# Oak Tree Systems - Display Enhancement Package for TI99/4a requires 32k and Extended Basic or Mini Memory or Ed/Assembler

INSTRUCTIONS for use of the Disk:

## DISPLAY ENHANCEMENT PACKAGE

This 40 track disk is FULL. It contains the subprograms you need to run DEP with either Extended Basic or Editor Assembler (or Mini Mem).

Also on the disk can be found demonstration programs for both modules and documentation for display on screen or on a printer with 40 or more columns.

### USING EDITOR/ASSEMBLER MODULE:

Advantages: You can recover from inadvertant program exits
The sub-programs will load quickly
Disadvantage: You can only use what is left of the VDP memory- about 14k, if you use CALL FILES(1).

For DEMONSTRATION program or documentation:

Insert module,select basic
OLD DSK1.DEMO
RUN

To use the subprogram:

Insert module,select basic
input: CALL LOAD("DSK1.BSCSUP")
& CALL LOAD("DSK1.DISPEN")

For MINI MEMORY MODULE as Ed/As **BUT**:

delete from DEMO program the line CALL LOAD("DSK1.BSCSUP") before you run the program.

Do not LOAD "BSCSUP"

### EXTENDED BASIC:

Advantage: You have about 20k to put your program in
You have almost full use of ExBas extra commands
Disadvantages: The sub program takes a while to load
It is not usually possible to recover from an accidental program exit.

Insert Module,select Extended Basic
OLD DSK1.DEMOX
RUN

Insert Module,select EXTENDED BASIC
input CALL LOAD("DSK1.DISPENX")

(The MiniMemory adds the advantage of using POKEV to amend the program in VDPRAM or to use limited sprites in MODE28.)

Please note & respect the copyright notice on p.12 of the documentation.

NB: The 32k RAM MUST be connected whatever module you choose.

DOCUMENTATION: **DISPLAY ENHANCEMENT DISK** Oak Tree Systems.

A full 90k disk containing a suite of programs for enhanced screen handling. In Assembler language.

REQUIRES TI99/4A plus 32k Expansion memory
plus Disk system
plus EITHER Extended Basic Module
OR Editor/Assembler Module.
OR Mini Memory Module

With EDITOR/ASSEMBLER you retain the full 15k for your TI BASIC program.
With EXTENDED BASIC you lose 4k of memory and have only 20k for your program.

---

## INTRODUCTION

The DISPLAY ENHANCEMENT PACKAGE (DEP) is an Assembler Language program which utilises the 32k memory expansion unit.
The package provides a 40x24 display format and other enhanced data entry and screen control functions.

Functions in the DEP are accessed from Basic or Extended Basic via CALL LINK. Depending on the function being used,parameters may be required to pass data to,or retrieve results from, LINKed subprograms.

## MEMORY USAGE

The memory expansion unit is used to store the DEP and the information to be displayed on the screen.
The 32k bytes of the unit are divided into an 8k segment and a 24k segment. Usage of the two segments depends on whether the Extended Basic version of the DEP or the console Basic/Editor/Assembler version is in use:

TI BASIC/EdAss: A minimum of 10000 bytes of the 24k segment are used for the program and data storage. This area would not normally be used by TI BASIC.

EXTENDED BASIC: Most of the 8k segment is used to store the DEP program. The first 3842 bytes of the 24k segment is used as a data storage area. Your Extended Basic program is also stored in this area- thus the available program size is reduced 3842 bytes.
When an Extended Basic program is RUNNING, the SIZE command must always show at least 3842 bytes of PROGRAM SPACE remaining.

## DATA STORAGE AREA

The data storage area normally consists of 96 lines of 40 characters each- a total of 3840 bytes. How these 96 lines are used depends on the MODE the DEP is operating under. There are three distinct operating modes:

A. Scroll Mode
B. Scroll Mode with locked area
C. Page Mode.

## A. SCROLL MODE

In Scroll mode,INPUT or PRINT functions cause the next available line to be used as an INPUT or PRINT area.
Additional lines will be used depending on the size of the input or printed data. A pointer is maintained by the DEP to the next available line to be used. This pointer is initially set to LINE 1 - and is also set to line one when the DATA area is CLEARed.
As each line is used,the pointer increments to the next line of the data storage area. When line 25 is reached, the WINDOW (the part of the data storage area displayed on the screen) is moved down one line,so that lines 2-25 are now displayed.
Each line added (via INPUT or PRINT) causes movement of the window downward. This appears on the screen as the data scrolling upward.

