1
F888  8E E798          202         LDX   #SCRBOT   * CLEAR BOTTOM LINE
F88B  B6 DFDE          203         LDA   COLL+2    * THIS SETS GENERAL OUTPUT
F88E  A7 80            204         STA   ,X+       * TO GREEN
F890  17 059C   (FE2F) 205 SCROL3  LBSR  XPLACE    * SET CURSOR TO EDGE
F893  CC 2020          206         LDD   #$2020    * SET SPACERS ALONG BOTTOM ROW
F896  ED 81            207 SCROL2  STD   0,X++
F898  8C E7E9          208         CMPX  #SCREND+1
F89B  25 F9     (F896) 209         BCS   SCROL2
F89D  35 96            210         PULS  A,B,X,PC
                       211 *
                       212 * * * PLACE CURSOR * * *
                       213 *
F89F  8E E798          214 CURSA   LDX   #SCRBOT   * THIS IS THE "0=" MONITOR CURSOR
F8A2  B6 DFDD          215         LDA   COLL+1    * SET TO RED
F8A5  A7 80            216         STA   ,X+
F8A7  CC 4F3D          217         LDD   #$4F3D    * ASCII O=
F8AA  ED 81            218         STD   ,X++
F8AC  16 0580   (FE2F) 219         LBRA  XPLACE    * PLACE THE SCREEN CURSOR
                       220 *
                       221 * * * DISPLAY 'X' AS A 4 DIGIT ASCII STRING ON SCREEN + SPACE * * *
                       222 *
F8AF  17 01C4   (FA76) 223 SPACEX  LBSR  SPACE
F8B2  34 16            224 DISX    PSHS  X,D
F8B4  1F 10            225         TFR   X,D
F8B6  17 017F   (FA38) 226         LBSR  HEXA      * DATA IN 'D' READY FOR HEXDIS PRINT 'A'
F8B9  1F 10            227         TFR   X,D       * RETRIEVE DATA GET 2nd BYTE FOR HEXDIS
F8BB  1F 98            228         TFR   B,A       * GET LOW BYTE INTO 'A'
F8BD  17 0178   (FA38) 229         LBSR  HEXA
F8C0  35 96            230         PULS  X,D,PC
                       231 *
                       232 * * * DISPLA * * * DISPLAY 'A' TO SCREEN EXCEPT $0D,$7F
F8C2  17 01B1   (FA76) 233 SPADIS  LBSR  SPACE     * DISPLAY A SPACE FIRST
F8C5  20 06     (F8CD) 234         BRA   DISPLA
F8C7  86 0D            235 CR      LDA   #$0D      * JUST CR IF ENTERED HERE
F8C9  8D 02     (F8CD) 236         BSR   DISPLA
F8CB  86 0A            237         LDA   #$0A
F8CD  34 36            238 DISPLA  PSHS  A,B,X,Y
F8CF  BE DFE4          239         LDX   CURSOR    * PICK UP CURSOR ADDRESS
F8D2  7D DFED          240         TST   STAT      * SEE IF IN CONTROL SEQUENCE
```

```
F8D5 102A 008F   (F968)  241          LBPL  FILT1
F8D9   4D                242          TSTA
F8DA   27 17     (F8F3)  243          BEQ   DIS
F8DC   81 20             244          CMPA  #$20      * LOWEST ASCII CODE
F8DE   25 39     (F919)  245          BLO   FILT      * GO SEE IF SPECIAL
F8E0   81 7F             246          CMPA  #$7F      * HIGHEST ASCII CODE
F8E2   24 35     (F919)  247          BHS   FILT      * GO SEE IF SPECIAL
F8E4   A7 80             248 DISP1    STA   ,X+       * WAS ASCII SO DISPLA
F8E6   8C E7E9           249          CMPX  #SCREND+1 * ? END OF SCREEN
F8E9   26 05     (F8F0)  250          BNE   DISEND
F8EB   8D 8C     (F879)  251          BSR   SCROLL
F8ED   8E E799           252          LDX   #SCRBOT+1
F8F0   17 053C   (FE2F)  253 DISEND   LBSR  XPLACE
F8F3   35 B6             254 DIS      PULS  A,B,X,Y,PC
                         255 *
F8F5 010708090A0C        256 CODTAB   FCB   01,07,08,09,$0A,$0C,$0D,$1A,$1B,$7F
F8FB 0D1A1B7F
F8FF B0B1                257          FCB   $B0,$B1
                         258 *
F901 05E8                259 CODADD   FDB   GOHOME-CODADD * 01 HOME
F903 0050                260          FDB   SBLEEP-CODADD * 07 SEND BLEEP
F905 0648                261          FDB   LEFT-CODADD * 08 CURSOR LEFT
F907 0654                262          FDB   RIGHT-CODADD * 09 CURSOR RIGHT
F909 05ED                263          FDB   LFEED-CODADD * 0A CURSOR DOWN LINE FEED
F90B 05E3                264          FDB   CSCREN-CODADD * 0C CLEAR SCREEN
F90D 0102                265          FDB   DOCR-CODADD * 0D DO C/R
F90F 05B3                266          FDB   CUP-CODADD * 1A CURSOR UP
F911 0060                267          FDB   ESCAPE-CODADD * 1B ESCAPE SEQUENCE
F913 011C                268          FDB   DELET-CODADD * 7F DELETE
F915 061A                269          FDB   CLINE-CODADD * B0 CLEAR TO END OF LINE
F917 0605                270          FDB   CEND-CODADD * B1 CLEAR TO END OF SCREEN
                         271 *
                         272 *
F919   81 80             273 FILT     CMPA  #$80
F91B   27 D6     (F8F3)  274          BEQ   DIS
F91D   25 04     (F923)  275          BLO   FILT00
F91F   81 A0             276          CMPA  #$A0
F921   25 1D     (F940)  277          BLO   VIDEO     All codes 81 ) 9F to VIDEO
F923   31 8C CF (F8F5)  278 FILT00    LEAY  CODTAB,PCR * ADDRESS OF CODE TABLE
F926   5F               279          CLRB            * CLEAR FOR COUNT
F927   A1 A5             280 FILT0    CMPA  B,Y       * DOES 'A' MATCH TABLE
F929   27 07     (F932)  281          BEQ   CODEDO    * IF SO FIND ADDRESS OF HANDLER
F92B   5C                282          INCB
F92C   C1 0C             283          CMPB  #$C       * No OF CODES IN TABLE + 1
F92E   26 F7     (F927)  284          BNE   FILT0     * NOT TO END OF TABLE YET
F930   20 B2     (F8E4)  285          BRA   DISP1     * SEND AS IS, NOT SPECIAL
                         286 *
F932   58               287 CODEDO   ASLB            * 'B' X 2
F933   31 8C CB (F901)  288          LEAY  CODADD,PCR * PICK UP CODE ADDRESS
F936   34 10             289          PSHS  X
F938   AE A5             290          LDX   B,Y
F93A   1F 10             291          TFR   X,D
F93C   35 10             292          PULS  X
F93E   6E AB             293          JMP   D,Y       * GO DO
                         294 *
                         295 *
                         296
F940   84 7F             297 VIDEO    ANDA  #$7F
F942   81 08             298          CMPA  #8        IF NOT A COLOUR
F944   24 9E     (F8E4)  299          BHS   DISP1     * NOW SEND
```

