# Manual for XTRA01.DOS
# Utilities and Disassembler

# List of extended DOS commands and their syntax

### !CODE / !LCODE

**Description:**
Similar to !FIND, but used to find things in (code) memory.
Can be used to find specified sequences of bytes / machine code.
It just lists out the memory addresses where the sequence(s) begin.
!LCODE is the same, but directs output to the printer.

**Syntax:**

| | |
|---|---|
| !CODE n1,n2,n3 etc. | will find the specified sequence of bytes. |

**Notes:**
- n1,n2,n3 etc. can be any mixture of single byte numbers, expressions or variables.
  E.g. !CODE 14, #14, N is OK.
- No strings allowed. Use the ASCII code sequence instead.
- !CODE 67,79,68,69 will find the string "CODE".
- All of memory is searched. Including the screen memory and ROM. (It's still quick.)
- A match will be found at 0035. This is where the sequence of bytes is set up in memory.
- There may also be a match in screen memory.
- Press a key to stop / start scrolling.

### !CONV

**Description:**
This command destroys all BASIC currently in memory.
It creates a new basic program, consisting only of DATA lines.
It converts a specified block of machine code or bytes into DATA statements ready for a BASIC loader.
It also creates its own check sums.

**Syntax:**

| | |
|---|---|
| !CONV n1 STEP n2,n3 TO n4 | Lists all lines |
| !CONV nl,n3 TO n4 | Lists all lines |

**Notes:**
- set HIMEM before use
- n1 is the first DATA line number
- n2 is the STEP between lines - default n2 is 10 if unspecified
- n3 is start address of block to be converted
- n4 is end address of block to be converted
- OUT OF MEMORY ERROR will be given if HIMEM is reached, (DATA lines to date survive).
- Check sums are generated at the end of each DATA line

### !DIS

**Description:**
Start a machine code disassembler, with simple but powerful features.

**Syntax:**

| | |
|---|---|
| !DIS | Starts it up, clears the screen, and gives a ># prompt at the top left. |

**Notes:**

- Type in a 4 digit start address in HEX. E.g. C000 for the start of ROM
- Follow it with - Optional, to specify an end address. The # is added automatically.
- Followed by a 4 digit end address also in HEX if an end address is required.
  E.g. >#C000-#C0FF for the first page of ROM.
  E.g. >#C000 for all of ROM
- HEX addresses only.
- Other keys which respond are:
  DEL Resets to the start screen and >#. Resets other flags too.
  P Toggles a printer on / off for printed output. Default is off.
  L Lists continuously, otherwise a key press is needed to scroll between screens. Default is off.
  M Toggles between Code and Data modes. Default is Code.
  Data mode displays data, ASCII characters instead of a disassembly.
  X Exit.
- Be careful in CONTinuous Mode, especially if the Printer is toggled on - It won't stop!
- Form feeds are required between listings - so that you can get several short ones onto one page.

## !DISK (or !DISC)

**Description:**
For transferring files from tape to disk.

**Syntax:**

| | |
|---|---|
| !DISK | Transfers 1 file from tape to the default disk drive. |
| !DISK ,S | Transfers 1 file from tape (SLOW speed) to the default disk drive. |
| !DISK TO 1 | Transfers 1 file from tape to drive 1. |
| !DISK 3 | Transfers 3 files from tape to the default disk drive. |
| !DISK 3 TO 1 | Transfers 3 files from tape to drive 1. |

**Notes:**
- !DISK and !DISC are interchangeable - use either.
- For SLOW tapes: all variations can be suffixed with ,S
- Make sure the tape is in the right place - filenames are not specified.
- Long files with short gaps - do them one at a time. If there is a long file, then a short space before the second file, the second file may be missed whilst the first is being sent to disk.

## !EDIT

**Description:**
This command deletes a specified block of lines from a BASIC program.

**Syntax:**

| | |
|---|---|
| !EDIT n1 | Deletes the single line numbered n1. |
| !EDIT n1 TO n2 | Deletes the block of lines from n1 to n2, inclusive. |
| !EDIT n1 TO | Deletes the block of lines from n1 to the end. |

**Notes:**
- n1 must always be specified !EDIT TO n2 is not allowed.
- n1 and n2 can be expressions
- No lines have to exist
- Variables are lost after execution just as in ordinary line deletion

## !FIND / !LFIND

**Description:**
Used to find things in basic programs.
It just lists out the lines containing the string you are looking for.

!LFIND is the same, but directs output to the printer.

**Syntax:**

| | |
|---|---|
| !FIND REM | will list all lines containing REM (tokenised). |
| !FIND "REM" | will list all lines containing the string REM (not tokenised). |
| !FIND GOTO | will list all lines containing the GOTO token. |
| !FIND * | will find the multiplication token. |
| !FIND "*" | will find the * character. |
| !FIND ? | will find the PRINT token |
| !FIND "?" | will find the ? character. |

**Notes:**

- "strings" can contain spaces, quotes, punctuation
- Complete matches only are reported
- No wildcards. Strings are taken exactly as typed, including * and ?
- Don't forget tokens! FORGET = F + OR + GET. "FORGET" = FORGET.
- Can be used in basic programs, but only at the end of a line, since everything to the end of the line will be taken as part of the string.

## !LIST / !LLIST

**Description:**
This is a variation on the standard LIST command, but much more flexible. It uses the TO token instead of -
This allows expressions to be used. It can be used from within a basic program, which will carry on after the LIST.

**Syntax:**

| | |
|---|---|
| !LIST | Lists all lines |
| !LIST TO | Lists all lines |
| !LIST n1 | Lists line n1 |
| !LIST n1 TO | Lists line n1 onwards |
| !LIST n1 TO n2 | Lists from n1 to n2 |
| !LIST TO n2 | Lists up to line n2 |
| !LLIST | All as for !LIST, but output to printer |

**Notes:**

- n1 and n2 can be expressions e.g
  FOR LI=100 TO 1000 STEP 100
  !LIST LI
  NEXT
  will list only lines 100,200,300 ..... 1000
- Control is returned to program. All can be used in basic programs

## !OLD

**Description:**
This is the opposite of NEW, and can save a disaster if used quickly enough.

**Syntax:**

| | |
|---|---|
| !OLD | Restores a basic program which has just been NEWed by mistake. |

**Notes:**

- Just hope you do it quickly enough.
- If it does not work first time, try LIST, then !OLD again.
- If it still does not work, you've lost it.

## !RESTORE

**Description:**
This is a computed RESTORE command, which restores the DATA pointer to the start of a specified line.

**Syntax:**

| | |
|---|---|
| !RESTORE n1 | Restores the DATA pointer to line numbered n1. |

**Notes:**
n1 can be an expression

# Disassembly for the extended DOS commands

**Table of keywords for extended DOS commands**

```
#D000     8E 00              DTA  LLIST
#D002     BC 00              DTA  LIST
#D004     43 B4 56 00        DTA  CONV
#D008     81 00              DTA  EDIT
#D00A     4F 4C 44 00        DTA  OLD
#D00E     9A 00              DTA  RESTORE
#D010     44 49 53 4B 00     DTA  DISK
#D015     44 49 53 43 00     DTA  DISC
#D01A     44 49 53 00        DTA  DIS
#D01E     4C 46 49 4E 44 00  DTA  LFIND
#D024     46 49 4E 44 00     DTA  FIND
#D029     43 4F 44 45 00     DTA  CODE
#D02E     4C 43 4F 44 45 00  DTA  LCODE

#D034 to  FF FF FF FF FF     DTA  Space for up to 19 more command words (total 32 and
#D0BB     FF FF FF FF FF     DTA  limited by the space for their addresses – see below)
```

**Address table for extended DOS commands**

Addresses for Extended DOS routines (1 less than the address for the routine as it is incremented by the RTS)

```
#D0C0     FF D0     DTA  To give #D100 for !LLIST
#D0C2     03 D1     DTA  To give #D104 for !LIST
#D0C4     79 D1     DTA  To give #D17A for !CONV
#D0C6     96 D2     DTA  To give #D297 for !EDIT
#D0C8     FA D2     DTA  To give #D2FB for !OLD
#D0CA     02 D3     DTA  To give #D303 for !RESTORE
#D0CC     19 D3     DTA  To give #D31A for !DISK
#D0CE     19 D3     DTA  To give #D31A for !DISC
#D0D0     46 D4     DTA  To give #D447 for !DIS
#D0D2     EE DB     DTA  To give #DBEF for !LFIND
#D0D4     F2 DB     DTA  To give #DBF3 for !FIND
#D0D6     9D DC     DTA  To give #DC9E for !CODE
#D0D8     98 DC     DTA  To give #DC99 for !LCODE

#D0DA to  FF FF     DTA  Space for 19 more addresses
#D0FE     FF FF     DTA
```

**!LLIST**

```
#D100     38          SEC       Enable Printer
#D101     6E F1 02    ROR 02F1
```

**!LIST**

```
#D104     38          SEC       Set EDIT mode flag
#D105     6E F2 02    ROR 02F2
#D108     A9 00       LDA #00
#D10A     85 33       STA 33    Set LIST pointers for Line 0
```

```
#D10C   85 34       STA 34
#D10E   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D111   B3 C6                   #C6B3 to look for line number and leave #CE/F pointing to it
#D113   20 E8 00    JSR 00E8    Re-fetch current character
#D116   F0 40       BEQ D158    Branch to LIST all lines if end of statement
#D118   C9 C3       CMP #C3     Test for TO
#D11A   F0 11       BEQ D12D    Branch if TO found
#D11C   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D11F   53 E8                   #E853 to evaluate expression into #0033/4
#D121   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D124   B3 C6                   # C6B3 to look for line number and leave #CE/F pointing to it
#D126   20 E8 00    JSR 00E2    Fetch the next character
#D129   C9 C3       CMP #C3     Test for TO
#D12B   D0 1A       BNE D147    Branch to list single line if TO not found
#D12D   20 E2 00    JSR 00E2    Fetch the next character
#D130   F0 26       BEQ D158    Branch to list to end if end of line
#D132   A5 CE       LDA CE      Save pointers to first line
#D134   A4 CF       LDY CF
#D136   85 00       STA 00
#D138   84 01       STY 01
#D13A   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D13D   53 E8                   #E853 to evaluate expression into #0033/4
#D13F   A5 00       LDA 00      Recover pointers to first required line
#D141   A4 01       LDY 01
#D143   85 CE       STA CE
#D145   84 CF       STY CF
#D147   AE F1 02    LDX 02F1    Test if printer flag is on
#D14A   10 05       BPL D151    Skip next command if printer off
#D14C   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D14F   1B C8                   #C81B to set output to printer.
#D151   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D154   6C C7                   #C76C to list the required lines
#D156   30 0F       BMI #D167   Branch always
#D158   AE F1 02    LDX 02F1    List to end (no last line specified)
#D15B   10 05       BPL D162    Skip next command if printer off
#D15D   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D160   1B C8                   #C81B to set output to printer.
#D162   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D165   66 C7                   #C766 to list to end of basic
#D167   A9 0D       LDA #0D     Carriage return
#D169   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D16C   D9 CC                   #CCD9 to print the character
#D16E   EA          NOP
#D16F   EA          NOP
#D170   EA          NOP
#D171   4E F2 02    LSR 02F2    Re-set list return flag
#D174   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D177   2F C8                   #C82F to reset output to screen
#D179   60          RTS         Exit
```