(scroll mode,continued...)

When the last line of the data storage area is occupied,additional INPUT or PRINT functions will result in removal of the data in line one from the area, and moving each line up.
This also appears on the screen as a scroll up,but takes slightly longer to perform.

B.

A LOCKed area of 1 to 18 lines can be established by a call to the LOCK function. In this mode, scrolling takes place as described above, except the locked lines do not take part in the scrolling and are always displayed at the top of the screen.
Note that the lines locked are always the first lines in the data storage area REGARDLESS OF WHAT WAS DISPLAYED ON THE SCREEN AT THE TIME THE LOCK CALL TOOK PLACE. Note also,the scrolled area and the locked area can each be CLEARed independently.

C.

In PAGE MODE the data area is divided into 4 segments (pages) of 24 lines each. Each page can be displayed, cleared, and modified independently.
Changes can be made to any page without affecting the other pages.
NO scrolling can take place in PAGE mode.

ADDITIONAL CALLS are provided to write data to any page and read data from any page, regardless of whether that page is currently displayed.

It is possible to expand the number of pages,assuming sufficient memory is available.

RESTRICTIONS DURING USE

DEP sets the Video Display Processor (the chip which generates the screen display: VDP) into 'text' mode, which displays 24 lines of 40 characters.
While in this mode, the following BASIC statements should NOT be used:

INPUT, PRINT, ACCEPT, DISPLAY, ACCEPT AT, DISPLAY AT, CALL COLOR,

CALL SCREEN, CALL CLEAR, CALL HCHAR, CALL VCHAR, CALL GCHAR, CALL CHAR,

ON ERROR - and all Extended Basic statements pertaining to SPRITES.

The DEP provides equivalent functions for most of the above,except Sprites. This is unavoidable in 'Text' mode.

Use of any of these statements will cause unusual or erroneous results, and may result in failure of the DEP, requiring the system to be turned off and on, to resume use.

In addition,there are restrictions that affect the use of breakpoints and the CLEAR key. These are explained under 'Interrupting a Running Program'

Use of TRACE is difficult because most calls to the DEP will write over the trace data on the screen.

## LOADING THE DISPLAY ENHANCEMENT PACKAGE

Prior to use, the DEP must be loaded into the memory expansion unit. It will remain loaded and ready to use as long as you do not return to the master title screen.(ie you remain within BASIC or EXTENDED BASIC).

The loading procedure depends on whether Console Basic/Ed/As or Ext Basic is being used. Both procedures assume the DEP is on the disk in Drive 1.

BASIC/EDITOR/ASSEMBLER:
The following statements should be the first executed in the Basic program:

```
100 CALL INIT
110 CALL LOAD ("DSK1.BSCSUP")
120 CALL LOAD ("DSK1.DISPEN")
```

EXTENDED BASIC:
The following statements should be the first executed in the Extended Basic Program:
(nb: The Extended Basic Call Load is quite slow-this will take about 100 seconds):-

```
100 CALL INIT
110 CALL LOAD("DSK1.DISPENX")
```

<u>NOTE</u>: Once the load has been done, subsequent BASIC programs may be loaded and run without repeating the CALL LOADs.
DEP will maintain its state,including the contents of the data storage area, even though a new BASIC program is loaded.

MINI MEMORY:
As Editor Assembler but omit loading BSCSUP as this is already resident.
Use CALL INIT before LOADing DISPEN !!!

continued......

## DESCRIPTION OF FUNCTIONS

In the descriptions below, variable names are only given as examples : you may use ANY variable name you choose.

Except where noted,you can use a constant in place of a variable.

You MUST adhere to data type- ie string or numeric.

### SET 40 COLUMN MODE

Accessed via CALL LINK ("MODE40")

The first time this entry is called,DEP clears the data storage area, displays a copyright notice, sets the VDP into text mode, and sets the display colour to white on blue.

No other calls to the DEP can be made unless the VDP has been set in text mode via this call.