```
F946  34 10            300         PSHS  X        ELSE SWAP FOR COLOUR IN TABLE
F948  8E DFDC          301         LDX   #COLL
F94B  A6 86            302         LDA   A,X
F94D  35 10            303         PULS  X
F94F  20 93     (F8E4) 304         BRA   DISP1
                       305 *
F951 10BE DFD9         306 SBLEEP  LDY   TBLEEP
F955  B7 BF95          307         STA   BLEEP    * TURN BLEEP ON
F958 313F              308 SBL1    DEY
F95A  26 FC     (F958) 309         BNE   SBL1
F95C  B7 BF95          310         STA   BLEEP    * TURN BLEEP OFF
F95F  20 92     (F8F3) 311         BRA   DIS
                       312 *
F961  86 05            313 ESCAPE  LDA   #5        * MARK START OF A SEQUENCE
F963  B7 DFED          314         STA   STAT     * 5 = 1B,5B,XX,YY, FOR CURSOR
F966  20 8B     (F8F3) 315         BRA   DIS      * POSTIONING
F968  F6 DFED          316 FILT1   LDB   STAT     * GET SEQUENCE STAGE
F96B  7A DFED          317         DEC   STAT
F96E  C1 05            318         CMPB  #5       * IS IT FIRST STAGE ie #$5B, OR $3B
F970  26 0D     (F97F) 319         BNE   FILT2    * NOTE IF 1B,3B IS SENT FOR CURSOR POSITIONING
F972  81 3B            320         CMPA  #$3B     * THE CURSOR DATA IS PRINTED OUT THIS IS A HELP
F974  27 04     (F97A) 321         BEQ   FILT1A   * WHEN TRYING TO FIND OUT WHAT CURSOR DATA IS BEING
F976  81 5B            322         CMPA  #$5B     * SENT BY A NEW PROGRAM
F978  26 50     (F9CA) 323         BNE   TABORT   * NO GO CHECK FOR LEFT OR RIGHT
F97A  B7 DFDC          324 FILT1A  STA   COLL     * TEMP STORE
F97D  20 50     (F9CF) 325         BRA   FILTEN
F97F  34 02            326 FILT2   PSHS  A
F981  B6 DFDC          327         LDA   COLL     * SEE WHICH SEQUENCE
F984  81 3B            328         CMPA  #$3B     * TEST SEQUENCE ?
F986  27 08     (F990) 329         BEQ   FTEST
F988  81 5B            330         CMPA  #$5B
F98A  27 0B     (F997) 331         BEQ   FHEX
F98C  35 02            332         PULS  A
F98E  20 3A     (F9CA) 333         BRA   TABORT
F990  35 02            334 FTEST   PULS  A
F992  7A DFED          335         DEC   STAT
F995  2D 33     (F9CA) 336         BLT   TABORT
F997  35 02            337 FHEX    PULS  A
F999  F6 DFED          338         LDB   STAT     * CURSOR INSTRUCTION IS IN 2 HEX BYTES
F99C  7A DFED          339         DEC   STAT
F99F  5D               340         TSTB
F9A0  27 0E     (F9B0) 341         BEQ   FHEX2    * WAS COLUMN
F9A2  C1 01            342         CMPB  #$1
F9A4  27 05     (F9AB) 343         BEQ   FHEX1    * WAS ROW
F9A6  C6 01            344         LDB   #1       * START OF SEQUENCE
F9A8  F7 DFED          345         STB   STAT
F9AB  B7 DFE8          346 FHEX1   STA   ADDHI    * ROW INPUT
F9AE  20 1F     (F9CF) 347         BRA   FILTEN
F9B0  B7 DFE9          348 FHEX2   STA   ADDLOW   * COLUMN INPUT
F9B3  FC DFE8          349         LDD   ADDHI    * PICK UP ROW & COLUMN
F9B6  4F               350         CLRA
F9B7  34 06            351         PSHS  D
F9B9  FC DFE8          352         LDD   ADDHI
F9BC  1F 89            353         TFR   A,B
F9BE  86 51            354         LDA   #LINOFF
F9C0  3D               355         MUL
F9C1  8B E0            356         ADDA  #$E0
F9C3  E3 E1            357         ADDD  ,S++
F9C5  1F 01            358         TFR   D,X
F9C7  16 FF26   (F8F0) 359         LBRA  DISEND
```

```
F9CA   86 FF             360 TABORT   LDA   #$FF
F9CC   B7 DFED           361          STA   STAT
F9CF   35 B6             362 FILTEN   PULS  A,B,X,Y,PC
                         363 *
F9D1 E798E747E6F6        364 LINSTA   FDB   $E798,$E747,$E6F6,$E6A5,$E654 * LINEASTARTS
F9D7 E6A5E654
F9DB E603E5B2E561        365          FDB   $E603,$E5B2,$E561,$E510,$E4BF,$E46E
F9E1 E510E4BFE46E
F9E7 E41DE3CCE37B        366          FDB   $E41D,$E3CC,$E37B,$E32A,$E2D9,$E288
F9ED E32AE2D9E288
F9F3 E237E1E6E195        367          FDB   $E237,$E1E6,$E195,$E144,$E0F3,$E0A2,$E051,$E000
F9F9 E144E0F3E0A2
F9FF E051E000
                         368 *
                         369 * * * DOCR * * *
                         370 *
FA03   8D 03      (FA08) 371 DOCR     BSR   DOCR0
FA05   16 FEE8    (F8F0) 372 DOCR3    LBRA  DISEND
FA08   31 8C C6   (F9D1) 373 DOCR0    LEAY  LINSTA,PCR
FA0B   AC A4             374 DOCR1    CMPX  0,Y
FA0D   24 09      (FA18) 375          BCC   DOCR2
FA0F   31 22             376          LEAY  2,Y
FA11 108C FA03           377          CMPY  #LINSTA+50,PCR * 50 IS 25 LINES X 2 BYTES
FA15   25 F4      (FA0B) 378          BCS   DOCR1
FA17   39                379          RTS
FA18   AE A4             380 DOCR2    LDX   ,Y
FA1A 3001                381          INX
FA1C   39                382          RTS
                         383 *
                         384 * * * DELET * * *
                         385 *
FA1D   BE DFE4           386 DELET    LDX   CURSOR
FA20   8C E000           387          CMPX  #SCRTOP
FA23   27 04      (FA29) 388          BEQ   DEL1
FA25   86 20             389          LDA   #$20
FA27   A7 82             390          STA   ,-X
FA29   20 DA      (FA05) 391 DEL1     BRA   DOCR3
                         392
                         393 *
                         394 * * * ASCII * * *
                         395 *
FA2B   81 09             396 ASCII    CMPA  #$9       * THIS CONVERTS LSB OF ACC INTO ASCII
FA2D   23 02      (FA31) 397          BLS   ASCZ
FA2F   8B 07             398          ADDA  #'A-'9-1  * 41-39-1 ie 7
FA31   8B 30             399 ASCZ     ADDA  #'0       * ADD 30 HEX
FA33   39                400          RTS
                         401 *
                         402 * * * HEXDISPLAY * * *
                         403 *                        * THIS FETCHES CHARACTERS POINTED TO BY 'X' INTO 'A'
                         404 *                        * THEN DISPLAYS 'A' AS 2 ASCII CHARACTERS
FA34   8D 40      (FA76) 405 SPAHEX   BSR   SPACE     * IF ENTERED HERE DISPLAY SPACE FIRST
FA36   A6 80             406 HEXDIS   LDA   ,X+       * LOADS 'A' AS ABOVE
FA38   34 02             407 HEXA     PSHS  A         *IF ENTERED HERE DISPLAYS 'A'
FA3A   44                408          LSRA
FA3B   44                409          LSRA
FA3C   44                410          LSRA
FA3D   44                411          LSRA
FA3E   8D EB      (FA2B) 412          BSR   ASCII     * GO CONVERT TO ASCII
FA40   17 FE8A    (F8CD) 413          LBSR  DISPLA
FA43   A6 E4             414          LDA   ,S        * RETRIEVE 'A'
```

```
FA45  84 0F            415          ANDA  #$0F      * NOW DO BITS 0 TO3
FA47  8D E2     (FA2B) 416          BSR   ASCII
FA49  17 FE81   (F8CD) 417          LBSR  DISPLA
FA4C  35 82            418          PULS  A,PC
                       419 *
                       420 * * * GETHEX * * *
                       421 *
FA4E  17 01A2   (FBF3) 422 GETHEX   LBSR  PDIS      * THIS EXCEPTS 1 HEX CHARACTER FROM K/B
FA51  81 0D            423 HEX      CMPA  #$0D      * CHECK FOR C/R ABORT
FA53  27 1E     (FA73) 424          BEQ   GEND      * IF SO RETURN TO 0=
FA55  81 39            425          CMPA  #$39      * ? LESS THEN 9
FA57  23 02     (FA5B) 426          BLS   HEX1
FA59  84 DF            427          ANDA  #$DF      * COVERT TO UPPER CASE
FA5B  80 30            428 HEX1     SUBA  #$30
FA5D  2B 0F     (FA6E) 429          BMI   NOTHEX
FA5F  81 09            430          CMPA  #$09
FA61  2F 0A     (FA6D) 431          BLE   GETEX     * WAS HEX EXIT
FA63  80 07            432          SUBA  #$7       * WAS IT A - F SUBTRACT BIAS
FA65  81 09            433          CMPA  #$9       * LESS THAN A ie 41
FA67  2F 05     (FA6E) 434          BLE   NOTHEX    * YES NOT HEX
FA69  81 0F            435          CMPA  #$F
FA6B  2E 01     (FA6E) 436          BGT   NOTHEX
FA6D  39               437 GETEX    RTS
FA6E  86 7F            438 NOTHEX   LDA   #$7F
FA70  17 FE5A   (F8CD) 439          LBSR  DISPLA
FA73  16 0144   (FBBA) 440 GEND     LBRA  WARM1
                       441 *
                       442 * * * SPACE * * *
                       443 *
FA76  34 02            444 SPACE    PSHS  A         * DISPLAY SPACE
FA78  86 20            445          LDA   #$20      * SPACE CODE
FA7A  17 FE50   (F8CD) 446          LBSR  DISPLA
FA7D  35 82            447          PULS  A,PC
                       448 *
                       449 * * * GET2 * * *
                       450 *
FA7F  8D CD     (FA4E) 451 GET2     BSR   GETHEX    * THIS FETCHES 2 HEX CODES INTO ACC
FA81  17 042B   (FEAF) 452 GET2A    LBSR  SHIFTL
FA84  1F 89            453          TFR   A,B       * PLACE HI NIBBLE INTO 'B'
FA86  8D C6     (FA4E) 454          BSR   GETHEX    * GET LOW NIBBLE INTO 'A'
FA88  34 04            455          PSHS  B         * PUSH 'B' ONTO STACK
FA8A  AB E0            456          ADDA  ,S+       * ADD 'B' TO 'A' AND POP STACK
FA8C  39               457          RTS
                       458 *
                       459 * * * GET4 * * *
                       460 *                        * THIS GETS 4 DIGITS FROM THE KEYBOARD
                       461 *                        * AND LEAVES 4 HEX VALUES
                       462 *                        * IN ADDHI AND ADDLOW AND IN 'X'
FA8D  8D F0     (FA7F) 463 GET4     BSR   GET2
FA8F  B7 DFE8          464          STA   ADDHI
FA92  8D EB     (FA7F) 465          BSR   GET2
FA94  B7 DFE9          466          STA   ADDLOW
FA97  BE DFE8          467          LDX   ADDHI     * THIS LOADS 'X' WITH ADDHI & ADDLOW
FA9A  39               468          RTS
                       469 *
                       470 * * * DATA TO SET COLOUR VDU BOARD * * *
                       471 *
FA9B 7F51640A1E02      472 SETIT    FCB   $7F,$51,$64,$0A,$1E,$02,$19,$1B,$00,$09,$69,$09,$00,$00,$07,$80
FAA1 191B00096909
FAA7 00000780
```