**!CONV**

```
#D17A   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D17D   53 E8                   #E853 to evaluate expression into #0033/4
#D17F   84 04       STY 04      Save it as the first line number to use
#D181   85 05       STA 05
#D183   A9 0A       LDA #0A     Set default STEP value to 10
#D185   85 0A       STA 0A      Save step value
#D187   20 E8 00    JSR 00E8    Re-fetch current character
```

```
#D18A   C9 CB       CMP #CB     Test for STEP
#D18C   D0 13       BNE D1A1    Branch if no step specified
#D18E   20 E2 00    JSR 00E2    Fetch next character
#D191   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D194   53 E8                   #E853 to evaluate expression into #0033/4
#D196   F0 0D       BEQ D1A5    Error if Step value = 0
#D198   84 0A       STY 0A      otherwise save the step value
#D19A   48          PHA
#D19B   68          PLA
#D19C   D0 07       BNE D1A5    Error if step > 255
#D19E   20 E8 00    JSR 00E8    Re-fetch current character
#D1A1   C9 2C       CMP #2C     Test for comma
#D1A3   F0 05       BEQ D1AA
#D1A5   A2 02       LDX #02     Code for 'Invalid command end' error
#D1A7   4C 02 F7    JMP F702    Print error message and exit
#D1AA   20 E2 00    JSR 00E2    Fetch next character
#D1AD   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D1B0   53 E8                   #E853 to evaluate expression into #0033/4
#D1B2   84 00       STY 00      Save the start address
#D1B4   85 01       STA 01      in #00/1
#D1B6   20 E8 00    JSR 00E8    Re-fetch current character
#D1B9   C9 C3       CMP #C3     Test for TO
#D1BB   D0 E8       BNE D1A5    Error if no TO
#D1BD   20 E2 00    JSR 00E2    Fetch next character
#D1C0   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D1C3   53 E8                   #E853 to evaluate expression into #0033/4
#D1C5   85 03       STA 03      Add 1 and save as end address
#D1C7   C8          INY         in #02/3
#D1C8   D0 02       BNE D1CC
#D1CA   E6 03       INC 03
#D1CC   84 02       STY 02
#D1CE   A5 9A       LDA 9A      Fetch Start of basic into #70/1
#D1D0   A4 9B       LDY 9B
#D1D2   85 70       STA 70      and save it into #70/1
#D1D4   84 71       STY 71
#D1D6   A2 00       LDX #00     Zero the checksum for current line
#D1D8   86 06       STX 06
#D1DA   86 07       STX 07
#D1DC   A0 02       LDY #02     Set line position counter to leave spaces for line link pointers
#D1DE   A5 04       LDA 04      Fetch the line number to use
#D1E0   91 70       STA (70),Y  Store it in the line
#D1E2   C8          INY
#D1E3   A5 05       LDA 05
#D1E5   91 70       STA (70),Y
#D1E7   C8          INY
#D1E8   A9 91       LDA #91     Add the token for DATA
#D1EA   91 70       STA (70),Y
#D1EC   C8          INY
#D1ED   A9 20       LDA #20
#D1EF   91 70       STA (70),Y  Add a space
#D1F1   C8          INY
#D1F2   A2 00       LDX #00
#D1F4   A1 00       LDA (00,X)  Fetch a byte from memory
#D1F6   48          PHA         Save it on the stack
#D1F7   18          CLC
#D1F8   65 06       ADC 06      Update the checksum
#D1FA   90 02       BCC D1FE
#D1FC   E6 07       INC 07
#D1FE   85 06       STA 06
#D200   68          PLA         Get the byte back
```

```
#D201   20 80 D2    JSR D280      Add the byte to the line as 2 ASCII characters
#D204   A9 2C       LDA #2C
#D206   91 70       STA (70),Y    Add a comma
#D208   C8          INY
#D209   E6 00       INC 00        Increment pointer to next data byte
#D20B   D0 02       BNE D20F
#D20D   E6 01       INC 01
#D20F   A5 01       LDA 01
#D211   C5 03       CMP 03        Test for reaching the end of code
#D213   90 06       BCC D21B
#D215   A5 00       LDA 00
#D217   C5 02       CMP 02
#D219   B0 53       BCS D26E      Branch if all done
#D21B   C0 35       CPY #35       Test the length of the line so far
#D21D   90 D3       BCC D1F2      Branch back to do 16 data bytes per line
#D21F   20 3D D2    JSR D23D      Do end of line
#D222   E8          INX
#D223   8A          TXA
#D224   C5 A7       CMP A7        Test proximity to HIMEM
#D226   90 AE       BCC D1D6      Carry on if HIMEM not approached
#D228   A9 FF       LDA #FF
#D22A   85 A9       STA A9
#D22C   A9 00       LDA #00
#D22E   A8          TAY
#D22F   91 70       STA (70),Y    Otherwise, finish off basic with 2 null bytes
#D231   C8          INY
#D232   91 70       STA (70),Y
#D234   20 6A E2    JSR E26A      Set line link pointers
#D237   20 A3 F7    JSR F7A3      Call routine in original ROM at
#D23A   7C C4                     #C47C to print error message and exit
#D23C   60          RTS

                                  Process the end of line
#D23D   18          CLC
#D23E   A5 04       LDA 04        Update #04/5 by STEP value to next line number
#D240   65 0A       ADC 0A
#D242   90 02       BCC D246
#D244   E6 05       INC 05
#D246   85 04       STA 04
#D248   A5 07       LDA 07        Fetch a byte of the checksum
#D24A   20 80 D2    JSR D280      Add it to the line as 2 ASCII characters
#D24D   A5 06       LDA 06        Same for other byte of checksum
#D24F   20 80 D2    JSR D280
#D252   A9 00       LDA #00
#D254   91 70       STA (70),Y    Add a null to end the line
#D256   98          TYA           Work out the line link pointer
#D257   A6 71       LDX 71
#D259   38          SEC
#D25A   65 70       ADC 70
#D25C   90 01       BCC D25F
#D25E   E8          INX
#D25F   A0 00       LDY #00
#D261   91 70       STA (70),Y    Put the line link pointer into the line
#D263   C8          INY
#D264   48          PHA
#D265   8A          TXA
#D266   91 70       STA (70),Y
#D268   68          PLA
#D269   85 70       STA 70        Save as pointer to the next line
#D26B   86 71       STX 71
#D26D   60          RTS           Return
```

```
                                        End reached
#D26E   20 3D D2        JSR D23D        Finish off the current line
#D271   A9 00           LDA #00
#D273   A8              TAY
#D274   91 70           STA (70),Y      Mark end of basic with 2 null bytes
#D276   C8              INY
#D277   91 70           STA (70),Y
#D279   A9 FF           LDA #FF
#D27B   85 A9           STA A9
#D27D   4C 6A E2        JMP E26A        Reset basic pointers and exit
                                        Add a byte to the line as 2 ASCII characters
#D280   48              PHA             Save the byte
#D281   4A              LSR
#D282   4A              LSR
#D283   4A              LSR
#D284   4A              LSR             Take the high nibble
#D285   20 8B D2        JSR D28B        Save it as ASCII character
#D288   68              PLA             Recover the byte
#D289   29 0F           AND #0F         Take the low nibble
#D28B   09 30           ORA #30         Test for 0 to 9 or A to F
#D28D   C9 3A           CMP #3A
#D28F   90 02           BCC D293
#D291   69 06           ADC #06         Add 6 for A to F
#D293   91 70           STA (70),Y      Save the character in the line
#D295   C8              INY
#D296   60              RTS             Exit
```

**!EDIT**

```
#D297   20 A3 F7        JSR F7A3        Call routine in original ROM at
#D29A   53 E8                           #E853 to evaluate expression into #0033/4
#D29C   20 A3 F7        JSR F7A3        Call routine in original ROM at
#D29F   B3 C6                           #C6B3 to locate the first line (or next one)
#D2A1   A5 CE           LDA CE          Save the start address of the first line
#D2A3   A4 CF           LDY CF          in #00/1
#D2A5   85 00           STA 00
#D2A7   84 01           STY 01
#D2A9   20 E8 00        JSR 00E8        Re-fetch current character
#D2AC   C9 C3           CMP #C3         Test for TO
#D2AE   D0 10           BNE D2C0        Branch if no TO (single line to delete)
#D2B0   A9 FA           LDA #FA         Set default end line number
#D2B2   85 33           STA 33
#D2B4   85 34           STA 34
#D2B6   20 E2 00        JSR 00E2        Fetch next character
#D2B9   F0 05           BEQ D2C0        Branch if end (delete to end of basic)
#D2BB   20 A3 F7        JSR F7A3        Call routine in original ROM at
#D2BE   53 E8                           #E853 to evaluate expression into #0033/4
#D2C0   E6 33           INC 33          Add 1 to the end line number specified
#D2C2   D0 02           BNE D2C6
#D2C4   E6 34           INC 34
#D2C6   20 A3 F7        JSR F7A3        Call routine in original ROM at
#D2C9   B3 C6                           #C6B3 to find the address of that line
#D2CB   A5 CE           LDA CE          Save the address as the start
#D2CD   A4 CF           LDY CF          of memory block to shift
#D2CF   85 02           STA 02
#D2D1   84 03           STY 03
#D2D3   A0 00           LDY #00         Initialise index
#D2D5   20 ED D2        JSR D2ED        Move line link pointer
#D2D8   20 ED D2        JSR D2ED        Move line link pointer
```

```
#D2DB   F0 0D       BEQ D2EA    Exit if end reached
#D2DD   20 ED D2    JSR D2ED    Move line number
#D2E0   20 ED D2    JSR D2ED    Move line number
#D2E3   20 ED D2    JSR D2ED    Move rest of line
#D2E6   D0 FB       BNE D2E3    Continue to end of line
#D2E8   F0 EB       BEQ D2D5    Branch to do next line when null reached
#D2EA   4C 6A E2    JMP E26A    Reset basic pointers and exit

#D2ED   B1 02       LDA (02),Y  Get one byte of data
#D2EF   48          PHA         Save it (for registers)
#D2F0   91 00       STA (00),Y  Save it into the line
#D2F2   C8          INY         Increment the pointers
#D2F3   D0 04       BNE D2F9
#D2F5   E6 01       INC 01
#D2F7   E6 03       INC 03
#D2F9   68          PLA         Recover the byte and registers
#D2FA   60          RTS         Return
```

**!OLD**

```
#D2FB   A9 01       LDA #01     Set the first line link pointer
#D2FD   A8          TAY         to be not null
#D2FE   91 9A       STA (9A),Y
#D300   4C 6A E2    JMP E26A    Reset basic pointers and exit
```

**!RESTORE**

```
#D303   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D306   53 E8                   #E853 to evaluate expression into #0033/4
#D308   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D30B   B3 C6                   #C6B3 to find the specified (or next) line
#D30D   A4 CF       LDY CF
#D30F   A6 CE       LDX CE
#D311   D0 01       BNE D314    Decrement the address
#D313   88          DEY         of the specified line by 1
#D314   CA          DEX         and save it in the DATA pointer
#D315   86 B0       STX B0      at #B0/1
#D317   84 B1       STY B1
#D319   60          RTS         Exit
```

