

**SPIRIT-30 EDSP**

**USER MANUAL**

---

**Sonitech International Inc**  
14 Mica Lane, Suite 208  
Wellesley, MA 02181, USA

---

Copyright (C) 1989 Sonitech International Inc.

This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose, without written permission of Sonitech International Inc.

# TABLE OF CONTENTS

## CHAPTER 1: STARTING UP

1.1 Hardware and Software Requirements .....	1-1
1.2 EDSP Workstation Package .....	1-1
1.3 Installation .....	1-2
1.4 Memory Description and Addressing Conventions .....	1-3

## CHAPTER 2: BASICS

2.1 EDSP Program Execution .....	2-1
2.2 Menu / Window Strategy .....	2-1
2.5 EDSP parameters.....	2-4
2.3 Window Example .....	2-5
2.4 A Quick Run.....	2-7

## CHAPTER 3: DESCRIPTION

### Global Functions:

{Display} .....	3-2
{Reset} .....	3-3
{Read write append} .....	3-4

### [Edsp] Module

{Edsp software config} .....	3-5
{Save environment} .....	3-7
{Restore environment}.....	3-8
{Dos exit}.....	3-9

### [Xecute] Module

{Program load}.....	3-10
{Program execute}.....	3-11

### [Memory] Module

{Block fill} .....	3-12
{Single step fill} .....	3-14

### [Debug] Module

{Single step/Breakpoint/Rev Assembly}.....	3-15
--	------

### [Signal] Module

{Generate} .....	3-16
------------------	------

[Plot] Module

{Overview}.....	3-26
{Quick 2D}.....	3-27
{Configure 2D}.....	3-28
{Multiple plots}.....	3-31

[Utility] Module

{Edit}.....	3-32
{View file}.....	3-33
{View directory}.....	3-34
{Print file}.....	3-35
{NuMber conversion}.....	3-36
{Modify file header}.....	3-37
{Modify file sequence}.....	3-38

[Application] Module

{User Application}.....	3-39
-------------------------	------

**APPENDIX**

A-1 Data file header information.

## CHAPTER 1

### STARTING UP

---

#### 1.1 HARDWARE AND SOFTWARE REQUIREMENTS

The following hardware and software environment is necessary for EDSP:

- IBM PC, XT, AT and compatibles
- DOS 3.0 or later
- Hard disk
- 640K memory
- Hercules graphics card or IBM compatible EGA card

The following systems are optional depending on the applications:

- Microsoft compatible compiler (preferably 'C').

A recommended system for applications development and signal processing is the following:

- IBM AT or compatible
- 40 MEG Hard drive
- 640K memory
- Microsoft 'C' compiler

---

#### 1.2 EDSP WORKSTATION PACKAGE

SPIRIT-30 EDSP package consists of six 5.25 inch, 360K diskettes.

---

## 1.3 INSTALLATION

### STEP 1 : *Backup your Diskettes*

- Make a backup copy of all the diskettes in this package and store the original disks in a safe place. Disk copy can be done using the DOS command DISKCOPY (refer to your DOS manual for more information on this command).

### STEP 2 : *Board Installation*

- Please follow the instructions in the SPIRIT-30 SYSTEM *Technical Reference Manual* included with the SPIRIT-30 board.

### STEP 3: *EDSP Software Installation*

#### \* **Setting up the Environment**

In order to use the software, certain commands must be placed in the AUTOEXEC.BAT file in the root directory on the hard disk. To do this:

1. Use a text editor to edit the file AUTOEXEC.BAT in the root directory on the hard disk.
2. Within the file, there should already be a line of the form:

```
PATH=XXXXXX
```

If this line is present, then add the string ;C:\SPIRIT30\EDSP to the end of this line, so that it appears:

```
PATH=XXXXXX;C:\SPIRIT30\EDSP
```

If there was no PATH command already in the file, then add the line:

```
PATH=C:\SPIRIT30\EDSP
```

3. Add the following line to your AUTOEXEC.BAT file:

```
SET SPIRIT_DIR=C:\SPIRIT30
```

**Note:** If you choose not to use the base directory shown above, i.e., C:\SPIRIT30, then use the name of your SPIRIT-30 base directory.

**\* Installation**

To install the software on your hard drive, follow these steps:

1. Insert the disk labeled **SPIRIT-30 : EDSP and Debugger** diskette 1 in drive A and enter the command:

A:INSTALL

2. Now, follow the installation program prompts.

**1.4 MEMORY DESCRIPTION AND ADDRESSING CONVENTIONS**

The TMS320C30 on SPIRIT-30 board provides on-chip memory that includes two blocks of 1Kx32-bit RAM, which is referred to as 'internal RAM'. Apart from this, SPIRIT-30 board has 32K x 32-bit words of dual access on-board memory which is referred to as 'external memory'.

Following is the convention used by EDSP software to address SPIRIT-30 board memories:

NAME	DESCRIPTION	ADDRESS RANGE
EX	SPIRIT-30 board external memory	0x000000 to 0x007fff
M0	SPIRIT-30 board internal RAM bank-0	0x809800 to 0x809bff
M1	SPIRIT-30 board internal RAM bank-1	0x809c00 to 0x809fff



## CHAPTER 2

### BASICS

Refer to section 2.1 in SPIRIT-30 Debugger User Manual before proceeding any further.

---

#### 2.1 EDSP PROGRAM EXECUTION

Begin EDSP session by typing "EDSP" .

C:\> EDSP (Make sure that the environment variable SPIRIT\_DIR is set to SPIRIT-30 base directory and path is properly set. For more information refer to chapter 1.3)

This will set the EDSP window environment and start the EDSP session.

Hit 'ENTER' in the opening window and you will see the pull down menu options module. If you want to exit the EDSP session type <ESC> or select QUIT option in EDSP module.

Before we proceed further, let's take a minute to understand the Menu / Window selection strategy.

---

#### 2.2 MENU / WINDOW STRATEGY

Typical EDSP screen appears as shown below:

EDSP XECUTE MEMORY DEBUG...	
EDSP pull down menu	

The pull down menus (ex: EDSP, XECUTE, etc.) are referred to as [Modules] and can be easily identified in this manual by square brackets.

The selections in the pull down menu are referred to as functions (ex: {Program Load}, {Program execute}... are functions in [XECUTE] module). Functions can be easily identified in this manual by curly brackets such as {function}.

A typical function selection will open a window with several options and parameters. This is explained in more detail in the section WINDOW EXAMPLE.

## NOTATIONS:

The following notations are used throughout the documentation:

[module]

- [...] is module name.

{function}

- {...} is a function name.

<Parameters>

- <...> is a window parameter name.

Example:

<1 Source :> (window parameters in DISPLAY window)

<2 Address :>

'input'

- '...' is user input.

NOTE: all user input must follow 'ENTER' key.

## KEYBOARD CONTROL:

The following are the key controls for the screen:

<ESC>

- Will exit from the present screen menu to the previous state.

<ARROW KEYS>

- Menu level : (Left & Right arrow keys) will select the pull down menus. (Up & Down arrow keys) will select the function within the menu.