```
                        473 *
                        474 * * * INITALISE THE COLOUR VDU BOARD * * *
                        475 *
FAAB  C6 0F             476 INIT    LDB   #$F
FAAD  30 8C EB  (FA9B)  477         LEAX  SETIT,PCR
FAB0  A6 85             478 INIT1   LDA   B,X
FAB2  F7 E7FE           479         STB   $E7FE     * COLOUR BOARD REGISTER SELECT REGISTER
FAB5  B7 E7FF           480         STA   $E7FF     * COLOUR BOARD COMMAND REGISTER
FAB8  5A                481         DECB
FAB9  2A F5     (FAB0)  482         BPL   INIT1
FABB  39                483         RTS
                        484 *
                        485 * * * INPUT * * *
                        486 *
FABC  17 0139   (FBF8)  487 INPUT   LBSR  UPOLL     SET CARRY & ZERO IF L/F IS ENTERED
FABF  81 0A             488         CMPA  #$0A      CLEAR CARRY & SET ZERO IF ESCAPE
FAC1  27 0A     (FACD)  489         BEQ   SETC
FAC3  81 1B             490         CMPA  #$1B      ESCAPE CODE
FAC5  27 09     (FAD0)  491         BEQ   CLRC
FAC7  81 0D             492         CMPA  #$0D      C/R ABORT TO 0=
FAC9  26 0A     (FAD5)  493         BNE   CHEHEX
FACB  20 A6     (FA73)  494         BRA   GEND      NOT RTSing SO DROP STACK
FACD  1A 05             495 SETC    ORCC  #5        SET ZERO AND CARRY
FACF  39                496         RTS
FAD0  1C FE             497 CLRC    ANDCC #$FE      CLEAR CARRY
FAD2  1A 04             498         ORCC  #4        SET ZERO
FAD4  39                499         RTS
FAD5  17 FDF5   (F8CD)  500 CHEHEX  LBSR  DISPLA
FAD8  17 FF76   (FA51)  501         LBSR  HEX       * DISPLAY AND SEE IF CODE IS HEX
FADB  1C FB             502 CHEND   ANDCC #$FB      CLEAR ZERO
FADD  39                503         RTS
                        504 *
                        505 * * * MEMORY MODIFY ROUTINE * * *
                        506 *
FADE  8D AD     (FA8D)  507 MEM     BSR   GET4      * GET 4 HEX DIGITS IN 'X' AND DISPLAY
FAE0  8D 16     (FAF8)  508 MEM2    BSR   LIST0
FAE2  17 FF51   (FA36)  509         LBSR  HEXDIS    * DISPLAY 'X' TARGET
FAE5  17 01FC   (FCE4)  510         LBSR  COLL3     * INPUT IF ANY TO YELLOW
FAE8  8D D2     (FABC)  511         BSR   INPUT     * GET K/B INPUT
FAEA  26 06     (FAF2)  512         BNE   MEM5      * WAS CHARACTER
FAEC  25 F2     (FAE0)  513         BCS   MEM2      * WAS L/F
FAEE  30 1E             514 MEM1    LEAX  -2,X      * MUST BE ESCAPE
FAF0  20 EE     (FAE0)  515         BRA   MEM2      * WAS ESCAPE GO BACK 1
FAF2  8D 8D     (FA81)  516 MEM5    BSR   GET2A     * GO AND DISPLAY AS HEX
FAF4  A7 1F             517         STA   -1,X      * PLACE IN MEMORY
FAF6  20 E8     (FAE0)  518         BRA   MEM2
                        519 *
                        520 * * * LIST0 * * *
                        521 *
FAF8  17 FDCC   (F8C7)  522 LIST0   LBSR  CR
FAFB  17 01E3   (FCE1)  523         LBSR  COLL2     * SET COLOUR TO GREEN
FAFE  17 FDB1   (F8B2)  524         LBSR  DISX      * DISPLAY 'X' AS 4 ASCII DIGITS
FB01  16 01E9   (FCED)  525         LBRA  COLL6     * SET COLOUR TO CYAN
                        526 *
                        527 *
                        528 * * * LIST A LINE OF MEMORY LOCATIONS * * *
                        529 *
FB04  8D 87     (FA8D)  530 LIST    BSR   GET4      * GET ADDRESS DISPLAY SPACE FIRST
FB06  8D F0     (FAF8)  531 LIST3   BSR   LIST0
FB08  C6 0F             532         LDB   #$0F      * LINE COUNT
```

```
FB0A   17 FF27    (FA34)   533 LIST1   LBSR  SPAHEX    * PRINT SPACE AND 'X' TARGET AS 2 ASCII
FB0D   5A                  534           DECB
FB0E   2A FA      (FB0A)   535           BPL   LIST1
FB10   17 FDB4    (F8C7)   536           LBSR  CR        * NOW PRINT ALPHA OF ABOVE HEX LINE
FB13   17 01D1    (FCE7)   537           LBSR  COLL4     * SET ALPHA TO BLUE
FB16   C6 0F               538           LDB   #$F       * SET UP COUNT AGAIN
FB18   30 10               539           LEAX  -$10,X    * BACK TO FIRST LOCATION OF ABOVE LINE
FB1A 10BE DFE4             540           LDY   CURSOR    * POS CURSOR UNDER BEGINING OF ABOVE HEX OUTPUT
FB1E   31 27               541           LEAY  7,Y
FB20 10BF DFE4             542           STY   CURSOR
FB24   A6 80               543 AD2       LDA   ,X+       * PICK UP BYTE
FB26   8D 24      (FB4C)   544           BSR   VASCII    * VERIFY IF ASCII
FB28   25 02      (FB2C)   545           BCS   AD1       * IF CARRY CLEAR IT IS NOT SO PRINT '.'
FB2A   86 2E               546           LDA   #'.       * WAS NOT ASCII SO PRINT
FB2C   17 FD9E    (F8CD)   547 AD1       LBSR  DISPLA    * WAS ASCII SO DISPLAY
FB2F   5A                  548           DECB
FB30   2B 0E      (FB40)   549           BMI   LIST4     * ? FINISHED
FB32   34 10               550           PSHS  X         * POSITION CURSOR
FB34   BE DFE4             551           LDX   CURSOR
FB37   30 02               552           LEAX  2,X
FB39   BF DFE4             553           STX   CURSOR
FB3C   35 10               554           PULS  X
FB3E   20 E4      (FB24)   555           BRA   AD2
FB40   17 FF79    (FABC)   556 LIST4     LBSR  INPUT
FB43   26 FB      (FB40)   557           BNE   LIST4
FB45   25 BF      (FB06)   558           BCS   LIST3     * WAS L/F DO ANOTHER
FB47   30 88 E0            559           LEAX  -$20,X    * THIS DROPS X TO PREVIOUS LINE
FB4A   20 BA      (FB06)   560           BRA   LIST3
                           561 *
                           562 * * * VASCII * * *
                           563 *                         * SEE IF 'A' CONTAINS AN ASCII CHARACTER
FB4C   81 20               564 VASCII    CMPA  #$20      * THIS SETS CARRY IF 'A' CONTAINED
FB4E   2B 07      (FB57)   565           BMI   NOT       * AN ASCII CHARACTER OR CLEARS IT IF NOT
FB50   81 7F               566           CMPA  #$7F
FB52   2A 03      (FB57)   567           BPL   NOT
FB54 1A01                  568           SEC             * IS ASCII
FB56   39                  569           RTS
FB57 1CFE                  570 NOT       CLC             * NOT ASCII
FB59   39                  571           RTS
                           572 *
                           573 * * * ENTRY POINT ON RESET * * *
                           574 *
FB5A 10CE DFD0             575 RESET     LDS   #SSTACK   * SET SYSTEM STACK POINTER
FB5E   CE BC00             576           LDU   #USTACK   * SET UP USER STACK POINTER
FB61   4F                  577           CLRA            * SET UP DIRECT PAGE POINTER
FB62   1F 8B               578           TFR   A,DP
FB64   86 10               579           LDA   #$10      * PART OF K/BOARD BLEEP TIME DELAY
FB66   B7 DFD6             580           STA   KBLEEP
FB69   86 30               581           LDA   #$30      * PART OF TERMINAL BLEEP TIME DELAY
FB6B   B7 DFD9             582           STA   TBLEEP
FB6E   86 FF               583           LDA   #$FF      * MARK STAT FOR FILTER SUB
FB70   B7 DFED             584           STA   STAT
FB73   B7 DFDA             585           STA   TBLEEP+1  * SET TERMINAL BLEEP
FB76   B7 DFD7             586           STA   KBLEEP+1  * AND K/B BLEEP
FB79   7F DFD5             587           CLR   KBFLAG    * ZERO = NO BLEEP
FB7C   7F FE00             588           CLR   $FE00     * CLEAR 128/512K RAM CARD IF FITTED
FB7F   7F FFFF             589           CLR   $FFFF
FB82   1C AF               590           ANDCC #$AF      * ENABLE IRQ & FIRQ
FB84   B7 BFF0             591           STA   KBCLR     * THIS CLEARS THE KEY BOARD ON RESET
FB87   17 FF21    (FAAB)   592           LBSR  INIT      * INITIALSE COLOUR BOARD
```