**!DISK or !DISC**

```
#D31A   A2 00       LDX #00     Set tape defaults
#D31C   8E 4D 02    STX 024D    Fast tape speed
#D31F   8E 5A 02    STX 025A    Join off
#D322   8E 5B 02    STX 025B    Verify off
#D325   8E 7F 02    STX 027F    No filename
#D328   E8          INX
#D329   8E B2 02    STX 02B2    Set counter for files to copy (default = 1)
#D32C   AD 0C C0    LDA C00C    Get default drive number
#D32F   8D 2B C1    STA C12B
#D332   20 E8 00    JSR 00E8    Re-fetch current character
#D335   F0 43       BEQ D37A    Start if end of statement
#D337   C9 2C       CMP #2C     Test for comma ,
#D339   F0 2E       BEQ D369    Branch to test for S (Slow tape speed)
#D33B   C9 C3       CMP #C3     Test for TO
#D33D   F0 17       BEQ D356    Branch if TO found
```

```
#D33F  20 A3 F7   JSR F7A3     Call routine in original ROM at
#D342  C8 D8                  #D8C8 to get single byte expression
#D344  8E B2 02   STX 02B2     Save number of files to copy
#D347  F0 31      BEQ D37A     Start if end of statement
#D349  C9 2C      CMP #2C      Test for comma ,
#D34B  F0 1C      BEQ D369     Branch to test for S (Slow tape speed)
#D34D  C9 C3      CMP #C3      Test for TO
#D34F  F0 05      BEQ D356     Branch if TO found
#D351  A2 02      LDX #02      Code for 'Invalid command end'
#D353  4C 02 F7   JMP F702     Print error message and exit
#D356  20 E2 00   JSR 00E2     Fetch next character (after TO)
#D359  B0 F6      BCS D351     Error if not a number 0 to 9
#D35B  E9 2F      SBC #2F      Turn the ASCII into its number
#D35D  8D 2B C1   STA C12B     Save as the destination drive number
#D360  20 E2 00   JSR 00E2     Fetch next character
#D363  F0 15      BEQ D37A     Start if end of statement
#D365  C9 2C      CMP #2C      Test for comma , again
#D367  D0 E8      BNE D351     Error if not
#D369  20 E2 00   JSR 00E2     Fetch next character to test for Slow
#D36C  C9 53      CMP #53      Test for S
#D36E  D0 E1      BNE D351     Error if comma not followed by S
#D370  A9 01      LDA #01      Set tape speed to Slow
#D372  8D 4D 02   STA 024D
#D375  20 E2 00   JSR 00E2     Fetch next character
#D378  D0 D7      BNE D351     Error if not end of statement
#D37A  AE 2B C1   LDX C12B     Start the copying: get the drive number
#D37D  20 E1 E5   JSR E5E1     Test for valid drive number
#D380  20 26 EB   JSR EB26     Close files for writing
#D383  08         PHP
#D384  20 A3 F7   JSR F7A3     Call routine in original ROM at
#D387  6A E7                  #E7A7 to set 6522 for cassette
#D389  20 A3 F7   JSR F7A3     Call routine in original ROM at
#D38C  7D E5                  #E57D to print 'Searching …'
#D38E  20 A3 F7   JSR F7A3     Call routine in original ROM at
#D391  AC E4                  #E4AC to load file header and print 'Found …'
#D394  2C AE 02   BIT 02AE     Ignore file if it's an array
#D397  70 F6      BVS D38E
#D399  A9 04      LDA #04      Test the Start address for loading
#D39A  CD AA 02   CMP 02AA     If it is below #0400,
#D39D  90 09      BCC D3A8
#D39F  EE AA 02   INC 02AA
#D3A2  EE AC 02   INC 02AC
#D3A5  E8         INX          increase it until it is in #0500
#D3A6  10 F2      BPL D39A
#D3A8  8E B3 02   STX 02B3     and save the displacement
#D3AB  20 A3 F7   JSR F7A3     Call routine in original ROM at
#D3AE  E0 E4                  E4E0 to load the file
#D3B0  20 A3 F7   JSR F7A3     Call routine in original ROM at
#D3B3  3D E9                  E93D to re-set the 6522
#D3B5  28         PLP
#D3B6  AD AE 02   LDA 02AE     Test for basic file
#D3B9  D0 03      BNE D3BE
#D3BB  20 6A E2   JSR E26A     If basic, do the line link pointers
#D3BE  A9 20      LDA #20
#D3C0  A2 09      LDX #09
#D3C2  9D 2B C1   STA C12B,X   Clear the filename for file to save to disk
#D3C5  CA         DEX
#D3C6  D0 FA      BNE D3C2
#D3C8  BD 93 02   LDA 0293,X   Copy up the tape filename for
#D3CB  F0 10      BEQ D3DD     up to 9 characters
```

```
#D3CD   C9 3F       CMP #3F     replacing any ? and *
#D3CF   F0 07       BEQ D3D8    with spaces
#D3D1   C9 2A       CMP #2A
#D3D3   F0 03       BEQ D3D8
#D3D5   9D 2C C1    STA C12C,X
#D3D8   E8          INX
#D3D9   E0 09       CPX #09
#D3DB   D0 EB       BNE D3C8    Carry on for up to 9 characters
#D3DD   20 C2 F5    JSR F5C2    Look for the specified file on disk
#D3E0   F0 05       BEQ D3E7    Carry on if not already present
#D3E2   A2 09       LDX #09     Error code for 'File already exists' error
#D3E4   4C 02 F7    JMP F702    Print error and exit
#D3E7   A2 02       LDX #02
#D3E9   20 60 F5    JSR F560    Print 'Saving …' and filename
#D3EC   20 02 F6    JSR F602    Set up first sector for save
#D3EF   D0 05       BNE D3F6    Carry on if sufficient disk space
#D3F1   A2 0A       LDX #0A     Code for 'Insufficient disk space' error
#D3F3   4C 02 F7    JMP F702    Print error and exit
#D3F6   A2 FF       LDX #FF     Set default values for write protect,
#D3F8   8E 25 C0    STX C025    Transfer address etc
#D3FB   E8          INX
#D3FC   8E 26 C0    STX C026
#D3FF   8E 2B C0    STX C02B
#D402   8E 2C C0    STX C02C
#D405   8E 3B C1    STX C13B
#D408   AE AE 02    LDX 02AE    Test for Basic
#D40B   D0 04       BNE D411    Branch if not
#D40D   E8          INX
#D40E   8E 2B C0    STX C02B    Mark file if basic
#D411   38          SEC
#D412   AC A9 02    LDY 02A9    Set pointers for start of data transfer
#D415   AD AA 02    LDA 02AA
#D418   84 0C       STY 0C
#D41A   85 0D       STA 0D
#D41C   ED B3 02    SBC 02B3    Put the start address into the header information,
#D41F   8C 27 C0    STY C027    less any displacement from #02B3
#D422   8D 28 C0    STA C028
#D425   38          SEC
#D426   AC AB 02    LDY 02AB    Set pointers for end of data transfer
#D429   AD AC 02    LDA 02AC
#D42C   8C 4B C1    STY C14B
#D42F   8D 4C C1    STA C14C
#D432   ED B3 02    SBC 02B3    Put the end address into the header information,
#D435   8C 29 C0    STY C029    less any displacement from #02B3
#D438   8D 2A C0    STA C02A
#D43B   20 5B E3    JSR E35B    Do the !SAVE
#D43E   CE B2 02    DEC 02B2    Decrement the number of files to copy
#D441   F0 03       BEQ D446    Exit if finished
#D443   4C 83 D3    JMP D383    otherwise go back for the next one
#D446   60          RTS         Exit
```

**!DIS**

```
#D447   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D44A   CE CC                   #CCCE to clear the screen
#D44C   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D44F   F5 E5                   #E5F5 to clear the status line
#D451   A9 02       LDA #02     Enable printout to screen turn cursor off.
#D453   8D 6A 02    STA 026A
```

```
#D456   A9 FA       LDA #FA
#D458   85 06       STA 06      Set default end address for disassembly
#D45A   A9 FF       LDA #FF
#D45C   8D 0C 02    STA 020C    Set CAPS on
#D45F   85 05       STA 05      Set flag for – key being pressed
#D461   85 07       STA 07      Set default end address for disassembly
#D463   85 0A       STA 0A      Set Code / Data flag to Code (default)
#D465   A9 7F       LDA #7F
#D467   85 0B       STA 0B      Set default flag for continuous mode to OFF
#D469   8D F0 02    STA 02F0    Set printer flag
#D46C   A9 D2       LDA #D2     Set vector for start of screen
#D46E   85 70       STA 70      to #BBD2
#D470   A9 BB       LDA #BB
#D472   85 71       STA 71
#D474   A0 00       LDY #00
#D476   A9 3E       LDA #3E     > character (prompt)
#D478   91 70       STA (70),Y  put it on the screen
#D47A   C8          INY         Increment screen (column) counter
#D47B   A9 23       LDA #23     # character
#D47D   91 70       STA (70),Y  put it on the screen
#D47F   C8          INY         Increment screen (column) counter
                                Get a start address from the keyboard
#D480   A2 01       LDX #01     Key counter
#D482   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D485   E8 C5                   #C5E8 to get a key from keyboard into A
#D487   85 00       STA 00      Save it in #00
#D489   E0 05       CPX #05     Test for 5th key press
#D48B   F0 21       BEQ D4AE    Branch on 5th key
#D48D   E0 0A       CPX #0A     Test for 10th key press
#D48F   F0 1D       BEQ D4AE    Branch on 10th key
#D491   38          SEC         Process the key
#D492   E9 30       SBC #30
#D494   90 18       BCC D4AE    Reject keys < 0
#D496   C9 0A       CMP #0A
#D498   90 0A       BCC D4A4    Accept if between 0 and 9
#D49A   E9 07       SBC #07     Test for A to F
#D49C   C9 0A       CMP #0A
#D49E   90 0E       BCC D4AE    Reject if key < A
#D4A0   C9 10       CMP #10
#D4A2   B0 0A       BCS D4AE    Reject if key > F
#D4A4   95 00       STA 00,X    Accept the key and save it in
#D4A6   E8          INX         #01 to #09 for the first 9 keys pressed
#D4A7   A5 00       LDA 00      Get the character back again
#D4A9   91 70       STA (70),Y  put it on the screen
#D4AB   C8          INY         Increment screen (column) counter
#D4AC   10 D4       BPL D482    Go back for next key

#D4AE   A5 00       LDA 00      Get the key back
#D4B0   C9 2D       CMP #2D     Test for  -  key
#D4B2   D0 15       BNE D4C9    Branch if it isn't
#D4B4   24 05       BIT 05      Test whether it is there already
#D4B6   10 CA       BPL D482    Reject it if already there
#D4B8   E0 05       CPX #05     Is this the 5th key?
#D4BA   D0 C6       BNE D482    Reject the – if it isn't
#D4BC   85 05       STA 05      otherwise, accept it and store it in #05
#D4BE   E8          INX         Increment key counter
#D4BF   91 70       STA (70),Y  put the - on the screen
#D4C1   C8          INY         Increment screen (column) counter
#D4C2   A9 23       LDA #23     # character to follow the -
#D4C4   91 70       STA (70),Y  put it on the screen
```

| | | | |
|---|---|---|---|
| #D4C6 | C8 | INY | Increment screen (column) counter |
| #D4C7 | 10 B9 | BPL D482 | Go back for next key |
| | | | |
| #D4C9 | C9 0D | CMP #0D | Test for Return key |
| #D4CB | D0 0A | BNE D4D7 | Branch if it isn't |
| #D4CD | E0 05 | CPX #05 | Is this the 5th key? |
| #D4CF | F0 0C | BEQ D4DD | Accept it if it is the 5th key |
| #D4D1 | E0 0A | CPX #0A | |
| #D4D3 | F0 08 | BEQ D4DD | Accept also as 10th key |
| #D4D5 | D0 AB | BNE D482 | Otherwise reject it and go back for another |
| | | | |
| #D4D7 | 20 7E D6 | JSR D67E | Test for other valid keys (P M X L Del) |
| #D4DA | 4C 82 D4 | JMP D482 | Go back for another key |