- Function level: Will move the current selection item to the next item.

<ENTER>

- Once the selection is highlighted, this key will invoke (confirm that selection) the item.

- All input must be followed by this key.

<BACK SPACE>

- To delete the inputs

<ALT high\_lighted character>

At menu selection level, hitting ALT and high\_lighted character will open selected pull down window. This is an alternate to using left and right arrow keys.

Example:

In {XECUTE} menu, <ALT p> will open {PLOT} menu window.

## Function keys

F3,F4,F6 keys are active at all the menu and option levels and will select DISPLAY, RESET and READ WRITE APPEND (RWA) function options respectively. These functions are referred to as global functions.

**Highlighted Character**

The selection is invoked (this is equivalent to moving using the Arrow Keys to the item and hitting 'ENTER').

**Example 1:**

[XECUTE] {program Load} - this option can be selected by typing 'L' or 'I' in XECUTE menu.

**WINDOW ERROR MESSAGE:**

Error messages are displayed on the bottom of the screen. The following are two commonly observed error messages in EDSP.

***"cannot open 'window name'"***

If you have a resident program, you may need to uninstall it for the EDSP window environment to function properly. To correct the problem, exit EDSP, uninstall resident program and enter EDSP again.

***"Memory allocation error" or "Not enough memory to allocate".***

You may see this message if there is a resident program. To correct the problem, exit EDSP, uninstall resident program and reenter EDSP.

## 2.3 EDSP PARAMETERS

Following are the frequently used parameters in EDSP:

### <Source:>

This signifies the origin of the data in the memory. The valid options for <source> are:

(see MEMORY ADDRESSING for following on-chip memories and their respective addresses)

'EX' - refers to the SPIRIT-30 external memory.

'M0' - refers to the TMS320C30 internal RAM bank-0.

'M1' - refers to the TMS320C30 internal RAM bank-1.

'FI' - File, refers to IBMPC hard disk or floppy files.

'fi test.dat' - refers to source as file and filename as test.dat.

### <Destination:>

This signifies the target memory or file for writing data. The valid options for the <destination> are same as options for <source>. (see options for <Source> above).

### <Address :>

Address of the memory location. In case the <Source> is 'FI'(file), <Address> corresponds to offset of the file. (see section on MEMORY DESCRIPTION AND ADDRESSING CONVENTIONS for valid addresses)

### <Datatype:>

These are valid inputs for data type :

'H'	- Hex	80000000
'F'	- IEEE float	0.0000000E+000
'I'	- Integer	-2147483648
'U'	- Unsigned Integer	3458718999
'T'	- TI float	1.0000000E+000

### <Filename:>

Files used in EDSP are program files and data files. In {Program Load} option, the valid filename must be DSP executable 'COFF' file prepared by TMS320C30 linker. The default extension for program files is '.out' and for data files is '.dat'. The data file should have the header or else the error message 'invalid file header info' will appear. See Appendix A-1 for the format of data file header.

## 2.4 WINDOW EXAMPLE

STEP 1: Start the EDSP session by typing "EDSP" and press 'ENTER'

STEP 2: Hit 'ENTER' at first introduction window.

STEP 3: Select the [Memory] module as follows:

The desired item [module] can be selected in two ways:

- By use of <ARROW KEYS> (Left and Right) keys; Select [Memory] menu by left or right arrow keys or
- By typing the <ALT high\_lighted menu character>; [Memory] can be selected by typing <ALT 'm'>.

STEP 4: Select {Reset} option to clear SPIRIT-30 board memory.

(NOTE: this option is accessible in all modules.)

This option can be selected in three ways:

1. Using Up or Down arrow keys to high\_light the selection;
2. Typing function key <F4>
3. Type the high\_lighted selection character. Here 'R' is the selection character for {Reset} option.

After selecting reset option, you will see the {Reset} window. (See section RESET under chapter 3 DESCRIPTION for detail explanation of the options).

- Type '5'(or use <ARROW KEY> to get to parameter < 5. ....> and hit 'ENTER' ) to initialize external memory. Try other options as well.
- <ESC> will exit this window and bring you back to the {Memory} module menu.

STEP 5: Select {Block Fill} option.

- Use the same procedure as described above. Here 'K' is the key selection character. You will now see the Block Fill window.

STEP 6: Fill 100 DSP words in External memory with a value of -1.23e15 (IEEE float) starting at address 1200H.

- Select item 1 <source> by typing '1' 'ENTER'.
- Valid options for source will be displayed. Type 'EX' 'ENTER' to select Bank0 memory as source.
- Select item 2 <Start Address> by typing '2' 'ENTER'.
- Type '1200' and hit 'ENTER' and the default value should get updated to 1200.
- Keep the default value for item 3 <Address Increment>.

- EDSP Workstation - Basics 2-6

- Select item 4 <Count> by typing '4' 'ENTER'.
- Type '100' for the <Count> value hit 'ENTER'. The <End Address> should get updated to '1263'.
- Select item 6 <Data Type> by typing '6' 'ENTER'.
- The <Data Type> options are displayed. Type 'f' and hit 'ENTER' for float type.
- Select item 7 <Data Value> by typing '7' 'ENTER'.
- Type '-1.23e15' for <Data Value> and hit 'ENTER'.
- keep the default values for all the other options.
- Select item 0 <Fill> by typing '0' 'ENTER'. This will fill the given value in the external board memory starting at address 1200H.
- Hit <ESC> to exit from this window.

STEP 7: Check the memory contents to be sure the above Block Fill operation was successful.

- Select {Display} option by typing 'D'(or 'd') or hitting function key 'F3'
- Enter <Source> as 'EX' (be sure to hit 'ENTER')
- Give <Start Address> as '11F0'
- Give <Data Type> as 'f'
- On the screen you should see the External memory values in decimal. Hit <ALT> H to change the format to Hex; Hit F9 to see other data display options.
- Hit <Up or Down arrow or Page Up or Page Down> to scroll the screen. Hit <ALT> F to display in decimal. Compare the displayed data with the data you loaded in previous STEPs.
- <ESC> will return to {XECUTE} menu module.

## 2.5 A QUICK RUN

The objective of the following tutorial is to familiarize you with functions of several key modules. For details on the functions, refer to chapter 3 DESCRIPTION in this manual.

### TUTORIAL

In this tutorial, we will

- Load data file (sine wave with hamming window applied) to external memory
- Display the content of external memory.
- Plot data from external memory and files.

**STEP 1:** *Initialize DSP external memory and reset the DSP.*

- Type "EDSP" from DOS and press 'ENTER'
- Select {Reset} function by pressing 'R' or function key F4.
- Choose option 6 from reset menu (This will reset DSP and initialize all DSP internal & external memory locations to zero.).
- Quit from reset menu by pressing <Esc>.