Subsequent calls to this entry set text mode, restore the previously set colours, and re-display the last screen, but do not clear the data.

### SET 28 COLUMN MODE

Accessed via CALL LINK ("MODE28")

This entry returns the display to the 28 column (graphics) mode used by the BASIC interpreters. It also sets the colour for all character sets to black on green.

This entry MUST be called prior to opening or closing cassette files,
prior to using the BASIC CHAR function,&
prior to terminating the program IN ANY WAY.

### CLEAR DATA STORAGE AREA

Accessed via CALL LINK ("CLEAR")

This entry clears the entire data storage area, and displays a blank screen. Subsequent data entries in scroll mode start at the first available non-locked line of the data storage area.

### CLEAR LOCKED PORTION OF DISPLAY

Accessed via CALL LINK ("CLEARL")

This entry clears the locked section of display,while leaving the scrolled section intact. If NO lines are locked,the call is ignored.

### CLEAR SCROLLED PORTION OF DISPLAY

Accessed via CALL LINK ("CLEARS")

This entry clears the scrolled section of the display,while leaving the locked section intact. If no lines are locked, all 96 lines of the scroll area are cleared.

### CLEAR PAGE

Accessed via CALL LINK("CLEARP")

This entry clears the currently displayed page of the display. This would normally be used in PAGE mode.
In SCROLL mode, the first 24 lines of the data storage area are cleared - not the data currently displayed.

### RETRIEVE A STRING FROM THE DATA AREA

Accessed via CALL LINK ("READ",Y,X,P,N,V$)

This call allows retrieval of a string of data from the data storage area. It allows the area to be used for temporary storage or as an area to pass data from program to program.
Parameters for this call are as follows:

Y,X = same as HCHAR (page 5)
P = An integer constant or variable with a value from 0 to the number of pages available (see NPAGES).
Indicates the page on which the desired data is located,which need not be the presently displayed page.
Page 0 and 1 are equivalent and refer to the first 24 lines of the data storage area.
N = Number of characters to be read,from 1 to 255. Must be an integer. (constant or variable).
V$ = A STRING VARIABLE ONLY which will contain the data retrieved.

### MOVE A STRING TO THE DATA STORAGE AREA

Accessed via CALL LINK ("WRITE",Y,X,P,N,V$)

This call allows a string of data to be moved to the data storage area.
Allows the area to be used for temporary storage,or as an area to pass data from program to program.
It also allows building a display page without having that display show on the screen during the building process.
Parameters are as follows:
Y,X,P = Same as READ above.
N = Number of characters to be written.
If V$ contains less than N characters,blanks will be added at the end. If V$ contains more than N characters,it will be truncated to fit. N must be an integer constant or variable.
If the data would extend off the page,it is truncated.
V$ = A string constant or variable containing the data to be written.

### SET PAGE MODE/SCROLL MODE/DISPLAY PAGE

Accessed via CALL LINK ("DIPAGE",N)

For N greater than 0, sets the DEP in page mode and displays Page N.
Each page is a block of 24 lines of the data storage area.
NB: In page mode, INPUT,PRINT,SCRUP,SCRDN are NOT valid. All other calls are allowed.
Switching pages does not destroy the contents of any page.
If N = 0 (zero), the display is set in SCROLL mode. All calls are valid in scroll mode. The DEP is in scroll mode when initially loaded.
Switching from PAGE MODE to SCROLL MODE results in an unpredictable display and should generally be followed by a CLEAR.

## LOCK DISPLAY

Accessed via CALL LINK ("LOCK",N)

This entry locks the first N lines of the data storage area and displays them at the top of the screen. This allows the upper portion of the screen to be fixed in place while the bottom portion is scrolled.

The lines locked are the first N lines of the data storage area,regardless of what lines were currently displayed on the screen.
N may be zero - in which case none of the display is locked and the entire screen can scroll.
In order to allow room for entries in the scrolled area,no more than 18 of the 24 lines may be locked.

## SCROLL SCREEN UP

Accessed via CALL LINK("SCRLUP")

This call scrolls the screen up one line, or equivalently moves the window down one line.
If the window is at the end of the data storage area,no movement takes place,and the call is ignored.

## SCROLL SCREEN DOWN

Accessed via CALL LINK ("SCRLDN")

This scrolls the screen down one line / moves the window up one line.