Carry on getting valid keys until Return is pressed after the 5th or 10th key

| | | | |
|---|---|---|---|
| #D4DD | A2 01 | LDX #01 | Set key counter to first key |
| #D4DF | 20 07 D7 | JSR D707 | Turn the 4 numbers to an address at #01/2 |
| #D4E2 | 24 05 | BIT 05 | Test if there is a second address |
| #D4E4 | 30 05 | BMI D4EB | Branch if there isn't |
| #D4E6 | A2 06 | LDX #06 | Set key counter to 6th key |
| #D4E8 | 20 07 D7 | JSR D707 | Turn the 4 numbers to an address at #06/7 |
| #D4EB | 20 31 D6 | JSR D631 | Print a line if LPRINT is on, and move to next line of screen |
| #D4EE | A5 02 | LDA 02 | Test if the end address has been reached yet |
| #D4F0 | C5 07 | CMP 07 | |
| #D4F2 | 90 10 | BCC D504 | Branch if end not reached |
| #D4F4 | D0 06 | BNE D4FC | Branch if at end |
| #D4F6 | A5 06 | LDA 06 | |
| #D4F8 | C5 01 | CMP 01 | |
| #D4FA | B0 08 | BCS D504 | Branch if end not reached |
| #D4FC | 20 A3 F7 | JSR F7A3 | Call routine in original ROM at |
| #D4FF | E8 C5 | | #C5E8 to wait for a key from the keyboard |
| #D501 | 4C 47 D4 | JMP D447 | Go back to the start of !DIS routine to start again, |
| | | | |
| | | | Start a new line on screen |
| #D504 | A0 01 | LDY #01 | Set screen position (column) |
| #D506 | A9 23 | LDA #23 | # character |
| #D508 | 91 70 | STA (70),Y | put it on the screen |
| #D50A | C8 | INY | Increment screen (column) counter |
| #D50B | A5 02 | LDA 02 | Fetch high byte of start address |
| #D50D | 20 1C D7 | JSR D71C | Place it on screen as 2 ASCII |
| #D510 | A5 01 | LDA 01 | Fetch the low byte of the start address |
| #D512 | 20 1C D7 | JSR D71C | Place it on screen as 2 ASCII |
| #D515 | A2 04 | LDX #04 | Set counters |
| #D517 | A0 04 | LDY #04 | |
| #D519 | B1 01 | LDA (01),Y | Copy 5 bytes of code from their location |
| #D51B | 95 74 | STA 74,X | into a buffer at #74 - #78 |
| #D51D | CA | DEX | |
| #D51E | 88 | DEY | |
| #D51F | 10 F8 | BPL D519 | |
| #D521 | A5 0A | LDA 0A | Test the flag to disassemble as code or data |
| #D523 | 30 12 | BMI D537 | Branch for code |
| #D525 | A9 44 | LDA #44 | D |
| #D527 | 85 7B | STA 7B | |
| #D529 | A9 54 | LDA #54 | T |
| #D52B | 85 7C | STA 7C | |
| #D52D | A9 41 | LDA #41 | A |
| #D52F | 85 7D | STA 7D | Put DTA as the 'OPCODE' in #7B-D |
| #D531 | A9 05 | LDA #05 | Set the number of bytes per line to 5 |
| #D533 | 85 7E | STA 7E | |
| #D535 | 10 34 | BPL D56B | Branch always |

```
#D537   A9 D7       LDA #D7     Set high byte of address for OPCODE lookup table
#D539   85 7A       STA 7A
#D53B   A9 EB       LDA #EB     Low byte of start of OPCODE lookup table
#D53D   A2 03       LDX #03     When disassembling code, the OPCODE is the
#D53F   18          CLC         first byte, stored in #74.  This routine works out
#D540   65 74       ADC 74      where to find it in the OPCODE lookup table
#D542   90 02       BCC D546    by adding 4x the OPCODE to the start address
#D544   E6 7A       INC 7A      of the table.  Each entry is 4 bytes long.
#D546   CA          DEX         Bytes 1-3 are the ASCII for the mnemonic
#D547   10 F6       BPL D53F    and the 4th byte is a code to indicate its type.
#D549   85 79       STA 79      The lookup table address ends up in #79/A
#D54B   A2 04       LDX #04     Set counters
#D54D   A0 03       LDY #03
#D54F   B1 79       LDA (79),Y  Get the 4th byte from the OPCODE table.
#D551   48          PHA         Save it
#D552   29 0F       AND #0F     Take the low nibble
#D554   85 7F       STA 7F      Save it as a code for formatting
#D556   CA          DEX
#D557   68          PLA         Get the byte back
#D558   4A          LSR         Take the high nibble
#D559   4A          LSR
#D55A   4A          LSR
#D55B   4A          LSR
#D55C   38          SEC
#D55D   E9 02       SBC #02
#D55F   85 7E       STA 7E      Save it as the number of bytes used by this OPCODE
#D561   CA          DEX
#D562   88          DEY
#D563   B1 79       LDA (79),Y  Copy 3 bytes for the OPCODE mnemonic
#D565   95 7B       STA 7B,X    into #7B-D
#D567   CA          DEX         End up with mnemonic in #7B-D
#D568   88          DEY         Number of bytes used by this OPCODE in #7E
#D569   10 F8       BPL D563    and a formatting code in #7F
```