```
FB8A  86 3B            593         LDA  #$3B     * RTI CODE, SET UP INTERUPTS
FB8C  8E DFEE          594         LDX  #SWI3    * BOTTOM OF INTERUPT TABLE
FB8F  A7 80            595 RES1    STA  0,X+
FB91  8C DFFD          596         CMPX #NMI     * TOP OF TABLE
FB94  26 F9     (FB8F) 597         BNE  RES1
FB96  86 7E            598         LDA  #$7E     * JUMP CODE
FB98  B7 DFD2          599         STA  REDIRCT  * DISPLAY REDIRECTION
FB9B  B7 DFFD          600         STA  NMI      * SET UP NMI VECTOR TO POINT TO KEY BOARD
FB9E  CC F8CD          601         LDD  #DISPLA  * PICK UP DISPLAY POSTION
FBA1  FD DFD3          602         STD  REDIRCT+1 * DISPLAY PINTER
FBA4  CC F832          603         LDD  #GETKEY
FBA7  FD DFFE          604         STD  NMI+1    * GETKEY POINTER
FBAA  17 02A1   (FE4E) 605         LBSR STDCOL   * SET STD MONITOR COLOURS
FBAD  17 FCB3   (F863) 606         LBSR CLRSCR
FBB0  17 FCC6   (F879) 607         LBSR SCROLL
FBB3  30 8D 040C (FFC3) 608        LEAX >VERSIO,PCR
FBB7  17 FC9B   (F855) 609         LBSR STRING   * PRINT VERSION HEADER FROM XBUG SPACE
FBBA  17 FCBC   (F879) 610 WARM    LBSR SCROLL
FBBD 10CE DFD0         611 WARM1   LDS  #SSTACK  * RESET STACK
FBC1  17 FCDB   (F89F) 612         LBSR CURSA    * DISPLAY CURSOR
FBC4  17 011D   (FCE4) 613 WARM2   LBSR COLL3
FBC7  8E DFEA          614         LDX  #TEMP1   * POINT AT TEMP1
FBCA  5F               615         CLRB          * CLEAR COUNT
FBCB  8D 37     (FC04) 616 P1      BSR  POLL     * POLL KEYBOARD & GET MONITOR COMMAND
FBCD  81 0D            617         CMPA #$0D     * C/R ABORT ENTRY
FBCF  27 E9     (FBBA) 618         BEQ  WARM
FBD1  81 7F            619         CMPA #$7F     * DELETE CODE
FBD3  27 11     (FBE6) 620         BEQ  P7
FBD5  A7 85            621         STA  B,X      * SAVE CHARACTER IN TEMP
FBD7  5C               622         INCB          * BUMP COUNT
FBD8  17 FCF2   (F8CD) 623         LBSR DISPLA
FBDB  C1 02            624         CMPB #2       * CHECK FOR 3 LETTERS
FBDD  23 EC     (FBCB) 625         BLS  P1
FBDF  17 FE94   (FA76) 626         LBSR SPACE
FBE2  8D 35     (FC19) 627         BSR  COMAND
FBE4  20 D4     (FBBA) 628         BRA  WARM
FBE6  5D               629 P7      TSTB
FBE7  27 E2     (FBCB) 630         BEQ  P1
FBE9  17 FCE1   (F8CD) 631         LBSR DISPLA
FBEC  86 20            632         LDA  #$20
FBEE  5A               633         DECB
FBEF  A7 85            634         STA  B,X
FBF1  20 D8     (FBCB) 635         BRA  P1
                       636
                       637 *
                       638 * * * POLL-DISPLAY
                       639 *
                       640
FBF3  8D 03     (FBF8) 641 PDIS    BSR  UPOLL
FBF5  16 FCD5   (F8CD) 642         LBRA DISPLA
                       643
FBF8  7D DFDB          644 UPOLL   TST  KBFLG
FBFB  27 07     (FC04) 645         BEQ  POLL
FBFD  B6 DFD8          646         LDA  ICHAR
FC00  7F DFDB          647 UPOLL2  CLR  KBFLG
FC03  39               648         RTS
                       649
                       650 *
                       651 * * * POLL KEYBOARD * * *
                       652 *
```

```
FC84   7F DFD8           653 POLL    CLR   ICHAR
FC87   B6 DFD8           654 KB1     LDA   ICHAR
FC8A   27 FB     (FC87)  655         BEQ   KB1
FC8C   20 F2     (FC80)  656         BRA   UPOLL2
                         657 *
                         658 * * * PRINT 'D' * * * PRINT THE ASCII OF 'A','B', ie D
                         659 *
FC8E   17 FE65   (FA76)  660 SPAPD   LBSR  SPACE     * PRINT SPACE BEFORE OUTPUTTING D
FC11   17 FCB9   (F8CD)  661 PD      LBSR  DISPLA
FC14   1F 98             662         TFR   B,A
FC16   16 FCB4   (F8CD)  663         LBRA  DISPLA
                         664 *
                         665 * * * COMMAND HANDLER * * *
                         666 *
FC19   30 8C 48  (FC64)  667 COMAND  LEAX  CMD,PCR
FC1C   C6 4B             668         LDB   #75       * NUMBER OF MONITOR COMMANDS X 3
FC1E 108E 0019           669         LDY   #25       * NUMBER OF COMMANDS 'Y' = COUNT
FC22   B6 DFEA           670 NXTCMD  LDA   TEMP1
FC25   84 DF             671         ANDA  #$DF      * SET TO UPPER CASE
FC27   A1 85             672         CMPA  B,X       * COMPARE 'A' TO COMMAND LIST
FC29   26 2A     (FC55)  673         BNE   COO1
FC2B   B6 DFEB           674         LDA   TEMP2     * TEST SECOND COMMAND LETTER
FC2E   84 DF             675         ANDA  #$DF      * CHANGE TO UPPER CASE
FC30   30 01             676         LEAX  1,X       * MOVE BASE POINTER TO NEXT LETTER
FC32   A1 85             677         CMPA  B,X
FC34   26 1D     (FC53)  678         BNE   CO2       * NO DID NOT MATCH
FC36   B6 DFEC           679         LDA   TEMP3     * LAST ONE MATCHED TRY NEXT
FC39   84 DF             680         ANDA  #$DF      * CHANGE TO UPPERCASE
FC3B   30 01             681         LEAX  1,X       * MOVE BASE POINTER AGAIN
FC3D   A1 85             682         CMPA  B,X
FC3F   26 10     (FC51)  683         BNE   COM3      * NO MUST BE ANOTHER STRING
FC41   30 8C 6B  (FCAF)  684         LEAX  ADDRES,PCR
FC44   1F 20             685         TFR   Y,D       * GET COUNT INTO 'B'
FC46   58                686         ASLB            * B TIMES 2
FC47 10AE 85             687         LDY   B,X       * Y HAS OFFSET FROM ADDRES TO COMMAND
FC4A   1F 20             688         TFR   Y,D
FC4C   30 8C 60  (FCAF)  689         LEAX  ADDRES,PCR
FC4F   6E 8B             690         JMP   D,X
FC51   30 1F             691 COM3    LEAX  -1,X      * DROP ASCII COMAND BASE
FC53   30 1F             692 CO2     LEAX  -$1,X     * SAME AGAIN
FC55   31 3F             693 COO1    LEAY  -$1,Y     * DECREMENT COMMAND COUNT
FC57   C0 03             694         SUBB  #3
FC59   2A C7     (FC22)  695         BPL   NXTCMD
FC5B   17 0249   (FEA7)  696         LBSR  CHECK1
FC5E   BE F004           697         LDX   INVAL     * PICK UP INVALID WARNING STRING ADDRESS
FC61   16 FBF1   (F855)  698         LBRA  STRING
                         699 *
                         700 * * * COMMAND VALIDATION TABLE * * *
                         701 *
FC64 434F50              702 CMD     FCC   'COP'     * COPY
FC67 4D454D              703         FCC   'MEM'     * MODIFY MEMORY
FC6A 535957              704         FCC   'SYW'     * WARM START TO DOS SYSTEM
FC6D 535943              705         FCC   'SYC'     * COLD START TO DOS SYSTEM
FC70 535942              706         FCC   'SYB'     * BOOT DOS SYSTEM
FC73 4C4953              707         FCC   'LIS'     * LIST A LINE OF MEMORY
FC76 4A4D50              708         FCC   'JMP'     * GO TO ADDRESS THAT FOLLOWS
FC79 4A5352              709         FCC   'JSR'     * JSR TO FOLLOWING ADDRESS
FC7C 4A4D49              710         FCC   'JMI'     * JUMP INDIRECT
FC7F 4A5349              711         FCC   'JSI'     * JSR INDIRECT
FC82 544553              712         FCC   'TES'     * MEMORY TEST
```