**STEP 2:** *Read a data file and write to DSP memory.*

- Choose option {read Write append} by pressing 'W' or function key F6.
- Enter the parameters as shown below. (To select <Source>, move the cursor to <Source> and hit 'Enter'. Type 'F1' at the blinking cursor and hit 'Enter' again. Similarly, move cursor to other options and enter the values as shown below)

Source	:'F1'	/* Source is a file */
Address	:'0'	/* Offset from 1st data point in file */
Count	:'256'	/* No. of data points to read */
Filename	:'SINHAMM.DAT'	/* Name of the input file */
Destination	:'EX'	/* External memory locations */
Address	:'1000'	/* Address of destination in hex */

Note: File 'SINHAMM.DAT' is provided with EDSP and loaded in the same directory during installation.

- Choose option 9 <RWA operation>  
(This will read 256 points from file SINHAMM.DAT and write to DSP external memory locations starting from 0x1000. File SINHAMM.DAT has 256 points and <data type> of the file is IEEE floating point)
- Quit from {read write append} menu by pressing <Esc>.

STEP 3: *Display contents of memory locations filled in step 2.*

- Choose {Display} option by pressing 'D' or function key F3.
- Enter the parameters as follows.

Source	: 'EX'
Address	: '1000'
Datatype	: 'F'

Memory locations starting from 1000 will be displayed in IEEE floating point data type.

- Observe that the memory contents of location 1000 are same as file SINHAMM.DAT.
- To display contents of file SINHAMM.DAT press <Esc> key, this will bring you back to Display window, and enter the parameters as follows

Source	: 'F'
Address	: '0'
Datatype	: 'F' /*IEEE float */
Filename	: 'SINHAMM.DAT'

- Quit display function by pressing <Esc>.

STEP 4: *Plot data from DSP memory*

- Press <ALT> P to go to [PLOT] module.
- Choose {Configur 2D} by pressing 'C'.
- You will see default plot parameters for plot number 0.
- Enter parameters as follows by selecting the parameter select character.

Source	: 'EX'
Address	: '1000'
Datatype	: 'F'
Count	: '256'

- Choose title, xlabel, ylabel and plot color as you desire.
- Press 'P' to plot.
- You will see a plot of a sine wave with hamming window applied to it.
- Press 'Esc' to quit the sine wave display and get back to parameter window.
- Quit from configure plot window by pressing 'Esc'

## CHAPTER 3

### DESCRIPTION

#### 3.0 DESCRIPTION

EDSP has 8 major modules with each module having several functions and one or more global functions which are often needed within a module.

#### {Global Functions}

- {Display}
- {Reset}
- {Read Write Append}

#### [EDSP] Module Functions

- {Edsp Software Config}
- {Save Environment}
- {Restore Environment}
- {DOS Exit}

#### [Xecute] Module Functions

- {Program Load}
- {Program Execute}

#### [Memory] Module Functions

- {Block Fill}
- {Single Step Fill}

#### [Debug] Module Functions

- {Single step/Breakpoint}

#### [Signal] Module Functions

- {Generate/Modify/Analyze}

#### [Plot] Module Functions

- {Overview}
- {Quick 2D}
- {Multiple Plots}
- {Configure 2D}

#### [Utility] Module Functions

- {Edit}
- {View File}
- {View Directory}
- {Print File}
- {Number Conversion}
- {Modify file header}
- {Modify file sequence}

#### [Application] Module Functions

- {User Application}

**Global Function: Display**

---

**NAME: {Display}**

**OPERATION:**

Show contents of SPIRIT-30 internal & external memory or file in various data types. This option can be activated in all pull down windows by either hitting 'd' or function key F3.

**PARAMETERS:**

- <Source> Refer to section on EDSP PARAMETERS.
- <Address> Refer to section on EDSP PARAMETERS.
- <Data Type> Refer to section on EDSP PARAMETERS.  
If the source is a file then this input is ignored and file is displayed in the data type of the file (as given in file header)
- <Filename> Refer to section on EDSP PARAMETERS.  
This input is meaningful if source is of type file.

**MESSAGES:**

- F9 - see available data types.  
Shows how to change the current display data type. Opens a window having the following information.

Press To change display data type to

- ALT-F IEEE floating point
- ALT-H Hex
- ALT-I Integer(32 bit long)
- ALT-U Unsigned Integer(32 bit long)
- ALT-T TI floating point

See also: EDSP PARAMETERS.

**Global Function: Reset**

---

**NAME: {Reset}**

**OPERATION:**

Reset/Clear DSP, DSP registers and DSP memories.

This option can be activated in all pull down windows by either hitting 'r' or function key F4.

**PARAMETERS:**

- <1. Reset DSP> Clears extended precision & auxiliary registers, clears interrupts & repeat registers. Configures peripherals and sets up stack pointers.
- <2. Clear DSP registers> DSP registers R0-R7, AR0-AR7, IR0-IR1 and ST are initialized to zero.
- <3. Clear DSP RAMs> Initialize DSP RAMs (RAM0 and RAM1) to 'memory init' value specified in EDSP SOFTWARE CONFIGURATION. Default value for initialization is zero.
- <4. All of the above(1,2,3)>  
Above operations are done in sequential order of 1,2 and 3.
- <5. Clear board memory> Initialize SPIRIT-30 external memory locations to 'memory init' value specified in EDSP SOFTWARE CONFIGURATION.  
Default value for initialization is zero.
- <6. All of the above(1,2,3,5)>  
All of the above operations are done in sequential order of 1,2,3 and 5.

**Global Function: Read Write Append**

---

**NAME:** {read Write append}

**OPERATION:**

Read data from DSP memory / file and write to DSP memory or write/append to a file.

This option can be activated in all pull down windows either by hitting 'w' or function key F6.

**PARAMETERS:**

<Source>	Refer to section on EDSP PARAMETERS.
<Address>	Refer to section on EDSP PARAMETERS. Address for memory or offset for file.
<Count>	Enter number of DSP words(32 bit) to be read.
<Filename>	Refer to section on EDSP PARAMETERS. This input is meaningful if source is of type file[FI].
<Destination>	Refer to section on EDSP PARAMETERS.
<Address>	Refer to section on EDSP PARAMETERS. This input is meaningful only if destination is DSP memory.
<Data Type>	Refer to section on EDSP PARAMETERS. This input is meaningful only if destination is file.
<Filename>	Refer to section on EDSP PARAMETERS. This input is meaningful if destination is of type file[FI].
<rWa operation>	Invoke read/Write/append operation after source and destination PARAMETERS have been specified.

If <source> and <destination> are both DSP memory then data is copied from source locations to destination locations.

If <source> is DSP memory and <destination> is a file then data is interpreted by the datatype specified here and output is written to the file.

If <source> is file and <destination> is DSP memory then data is copied from file to memory locations according to datatype of the file (as given in the file header).

**MESSAGES:**

File already exists, append, overwrite, quit(A/O/Q):

If destination is a file and file already exists then this message will appear.

Option 'O' (Overwrite) will create a new file.

Option 'A' (Append) will append data to existing file.

(Make sure that data type of old file is same as data type specified here)

Option 'Q' (Quit) will quit the operation

**[EDSP] module:**

---

**NAME: {Edsp Software Config} (software configuration)**

**OPERATION:**

This option allows you to configure some of the EDSP parameters to suit your environment. These parameters can be saved and restored back using {save environment} and {restore environment} functions respectively. Choosing this option will open SOFTWARE CONFIGURATION window displaying the parameters with default / restored values. Parameters can be modified by selecting the parameter high lighted number at <option #:> prompt.