If the window is at the beginning of the (non-locked) data storage area, no movement takes place and the call is ignored.

## INPUT FROM THE KEYBOARD

Accessed via CALL LINK ("INPUT","PROMPT",CTL$,V$,STAT)

This call provides the function of the INPUT statement in BASIC,except that only one data item may be entered,and the data entered is always placed in a STRING variable. Four parameters are required for this call and are defined as follows:

"PROMPT" - May be a string constant or variable. This is displayed prior to entry of the INPUT. The parameter IS REQUIRED but may be a null string (i.e. "").

CTL$ - A string constant or variable. In this string each character has a meaning as follows:

- a A = ANY CHARACTER MAY BE ENTERED
- B = BEEP prior to INPUT
- C = CHECK that input can be evaluated with VAL function
- D = Permits only digits 0 to 9 to be entered.
- N = Numeric- permits 0 to 9 AND E,+,-
- U = Permits Uppercase Alphabetic characters only
- ' = (single quote- key FCTN O)= Any of the characters which follow the ' may be entered.

V$ = MUST BE A STRING VARIABLE. If numeric input is desired,the VAL function can be used to obtain it. If a "C" is included in the control string,the value input will always be convertable to a numeric via VAL.

STAT- MUST be a numeric variable only. STAT is set to the key code of the key pressed to terminate the input. This can be:
ENTER, BACK, REDO, BEGIN, PROCEED, or AID.
Except from ENTER, these keys are ONLY enabled when CTLON has been called.(See CTLON- next page).

IMPORTANT NOTE: If any key except ENTER is pressed, numeric validity checking ('C' option) is NOT performed. The program has to check STATUS before it can attempt to convert an input value to a numeric.

## ENABLE CONTROL KEYS

Accessed via  CALL LINK ("CTLON")

This call allows use of the BEGIN,PROCEED,AID,REDO and BACK keys during INPUT or ACCEPT.
The codes for these keys (14,12,1,6 and 15 respectively) are passed to the BASIC program in the STAT variable. If CTLON has been called the BASIC program should check STAT after each ACCEPT or INPUT to determine the required action.

## DISABLE CONTROL KEYS

Accessed via  CALL LINK ("CTLOFF")

This call disables the 5 keys which are enabled by CTLON.
When loaded, the DEP has these keys disabled.
Note that ENTER  CLEAR  and  QUIT are  ALWAYS enabled,and perform their usual functions.

## PRINT TO SCREEN

Accessed via  CALL LINK ("PRINT",V1$,CTL1,V2$,CTL2.......)

This call performs the function of the PRINT statement in BASIC.
A total of 16 parameters may be specified.
The parameters consist of alternating string variables/constants and integer numeric variables/constants.
To print NUMERIC items,convert them to strings using the STR$ function prior to this call.
The parameters are:
V1$,V2$ etc - contain the character string to be printed,up to 255 characters. MAXIMUM of 8 values.
CTL1,CTL2.etc- Specifies action to be taken between each string- the equivalent to the BASIC print separators, as follows:

| VALUE: | EQUIVALENT: | ACTION: |
|---|---|---|
| 1 | ; | Next string follows immediately after previous string |
| 2 | , | Next string appears in next print zone- there are 3 zones of 13 characters. |
| 3 | : | Next string starts on next line. |

You may perform PENDING print operations by ending the parameter list with a CTL parameter.
If the parameter list ends with a data string,the next call to PRINT will always start a new line.
Intervening calls to INPUT will also cause the next PRINT to start on a new line.

## ALLOCATE ADDITIONAL PAGES

Accessed via  CALL LINK ("NPAGES", N)

With some restrictions you can create more pages of data storage area.
N must be an integer from 4 to 16, equal to the total pages to be allocated

These pages can be accessed in PAGE mode only,or via WRITE or READ.

The following restrictions apply:
EXTENDED BASIC : Each page beyond 4 uses an additional 960 bytes of Program Area.

Basic/Editor/Assembler: Additional pages occupy memory immediately following the DEP in the 24k memory segment. This does not present a problem unless other assembler lanuage programs are used with the DEP.
In that case,loading the other programs BEFORE loading DEP will avoid any conflict. You must make certain that sufficient memory is available in the 24k segment.