```
FC85 434C52              713          FCC   'CLR'     * CLR SCREEN
FC88 434F4C              714          FCC   'COL'     * COLOUR CHANGE
FC8B 48454C              715          FCC   'HEL'     * HELP DISPLAY MENU
FC8E 564543              716          FCC   'VEC'     * LIST VECTOR JUMP TABLE
FC91 46494C              717          FCC   'FIL'     * FILL MEMOYRY
FC94 435552              718          FCC   'CUR'     * CURSOR MOVE
FC97 415343              719          FCC   'ASC'     * ASCII TABLE DISPLAY
FC9A 53594E              720          FCC   'SYN'     * COMMAND SYNTAX
FC9D 484F4D              721          FCC   'HOM'     * HOME CURSOR
FCA0 574152              722          FCC   'WAR'     * WARM START $0003
FCA3 535441              723          FCC   'STA'     * START $0000
FCA6 455850              724          FCC   'EXP'     * EXPAND MONITOR
FCA9 424C45              725          FCC   'BLE'     * KEYBOARD BLEEP
FCAC 4B494C              726          FCC   'KIL'     * KILL OFF THE CACHE
                         727 *
                         728 * * * MONITOR COMMANDS ADDRESS TABLE * * *
                         729 *
FCAF 00F8                730 ADDRES   FDB   COPY-ADDRES * THIS IS POS IND CODE
FCB1 FE2F                731          FDB   MEM-ADDRES * MODIFY MEMORY
FCB3 0098                732          FDB   SW-ADDRES  * WARM START DOS
FCB5 009B                733          FDB   SC-ADDRES  * DOS COLD START
FCB7 009E                734          FDB   SYB-ADDRES * BOOT DOS SYSTEM
FCB9 FE55                735          FDB   LIST-ADDRES * LIST A LINE OF MEMORY
FCBB 008E                736          FDB   JMP-ADDRES * JUMP TO ADDRESS THAT FOLLOWS
FCBD 008E                737          FDB   JMP-ADDRES * JSR TO ADDRESS THAT FOLLOWS
FCBF 0093                738          FDB   JMI-ADDRES * JUMP INDIRECT
FCC1 0093                739          FDB   JMI-ADDRES * JSR  INDIRECT
FCC3 0178                740          FDB   TEST-ADDRES * TEST MEMORY
FCC5 FBB4                741          FDB   CLRSCR-ADDRES * CLEAR SCREEN
FCC7 030C                742          FDB   COL-ADDRES * CHANGE COLOURS
FCC9 014A                743          FDB   HELP-ADDRES * DISPLAY MENU
FCCB 0302                744          FDB   VECTOR-ADDRES * DISPLAY VECTOR TABLE
FCCD 0164                745          FDB   FILL-ADDRES * FILL MEMORY
FCCF 02B2                746          FDB   CORSER-ADDRES * CURSOR MOVE
FCD1 028C                747          FDB   ASC-ADDRES * ASCII TABLE DISPLAY
FCD3 01AA                748          FDB   SYNTAX-ADDRES * COMMAND SYNTAX
FCD5 02DC                749          FDB   HOME-ADDRES * HOME CURSOR
FCD7 0088                750          FDB   UWARM-ADDRES * WARM START $0003
FCD9 008B                751          FDB   STAR-ADDRES * COLD START $0000
FCDB 02F9                752          FDB   EXP-ADDRES * EXPAND MONITOR
FCDD 0078                753          FDB   BLE-ADDRES * KEY PRESS BLEEPER
FCDF 0059                754          FDB   KCACHE-ADDRES * KILL OFF THE CACHE
                         755 *
                         756 * * * COLL2 TO 7 * * *
                         757 *
FCE1   C6 02             758 COLL2    LDB   #2
FCE3 B5                  759          FCB   $B5
FCE4   C6 03             760 COLL3    LDB   #3
FCE6 B5                  761          FCB   $B5
FCE7   C6 04             762 COLL4    LDB   #4
FCE9 B5                  763          FCB   $B5
FCEA   C6 05             764 COLL5    LDB   #5
FCEC B5                  765          FCB   $B5
FCED   C6 06             766 COLL6    LDB   #6
FCEF B5                  767          FCB   $B5
FCF0   C6 07             768 COLL7    LDB   #7
                         769 *
                         770 * * * COLOUR * * *
                         771 *
FCF2   34 36             772 COLOUR   PSHS  A,B,X,Y   * COLOUR OFFSET MUST BE IN
```

```
FCF4  5D              773          TSTB          * 'B' ON ENTRY
FCF5  27 0F    (FD06) 774          BEQ   COLEND
FCF7  C1 08           775          CMPB  #8
FCF9  24 0B    (FD06) 776          BHS   COLEND
FCFB  8E DFDC         777          LDX   #COLL
FCFE  A6 85           778          LDA   B,X
FD00  BE DFE4         779          LDX   CURSOR
FD03  16 FBDE  (F8E4) 780          LBRA  DISP1
FD06  35 B6           781 COLEND   PULS  A,B,X,Y,PC
                      782 *
                      783 * * * KILL CACHE * * *
                      784 *
FD08  86 33           785 KCACHE   LDA   #$33     FIRST CACHE CARD ADDRESS
FD0A  8D 0A    (FD16) 786          BSR   CACHE1
FD0C  86 44           787          LDA   #$44     SECOND CACHE CARD SLOT
FD0E  8D 06    (FD16) 788          BSR   CACHE1
FD10  4F              789          CLRA
FD11  B7 FFFF         790          STA   $FFFF    RESET PAGEING TO SLOT 0
FD14  20 19    (FD2F) 791          BRA   BLEOUT
                      792
FD16  B7 FFFF         793 CACHE1   STA   $FFFF    SET PAGE TO 'A'
FD19  4F              794          CLRA
FD1A 108E 0000        795          LDY   #$0000   START ADDRESS OF 32K PAGE
FD1E  8E 8000         796          LDX   #$8000   END ADDRESS OF 32K PAGE
FD21  BF DFEA         797          STX   TEMP1
FD24  16 00F7  (FE1E) 798          LBRA  FILL1
                      799
                      800 *
                      801 * * * BLE * * *
                      802 *
FD27  7D DFD5         803 BLE      TST   KBFLAG   * TOGGLE KEYBOARD BLEEPER ON/OFF
FD2A  27 06    (FD32) 804          BEQ   KBSET
FD2C  7F DFD5         805          CLR   KBFLAG
FD2F  16 FE88  (FBBA) 806 BLEOUT   LBRA  WARM
FD32  7C DFD5         807 KBSET    INC   KBFLAG
FD35  20 F8    (FD2F) 808          BRA   BLEOUT
                      809 *
                      810 * * * WARM * * *
                      811 *
FD37  7E 0003         812 UWARM    JMP   )$0003   * USER WARM START
                      813 *
                      814 * * * STAR * * *
                      815 *
FD3A  7E 0000         816 STAR     JMP   )$0000   * USER COLD START
                      817 *
                      818 * * * JMP * * * JSR * * *
                      819 *
FD3D  17 FD4D  (FA8D) 820 JMP      LBSR  GET4     * OBTAIN ADDRESS
FD40  1F 15           821          TFR   X,PC     * PLACE ADDRESS INTO PROGRAME COUNTER
                      822 *
                      823 * * * JMI * * *
                      824 *
FD42  17 FD48  (FA8D) 825 JMI      LBSR  GET4     * THIS JUMPS INDIRECT TO USER
FD45  6E 94           826          JMP   [0,X]
                      827 *
                      828 * * * SW (DOS WARM) * * *
                      829 *
FD47  7E CD03         830 SW       JMP   $CD03    * DOS WARM START ENTRY POINT
FD4A  7E CD00         831 SC       JMP   $CD00    * DOS COLD START ADDRESS
                      832 *
```