Valid inputs for each parameter are displayed in the option window when selected.

**PARAMETERS:**

'1' <1 Program Directory>

Choosing this option will prompt for <path name> for program directory. Program directory is where user program files are located and will be searched for loading in {load program} function.

Eg: path name: '\EDSP\PROGRAM'

With above configuration, EDSP will load all program files from "\EDSP\PROGRAM" directory. You can give any path name, however it must begin with "\". Default program directory is the current working directory (".").

'2' <2 Data Directory>

(CURRENTLY NOT SUPPORTED)

Choosing this option will prompt for <path name> for data directory. Data directory is where user data files are located and will be accessed from.

Eg: path name: 'EDSP\DATA'

With above configuration, EDSP will access all data files from "\EDSP\DATA" directory. You can give any path name, however it must begin with "\". Default data directory is the current working directory (".").

'3' <3 Editor>

Choosing this option will prompt for <editor> name that invokes the editor on your system. Choosing the {EDIT} function in [Utility] module will invoke this editor. (see section on {EDIT} under UTILITY)

'4' <4 Assembler>

(NOT SUPPORTED FOR THIS BOARD)

Choosing this option will prompt for <assembler> name that invokes the assembler on your system. Choosing the {ASSEMBLE} function in [Utility] module will invoke this assembler. (see section on {ASSEMBLE} under UTILITY)

'5' <5 Printer>

Choosing this option will prompt for <printer> number identifying the printer you want to use on your system. Valid options for printer number are displayed in the option window and on the message line. Following are the valid printer numbers:

'0' for EPSON FX, JX-80, LQ1500

'1' for IBM Graphics, ProPrinter, EPSON LX & MX

- EDSP Workstation - Description 3-6

- '2' for Okidata dot matrix
- '3' for LaserJet, LaserJet+, Series II
- '4' for Texas Instruments.

'6' <6 Memory init value>

Choosing this option will prompt for memory initialization value. This value is used for DSP reset option in {reset} function to clear DSP memories. By default, this value is 0x0L. The valid input for <memory init value> is long hex number such as: 00000000 or 00000001 or FFFFFFFF, etc.

(see section on RESET)

'7' <7 Axis color> (color of the plot axis)

'8' <8 Plot color> (color of the plot graph)

'9' <9 Title color> (color of the title on the plot)

'10' <10 Label color> (colors of X & Y labels on the plot)

(Above parameters are used by EDSP plot module for its plot display)

Choosing any of these options will display the color block string with arrow pointing to current color. Select the color by moving the arrow left or right and pointing the arrow to the color block of choice. 'Enter' key will confirm the selection. However, these options are not valid if you do not have a color video adapter.

**[EDSP] module:**

---

**NAME: {saVe environment} (save the current EDSP environment)**

**OPERATION:**

To save following EDSP environments: "Software Configuration " "Plot Configuration "

Choosing any of the options will display the default filename and prompt for new file name to save the environment. All the files are saved in binary format. If the file name already exists then it will be overwritten. These files can be used to restore corresponding environment using {resTore environment} function.

**PARAMETERS:**

- '1' <1 Save Data Block Definitions >  
(Option not available)
- '2' <2 Save Software Configuration >  
This option will save the software configuration. All the parameters defined in {EDSP software config} function will be saved. Default file name will be displayed and new file name can be entered.
- '3' <3 Save Plot Configuration >  
This option will save the plot configuration( 0 - 15) information. All the parameters defined in (configure 2D} function will be saved. Default file name will be displayed and new file name can be entered
- '4' <4 All of the above(2,3) >  
This option will perform the options 2 and 3 in sequence.

**[EDSP] module:**

---

**NAME: {Restore Environment} (restore the EDSP environment)**

**OPERATION:**

Restore following EDSP environments. "Software Configuration " "Plot Configuration "
--

Choosing any of the options will display the default filename and prompt for new file name to restore the environment. The files used in {save environment} are used for restoring environment.

**PARAMETERS:**

- <1 Restore Data Block Definitions>  
(Option not available)
- <2 Restore Software Configuration>  
This option will restore the software configuration. Default file name will be displayed and new file name can be entered.
- <3 Restore Plot Configuration>  
This option will restore the plot configuration( 0 - 15) information. Default file name will be displayed and new file name can be entered
- <4 All of the above(2,3) >  
This option will perform the options 2 and 3 in sequence.

*- EDSP Workstation - Description 3-9*

**[EDSP] Module:**

---

**NAME:** {DOS Exit} (Temporarily exit to Dos)

**OPERATION:**

This option allows temporary exit to dos from EDSP. To return to EDSP from dos, type 'exit' at dos prompt.

**[Xecute] Module:**

---

**NAME:** {Program Load} (Load program)

**OPERATION:**

This option allows loading of DSP executable 'COFF' file into SPIRIT-30 memory. Upper half of the LOAD window displays the currently loaded program. Load and Execute options appear on the lower half of the Load window.

**Load option:**

Choosing this option displays the current program directory path and prompts for file name to be loaded. Current program directory is where user program files are located and this path can be configured through EDSP SOFTWARE CONFIG option in EDSP module. By default, this path is set to current working directory.

Files are always loaded from current program directory unless full path name is entered. Hitting 'F2' key will display and load from program directory.

**PARAMETERS:**

<File Name>: [path]filename or

<File Name>: 'F2' (this will display program files in program directory and file can be selected by highlighting the file and hitting <ENTER>)

EDSP expects 'COFF' files that are output of TI's TMS320C30 Assembler/Linker. Files are loaded in memory at address specified by the 'COFF' program file.

If the file specified above already exists and is displayed, it will be reloaded. EDSP only remembers the last file loaded.

**Execute option:**

Choosing this option will execute the loaded program. (see next section Program Execute for more details)

**[Xecute] Module:**

---

**NAME:** {Program Execute} (Execute Program)

**OPERATION:**

This option allows execution of program file at the specified address. Upper half of the EXECUTION window displays the map of the currently loaded program file. Execute parameters appear on the lower half of the EXECUTION window.

**PARAMETERS**

- <1> Execute > Choosing this option will start execution of the file from Start Address. EDSP detects the end of the program execution and halts the DSP when it reads the value 0x15 at memory location 0x300. If the program does not end execution or is in loop, <ESC> will Halt Execution after reconfirmation.
- <2> Start Address[ ]>: Start address of the file is displayed and can be changed to start execution at new address. (see section on MEMORY ADDRESSING)

**[Memory] Module:**

---

**NAME: {block fill} (Fill block of memory)**

**OPERATION:**

Fills the selected memory locations with given value of given data type.

**PARAMETERS:**

Following are the valid input parameters at Option: prompt:

Valid options for each parameter are displayed in option window.