The next part is used to process code or data.

```
#D56B   A5 7E       LDA 7E      Save a copy of the number of bytes
#D56D   85 80       STA 80      used by the current OPCODE as a counter.
#D56F   18          CLC         Move the start address pointer in #01/2 on by
#D570   65 01       ADC 01      this number of bytes, ready for the next one.
#D572   90 02       BCC D576
#D574   E6 02       INC 02
#D576   85 01       STA 01
#D578   A0 07       LDY #07
#D57A   A9 00       LDA #00
#D57C   85 72       STA 72
#D57E   A6 72       LDX 72
#D580   B5 74       LDA 74,X    Get a byte of code
#D582   20 1C D7    JSR D71C    Place it on screen as 2 ASCII
#D585   C8          INY         Increment screen pointer to make a space.
#D586   E6 72       INC 72
#D588   C6 80       DEC 80      Decrement the byte counter
#D58A   D0 F2       BNE D57E    Loop back to continue
#D58C   A2 00       LDX #00     Set counter
#D58E   A0 16       LDY #16     Set screen position (column)
#D590   B5 7B       LDA 7B,X    Get a character from the mnemonic.
#D592   91 70       STA (70),Y  put it on the screen
#D594   C8          INY
#D595   E8          INX
#D596   E0 03       CPX #03     Carry on for the 3 characters of the mnemonic.
```

```
#D598    D0 F6        BNE D590
#D59A    C8           INY          move along screen
#D59B    A5 0A        LDA 0A       Test the data / code flag
#D59D    30 1A        BMI D5B9     branch if code.

Process as DATA
#D59F    A2 00        LDX #00      Set counter
#D5A1    B5 74        LDA 74,X     Get a byte of data
#D5A3    C9 20        CMP #20      If below #20, use #20 (a space)
#D5A5    10 02        BPL D5A9
#D5A7    A9 20        LDA #20
#D5A9    C9 7E        CMP #7E      If above #7E, use #20 (a space)
#D5AB    30 02        BMI D5AF
#D5AD    A9 20        LDA #20
#D5AF    91 70        STA (70),Y   put it on the screen as ASCII (or a space)
#D5B1    C8           INY
#D5B2    E8           INX
#D5B3    E0 05        CPX #05
#D5B5    D0 EA        BNE D5A1     Carry on for 5 bytes of data
#D5B7    F0 46        BEQ D5FF     Branch when done.

#D5B9    A6 7F        LDX 7F       Test the formatting code.
#D5BB    F0 42        BEQ D5FF     Branch if 0 (nothing to follow it)
#D5BD    CA           DEX
#D5BE    D0 06        BNE D5C6
#D5C0    20 33 D7     JSR D733     Process format code 1
#D5C3    4C FF D5     JMP D5FF     and move on.
#D5C6    CA           DEX
#D5C7    D0 06        BNE D5CF
#D5C9    20 3D D7     JSR D73D     Process format code 2
#D5CC    4C FF D5     JMP D5FF     and move on.
#D5CF    CA           DEX
#D5D0    D0 06        BNE D5D8
#D5D2    20 4D D7     JSR D74D     Process format code 3
#D5D5    4C FF D5     JMP D5FF     and move on.
#D5D8    CA           DEX
#D5D9    D0 06        BNE D5E1
#D5DB    20 5B D7     JSR D75B     Process format code 4
#D5DE    4C FF D5     JMP D5FF     and move on.
#D5E1    CA           DEX
#D5E2    D0 06        BNE D5EA
#D5E4    20 69 D7     JSR D769     Process format code 5
#D5E7    4C FF D5     JMP D5FF     and move on.
#D5EA    CA           DEX
#D5EB    D0 06        BNE D5F3
#D5ED    20 77 D7     JSR D777     Process format code 6
#D5F0    4C FF D5     JMP D5FF     and move on.
#D5F3    CA           DEX
#D5F4    D0 06        BNE D5FC
#D5F6    20 7D D7     JSR D77D     Process format code 7
#D5F9    4C FF D5     JMP D5FF     and move on.
#D5FC    20 8B D7     JSR D78B     Process format code 8.

#D5FF    20 31 D6     JSR D631     Print a line if LPRINT is on, and move to next line of screen.
#D602    24 0B        BIT 0B       Test the CONT (continuous mode) flag.
#D604    30 15        BMI D61B     Branch if in continuous mode.
#D606    A5 71        LDA 71       Test screen position.
#D608    C9 BF        CMP #BF      Near bottom of screen?
#D60A    D0 22        BNE D62E     Branch if not (to do a screen full)
#D60C    20 A3 F7     JSR F7A3     Call routine in original ROM at
```

```
#D60F   E8 C5                   #C5E8 to read a key from the keyboard.
#D611   C9 20       CMP #20     Space bar
#D613   F0 06       BEQ D61B    Carry on if space bar pressed.
#D615   20 7E D6    JSR D67E    Test and process other valid keys
#D618   4C 0C D6    JMP D60C    until a valid key is pressed.

#D61B   A5 71       LDA 71      Test screen position
#D61D   C9 BF       CMP #BF     Near bottom of screen?
#D61F   D0 0D       BNE D62E    Branch if not to go back for more.
#D621   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D624   CE CC                   #CCCE to Clear the screen.
#D626   A9 D2       LDA #D2     Re-set the screen pointers
#D628   85 70       STA 70      in #70/1
#D62A   A9 BB       LDA #BB
#D62C   85 71       STA 71
#D62E   4C EE D4    JMP D4EE    Go back for more.

#D631   AD F0 02    LDA 02F0    Test if the LPRINT flag is on
#D634   10 3C       BPL D672    Branch if it isn't
#D636   8D F1 02    STA 02F1    Toggle output to printer to on
#D639   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D63C   6A E7                   #E76A to set 6522 to cassette
#D63E   A9 1B       LDA #1B     ESC character
#D640   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D643   C1 F5                   #F5C1 to send a character to the printer
#D645   A9 6C       LDA #6C     l character
#D647   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D64A   C1 F5                   #F5C1 to send a character to the printer
#D64C   A9 05       LDA #05     ESC l 5 string to initialise printer
#D64E   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D651   C1 F5                   #F5C1 to send a character to the printer
#D653   A0 00       LDY #00     Counter for screen address
#D655   84 73       STY 73
#D657   B1 70       LDA (70),Y  Get a character from the screen
#D659   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D65C   C1 F5                   #F5C1 to send a character to the printer
#D65E   A4 73       LDY 73
#D660   C8          INY
#D661   C0 26       CPY #26     Move along the screen for 38 characters
#D663   D0 F0       BNE D655    Loop to copy a screen line to printer.
#D665   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D668   F0 CB                   #CBF0 to print a new line
#D66A   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D66D   40 E9                   #E940 to re-set 6522 and some counters
#D66F   4E F1 02    LSR #02F1   Toggle output to printer off

#D672   A5 70       LDA 70      Come in here if LPRINT flag is off
#D674   18          CLC         Move pointers onto next line
#D675   69 28       ADC #28     by adding 40
#D677   90 02       BCC D67B
#D679   E6 71       INC 71
#D67B   85 70       STA 70
#D67D   60          RTS         Return

Test and process other valid key presses.
#D67E   C9 58       CMP #58     Test for X (for exit)
#D680   D0 0D       BNE D68F
#D682   20 A3 F7    JSR F7A3    Call routine in original ROM at
#D685   F5 E5                   #E5F5 to clear status message line
#D687   A9 03       LDA #03
```

```
#D689   8D 6A 02    STA 026A    Reset screen output and cursor.
#D68C   68          PLA         Lose the last return address
#D68D   68          PLA
#D68E   60          RTS         Exit disassembler.

#D68F   C9 7F       CMP #7F     Test for DEL key – re-start
#D691   D0 05       BNE D698
#D693   68          PLA         Lose the last return address.
#D694   68          PLA
#D695   4C 47 D4    JMP D447    Go back to the start of disassembler.

#D698   86 72       STX 72      Save X
#D69A   84 73       STY 73      and Y in temporary store.
#D69C   C9 50       CMP #50     Test for P – toggle LPRINT mode
#D69E   D0 22       BNE D6C2
#D6A0   AD F0 02    LDA 02F0    Test LPRINT flag.  #7F is off.
#D6A3   30 0B       BMI D6B0    Branch if it's on (to toggle off)
#D6A5   09 80       ORA #80
#D6A7   8D F0 02    STA 02F0    Toggle LPRINT mode to ON
#D6AA   A9 C8       LDA #C8
#D6AC   A0 D7       LDY #D7     #D7C8 is address for LPRINT message
#D6AE   D0 0B       BNE D6BB    Branch to put message on status line.
#D6B0   29 7F       AND #7F
#D6B2   8D F0 02    STA 02F0    Toggle LPRINT mode to OFF
#D6B5   85 0B       STA 0B      Re-set the CONTinuous mode to OFF
#D6B7   A9 DE       LDA #DE     #D7DE is the address for a blank message
#D6B9   A0 D7       LDY #D7     to clear the LPRINT message.
#D6BB   A2 0B       LDX #0B     Set position of message on status line.
#D6BD   20 B6 D7    JSR D7B6    Put the message on the status line.
#D6C0   F0 40       BEQ D702    Branch to recover X and Y, then return.

#D6C2   C9 4D       CMP #4D     Test for M  - mode flag for data or code.
#D6C4   D0 1D       BNE D6E3
#D6C6   A5 0A       LDA 0A      Test current flag.  #7A=DATA.  #FF=CODE.
#D6C8   30 0A       BMI D6D4    Branch if CODE to toggle to DATA.
#D6CA   09 80       ORA #80
#D6CC   85 0A       STA 0A      Toggle to CODE
#D6CE   A9 CF       LDA #CF
#D6D0   A0 D7       LDY #D7     #D7CF is address for CODE message.
#D6D2   D0 08       BNE D6DC    Branch to put the message on the status line.
#D6D4   29 7F       AND #7F
#D6D6   85 0A       STA 0A      Toggle to DATA
#D6D8   A9 D4       LDA #D4
#D6DA   A0 D7       LDY #D7     #D7D4 is the address for DATA message.
#D6DC   A2 05       LDX #05     Set position of the message on the status line.
#D6DE   20 B6 D7    JSR D7B6    Put the message on the status line.
#D6E1   F0 1F       BEQ D702    Branch to recover X and Y, then return.

#D6E3   C9 4C       CMP #4C     Test for L
#D6E5   D0 1B       BNE D702    Branch to recover X and Y, then return.
#D6E7   A5 0B       LDA 0B      Test current flag.  #FF=CONT mode, #7F not.
#D6E9   30 0A       BMI D6F5    Branch if in CONT mode to toggle off.
#D6EB   09 80       ORA #80
#D6ED   85 0B       STA 0B      Set CONT mode to ON
#D6EF   A9 D9       LDA #D9
#D6F1   A0 D7       LDY #D7     #D7D9 is the address for the CONT message.
#D6F3   D0 08       BNE D6FD    Branch to put the message on the status line.
#D6F5   29 7F       AND #7F
#D6F7   85 0B       STA 0B      Set CONT mode to OFF
#D6F9   A9 E6       LDA #E6     #D7E6 is the address for a blank message
```

```
#D6FB   A0 D7       LDY #D7     to clear the CONT
#D6FD   A2 13       LDX #13     Set position of the message on the status line.
#D6FF   20 B6 D7    JSR D7B6    Put the message on the status line.
#D702   A6 72       LDX 72      Recover X
#D704   A4 73       LDY 73      and Y registers
#D706   60          RTS         and return.
```

The following routine processes 4 numbers between 0 and F (e.g. in #01 to #04) and turns them into a 2 byte address stored in the first 2 places (#01/2)

```
#D707   B5 00       LDA 00,X    Fetch the number represented by the 1st key
#D709   0A          ASL         Shift the 4 bits to the high nibble
#D70A   0A          ASL         to turn 0 to F into #00 to #F0
#D70B   0A          ASL
#D70C   0A          ASL
#D70D   75 01       ADC 01,X    Add the 2nd key (low nibble) to give the
#D70F   95 01       STA 01,X    high byte of the address and store it.
#D711   B5 02       LDA 02,X    Fetch the number represented by the 3rd key
#D713   0A          ASL         Shift the 4 bits to the high nibble
#D714   0A          ASL
#D715   0A          ASL
#D716   0A          ASL
#D717   75 03       ADC 03,X    Add the 4th key (low nibble) to give the
#D719   95 00       STA 00,X    low byte of the address and store it.
#D71B   60          RTS         Return
```

The following routine takes a single byte, turns it into 2 ASCII characters and places them on screen.

```
#D71C   48          PHA         Save the byte
#D71D   4A          LSR         Shift the 4 high bits to the low nibble
#D71E   4A          LSR
#D71F   4A          LSR
#D720   4A          LSR
#D721   20 27 D7    JSR D727    Turn it into ASCII for 0-9 A-F and put it on screen
#D724   68          PLA         Get the byte back
#D725   29 0F       AND #0F     Strip off the 4 high bits
#D727   09 30       ORA #30     Turn it into its ASCII character
#D729   C9 3A       CMP #3A
#D72B   90 02       BCC D72F
#D72D   69 06       ADC #06     Add 6 for A to F
#D72F   91 70       STA (70),Y  Place it on the screen
#D731   C8          INY         Increment screen (column) counter
#D732   60          RTS         Return

Process format code 1           Immediate   Example LDA #23
#D733   A9 23       LDA #23     #
#D735   91 70       STA (70),Y  Put # on screen
#D737   C8          INY
#D738   A5 75       LDA 75
#D73A   4C 1C D7    JMP D71C    Do a single byte number as 2 ASCII

Process format code 2           Zero page / Abs Example LDA 7E or JSR D71C.
#D73D   A5 7E       LDA 7E      Test number of bytes used by OPCODE
#D73F   C9 02       CMP #02
#D741   F0 05       BEQ D748    Branch if 2, continue if 3
#D743   A5 76       LDA 76      Get the high byte if there is one.
#D745   20 1C D7    JSR D71C    Do a single byte number as 2 ASCII
#D748   A5 75       LDA 75      Get the low byte
#D74A   4C 1C D7    JMP D71C    Do a single byte number as 2 ASCII

Process format code 3           ZP,X or Abs,X   Example LDA 7E,X  or STA BB80,X
#D74D   20 3D D7    JSR D73D    Do the 1 or 2 byte number
```

```
#D750    A9 2C          LDA #2C          add a comma
#D752    91 70          STA (70),Y
#D754    C8             INY
#D755    A9 58          LDA #58          add an X
#D757    91 70          STA (70),Y
#D759    C8             INY
#D75A    60             RTS

Process format code 4   ZP,Y or Abs,Y    Example LDA 7E,Y  or  STA BB80,Y
#D75B    20 3D D7       JSR D73D         Do the 1 or 2 byte number
#D75E    A9 2C          LDA #2C          add a comma
#D760    91 70          STA (70),Y
#D762    C8             INY
#D763    A9 59          LDA #59          add a Y
#D765    91 70          STA (70),Y
#D767    C8             INY
#D768    60             RTS

Process format code 5   (ZP,X)           Example LDA (70,X)
#D769    A9 28          LDA #28          do a (
#D76B    91 70          STA (70),Y
#D76D    C8             INY
#D76E    20 4D D7       JSR D74D         add a 1 or 2 byte number plus ,X
#D771    A9 29          LDA #29          add the )
#D773    91 70          STA (70),Y
#D775    C8             INY
#D776    60             RTS

Process format code 6   (ZP),Y           Example STA (70),Y
#D777    20 7D D7       JSR D77D         Do (1 or 2 byte)
#D77A    4C 5E D7       JMP D75E         followed by ,Y

Process format code 7   (ZP) or (Abs)    Subroutine used as part format
#D77D    A9 28          LDA #28          do a (
#D77F    91 70          STA (70),Y
#D781    C8             INY
#D782    20 3D D7       JSR D73D         Do the 1 or 2 byte number.
#D785    A9 29          LDA #29          add the )
#D787    91 70          STA (70),Y
#D789    C8             INY
#D78A    60             RTS

Process format code 8                   Branches
#D78B    A5 01          LDA 01           Store a working copy of the address
#D78D    85 03          STA 03           of the next instruction.
#D78F    A5 02          LDA 02
#D791    85 04          STA 04
#D793    A5 75          LDA 75           Fetch the offset byte.
#D795    30 0C          BMI D7A3         Branch if negative.
#D797    18             CLC              Process if positive.
#D798    65 03          ADC 03           Add the offset to the next address
#D79A    90 02          BCC D79E         and save it in #03/4.
#D79C    E6 04          INC 04
#D79E    85 03          STA 03
#D7A0    18             CLC
#D7A1    90 09          BCC D7AC         Branch always
#D7A3    18             CLC              Process negative offset
#D7A4    65 03          ADC 03           and save the address in #03/4
#D7A6    B0 02          BCS D7AA
#D7A8    C6 04          DEC 04
```

```
#D7AA   85 03              STA 03
#D7AC   A5 04              LDA 04          Fetch the high byte of the branch address.
#D7AE   20 1C D7           JSR D71C        Do a single byte number as 2 ASCII
#D7B1   A5 03              LDA 03          then do the low byte.
#D7B3   4C 1C D7           JMP D71C        Do a single byte number as 2 ASCII

Place a message on the status line.
#D7B6   85 0C              STA 0C          Save the start address for the required message.
#D7B8   84 0D              STY 0D
#D7BA   A0 00              LDY #00         Set counter
#D7BC   B1 0C              LDA (0C),Y      Get a character of the message
#D7BE   F0 07              BEQ D7C7        Exit if end of message
#D7C0   9D 80 BB           STA BB80,X      Place character on status line
#D7C3   E8                 INX
#D7C4   C8                 INY
#D7C5   D0 F5              BNE D7BC        Go back for next character.
#D7C7   60                 RTS             Return.

#D7C8   4C 50 52 49 4E 54 00  DTA LPRINT   Messages for status line.
#D7CF   43 4F 44 45 00        DTA CODE
#D7D4   44 41 54 41 00        DTA DATA
#D7D9   43 4F 4E 54 00        DTA CONT
#D7DE   20 20 20 20 20        DTA           Blank messages to overwrite the above.
#D7E3   20 20 20 20 20        DTA           Pick the start address to get the required length
#D7E8   20 20 00              DTA           of message, ending with null.

OPCODE Table       Op Code    Mnemonic and formatting code / bytes used.
#D7EB   42 52 4B 30   #00     BRK  0
#D7EF   4F 52 41 45   #01     ORA  E
#D7F3   3F 3F 3F 30   #02     ???  0        No such Op Code
#D7F7   3F 3F 3F 30   #03     ???  0
#D7FB   3F 3F 3F 30   #04     ???  0
#D7FF   4F 52 41 42   #05     ORA  B
#D803   41 53 4C 42   #06     ASL  B
#D807   3F 3F 3F 30   #07     ???  0
#D80B   50 48 50 30   #08     PHP  0
#D80F   4F 52 41 41   #09     ORA  A
#D813   41 53 4C 30   #0A     ASL  0
#D817   3F 3F 3F 30   #0B     ???  0
#D81B   3F 3F 3F 30   #0C     ???  0
#D81F   4F 52 41 52   #0D     ORA  R
#D823   41 53 4C 52   #0E     ASL  R
#D827   3F 3F 3F 30   #0F     ???  0
#D82B   42 50 4C 48   #10     BPL  H
#D82F   4F 52 41 46   #11     ORA  F
#D833   3F 3F 3F 30   #12     ???  0
#D837   3F 3F 3F 30   #13     ???  0
#D83B   3F 3F 3F 30   #14     ???  0
#D83F   4F 52 41 43   #15     ORA  C
#D843   41 53 4C 43   #16     ASL  C
#D847   3F 3F 3F 30   #17     ???  0
#D84B   43 4C 43 30   #18     CLC  0
#D84F   4F 52 41 54   #19     ORA  T
#D853   3F 3F 3F 30   #1A     ???  0
#D857   3F 3F 3F 30   #1B     ???  0
#D85B   3F 3F 3F 30   #1C     ???  0
#D85F   4F 52 41 53   #1D     ORA  S
#D863   41 53 4C 53   #1E     ASL  S
#D867   3F 3F 3F 30   #1F     ???  0
#D86B   4A 53 52 52   #20     JSR  R
```