```
                        833 * * * SYB (DOS BOOT) * * *
                        834 *
FD4D   86 BF            835 SYB     LDA   #$BF
FD4F   1F 8B            836         TFR   A,DP
FD51   96 90            837         LDA   $90
FD53   86 20            838         LDA   #$20
FD55   97 94            839         STA   $94
FD57   86 0B            840         LDA   #$0B
FD59   97 90            841         STA   $90
FD5B   8D 3C      (FD99)842         BSR   DEL28
FD5D   8D 41      (FDA0)843         BSR   TSTBSY
FD5F   8E C100          844         LDX   #$C100
FD62   86 01            845         LDA   #$01
FD64   97 92            846         STA   $92
FD66   86 04            847         LDA   #$04
FD68   97 90            848         STA   $90
FD6A 108E 0000          849         LDY   #$0000
FD6E 313F               850 FLB2    DEY
FD70   26 FC      (FD6E)851         BNE   FLB2
FD72   96 94            852         LDA   $94
FD74   84 3E            853         ANDA  #$3E
FD76   97 94            854         STA   $94
FD78   96 94            855 BOOT1   LDA   $94
FD7A   85 40            856         BITA  #$40
FD7C   27 FA      (FD78)857         BEQ   BOOT1
FD7E   84 7E            858         ANDA  #$7E
FD80   97 94            859         STA   $94
FD82   96 90            860 BOOT2   LDA   $90
FD84   85 01            861         BITA  #$01
FD86   27 0A      (FD92)862         BEQ   BOTEND
FD88   85 02            863         BITA  #$02
FD8A   27 F6      (FD82)864         BEQ   BOOT2
FD8C   96 93            865         LDA   $93
FD8E   A7 80            866         STA   ,X+
FD90   20 F0      (FD82)867         BRA   BOOT2
FD92   86 00            868 BOTEND  LDA   #DIRPAG
FD94   1F 8B            869         TFR   A,DP
FD96   7E C100          870         JMP   $C100
FD99   17 0000    (FD9C)871 DEL28   )LBSR  DEL14
FD9C   17 0000    (FD9F)872 DEL14   )LBSR  DEL
FD9F   39               873 DEL     RTS
FDA0   D6 90            874 TSTBSY  LDB   $90
FDA2   C5 01            875         BITB  #$01
FDA4   26 FA      (FDA0)876         BNE   TSTBSY
FDA6   39               877         RTS
                        878 *
                        879 * * * COPY * * *
                        880 *
FDA7   8D 5F      (FE08)881 COPY    BSR   GET8     * RETURNS WITH START ADR IN Y END IN X
FDA9 10BF DFE6          882         STY   USAVE
FDAD   BF DFEA          883         STX   TEMP1    * SAVE END ADDRESS
FDB0   17 FF31    (FCE4)884         LBSR  COLL3    * SET TO YELLOW
FDB3   17 FCD7    (FA8D)885         LBSR  GET4     * FETCH ADDRESS TO COPY TO INTO 'X'
FDB6   BC DFE6          886         CMPX  USAVE    GOING UP MEMORY ?
FDB9   23 22      (FDDD)887         BLS   C01      NO COPY LOWEST BYTE FIRST
FDBB   1F 10            888         TFR   X,D      CALC NEW ADDRESS
FDBD   F3 DFEA          889         ADDD  TEMP1
FDC0   B3 DFE6          890         SUBD  USAVE
FDC3   1F 01            891         TFR   D,X
FDC5 3001               892         INX
```

```
FDC7   34 10              893          PSHS  X
FDC9 10BE DFEA            894          LDY   TEMP1
FDCD 3121                 895          INY
FDCF   A6 A2              896 COP2     LDA   0,-Y       COPY HIGHEST BYTE FIRST
FDD1   A7 82              897          STA   0,-X
FDD3 10BC DFE6            898          CMPY  USAVE
FDD7   22 F6      (FDCF)  899          BHI   COP2
FDD9   35 10              900          PULS  X
FDDB   20 0A      (FDE7)  901          BRA   CO3
FDDD   A6 A0              902 CO1      LDA   ,Y+        * PICK UP BYTE
FDDF   A7 80              903          STA   ,X+        * PLACE INTO COPY AREA
FDE1 10BC DFEA            904          CMPY  TEMP1
FDE5   23 F6      (FDDD)  905          BLS   CO1
FDE7   17 FEF7    (FCE1)  906 CO3      LBSR  COLL2      * SET TO GREEN
FDEA   30 1F              907          LEAX  -1,X
FDEC   16 FAC3    (F8B2)  908          LBRA  DISX       * GO DISPLAY END ADDRESS
                          909 *
                          910 * * * NEWSTRING * * *
                          911 *
FDEF   17 FAD5    (F8C7)  912 NEWSTR   LBSR  CR         * USER VECTOR CALLED
FDF2   16 FA60    (F855)  913          LBRA  STRING     * OUTPUT STRING POINTED TO 'X'
                          914 *
                          915 * * * CINPUT * * *
                          916 *
FDF5   7D DFDB            917 CINPUT   TST   KBFLG      * SEE IF KBFLG IS STILL ZERO
FDF8   39                 918          RTS              * ie HAS CHARACTER BEEN ENTERED
                          919 *
                          920 * * * HELP * * *
                          921 *
FDF9   17 FA67    (F863)  922 HELP     LBSR  CLRSCR     * CLEAR SCREEN
FDFC   17 00A8    (FEA7)  923          LBSR  CHECK1     * PULL DOWN DATA EPROM
FDFF   BE F006            924          LDX   HELPAS     * HELP ASCII TEXT
FE02   17 FA50    (F855)  925 DROP2    LBSR  STRING     * DISPLAY TEXT STRING
FE05   16 FDB5    (FBBD)  926 DROP     LBRA  WARM1      * STOPS SCREEN FROM JUMPING
                          927 *
                          928 * * * GET8 * * *
                          929 *
FE08   17 FC82    (FA8D)  930 GET8     LBSR  GET4       * GET START ADDRESS INTO 'Y'
FE0B   1F 12              931          TFR   X,Y        * AND END ADDRESS INTO 'X' ON EXIT
FE0D   17 FED1    (FCE1)  932          LBSR  COLL2
FE10   16 FC7A    (FA8D)  933          LBRA  GET4
                          934 *
                          935 * * * FILL * * *
                          936 *
FE13   8D F3      (FE08)  937 FILL     BSR   GET8       * FILL MEMORY WITH THE CONTENTS OF 'A'
FE15   BF DFEA            938          STX   TEMP1      * SAVE END ADDRESS
FE18   17 FC5B    (FA76)  939          LBSR  SPACE
FE1B   17 FC61    (FA7F)  940          LBSR  GET2
FE1E   A7 A0              941 FILL1    STA   ,Y+
FE20 10BC DFEA            942          CMPY  TEMP1
FE24   23 F8      (FE1E)  943          BLS   FILL1
FE26   39                 944          RTS
                          945 *
                          946 * * * TEST * * *
                          947 *
FE27   86 30              948 TEST     LDA   #$30       ASCII EPROM CODE ADDRESS
FE29   8D 50      (FE7B)  949          BSR   PULL0      GO PULL DOWN EPROM
FE2B   6E 9F F001         950          JMP   [$F001]    JUMP TO EPROM
                          951 *
                          952 * * * XPLACE * * *
```

```
                        953 *
FE2F   BF DFE4          954 XPLACE   STX   CURSOR
FE32   34 02            955 PLACE    PSHS  A
FE34   86 0E            956          LDA   #$0E      * SET 6845 REG REG TO CURSOR REG
FE36   B7 E7FE          957          STA   $E7FE     * SET CURSOR HIGH ADDRESS
FE39   B6 DFE4          958          LDA   CURSOR    * GET NEW CURSOR POSITION
FE3C   84 0F            959          ANDA  #$0F      * REAL ADDRESS TO SCREEN ADD
FE3E   B7 E7FF          960          STA   $E7FF     * PLACE INTO 6845 COM REGISTER
FE41   86 0F            961          LDA   #$0F      * SET CURSOR LOW ADDRESS
FE43   B7 E7FE          962          STA   $E7FE
FE46   B6 DFE5          963          LDA   CURSOR+1
FE49   B7 E7FF          964          STA   $E7FF
FE4C   35 82            965          PULS  A,PC
                        966 *
                        967 * * * STDCOL * * *
                        968 *
FE4E   8E DFDC          969 STDCOL   LDX   #COLL     * THIS SETS STD MON COLOURS
FE51   C6 07            970          LDB   #7
FE53   E7 85            971 STD1     STB   B,X
FE55   5A               972          DECB
FE56   26 FB     (FE53) 973          BNE   STD1
FE58   39               974 RET1     RTS
                        975 *
                        976 * * * SYNTAX * * *
                        977 *
FE59   17 FC1A   (FA76) 978 SYNTAX   LBSR  SPACE    * THIS SUPLIES THE SYNTAX OF ANY COMMAND
FE5C   CC 2020          979          LDD   #$2020
FE5F   FD DFEB          980          STD   TEMP2
FE62   5F               981          CLRB
FE63   8E DFEA          982          LDX   #TEMP1
FE66   17 FD8A   (FBF3) 983 SYN1     LBSR  PDIS     * POLL & DISPLAY
FE69   81 0D            984          CMPA  #$0D     * END IF CR
FE6B   27 EB     (FE58) 985          BEQ   RET1
FE6D   A7 85            986          STA   B,X      * SAVE INTO TEMPS
FE6F   5C               987          INCB
FE70   C1 02            988          CMPB  #2       * ONLY NEED 3 CHARACTERS
FE72   23 F2     (FE66) 989          BLS   SYN1
FE74   86 28            990          LDA   #$28     * SYNTAX EPROM ADDRESS ie $2800
FE76   8D 03     (FE7B) 991          BSR   PULL0    * PULL DOWN DATA
FE78   7E F001          992          JMP   EXTENS   * GO TO EPROM JUST DOWN LOADED
                        993 *
                        994 * * * PULL0 * * *
                        995 *
FE7B   8D 08     (FE85) 996 PULL0    BSR   PULL
FE7D   B1 F000          997          CMPA  EPMARK   IF PULL UNSUCCESFUL
FE80 1026 FD36   (FBBA) 998          LBNE  WARM     ABORT
FE84   39               999          RTS
                       1000
                       1001 *
                       1002 * * * PULL * * *
                       1003 *
FE85   34 36           1004 PULL     PSHS  A,B,X,Y  * THIS LOADS DATA FROM EPROM BOARD AS DEFINED IN 'A' ON ENTRY
FE87   B1 F000         1005          CMPA  EPMARK   * IS 'A' THE SAME EPROM CODE AS IS PRESENTLY LOADED
FE8A   27 19     (FEA5)1006          BEQ   PULEND   * IF SO RETURN
FE8C   5F              1007          CLRB           * 'D' NOW HAS THE EPROM ADDRESS
FE8D   1F 01           1008          TFR   D,X      * 'X' NOW HAS THE EPROM ADDRESS
FE8F 108E F000         1009          LDY   #$F000   * TARGET FOR EPROM DATA
FE93   86 70           1010          LDA   #$70     * SWITCH PAGE CODE
FE95   B7 FFFF         1011          STA   $FFFF    * PAGE SWITCH ADDRESS
FE98   EC 81           1012 PULL1    LDD   ,X++     * THIS DOWNLOADS THE EPROM
```