**<1 Source>**

Valid source inputs are 'EX','M0','M1'. (see section SOURCE under EDSP PARAMETERS)

**<2 Start address>**

Address in hex (see section MEMORY ADDRESSING for valid address range)

**<3 Address increment>**

Address increment in +/- integer is the number of words skipped before two consecutive words are filled. A value of '1' will cause a contiguous block to be filled with address increment and value of '-1' will cause a contiguous block to be filled with address decrement.

**<4 count>**

Count in decimal is the number of words to be filled. If you are not sure of the count but know the end address you want to reach, you can skip this option and go to the next option. However, if count is given, end address will get updated accordingly.

**<5 End address>**

End address in hex is the end of the memory block to be filled. Option 4 (count) updates the end address, however, if end address input is given then count is updated accordingly.

**<6 Data type>**

Data type format of the data value to be filled. (see section DATATYPE under EDSP PARAMETERS for valid data types)

**<7 Data value>**

Data value to be filled in above data type format.

**<8 Data increment>**

Data increment value in above data type format will increment data value at each address increment as follows:  $data = data + data\_increment;$

<9 Address fill width>

Address fill width in decimal is the number of memory words to be filled at every address increment. Width of '2' will fill the data at current address and next address with same data. Address fill width must not be greater than address increment.

<0 Fill>

Choose this to perform block fill.

Example:

1 source	: 'EX'
2 start address	: '1000'
3 address inc	: '3'
4 count	: '5'
5 end address	: '1038'
6 data type	: 'f'
7 data value	: '125.0'
8 data increment	: '10.0'
9 addr fill width	: '2'

Memory Map

```
000fff -> xxxxxxxxxxxxxxxx
001000 -> 125
001004 -> 125
001008 -> xxxxxxxxxxxxxxxx
00100C -> 135
001010 -> 135
001014 -> xxxxxxxxxxxxxxxx
00102C -> 165
```

**[Memory] Module**

---

**NAME:** {single sTep fill}

**OPERATION:**

Fills a selected DSP memory location in specified data type. The fill address is highlighted and location is filled with the value entered.

**PARAMETERS:**

<Source> Refer to section on EDSP PARAMETERS.  
Error message is displayed if source is file.

<Address> Refer to section on EDSP PARAMETERS.

<Data Type> Refer to section on EDSP PARAMETERS

Memory locations are displayed in the specified data type.

**MESSAGES:**

F9 - see available data types. Tells how to change the input and display data type by opening a window having following information..

Press To change data type to

ALT-F IEEE floating point

ALT-H Hex

ALT-I Integer(32 bit long)

ALT-U Unsigned Integer(32 bit long)

ALT-T TI floating point

See also: EDSP PARAMETERS.

**[Debug] Module:**

---

**NAME: {Single step/Breakpoint}**

**OPERATION:**

Choosing this option will invoke SPIRIT-30 Debugger. Please refer to SPIRIT-30 Debugger User Manual.

**[signal] Module:**

**NAME: {Generate}**

**OPERATION:**

```
Generate <1 Signal>
Generate <2 Noise>
Generate <3 Window>
Generate <4 Signal + Noise>
Generate <5 Signal * Window>
Generate <6 Noise * Window>
Generate <7 (Signal + Noise) * Window>
Perform <8 Add-Multiply Signals>
```

Choosing {Generate} option in [signal] module will open 'GENERATE' window displaying all of the above options. All the options can be selected either by scrolling with up & down arrow keys and 'Enter' or by typing the highlighted characters.

**PARAMETERS:**

**<1 Signal> :**

Choosing this option will open 'GENERATE SIGNAL' window, displaying following signal options and parameters with default values. All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

**<1 Overall length [ ] :>**

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.

```
<---0's--->Ê<--- signal values --->Ê<---0's--->
                Ê<-number of samples->Ê
<----- overall length ----->
                starting position
```

**<2 Generate to [ ] :>**

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in external memory of SPIRIT-30. 'FI file.dat' will create the file file.dat and store the result in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

**<3 Quantize (bits) [ ] :>**

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

**<Generate 'signal name' :>**

Generate the requested signal with given parameters and store in the destination specified. The signal values are stored in IEEE floating point (32 bit) data type.

- <Sine>
- <Cosine>
- <squarE>
- <triAngle>
- <saWtooth>

Choosing any of the above options will open 'SAMPLED SIGNAL: ' window, displaying the following parameters with default values. All the parameters can be modified by typing the highlighted characters or scrolling with arrow keys and 'Enter'.

- <1 Amplitude [ ]:>

Amplitude of the signal.

- <2 DC offset [ ]:>

This value will be added to the signal value.

- <3 Frequency (Khz) [ ]:>

Frequency of the signal in Khz. This value along with sampling rate determines the interval at which signal is sampled.

- <4 Shift (rad) [ ]:>

Shift from origin in radian.

- <5 Sample rate (Khz) [ ]:>

The rate at which signal is sampled. This value along with frequency of the signal determines the interval at which signal is sampled.

- <6 Number of samples [ ]:>

Integer value signifies the number of signal samples to be generated. This value should be typically less than overall length of signal.

- <7 Start position [ ]:>

Integer value signifies the starting position within the overall length to generate number of signal samples given in option <6> above.

NOTE To generate signal with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE > option.

- <Chirp>

Choosing this option will open 'SAMPLED SIGNAL: CHIRP' window, displaying the following parameters with default values. All the parameters can be modified by typing the highlighted characters or scrolling with arrow keys and 'Enter'.

- <1 Amplitude [ ]:>

Amplitude of the signal.

- <2 Start freq (Khz) [ ]:>

The starting frequency of the signal.

- <3 Stop freq (Khz) [ ]:>

The ending frequency of the signal.

<4 Sample rate (Khz) [ ]:>

The rate at which signal is sampled. This value along with frequency of the signal determines the interval at which signal is sampled.

<5 Number of samples [ ]:>

Integer value signifies the number of signal samples to be generated. However, if number of samples are greater than <Overall length>, only overall length of signals will be generated.

<6 Start position [ ]:>

Integer value signifies the starting position within the Overall length to generate number of signals given in option <5> above.

NOTE To generate signal with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE> option.

<Dual sine>

Choosing this options will open 'SAMPLED SIGNAL: DUAL SINE = SINE\_1 + SINE\_2' window. All the parameters can be modified by typing the highlighted characters or scrolling with arrow keys and 'Enter'.

<1 Amplitude\_1 [ ]:>

Amplitude of the signal\_1.

<2 Amplitude\_2 [ ]:>

Amplitude of the signal\_2.

<3 DC offset [ ]:>

This value will be added to the signal value.

<4 Frequency\_1 (Khz) [ ]:>

Frequency of the signal\_1 in Khz. This value along with sampling rate determines the interval at which signal is sampled.

<5 Frequency\_2 (Khz) [ ]:>

Frequency of the signal\_2 in Khz. This value along with sampling rate determines the interval at which signal is sampled.

<6 Shift\_1 (rad) [ ]:>

Shift from origin in radian for the signal\_1.

<7 Shift\_2 (rad) [ ]:>

Shift from origin in radian for the signal\_2.