| | | | |
|---|---|---|---|
| #D86F | 41 4E 44 45 | #21 | AND E |
| #D873 | 3F 3F 3F 30 | #22 | ??? 0 |
| #D877 | 3F 3F 3F 30 | #23 | ??? 0 |
| #D87B | 42 49 54 42 | #24 | BIT B |
| #D87F | 41 4E 44 42 | #25 | AND B |
| #D883 | 52 4F 4C 42 | #26 | ROL B |
| #D887 | 3F 3F 3F 30 | #27 | ??? 0 |
| #D88B | 50 4C 50 30 | #28 | PLP 0 |
| #D88F | 41 4E 44 41 | #29 | AND A |
| #D893 | 52 4F 4C 30 | #2A | ROL 0 |
| #D897 | 3F 3F 3F 30 | #2B | ??? 0 |
| #D89B | 42 49 54 52 | #2C | BIT R |
| #D89F | 41 4E 44 52 | #2D | AND R |
| #D8A3 | 52 4F 4C 52 | #2E | ROL R |
| #D8A7 | 3F 3F 3F 30 | #2F | ??? 0 |
| #D8AB | 42 4D 49 48 | #30 | BMI H |
| #D8AF | 41 4E 44 46 | #31 | AND F |
| #D8B3 | 3F 3F 3F 30 | #32 | ??? 0 |
| #D8B7 | 3F 3F 3F 30 | #33 | ??? 0 |
| #D8BB | 3F 3F 3F 30 | #34 | ??? 0 |
| #D8BF | 41 4E 44 43 | #35 | AND C |
| #D8C3 | 52 4F 4C 43 | #36 | ROL C |
| #D8C7 | 3F 3F 3F 30 | #37 | ??? 0 |
| #D8CB | 53 45 43 30 | #38 | SEC 0 |
| #D8CF | 41 4E 44 54 | #39 | AND T |
| #D8D3 | 3F 3F 3F 30 | #3A | ??? 0 |
| #D8D7 | 3F 3F 3F 30 | #3B | ??? 0 |
| #D8DB | 3F 3F 3F 30 | #3C | ??? 0 |
| #D8DF | 41 4E 44 53 | #3D | AND S |
| #D8E3 | 52 4F 4C 53 | #3E | ROL S |
| #D8E7 | 3F 3F 3F 30 | #3F | ??? 0 |
| #D8EB | 52 54 49 30 | #40 | RTI 0 |
| #D8EF | 45 4F 52 45 | #41 | EOR E |
| #D8F3 | 3F 3F 3F 30 | #42 | ??? 0 |
| #D8F7 | 3F 3F 3F 30 | #43 | ??? 0 |
| #D8FB | 3F 3F 3F 30 | #44 | ??? 0 |
| #D8FF | 45 4F 52 42 | #45 | EOR B |
| #D903 | 4C 53 52 42 | #46 | LSR B |
| #D907 | 3F 3F 3F 30 | #47 | ??? 0 |
| #D90B | 50 48 41 30 | #48 | PHA 0 |
| #D90F | 45 4F 52 41 | #49 | EOR A |
| #D913 | 4C 53 52 30 | #4A | LSR 0 |
| #D917 | 3F 3F 3F 30 | #4B | ??? 0 |
| #D91B | 4A 4D 50 52 | #4C | JMP R |
| #D91F | 45 4F 52 52 | #4D | EOR R |
| #D923 | 4C 53 52 52 | #4E | LSR R |
| #D927 | 3F 3F 3F 30 | #4F | ??? 0 |
| #D92B | 42 56 43 48 | #50 | BVC H |
| #D92F | 45 4F 52 46 | #51 | EOR F |
| #D933 | 3F 3F 3F 30 | #52 | ??? 0 |
| #D937 | 3F 3F 3F 30 | #53 | ??? 0 |
| #D93B | 3F 3F 3F 30 | #54 | ??? 0 |
| #D93F | 45 4F 52 43 | #55 | EOR C |
| #D943 | 4C 53 52 43 | #56 | LSR C |
| #D947 | 3F 3F 3F 30 | #57 | ??? 0 |
| #D94B | 43 4C 49 30 | #58 | CLI 0 |
| #D94F | 45 4F 52 54 | #59 | EOR T |
| #D953 | 3F 3F 3F 30 | #5A | ??? 0 |
| #D957 | 3F 3F 3F 30 | #5B | ??? 0 |
| #D95B | 3F 3F 3F 30 | #5C | ??? 0 |

| | | | |
|---|---|---|---|
| #D95F | 45 4F 52 53 | #5D | EOR S |
| #D963 | 4C 53 52 53 | #5E | LSR S |
| #D967 | 3F 3F 3F 30 | #5F | ??? 0 |
| #D96B | 52 54 53 30 | #60 | RTS 0 |
| #D96F | 41 44 43 45 | #61 | ADC E |
| #D973 | 3F 3F 3F 30 | #62 | ??? 0 |
| #D977 | 3F 3F 3F 30 | #63 | ??? 0 |
| #D97B | 3F 3F 3F 30 | #64 | ??? 0 |
| #D97F | 41 44 43 42 | #65 | ADC B |
| #D983 | 52 4F 52 42 | #66 | ROR B |
| #D987 | 3F 3F 3F 30 | #67 | ??? 0 |
| #D98B | 50 4C 41 30 | #68 | PLA 0 |
| #D98F | 41 44 43 41 | #69 | ADC A |
| #D993 | 52 4F 52 30 | #6A | ROR 0 |
| #D997 | 3F 3F 3F 30 | #6B | ??? 0 |
| #D99B | 4A 4D 50 57 | #6C | JMP W |
| #D99F | 41 44 43 52 | #6D | ADC R |
| #D9A3 | 52 4F 52 52 | #6E | ROR R |
| #D9A7 | 3F 3F 3F 30 | #6F | ??? 0 |
| #D9AB | 42 56 53 48 | #70 | BVS H |
| #D9AF | 41 44 43 46 | #71 | ADC F |
| #D9B3 | 3F 3F 3F 30 | #72 | ??? 0 |
| #D9B7 | 3F 3F 3F 30 | #73 | ??? 0 |
| #D9BB | 3F 3F 3F 30 | #74 | ??? 0 |
| #D9BF | 41 44 43 43 | #75 | ADC C |
| #D9C3 | 52 4F 52 43 | #76 | ROR C |
| #D9C7 | 3F 3F 3F 30 | #77 | ??? 0 |
| #D9CB | 53 45 49 30 | #78 | SEI 0 |
| #D9CF | 41 44 43 54 | #79 | ADC T |
| #D9D3 | 3F 3F 3F 30 | #7A | ??? 0 |
| #D9D7 | 3F 3F 3F 30 | #7B | ??? 0 |
| #D9DB | 3F 3F 3F 30 | #7C | ??? 0 |
| #D9DF | 41 44 43 53 | #7D | ADC S |
| #D9E3 | 52 4F 52 53 | #7E | ROR S |
| #D9E7 | 3F 3F 3F 30 | #7F | ??? 0 |
| #D9EB | 3F 3F 3F 30 | #80 | ??? 0 |
| #D9EF | 53 54 41 45 | #81 | STA E |
| #D9F3 | 3F 3F 3F 30 | #82 | ??? 0 |
| #D9F7 | 3F 3F 3F 30 | #83 | ??? 0 |
| #D9FB | 53 54 59 42 | #84 | STY B |
| #D9FF | 53 54 41 42 | #85 | STA B |
| #DA03 | 53 54 58 42 | #86 | STX B |
| #DA07 | 3F 3F 3F 30 | #87 | ??? 0 |
| #DA0B | 44 45 59 30 | #88 | DEY 0 |
| #DA0F | 3F 3F 3F 30 | #89 | ??? 0 |
| #DA13 | 54 58 41 30 | #8A | TXA 0 |
| #DA17 | 3F 3F 3F 30 | #8B | ??? 0 |
| #DA1B | 53 54 59 52 | #8C | STY R |
| #DA1F | 53 54 41 52 | #8D | STA R |
| #DA23 | 53 54 58 52 | #8E | STX R |
| #DA27 | 3F 3F 3F 30 | #8F | ??? 0 |
| #DA2B | 42 43 43 48 | #90 | BCC H |
| #DA2F | 53 54 41 46 | #91 | STA F |
| #DA33 | 3F 3F 3F 30 | #92 | ??? 0 |
| #DA37 | 3F 3F 3F 30 | #93 | ??? 0 |
| #DA3B | 53 54 59 43 | #94 | STY C |
| #DA3F | 53 54 41 43 | #95 | STA C |
| #DA43 | 53 54 58 44 | #96 | STX D |
| #DA47 | 3F 3F 3F 30 | #97 | ??? 0 |
| #DA4B | 54 59 41 30 | #98 | TYA 0 |