## DISPLAY AT SCREEN LOCATION

Accessed via     CALL LINK ("DISP",Y,X,L,CTL$,V$)

This call allows displaying data at a specified screen location and provides certain formatting capabilities.

DISP should be done in PAGE mode or
                    in a LOCKed area ONLY to avoid positioning problems.
Parameters are as follows:

Y,X - identical to HCHAR
  L - the length of the field - from 1 to 255
CTL$- A STRING constant or variable,in which each character performs certain functions as defined below:

- B - Beep when displaying
- E - Erase the field prior to displaying the string
- L - Left justify in field (default is set to this)
- R - Right justify in specified field
- P - Decimal aligned - immediately after the P must be a number 1-9 signifying the number of places to the right of the decimal to be displayed. Unused positions are filled with zeros.

V$  - A string constant or variable,containing the data  to be displayed. To display numeric data,use STR$ to convert from numeric to string prior to this call.

EXAMPLE:   CALL LINK ("DISP",10,20,12,"BEP4",NUM$) -
this call displays the variable NUM$ at row 10, column 20, in a field 12 characters long, sounds a beep, erases the field and positions the data such that there are 4 positions to the right of the decimal point.

### Exception Conditions :-

If the data string is longer than the field specified- it will be truncated on the right - regardless of justification.

If the field would extend beyond the edge of the page, it will be truncated.

If the number of decimals specified is greater than the field length minus one, decimal alignment is ignored.

continued....

## ACCEPT AT SCREEN LOCATION

Accessed via    CALL LINK ("ACCEPT",Y,X,L,CTL$,V$,STAT)

This call allows entry of data at any location on the screen, using PAGE mode or a LOCKed area.
Several options are available when using ACCEPT. Parameters are as follows:

Y,X,L - same as DISP (page 9).

CTL$ - A string constant or variable in which each character performs certain control functions as described below:

A - Allows any character to be entered (All)

B - Sounds a Beep prior to entry

C - Checks entry to ensure VAL will work.

D - Allows only digits to be entered (0 to 9)

E - Erases the field prior to entry

L - Left justification- data displayed as entered.(Default value).

N - Permits digits 0 to 9 and +-. & E to be entered.

P - Decimal alignment - a number 1 to 9 must follow immediately after the P, signifying the number of places after the decimal point.    'E' MUST be used.
During entry, numbers are placed before the decimal point until the period (.) is entered - the cursor then moves to the right of the decimal point.
The decimal point will always be in the same location no matter how many digits are entered on either side of it.

R - Data is right justified during entry. Each character entered pushes the previous characters to the left.Use with 'E'.

T - Trigger entry on a filled field.
When T is specified, entry of a character in the last position of a field causes the data to be entered without the ENTER key.
NB: Cannot be used with 'R'.    'E' MUST be used.

U - Permits Uppercase alphabetical characters only to be entered.

NOTE: When using P R or T, the screen must be blank, or the field cleared in some other way - eg by using the control E - which must occur **before** P R or T in the control string.

V$ - A string VARIABLE only - this variable will contain the entered data. If numeric input is required,the VAL function can be used to convert the string to a numeric variable.

STAT - A numeric VARIABLE - STAT is set to the code of the key which terminated the input.
If option T terminates the entry, a value of 13 is placed in STAT- the same as for ENTER. See CTLON on page 8.

continued...

## KEY FUNCTIONS DURING DATA ENTRY

While data is being entered, several function keys can be used to assist in editing the data being entered.

Each of these keys is listed below,along with the function it provides.

**INSERT** - After pressing INSERT,all subsequent characters entered are entered into the location of the cursor,forcing any following characters to the right.
Reset by using the right or left arrow keys or by using a termination key (eg ENTER), or by using ERASE or DELETE.

**DELETE** - Deletes the character at the cursor position and moves the following characters to the left.

**CLEAR** - Interrupts the program and switches to 28 column mode.
CLEAR should ONLY be used: During Input or Accept.

See 'Interrupting a Running Program' -following.

**UP & DOWN ARROWS** - Scroll the screen up and down when in SCROLL mode (with or without locked area).
Have no effect in page mode.
During INPUT: The screen can be scrolled while entry is taking place,but pressing any key other than the UP or DOWN arrows returns the screen to the original position so that entry can continue.