```
FE9A   ED A1            1013          STD   ,Y++      % TARGET FOR EPROM DATA
FE9C 108C F7FF          1014          CMPY  #$F7FF    % 2K YET
FEA0   23 F6     (FE98) 1015          BLS   PULL1
FEA2   7F FFFF          1016          CLR   $FFFF     % THIS SETS PAGING BACK TO 00
FEA5   35 B6            1017 PULEND   PULS  A,B,X,Y,PC
                        1018 %
                        1019 % % % CHECK1 % % %
                        1020 %
FEA7   34 02            1021 CHECK1   PSHS  A
FEA9   86 20            1022          LDA   #EPROM1   % DOWNLOAD $2000 FROM EPROM BOARD
FEAB   8D CE     (FE7B) 1023 CHECK    BSR   PULL0     % LOAD IN DATA AS 'A' + 00 ie 'A'=XX00
FEAD   35 82            1024          PULS  A,PC
                        1025 %
                        1026 %
                        1027 % % % SHIFTL % % %
                        1028 %
FEAF   48               1029 SHIFTL   ASLA
FEB0   48               1030          ASLA
FEB1   48               1031          ASLA
FEB2   48               1032          ASLA
FEB3   39               1033          RTS
                        1034 %
                        1035 % % % CURSOR UP OR SCROLL DOWN % % %
                        1036 %
FEB4   BE DFE4          1037 CUP      LDX   CURSOR    PICK UP CURSOR ADDRESS
FEB7   30 88 AF         1038          LEAX  -LINOFF,X SUBTRACT LINE LENGTH
FEBA   8C E001          1039          CMPX  #SCRTOP+1
FEBD   22 22     (FEE1) 1040          BHI   CUPEND    NO SO JUST MOVE CURSOR
FEBF   8E E7EA          1041          LDX   #SCREND+2 SCROLL DOWN
FEC2   EC 88 AD         1042 CUP1     LDD   -LINOFF-2,X GET CHARACTERS ABOVE
FEC5   ED 83            1043          STD   0,--X     PLACE HERE
FEC7   8C E051          1044          CMPX  #SCRTOP+LINOFF
FECA   22 F6     (FEC2) 1045          BHI   CUP1
FECC   8E E000          1046          LDX   #SCRTOP
FECF   17 FF5D   (FE2F) 1047          LBSR  XPLACE
FED2   17 FE0C   (FCE1) 1048          LBSR  COLL2     GO SET COLOUR CODE
FED5   86 20            1049          LDA   #$20      CLEAR TOP LINE
FED7   A7 80            1050 CUP2     STA   ,X+
FED9   8C E051          1051          CMPX  #SCRTOP+LINOFF
FEDC   25 F9     (FED7) 1052          BCS   CUP2
FEDE   16 FA12   (F8F3) 1053 CUP3     LBRA  DIS
FEE1   16 FA0C   (F8F0) 1054 CUPEND   LBRA  DISEND
                        1055 %
                        1056 % % % CLR SCR ON $0C & HOME CURSOR % % %
                        1057 %
FEE4   17 F97C   (F863) 1058 CSCREN   LBSR  CLRSCR    CLEAR SCREEN AND HOME CURSER
FEE7   20 F5     (FEDE) 1059          BRA   CUP3
                        1060 %
                        1061 % % % GOHOME % % %
                        1062 %
FEE9   8E E001          1063 GOHOME   LDX   #SCRTOP+1 % USE FROM VECTOR
FEEC   20 F3     (FEE1) 1064          BRA   CUPEND    % GO PLACE CURSOR
                        1065 %
                        1066 % % % LFEED OR SCROLL % % %
                        1067 %
FEEE   BE DFE4          1068 LFEED    LDX   CURSOR
FEF1   34 10            1069          PSHS  X         SAVE MAY NEED LATER
FEF3   30 88 51         1070          LEAX  LINOFF,X  ADD LINE LENGTH
FEF6   8C E7E9          1071          CMPX  #SCREND+1
FEF9   24 04     (FEFF) 1072          BCC   LFEND
```

```
FEFB  32 62          1073          LEAS  2,S       DROP 'X' FROM STACK
FEFD  20 E2   (FEE1) 1074          BRA   CUPEND
FEFF  17 F977 (F879) 1075 LFEND    LBSR  SCROLL
FF02  35 10          1076          PULS  X         RETRIEVE CURSOR POSTION
FF04  20 DB   (FEE1) 1077          BRA   CUPEND
                     1078 *
                     1079 * * * CLEAR TO END OF SCREEN * * *
                     1080 *
FF06  34 16          1081 CEND     PSHS  D,X
FF08  CC 2020        1082          LDD   #$2020    space code
FF0B  BE DFE4        1083          LDX   CURSOR    pick up current cursor position
FF0E  ED 81          1084 CLR1     STD   0,X++
FF10  8C E7EA        1085          CMPX  #SCREND+2 end of screen
FF13  23 F9   (FF0E) 1086          BLS   CLR1      not yet
FF15  35 16          1087          PULS  D,X
FF17  8D 7B   (FF94) 1088          BSR   SETCOL
FF19  20 C6   (FEE1) 1089          BRA   CUPEND
                     1090 *
                     1091 * * * CLEAR TO END OF LINE * * *
                     1092 *
FF1B  34 36          1093 CLINE    PSHS  X,Y,D     CLEAR TO END OF LINE
FF1D  BE DFE4        1094          LDX   CURSOR
FF20  17 FAE5 (FA08) 1095          LBSR  DOCR0     GO AND SORT OUT NEXT LINE END
FF23  1F 10          1096          TFR   X,D       'X' has next line start
FF25  C3 0050        1097          ADDD  #LINOFF-1
FF28  FD DFEA        1098          STD   TEMP1     * SAVE START OF NEW LINE
FF2B  BE DFE4        1099          LDX   CURSOR
FF2E  86 20          1100          LDA   #$20
FF30  A7 80          1101 CLINE1   STA   ,X+       * CLEAR CURSOR POSITION
FF32  BC DFEA        1102          CMPX  TEMP1     * START OF NEXT LINE YET
FF35  25 F9   (FF30) 1103          BCS   CLINE1
FF37  35 36          1104          PULS  X,Y,D
FF39  20 A6   (FEE1) 1105          BRA   CUPEND
                     1106 *
                     1107 * * * ASC * * *
                     1108 *
FF3B  17 F925 (F863) 1109 ASC      LBSR  CLRSCR    * MOVE CURSOR TO TOP OF SCREEN
FF3E  86 30          1110          LDA   #$30      * ASCII DATA EPROM ADDRESS ie $3000
FF40  17 FF38 (FE7B) 1111          LBSR  PULL0     * DOWN-LOAD DATA
FF43  8E F003        1112          LDX   #EXTENS+2 * ie $F003 EPROM DOWNLOAD DATA ADDRESS
FF46  16 FEB9 (FE02) 1113          LBRA  DROP2     * GO DISPLAY
                     1114 *
                     1115 * * * MOVE CURSOR LEFT RIGHT * * *
                     1116 *
FF49  BE DFE4        1117 LEFT     LDX   CURSOR
FF4C  8C E001        1118          CMPX  #SCRTOP+1
FF4F  27 02   (FF53) 1119          BEQ   LEND
FF51  30 1F          1120          LEAX  -1,X
FF53  20 8C   (FEE1) 1121 LEND     BRA   CUPEND    * DONT MOVE
FF55  BE DFE4        1122 RIGHT    LDX   CURSOR
FF58  8C E7E8        1123          CMPX  #SCREND
FF5B  27 F6   (FF53) 1124          BEQ   LEND      * DONT MOVE
FF5D  30 01          1125          LEAX  1,X
FF5F  20 F2   (FF53) 1126          BRA   LEND
                     1127 *
                     1128 * * * CORSER * * *
                     1129 *
FF61  17 FB29 (FA8D) 1130 CORSER   LBSR  GET4      * GET 4 DECIMAL NUMBERS & SEND WITH CODES
FF64  8D 08   (FF6E) 1131          BSR   CONTROL   * SEND $1B,$5B
FF66  1F 10          1132          TFR   X,D
```