<8 Sample rate (Khz) [ ]:>

The rate at which signal is sampled. This value along with frequency\_1 and frequency\_2 of the signal determines the interval at which signal is sampled.

<9 Number of samples [ ]:>

Integer value signifies the number of signal samples to be generated. This value should be typically less than overall length of signal.

<0 Start position [ ]:>

Integer value signifies the starting position within the overall length to generate number of signal samples given in option <9> above.

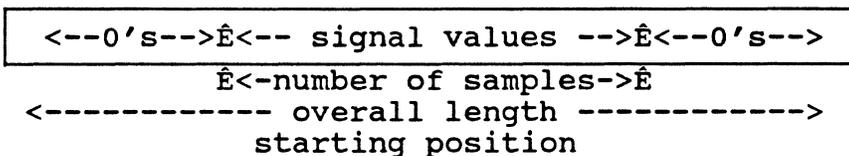
NOTE To generate signal with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE> option.

<2 Noise >

Choosing this option will open 'GENERATE NOISE ' window. All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

<1 Overall length [ ]:>

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.



<2 Generate to [ ]:>

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in external memory of SPIRIT-30. 'FI file.dat' will create the file file.dat and store the signal values in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

<3 Quantize (bits) [ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

<4 Add noise [ ]:>

Type 'Y' if noise is to be injected to an existing signal (generated earlier or acquired from somewhere). You will be asked for the source of the signal. It will be assumed that the source of the signal has data in IEEE floating point (32 bit) data type. Type 'N' to generate noise only.

<Generate noise :>

Generate the requested noise with given parameters and store in the destination specified. The data values are stored in IEEE floating point (32 bit) data type.

<uniForm >

<gaUssian >

<Poisson >

Choosing any of the above options will open 'NOISE: ' window. All the parameters can be modified by typing the highlighted characters or scrolling with arrow keys and 'Enter'.

<1 Standard deviation [ ]:>

Standard deviation of the noise.

<2 Noise length [ ]:>

Integer value signifies the number of noise samples to be generated. This value should be typically less than overall length of signal.

<3 Start position [ ]:>

Integer value signifies the starting position within the overall length to generate number of noise samples given in option <2> above.

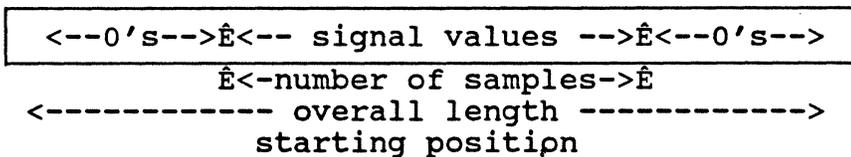
NOTE To generate noise with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE> option.

<3 Window >

Choosing this option will open 'GENERATE WINDOW ' window. All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

<1 Overall length [ ]:>

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.



<2 Generate to [ ]:>

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in SPIRIT-30 external memory. 'FI file.dat' will create the file file.dat and store the signal values in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

<3 Quantize (bits) [ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

<4 Source of signal [ ]:>

Specify the source of signal to be windowed. It will be assumed that the source of the signal has data in IEEE floating point (32 bit) data type. (refer section on SOURCE under EDSP PARAMETERS for more details)

<3 Quantize (bits) [ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

<Generate window:>

Generate the requested window with given parameters and store in the destination specified. The data values are stored in IEEE floating point (32 bit) data type.

- < Rectangular >
- < haMming >
- < haNning >
- < Blackman >

Choosing any of the above options will open 'WINDOW: ' All the parameters can be modified by typing the highlighted characters or scrolling with arrow keys and 'Enter'.

- <1 Window length = N [ ]:>

Integer value signifies the number of window samples to be generated. This value should be typically less than overall length of signal.

- <2 Start position [ ]:>

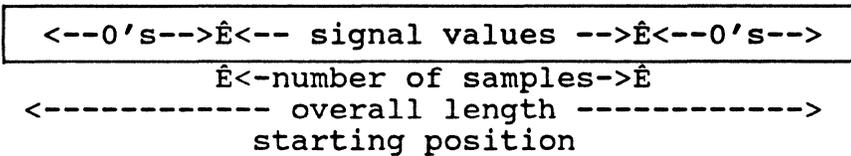
Integer value signifies the starting position within the overall length to generate number of window samples given in option <1> above. NOTE: to generate noise with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE> option.

#### < 4 Signal + Noise >

Choose this option to generate a specified signal and inject noise to it. Choosing this option will open 'GENERATE SIGNAL + NOISE' window. All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

- <1 Overall length [ ]:>

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.



- <2 Generate to [ ]:>

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in SPIRIT-30 external memory. 'FI file.dat' will create the file file.dat and store the signal values in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

- <3 Quantize (bits)[ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

- <4 Source of signal [ ]:>

Specify the source of signal to be windowed. It will be assumed that the source of the signal has data in IEEE floating point (32 bit) data type. (refer section on SOURCE under EDSP PARAMETERS for more details)

- <3 Quantize (bits)[ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

<Generate signal + noise:>

Generate the requested signal with given parameters and store in the destination specified. The data values are stored in IEEE floating point (32 bit) data type.

(refer to section on option <1 Signal> above for following signal options)

- <Sine>
- <Cosine>
- <squarE>
- <triAngle>
- <sawTooth>
- <Chirp>
- <Dual sine>

(refer to section on option <2 Noise> above for following noise options)

- <uniForm>
- <gaUssian>
- <Poisson>

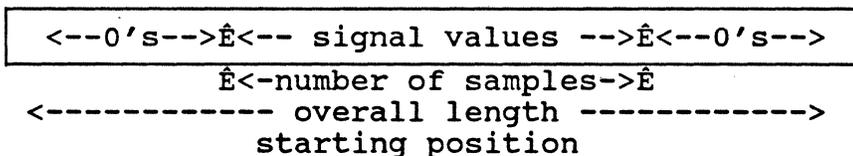
NOTE To generate signal with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE> option.

## <5 Signal x Window>

Choose this option to generate a specified signal and apply a specified window to it. Choosing this option will open 'GENERATE SIGNAL + WINDOW' window, All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

<1 Overall length [ ]:>

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.



<2 Generate to [ ]:>

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in SPIRIT-30 external memory. 'FI file.dat' will create the file file.dat and store the signal values in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

<3 Quantize (bits) [ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

<Generate signal x window:>

Generate the requested signal with given parameters and store in the destination specified. The data values are stored in IEEE floating point (32 bit) data type.

(refer to section on option <1 Signal> above for following signal options)

- <Sine>
- <Cosine>
- <squarE>
- <triAngle>
- <sawTooth>
- <Chirp>
- <Dual sine>

(refer to section on option <3 Window> above for following window options)

- <Rectangular>
- <haMming>
- <haNning>
- <Blackman>

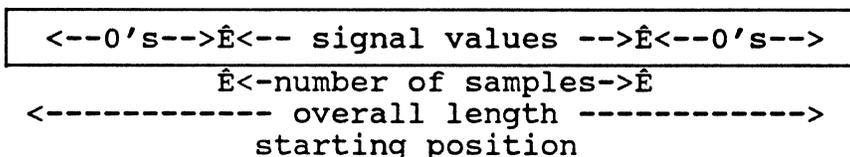
NOTE To generate signal with given parameters, hit escape and return to 'GENERATE SIGNAL' window and select <GENERATE> option.