| | | | |
|---|---|---|---|
| #DA4F | 53 54 41 54 | #99 | STA T |
| #DA53 | 54 58 53 30 | #9A | TXS 0 |
| #DA57 | 3F 3F 3F 30 | #9B | ??? 0 |
| #DA5B | 3F 3F 3F 30 | #9C | ??? 0 |
| #DA5F | 53 54 41 53 | #9D | STA S |
| #DA63 | 3F 3F 3F 30 | #9E | ??? 0 |
| #DA67 | 3F 3F 3F 30 | #9F | ??? 0 |
| #DA6B | 4C 44 59 41 | #A0 | LDY A |
| #DA6F | 4C 44 41 45 | #A1 | LDA E |
| #DA73 | 4C 44 58 41 | #A2 | LDX A |
| #DA77 | 3F 3F 3F 30 | #A3 | ??? 0 |
| #DA7B | 4C 44 59 42 | #A4 | LDY B |
| #DA7F | 4C 44 41 42 | #A5 | LDA B |
| #DA83 | 4C 44 58 42 | #A6 | LDX B |
| #DA87 | 3F 3F 3F 30 | #A7 | ??? 0 |
| #DA8B | 54 41 59 30 | #A8 | TAY 0 |
| #DA8F | 4C 44 41 41 | #A9 | LDA A |
| #DA93 | 54 41 58 30 | #AA | TAX 0 |
| #DA97 | 3F 3F 3F 30 | #AB | ??? 0 |
| #DA9B | 4C 44 59 52 | #AC | LDY R |
| #DA9F | 4C 44 41 52 | #AD | LDA R |
| #DAA3 | 4C 44 58 52 | #AE | LDX R |
| #DAA7 | 3F 3F 3F 30 | #AF | ??? 0 |
| #DAAB | 42 43 53 48 | #B0 | BCS H |
| #DAAF | 4C 44 41 46 | #B1 | LDA F |
| #DAB3 | 3F 3F 3F 30 | #B2 | ??? 0 |
| #DAB7 | 3F 3F 3F 30 | #B3 | ??? 0 |
| #DABB | 4C 44 59 43 | #B4 | LDY C |
| #DABF | 4C 44 41 43 | #B5 | LDA C |
| #DAC3 | 4C 44 58 44 | #B6 | LDX D |
| #DAC7 | 3F 3F 3F 30 | #B7 | ??? 0 |
| #DACB | 43 4C 56 30 | #B8 | CLV 0 |
| #DACF | 4C 44 41 54 | #B9 | LDA T |
| #DAD3 | 54 53 58 30 | #BA | TSX 0 |
| #DAD7 | 3F 3F 3F 30 | #BB | ??? 0 |
| #DADB | 4C 44 59 53 | #BC | LDY S |
| #DADF | 4C 44 41 53 | #BD | LDA S |
| #DAE3 | 4C 44 58 54 | #BE | LDX T |
| #DAE7 | 3F 3F 3F 30 | #BF | ??? 0 |
| #DAEB | 43 50 59 41 | #C0 | CPY A |
| #DAEF | 43 4D 50 45 | #C1 | CMP E |
| #DAF3 | 3F 3F 3F 30 | #C2 | ??? 0 |
| #DAF7 | 3F 3F 3F 30 | #C3 | ??? 0 |
| #DAFB | 43 50 59 42 | #C4 | CPY B |
| #DAFF | 43 4D 50 42 | #C5 | CMP B |
| #DB03 | 44 45 43 42 | #C6 | DEC B |
| #DB07 | 3F 3F 3F 30 | #C7 | ??? 0 |
| #DB0B | 49 4E 59 30 | #C8 | INY 0 |
| #DB0F | 43 4D 50 41 | #C9 | CMP A |
| #DB13 | 44 45 58 30 | #CA | DEX 0 |
| #DB17 | 3F 3F 3F 30 | #CB | ??? 0 |
| #DB1B | 43 50 59 52 | #CC | CPY R |
| #DB1F | 43 4D 50 52 | #CD | CMP R |
| #DB23 | 44 45 43 52 | #CE | DEC R |
| #DB27 | 3F 3F 3F 30 | #CF | ??? 0 |
| #DB2B | 42 4E 45 48 | #D0 | BNE H |
| #DB2F | 43 4D 50 46 | #D1 | CMP F |
| #DB33 | 3F 3F 3F 30 | #D2 | ??? 0 |
| #DB37 | 3F 3F 3F 30 | #D3 | ??? 0 |
| #DB3B | 3F 3F 3F 30 | #D4 | ??? 0 |

| | | | |
|---|---|---|---|
| #DB3F | 43 4D 50 43 | #D5 | CMP C |
| #DB43 | 44 45 43 43 | #D6 | DEC C |
| #DB47 | 3F 3F 3F 30 | #D7 | ??? 0 |
| #DB4B | 43 4C 44 30 | #D8 | CLD 0 |
| #DB4F | 43 4D 50 54 | #D9 | CMP T |
| #DB53 | 3F 3F 3F 30 | #DA | ??? 0 |
| #DB57 | 3F 3F 3F 30 | #DB | ??? 0 |
| #DB5B | 3F 3F 3F 30 | #DC | ??? 0 |
| #DB5F | 43 4D 50 53 | #DD | CMP S |
| #DB63 | 44 45 43 53 | #DE | DEC S |
| #DB67 | 3F 3F 3F 30 | #DF | ??? 0 |
| #DB6B | 43 50 58 41 | #E0 | CPX A |
| #DB6F | 53 42 43 45 | #E1 | SBC E |
| #DB73 | 3F 3F 3F 30 | #E2 | ??? 0 |
| #DB77 | 3F 3F 3F 30 | #E3 | ??? 0 |
| #DB7B | 43 50 58 42 | #E4 | CPX B |
| #DB7F | 53 42 43 42 | #E5 | SBC B |
| #DB83 | 49 4E 43 42 | #E6 | INC B |
| #DB87 | 3F 3F 3F 30 | #E7 | ??? 0 |
| #DB8B | 49 4E 58 30 | #E8 | INX 0 |
| #DB8F | 53 42 43 41 | #E9 | SBC A |
| #DB93 | 4E 4F 50 30 | #EA | NOP 0 |
| #DB97 | 3F 3F 3F 30 | #EB | ??? 0 |
| #DB9B | 43 50 58 52 | #EC | CPX R |
| #DB9F | 53 42 43 52 | #ED | SBC R |
| #DBA3 | 49 4E 43 52 | #EE | INC R |
| #DBA7 | 3F 3F 3F 30 | #EF | ??? 0 |
| #DBAB | 42 45 51 48 | #F0 | BEQ H |
| #DBAF | 53 42 43 46 | #F1 | SBC F |
| #DBB3 | 3F 3F 3F 30 | #F2 | ??? 0 |
| #DBB7 | 3F 3F 3F 30 | #F3 | ??? 0 |
| #DBBB | 3F 3F 3F 30 | #F4 | ??? 0 |
| #DBBF | 53 42 43 43 | #F5 | SBC C |
| #DBC3 | 49 4E 43 43 | #F6 | INC C |
| #DBC7 | 3F 3F 3F 30 | #F7 | ??? 0 |
| #DBCB | 53 45 44 30 | #F8 | SED 0 |
| #DBCF | 53 42 43 54 | #F9 | SBC T |
| #DBD3 | 3F 3F 3F 30 | #FA | ??? 0 |
| #DBD7 | 3F 3F 3F 30 | #FB | ??? 0 |
| #DBDB | 3F 3F 3F 30 | #FC | ??? 0 |
| #DBDF | 53 42 43 53 | #FD | SBC S |
| #DBE3 | 49 4E 43 53 | #FE | INC S |
| #DBE7 | 3F 3F 3F 30 | #FF | ??? 0 |
| #DBEB | 42 52 4B 30 | #00 | BRK 0 |

**!LFIND**

| | | | |
|---|---|---|---|
| #DBEF | EA | NOP | |
| #DBF0 | 20 92 DC | JSR DC92 | Set output to printer |