```
FF68  17 FCA6   (FC11)  1133          LBSR  PD        * SEND
FF6B  16 FC56   (FBC4)  1134          LBRA  WARM2
                        1135 *
                        1136 * * * CONTROL * * *
                        1137 *
FF6E  CC 1B5B           1138 CONTROL  LDD   #$1B5B    * SEND CURSOR CONTROL CODES ESC & '['
FF71  16 FC9D   (FC11)  1139          LBRA  PD
                        1140 *
                        1141 * * * DECIMAL TO HEX * * *
                        1142 *
FF74  34 04             1143 DECIMA   PSHS  B         * CONVERT DECIMAL NUMBER IN 'A' TO HEX
FF76  1F 89             1144          TFR   A,B
FF78  84 0F             1145          ANDA  #$0F      * GET LOWER NIBBLE
FF7A  34 02             1146          PSHS  A         * SAVE IT
FF7C  C4 F0             1147          ANDB  #$F0      * GET HIGH NIBBLE
FF7E  54                1148          LSRB            * MOVE IT TO LOWER NIBBLE
FF7F  54                1149          LSRB
FF80  54                1150          LSRB
FF81  54                1151          LSRB
FF82  86 0A             1152          LDA   #$0A      * HEX WEIGHT OF DECIMAL 10
FF84  3D                1153          MUL             * #$A X DECIMAL NUMBER OF 10s
FF85  EB E0             1154          ADDB  ,S+       * ADD IN DECIMAL UNITS
FF87  1F 98             1155          TFR   B,A       * PLACE ANSWER IN 'A'
FF89  35 84             1156          PULS  B,PC      * RESTORE 'B' AS ENTERED
                        1157 *
                        1158 * * * HOME * * *
                        1159 *
FF8B  8E E001           1160 HOME     LDX   #SCRTOP+1 * ALLOW FOR COLOUR CODE
FF8E  17 FE9E   (FE2F)  1161          LBSR  XPLACE    * MOVE CURSOR
FF91  16 FC30   (FBC4)  1162          LBRA  WARM2     * RETURN TO MONITOR
                        1163 *
                        1164 * * * SET COL * * *
                        1165 *
FF94  34 12             1166 SETCOL   PSHS  A,X
FF96  8E E000           1167          LDX   #SCRTOP   * THIS SETS COLOUR CODES DOWN THE LEFT HAND SIDE
FF99  B6 DFDE           1168          LDA   COLL+2    * OF THE SCREEN USED IF HOME IS USED BEFORE
FF9C  A7 84             1169 SET1     STA   0,X       * DISPLAYING AS OPPOSED TO SCROLLING
FF9E  30 88 51          1170          LEAX  LINOFF,X  * ADD SCREEN LINE LENGTH
FFA1  8C E7D8           1171          CMPX  #SCRBOT+$40
FFA4  25 F6     (FF9C)  1172          BCS   SET1
FFA6  35 92             1173          PULS  A,X,PC
                        1174 *
                        1175 * * * EXPAND * * *
                        1176 *
FFA8  17 FAD4   (FA7F)  1177 EXP      LBSR  GET2      * EXPAND THE MONITOR
FFAB  17 FECD   (FE7B)  1178          LBSR  PULL0     * PULL DOWN EPROM SPECIFIED
FFAE  7E F001           1179          JMP   EXTENS    * AND GO TO IT
                        1180 *
                        1181 * * * VECTOR * * *
                        1182 *
FFB1  17 FEF3   (FEA7)  1183 VECTOR   LBSR  CHECK1
FFB4  AD 9F F008        1184          JSR   [VECTO]
FFB8  16 FC02   (FBBD)  1185          LBRA  WARM1
                        1186 *
                        1187 * * * COL * * *
                        1188 *
FFBB  86 28             1189 COL      LDA   #$28      * SYNTAX EPROM ADDRESS
FFBD  17 FEBB   (FE7B)  1190          LBSR  PULL0
FFC0  7E F004           1191          JMP   EXTENS+3
                        1192 *
```

```
FFC3 05                 1193 VERSIO  FCB   $05
FFC4 363830392052       1194         FCC   "6809 RALBUG V1.4",$0D,$0A,$04
FFCA 414C42554720
FFD0 56312E340D0A
FFD6 04
                        1195
                        1196         END
```

AD1	FB2C	AD2	FB24	ADDHI	DFE8	ADDLOW	DFE9	ADDRES	FCAF
ASC	FF3B	ASCII	FA2B	ASC2	FA31	BLE	FD27	BLEEP	BF95
BLEOUT	FD2F	BOOT1	FD78	BOOT2	FD82	BOTEND	FD92	CACHE1	FD16
CEND	FF06	CHECK	FEAB	CHECK1	FEA7	CHEHEX	FAD5	CHEND	FADB
CINPUT	FDF5	CLINE	FF1B	CLINE1	FF30	CLR1	FF0E	CLRC	FAD0
CLRSCR	F863	CMD	FC64	CO1	FDDD	CO2	FC53	CO3	FDE7
CODADD	F901	CODEDO	F932	CODTAB	F8F5	COL	FFBB	COLEND	FD06
COLL	DFDC	COLL2	FCE1	COLL3	FCE4	COLL4	FCE7	COLL5	FCEA
COLL6	FCED	COLL7	FCF0	COLOUR	FCF2	COM3	FC51	COMAND	FC19
CONTROL	FF6E	COO1	FC55	COP2	FDCF	COPY	FDA7	CORSER	FF61
CR	F8C7	CSCREN	FEE4	CUP	FEB4	CUP1	FEC2	CUP2	FED7
CUP3	FEDE	CUPEND	FEE1	CURSA	F89F	CURSOR	DFE4	DECIMA	FF74
DEL	FD9F	DEL1	FA29	DEL14	FD9C	DEL28	FD99	DELET	FA1D
DIRPAG	0000	DIS	F8F3	DISEND	F8F0	DISP1	F8E4	DISPLA	F8CD
DISX	F8B2	DOCR	FA03	DOCR0	FA08	DOCR1	FA0B	DOCR2	FA18
DOCR3	FA05	DROP	FE05	DROP2	FE02	EPMARK	F000	EPROM1	0020
EPROM2	0028	EPROM3	0030	ESCAPE	F961	EXP	FFA8	EXTENS	F001
FHEX	F997	FHEX1	F9AB	FHEX2	F9B0	FILL	FE13	FILL1	FE1E
FILT	F919	FILT0	F927	FILT00	F923	FILT1	F968	FILT1A	F97A
FILT2	F97F	FILTEN	F9CF	FIRQ	DFF4	FLB2	FD6E	FTEST	F990
GEND	FA73	GET2	FA7F	GET2A	FA81	GET4	FA8D	GET8	FE08
GETB	F84B	GETEX	FA6D	GETHEX	FA4E	GETKEY	F832	GETOUT	F854
GOCOL	F005	GOHOME	FEE9	HELP	FDF9	HELPAS	F006	HEX	FA51
HEX1	FA5B	HEXA	FA38	HEXDIS	FA36	HOME	FF8B	ICHAR	DFD8
INIT	FAAB	INIT1	FAB0	INPUT	FABC	INVAL	F004	IRQ	DFF7
I_O	00BF	JMI	FD42	JMP	FD3D	KB1	FC07	KBCLR	BFF0
KBFLAG	DFD5	KBFLG	DFDB	KBLEEP	DFD6	KBREAD	BFF3	KBSET	FD32
KCACHE	FD08	LEFT	FF49	LEND	FF53	LFEED	FEEE	LFEND	FEFF
LINOFF	0051	LINSTA	F9D1	LIST	FB04	LIST0	FAF8	LIST1	FB0A
LIST3	FB06	LIST4	FB40	MEM	FADE	MEM1	FAEE	MEM2	FAE0
MEM5	FAF2	NEWSTR	FDEF	NMI	DFFD	NOT	FB57	NOTHEX	FA6E
NXTCMD	FC22	P1	FBCB	P7	FBE6	PD	FC11	PDIS	FBF3
PLACE	FE32	POLL	FC04	PULEND	FEA5	PULL	FE85	PULL0	FE7B
PULL1	FE98	REDIRCT	DFD2	RENTR	F802	RES1	FB8F	RESET	FB5A
RET1	FE58	RIGHT	FF55	SBL1	F958	SBLEEP	F951	SC	FD4A
SCR1	F869	SCR2	F866	SCRBOT	E798	SCREND	E7E8	SCROL1	F87E
SCROL2	F896	SCROL3	F890	SCROLL	F879	SCRTOP	E000	SET1	FF9C
SETC	FACD	SETCOL	FF94	SETIT	FA9B	SHIFTL	FEAF	SPACE	FA76
SPACEX	F8AF	SPADIS	F8C2	SPAHEX	FA34	SPAPD	FC0E	SSTACK	DFD0
STAR	FD3A	STAT	DFED	STD1	FE53	STDCOL	FE4E	STR1	F857
STR2	F861	STRING	F855	SW	FD47	SWI	DFFA	SWI2	DFF1
SWI3	DFEE	SYB	FD4D	SYN1	FE66	SYNTAX	FE59	TABORT	F9CA
TBLEEP	DFD9	TEMP1	DFEA	TEMP2	DFEB	TEMP3	DFEC	TEST	FE27
TSTBSY	FDA0	UCHIN	F808	UCLRSC	F81E	UCOL	F828	UCR	F80E
UDIS	F80A	UGET	F822	UHOME	F870	UIN2HE	F818	UIN4HE	F81A
UIN8HE	F81C	UINPUT	F82E	UOUT2H	F814	UOUT4H	F816	UPD	F824
UPDISPL	F806	UPLACE	F830	UPOLL	FBF8	UPOLL2	FC00	UPSTR	F810
UPULL	F812	URESET	F800	USAVE	DFE6	USCROL	F820	USET	F82C
USTACK	BC00	USTRIN	F80C	UUHOME	F82A	UUPOLL	F804	UWARM	FD37
UXPLAC	F826	VASCII	FB4C	VECTO	F008	VECTOR	FFB1	VERSIO	FFC3
VIDEO	F940	WARM	FBBA	WARM1	FBBD	WARM2	FBC4	XPLACE	FE2F