### <6 Noise \* Window>

Choose this option to generate a specified noise and apply the specified window to it. Choosing this option will open 'GENERATE NOISE + WINDOW' window, All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

<1 Overall length [ ]:>

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.



<2 Generate to [ ]:>

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in SPIRIT-30 external memory. 'FI file.dat' will create the file file.dat and store the signal values in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

<3 Quantize (bits)[ ]:>

Simulates an A/D converter for quantization numbers between 1 and 23 (inclusive).

<Generate signal\*window:>

Generate the requested signal with given parameters and store in the destination specified. The data values are stored in IEEE floating point (32 bit) data type.

(refer to section on option <2 Noise> above for following noise options)

<uniForm>  
<gaUssian>  
<Poisson>

(refer to section on option <3 Window> above for following window options)

<Rectangular>  
<haMming>  
<haNning>  
<Blackman>

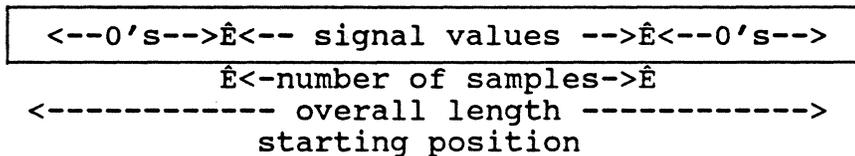
NOTE To generate signal with given parameters, escape back from above option to 'GENERATE NOISE + WINDOW' window and select option <GENERATE>.

### <7 (Signal + Noise) \* Window>

Choose this option to generate a signal , inject noise to the signal and apply window to it. Choosing this option will open 'GENERATE SIGNAL NOISE WINDOW' window. All the options can be selected by typing the highlighted characters or scrolling the options with arrow keys and 'Enter'.

<1 Overall length [ ] :>

Integer value signifies the total number of signal samples to be generated. This value is related to number of samples and starting position as follows.



<2 Generate to [ ] :>

Destination to store the generated signal. e.g. 'EX 1000' will store the signal starting from location 1000 in SPIRIT-30 memory. 'FI file.dat' will create the file file.dat and store the signal values in it. (refer section on DESTINATION under EDSP PARAMETERS for more details)

<Generate signal\*window:>

Generate the requested signal with given parameters and store in the destination specified. The data values are stored in IEEE floating point (32 bit) data type.

(refer to section on option <1 Signal> above for following signal options)

<Sine>  
<Cosine>  
<squarE>  
<triAngle>  
<sawTooth>  
<Chirp>  
<Dual sine>

(refer to section on option <2 Noise> above for following noise options)

<uniForm>  
<gaUssian>  
<Poisson>

(refer to section on option <3 Window> above for following window options)

<Rectangular>  
<haMming>  
<haNning>  
<Blackman>

### <8 Add-Multiply Signals>

Choose this option to add two signals or multiply two signals. The input signals will be assumed to be of IEEE floating point (32 bit) data type. Choosing this option will open 'SIGNAL ADD - MULTIPLY' All the parameters can be modified by typing the highlighted characters or scrolling the options with up-down arrow keys and 'Enter'.

<1 Signal\_1 [ ]:>  
Source of signal\_1.

<2 Signal\_2 [ ]:>  
Source of signal\_2.

(refer to section on SOURCE under EDSP PARAMETERS for more details)

<3 Signal\_3 [ ]:>  
Destination for add/multiply operation. Store the result destination specified here.  
(refer to section on DESTINATION under EDSP PARAMETERS for more details)

<4 Weight\_1 [ ]:>  
Scale of signal\_1.

<5 Weight\_2 [ ]:>  
Scale of signal\_2.

<6 Count [ ]:>  
Number of data points to add or multiply.

<7  $S_3 = (W_1 * S_1) + (W_2 * S_2)$ >  
Perform the operation of adding two signals and store the result in the destination specified by signal\_3.

<8  $S_3 = W_1 * S_1 * S_2$ >  
Perform the operation of multiplying two signals and store the result in destination specified by signal\_3.

---

**[Plot] Module:**

**OVERVIEW:** EDSP allows user the to define up to 16 plots numbered from 0 to 15. A plot configuration is collection of information which includes

- 1) The data source and its size,
- 2) The scaling information such as the maximum and the minimum values to be plotted,
- 3) The axes-labels, title of the plot and the color of the plot, and
- 4) The information about the preprocessing e.g. normalization or taking logarithms before plotting etc.

A plot is determined uniquely by a plot number between 0 and 15. These plots can be defined and modified in the 'Configure 2D' module.

**VIEWPORT:** A viewport is an area which is typically a subset of the entire screen. The screen can be divided in up to 4 viewports. Plots can be drawn in each viewport by overlapping them.

**MULTIPLE PLOTS** {Multiple Plots} function allows user to give inputs which decide the number of viewports to be drawn. This function also takes the inputs which decide which plot(s) should be drawn in which viewport.

**PLOT GRAPHIC SCREEN:** Plots are drawn in graphics mode. Screen switches to graphics mode when 'p' is pressed in either {Configure 2D} or {Multiple plots} window. Bottom line of the graphics screen shows the message 'p' to print, <Esc> to quit and the current date and time. Hitting 'p' will print graphics screen to the printer. If the printer is off-line a message 'check printer...' appears.

**[Plot] Module**

---

**NAME: {Quick 2D}**

**OPERATION:**

Draw a single plot with default parameters.

{Quick 2D} is activated by selecting 'Q' key in the [PLOT] module and it displays QUICK 2D window.

**PARAMETERS:**

<source :> following are the valid inputs for <source> prompt:

1. Integer value, between 0 - 15: an integer input between 0 - 15 refers to corresponding plot configured in the {CONFIGURE 2D} function. Eg: an input '9' will draw the plot defined by the plot number 9. Thus, choosing '9' in quick plot has the same effect as doing the following steps: i) choose {configure 2D} function. ii) choose plot number = 9. iii) press 'p'.
2. Function keys F2..F6: These keys will activate global functions {DISPLAY}, {RESET}, {DATA BLOCK CONFIG} and {READ WRITE APPEND} respectively.
3. 'Esc': hitting 'Esc' key will quit the {Quick 2D} function.
4. 'Enter': If 'Enter' key is hit then previously chosen input for the source is assumed and a graph is drawn.

This function thus can save key strokes for repetitive plotting of the same data.

**[Plot] Module**

---

**NAME: {Configure 2D}**

**OPERATION:**

Define new plot(s). Modify the parameters of existing plot(s). Draw a single plot with current parameters.

'Configure 2D' is activated by pressing the 'C' key in the 'PLOT' pull down menu. Window displays the current parameters for the displayed plot number. These parameters can be changed by selecting the appropriate option in the menu. An option may be selected by pressing the appropriate 'select key'. A 'select key' is the highlighted character in front of each option. E.g. Select key for count option is '5'.