**LEFT & RIGHT ARROWS** - Move the cursor left or right in the input area. Also cancels Insert mode, and Decimal Alignment mode in ACCEPT.
Using JUSTIFY,right justify is cancelled by using the LEFT arrow key.

**ERASE** - Blanks the input field and positions the cursor in the left most position. Resets Decimal,Right Justify & Insert.

**QUIT** - Returns to the Master Title Screen : All data in memory is lost.

## ERROR MESSAGES

Certain errors in parameters passed to the DEP will cause an error message to be displayed at the bottom centre of the screen. The message will remain on the screen about ten seconds and then disappear. The call that caused the error will be ignored and the program will continue. The meaning of these error codes is as follows:

| CODE: | MEANING: |
|---|---|
| A | Color- missing or invalid foreground parameter |
| B | Background Parameter Colour-Missing or invalid |
| C | Lock - Parameter missing, invalid, or greater than 18. |
| D | Y (row) Parameter missing or invalid (HCHAR,VCHAR,GCHAR,READ,WRITE, |
| E | Y (row) Parameter greater than 24. ( DISP, ACCEPT |
| F | X (col) Parameter missing or invalid. |
| G | X (col) parameter greater than 40. |
| H | Character code parameter missing or invalid (HCHAR,VCHAR) |
| I | Charactar code too large. |
| J | Page number missing,invalid,or exceeds number available. |
| K | Read-Error in Page Number or Length Parameter |
| L | Input-Called when NOT in scroll mode or parameters missing or invalid. |
| N | Print-Called when not in scroll mode. |
| O | DISP- missing or invalid parameter |
| P | Accept- Missing or invalid parameter |
| Q | NPAGES- Parameter missing or invalid |
| R | Write- Parameter missing or invalid or page number not available. |
| S | GCHAR- Parameter missing or invalid |
| T | Input or Accept- V$(RESULT) parameter missing. |

## INTERRUPTING A RUNNING PROGRAM

Because of its larger screen size, the DEP uses areas of the VDP memory which are normally reserved for the BASIC interpreters.
For most BASIC statements these are unused and thus present no problem.

However there are several functions which access these areas and therefore must be avoided when MODE40 is in effect.

Furthermore making the transition from RUN mode to COMMAND mode (as via BREKK, CLEAR or STOP ) will cause problems when in MODE40.

The following restrictions must be adhered to-

CONSOLE BASIC/EDITOR/ASSEMBER :-
(OR MINIM)

1. Always call MODE28 prior to a STOP or BREAK
2. You may use the CLEAR key during a DEP ACCEPT or INPUT and the normal environment for the Basic interpreter will be restored.
   Using CLEAR at any other point will NOT restore the interpreters environment.
3. In the event of the program breaking or stopping when in MODE40 the result will be a blank or oddly coloured screen. Type in CALL LINK("MODE28") and press enter.
   Note you will not see what you are typing,so use care!

EXTENDED BASIC :-

1. You must call MODE28 prior to STOP or BREAK. Otherwise a strange screen display and/or erroneous error messages will be generated.
2. You MAY use ON BREAK NEXT.
   You MAY NOT use ON ERROR...
3. You MAY use the clear key during a DEP INPUT or ACCEPT.
4. You MAY insert breakpoints as follows:

```
                        CALL LINK("MODE28")     )
                        BREAK                   )    =used as lines in
                        CALL LINK("MODE40")     )     your program
```

5. You cannot recover from an erroneous halt - always save your program before testing !!!!!!!
6. During DEP INPUT or ACCEPT a value of 2 is passed to STAT by the CLEAR key.

## COPYING THE DEP

This package is copyrighted by OAK TREE SYSTEMS and may not be reproduced for the purpose of conveying a copy to any person other than the original purchaser.
You may make copies for your own use - both as backup,and to place copies on other disks to be co-resident with programs that you write.

If you sell programs that you write, which use the DEP, you MUST obtain a sub-licence agreement if you wish to sell the DEP as part of your program. A copy of this agreement is available on request from:

The Distributor :-

This program has been imported from the USA by
Stephen Shaw,                    STOCKPORT, Cheshire,