**!FIND**

| | | | |
|---|---|---|---|
| #DBF3 | 38 | SEC | |
| #DBF4 | 6E F2 02 | ROR 02F2 | Set list return flag |
| #DBF7 | A9 9A | LDA #9A | Set marker for start of basic |
| #DBF9 | A0 00 | LDY #00 | |
| #DBFB | 85 00 | STA 00 | |
| #DBFD | 84 01 | STY 01 | |

```
#DBFF   84 07       STY 07      Clear flag for " in use
#DC01   20 E8 00    JSR 00E8    Re-fetch current character
#DC04   C9 22       CMP #22     Test for "
#DC06   D0 08       BNE DC10    Branch if not " (quotes)
#DC08   85 07       STA 07      Set flag for " in use
#DC0A   E6 E9       INC E9      Increment pointer to start of string
#DC0C   D0 02       BNE DC10
#DC0E   E6 EA       INC EA
#DC10   A5 E9       LDA E9      Copy text pointer as start of string
#DC12   A6 EA       LDX EA
#DC14   85 04       STA 04
#DC16   86 05       STX 05
#DC18   88          DEY         Count along the string until a null is reached
#DC19   C8          INY         This counts all characters,
#DC1A   B1 04       LDA (04),Y  including : and " etc
#DC1C   D0 FB       BNE DC19
#DC1E   98          TYA         Advance the text pointer to the null byte
#DC1F   18          CLC
#DC20   65 E9       ADC E9
#DC22   90 02       BCC DC26
#DC24   E6 EA       INC EA
#DC26   85 E9       STA E9
#DC28   A5 07       LDA 07
#DC2A   F0 08       BEQ DC34    If " are in use, decrement the length
#DC2C   88          DEY         of the string if the " have been closed.
#DC2D   B1 04       LDA (04),Y  But accept the line, even if
#DC2F   C9 22       CMP #22     the " were not closed
#DC31   F0 01       BEQ DC34
#DC33   C8          INY
#DC34   84 06       STY 06      Save the length of the string
#DC36   98          TYA
#DC37   F0 4B       BEQ DC84    Exit if the string is empty
#DC39   A0 01       LDY #01     Advance the marker at #00/1
#DC3B   B1 00       LDA (00),Y  to the start of the next
#DC3D   AA          TAX         line of basic
#DC3E   88          DEY
#DC3F   B1 00       LDA (00),Y
#DC41   85 00       STA 00
#DC43   86 01       STX 01
#DC45   18          CLC         Set another pointer at #02/3
#DC46   69 03       ADC #03     to point at the text in the current line
#DC48   90 01       BCC DC4B    and then used to move along it.
#DC4A   E8          INX
#DC4B   85 02       STA 02
#DC4D   86 03       STX 03
#DC4F   C8          INY
#DC50   B1 00       LDA (00),Y  Test for end of basic marker
#DC52   F0 30       BEQ DC84    Exit if end of basic
#DC54   E6 02       INC 02      Otherwise move along the basic line
#DC56   D0 02       BNE DC5A
#DC58   E6 03       INC 03
#DC5A   A0 00       LDY #00     Check the current line for the search string
#DC5C   B1 02       LDA (02),Y
#DC5E   F0 D9       BEQ DC39    New line if null reached
#DC60   D1 04       CMP (04),Y
#DC62   D0 F0       BNE DC54    Move along line if no match
#DC64   C8          INY
#DC65   C4 06       CPY 06      Test for length of required string
#DC67   D0 F3       BNE DC5C    Test next character if not at end of string
#DC69   A5 00       LDA 00      String found
```

```
#DC6B   A6 01       LDX 01      Copy the pointer for the start of this line
#DC6D   85 CE       STA CE      ready for the LIST / LLIST routine
#DC6F   86 CF       STX CF
#DC71   A0 02       LDY #02     Copy the line number for LIST / LLIST
#DC73   B1 00       LDA (00),Y  (i.e. single line LIST)
#DC75   85 33       STA 33
#DC77   C8          INY
#DC78   B1 00       LDA (00),Y
#DC7A   85 34       STA 34
#DC7C   20 A3 F7    JSR F7A3    Call routine in original ROM at
#DC7F   6C C7                   #C76C to LIST or LLIST the line
#DC81   18          CLC
#DC82   90 B5       BCC DC39    Go back to search for string again
#DC84   20 A3 F7    JSR F7A3    Call routine in original ROM at
#DC87   F0 CB                   #CBF0 to print a new line
#DC89   4E F2 02    LSR 02F2    Clear list return flag
#DC8C   20 A3 F7    JSR F7A3    Call routine in original ROM at
#DC8F   2F C8                   #C82F to reset output to screen
#DC91   60          RTS         and exit
#DC92   20 A3 F7    JSR F7A3    Call routine in original ROM at
#DC95   16 C8                   #C816 to set output to printer
#DC97   60          RTS         Return
#DC98   60          RTS
```

**!LCODE**

```
#DC99   20 A3 F7    JSR F7A3    Call routine in original ROM at
#DC9C   16 C8                   #C816 to set output to printer.
```

**!CODE**

```
#DC9E   20 98 F5    JSR F598    Print CR and LF for new line
#DCA1   A2 FF       LDX #FF     Re-set the counter for number of bytes to search for
#DCA3   86 06       STX 06      and set up the search 'string' of code
#DCA5   20 E8 00    JSR 00E8    Re-fetch current character
#DCA8   F0 79       BEQ DD23    Exit if nothing to find
#DCAA   E6 06       INC 06      Increment the byte counter
#DCAC   30 15       BMI DCC3    Exit if more than 128 bytes
#DCAE   20 A3 F7    JSR F7A3    Call routine in original ROM at
#DCB1   C8 D8                   #D8C8 to get a single byte expression into X
#DCB3   8A          TXA         Move it the A
#DCB4   A6 06       LDX 06
#DCB6   95 35       STA 35,X    Save it in the input buffer
#DCB8   20 E8 00    JSR 00E8    Re-fetch current character
#DCBB   F0 06       BEQ DCC3    Go search if end reached
#DCBD   20 B3 F5    JSR F5B3    Dispose of comma and fetch next character
#DCC0   4C AA DC    JMP DCAA    Loop back for next byte
                                Start the search
#DCC3   A2 FF       LDX #FF     Re-set the counter for matches
#DCC5   86 02       STX 02
#DCC7   86 03       STX 03
#DCC9   E8          INX         Re-set the memory pointer to #0000
#DCCA   86 00       STX 00
#DCCC   86 01       STX 01
#DCCE   A4 06       LDY 06      Fetch the number of bytes to search for
#DCD0   B1 00       LDA (00),Y  Test a byte
#DCD2   D9 35 00    CMP 0035,Y
#DCD5   D0 2D       BNE DD04    Branch if no match
```

```
#DCD7   88              DEY         otherwise carry on testing the bytes
#DCD8   10 F6           BPL DCD0    Loop until match found
#DCDA   E6 02           INC 02      Increment the match counter
#DCDC   D0 02           BNE DCE0
#DCDE   E6 03           INC 03
#DCE0   A6 02           LDX 02      Fetch the counter into A and X
#DCE2   A5 03           LDA 03
#DCE4   20 57 F7        JSR F757    Print out 2 byte number (counter) in decimal form
#DCE7   A9 20           LDA #20     Space character
#DCE9   A2 08           LDX #08
#DCEB   20 9F F5        JSR F59F    Print spaces to set a TAB
#DCEE   E4 30           CPX 30
#DCF0   90 F9           BCC DCEB
#DCF2   A9 23           LDA #23     # character
#DCF4   20 9F F5        JSR F59F    Print the # character
#DCF7   A5 01           LDA 01      Get the high byte of the address where the
#DCF9   20 8C F7        JSR F78C    match was found and print it out
#DCFC   A5 00           LDA 00      Same for low byte of the address
#DCFE   20 8C F7        JSR F78C
#DD01   20 98 F5        JSR F598    Print CR and LF for new line
                                    Move the pointers on
#DD04   AD DF 02        LDA 02DF    Test for key being pressed
#DD07   10 12           BPL DD1B    Branch if no key pressed
#DD09   29 7F           AND #7F     Test for CTRL-C
#DD0B   C9 03           CMP #03
#DD0D   F0 14           BEQ DD23    Exit if CTRL-C
#DD0F   C9 20           CMP #20     Test for space bar (pause)
#DD11   D0 08           BNE DD1B    Ignore any other keys
#DD13   4E DF 02        LSR 02DF    if space bar, wait for another key press
#DD16   AD DF 02        LDA 02DF    before moving on
#DD19   10 FB           BPL DD16
#DD1B   E6 00           INC 00      Increment the memory counter
#DD1D   D0 AF           BNE DCCE    Loop back to continue
#DD1F   E6 01           INC 01      the search, until
#DD21   D0 AB           BNE DCCE    the end of memory is reached
#DD23   20 98 F5        JSR F598    Print CR and LF for new line
#DD26   20 A3 F7        JSR F7A3    Call routine in original ROM at
#DD29   2F C8                       #C82F to re-set output to the screen
#DD2B   60              RTS         Exit

#DD2E   00 FF 00 FF 00  DTA         Free space
#DDFB   FF 00 FF 00 FF  DTA         Alternate bytes of 00 FF
```

# Memory usage by the extended DOS commands

**!CODE / !LCODE**

| Memory location | Use |
|---|---|
| 00/1 | Memory address being tested for the code match. |
| 02/3 | Counter for the number of matches found. |
| 06 | Counter for the length of the code string being searched for. |
| 35 > | Used to set up the code bytes in the input buffer. |
| 02DF | Used to test if a key is being pressed. |

**!CONV**

| Memory location | Use |
|---|---|
| 00/1 | Start address of code to convert to data lines |
| 02/3 | End address of code for conversion (+1) |
| 04/5 | First line number to use for DATA lines. Updated in use, to give current line number in use. |
| 06/7 | Checksum for current line |
| 0A | STEP value. Default is 10 |
| 70/1 | Vector to current basic line |

**!DIS**

| Memory location | Use | Other uses |
|---|---|---|
| 00 | Key pressed (temporary) | |
| 01 | Key press 1 | Start address for disassembly, low byte |
| 02 | Key press 2 | Start address for disassembly, high byte |
| 03 | Key press 3 | Temporary for calculating relative addresses |
| 04 | Key press 4 | Temporary for calculating relative addresses |
| 05 | Key press 5 '-' flag | #FF for off #2D for on |
| 06 | Key press 6 | End address for disassembly, low byte |
| 07 | Key press 7 | End address for disassembly, high byte |
| 08 | Key press 8 | |
| 09 | Key press 9 | |
| 0A | Data / code flag. #FF for Code. #7F for data | |
| 0B | CONTinuous mode flag. #FF for on, #7F for off | |
| 70 | Screen address for current line, low byte | |
| 71 | Screen address for current line, high byte | |

| Memory location | Use | Other uses |
|---|---|---|
| 72 | Temporary store for X register | |
| 73 | Temporary store for Y register | |
| 74 | 5 bytes of code or data copied down from memory | OPCODE for code disassembly |
| 75 | 5 bytes of code or data copied down from memory | |
| 76 | 5 bytes of code or data copied down from memory | |
| 77 | 5 bytes of code or data copied down from memory | |
| 78 | 5 bytes of code or data copied down from memory | |
| 79 | Address for OPCODE in lookup table (low byte) | |
| 7A | Address for OPCODE in lookup table (high byte) | |
| 7B | 5 bytes copied down from lookup table | First character of mnemonic |
| 7C | 5 bytes copied down from lookup table | Second character of mnemonic |
| 7D | 5 bytes copied down from lookup table | Third character of mnemonic |
| 7E | 5 bytes copied down from lookup table | Number of bytes used by this OPCODE |
| 7F | 5 bytes copied down from lookup table | Formatting code |
| 80 | Temporary copy of #7E for counting | |
| 02F0 | Printer flag for #02F1. #FF for LPRINT ON | |

**!DISK / !DISC**

| Memory location | Use |
|---|---|
| 02B2 | Number of files to copy to disk |
| 02B3 | Displacement, if the normal loading address is belo #0500 |

**!EDIT**

| Memory location | Use |
|---|---|
| 00/1 | Destination pointer for memory move |
| 02/3 | Source pointer for memory move |

**!FIND / !LFIND**

| Memory location | Use |
|---|---|
| 00/1 | Pointer – start address of current basic line |
| 02/3 | Pointer to text in the current line (moves along the line) |
| 04/5 | Pointer to start of the string being searched for |
| 06 | Length of string being sought. 0 = empty string. |
| 07 | Flag for whether quotes (“) are in use. Default is 0 (not in use) |
| 33/4 | Last line number, for the end of LIST |

| Memory location | Use |
|---|---|
| CE/F | Address pointer for the start of LIST |
| 02F1 | Printer flag |
| 02F2 | List return flag |

**!LIST / !LLIST**

| Memory location | Use |
|---|---|
| 00 | To save #CE whilst evaluating an expression |
| 01 | To save #CF whilst evaluating an expression |
| 33/34 | Last line number for LIST |
| CE/F | Pointer to start for LIST |
| 02F1 | Printer on flag |
| 02F2 | List return flag |