Hit <Esc> to retain the previous parameter value. Pressing the <Enter>, the UP arrow, or the DOWN arrow keys will cause the parameter to assume the new typed value. If the value is not a valid value, an error message will appear.

**PARAMETERS:**

<[1] Plot Number:>

Valid inputs: 0 to 15

This parameter defines the current plot of interest. The defaults for the chosen plot are displayed.

<[2] Source:>

Valid inputs: See section on EDSP PARAMETERS.

<[3] Address:>

Valid inputs: See the section on EDSP PARAMETERS.

<[4] Filename:>

Valid inputs: Character strings for filename. A filename without extension is given the default '.dat' extension.

A warning message appears if the file does not exist, or is read protected.

<[5] Datatype:>

Valid inputs: See section DATATYPE under EDSP PARAMETERS.

< [6] Count: >

Valid inputs: Number of data points to be plotted (integer).

< [7] Autoscale: >

Valid inputs: 'Y', 'N' (character)

The plot is scaled according to the maximum and the minimum values in the data if the autoscale parameter is 'Y'. Else, the plot uses the maximum and the minimum values from the 'Max' and the 'Min' parameters.

< [8] Max: >

Valid inputs: Number greater than the 'Min' value.

The data points in the plot above this value are not drawn, if the Autoscale parameter is 'N'.

< [9] Min: >

Valid inputs: Floating point number smaller than the 'Max' value. The data points in the plot below this value are not drawn, if the Autoscale parameter is 'N'.

< [A] X - Label: >

Valid inputs: A printable ASCII string.

The label for the horizontal axis. This label is centered in the plot. If this label is too big, it may be truncated to fit the screen.

< [B] Y - Label: >

Valid inputs: A printable ASCII string. The label for the Vertical axis. This label is centered in the plot. If this label is too big, it may be truncated to fit the screen.

< [C] Title: >

Valid inputs: A printable ASCII string. The title for the plot. This title is centered in the plot. If the title is too big, it may be truncated to fit the screen.

< [D] Color: >

A color palette appears on the screen with an arrow pointing to the current plot color. Select the new plot color and hit <ENTER>. Some colors may not appear on monochrome displays and printers. Suggested color in this case is 'white'.

< [E] Process: >

Valid inputs: 'none', 'db10', 'db20', 'bnorm', 'unorm', 'normdb10', 'normdb20'. This parameter specifies the processing to be performed on the data before plotting. The meaning of various processing are as below.

'none': No processing

'db10': Plot  $10 \cdot \log()$  of the data

'db20': Plot  $20 \cdot \log()$  of the data

'unorm': (unipolar) Normalize the data between 0.0, and +1.0

'bnorm': (bipolar) Normalize the data between -1.0, and +1.0

'normdb10': First (unipolar) normalize, and then take  $10 \cdot \log()$

'normdb20': First (unipolar) normalize, and then take  $20 \cdot \log()$

Before the  $\log()$  operation is performed on the data, value of the smallest positive data point is assigned to all the non-positive data.

<[F] Config plot:>

Valid inputs: 0 to 15

This is not a parameter of a plot. All the parameters of the current plot are assigned to the plot corresponding to the number selected in the 'config plot' option.

e.g. Let the number displayed in the '[1] Plot number' option be 0. If we enter '3' at the 'Config plot' option, the plot number 3 will have the same parameters as the plot number 0.

When 'p' is pressed, the plot with current (displayed) parameters is drawn.

Pressing <Esc> will return the user to the 'PLOT' pull down menu.

**[Plot] module**

---

**NAME: {Multiple plots}**

**OPERATION:**

This module allows drawing of multiple plots in all of the 4 viewports.

Plots can be overlapped in any/all of the viewports. To get into the 'Multiple plots' module, press 'M' key in the 'PLOT' pull down menu. This will put the 'MULTIPLE PLOTS' window on the screen. This window accepts data which allows user to put different plots in different windows in any order. An option may be selected in either of the following ways.

- 1) Use the UP/DOWN arrow keys to highlight the required option. Hit <Enter> to change the parameter.
- 2) Hit the key for the highlighted character to change the parameter of that option. If you change your mind during typing, hit <Esc> to retain the previous parameter value. Pressing <Enter>, UP arrow, or DOWN arrow keys will cause the parameter to assume the new typed value. If value is not a valid value, an error message will appear.

**PARAMETERS:**

<Viewport Placement:>

Valid inputs: A,B,C or 4

Four types of viewport placements are possible.

'A': One viewport, full screen.

'B': Two viewports. Viewport 1 in upper half and viewport 2 in the lower half of the screen.

'C': Three viewports. Viewport 1 in the upper half, viewport 2 in the lower left quarter and viewport 3 in the lower right quarter of the screen.

'D': Four viewports. Viewport 1 in the upper left, viewport 2 in the lower left, viewport 3 in the upper right and viewport 4 in the lower right quarter of the screen.

To select a viewport placement, enter A,B,C or D or use the cursor keys to highlight A,B,C or D and hit <ENTER>.

<Plots in viewport .....[1]>

Select character: '1'

Valid inputs: A list of integers between 0 and 15, separated by spaces.

The list of numbers decides which plots should be plotted in an overlapped fashion in the viewport '1'. The scaling of the first plot in the list is used for all the consecutive plots. The plots to be plotted in viewports 2, 3 and 4 can also be selected in the same fashion.

When 'p' is pressed, the plots are drawn.

Pressing <Esc> will return the user to the 'PLOT' pull down menu.

**[Utility] Module:**

---

**NAME:** {Edit}

**OPERATION:**

Choosing this option will prompt for filename and will display the file. With file input, it will invoke the editor that has been configured in {EDSP SOFTWARE CONFIG} function in [EDSP]

**PARAMETERS:**

<Filename:> filename

see {EDSP SOFTWARE CONFIG}

**[Utility] Module:**

---

**NAME:** {View File}

**OPERATION:**

Choosing this option will prompt for filename and will display the file. (dos operation "TYPE  
<filename> | more")

**PARAMETERS:**

<Filename:> filename

[Utility] module

---

**NAME:** {View Directory}

**OPERATION:**

Choosing this option will prompt for mask and will display the directory. (dos operation "DIR  
<mask> /p")

**PARAMETERS:**

<mask:> [drive][pathname] (you can use wildcards in pathname option)

**[Utility] module**

---

**NAME: {Print File}**

**OPERATION:**

Choosing this option will prompt for filename and will print the file to printer connected to your computer. (dos operation "PRINT <filename>")

**PARAMETERS:**

<Filename:> filename

**[Utility] module**

---

**NAME: {nuMber conversion}**

**OPERATION:**

Display a number in various data types . Select the data type in which you wish to input the data. Input data is displayed in various other data types.

**PARAMETERS:**

<Input Data Type>

Select the data type in which you wish to input the data.

<Input>

Enter the value you want to see in other data type(s). Value entered will be displayed in various other data types.

**MESSAGES:**

See also: DATA TYPES in EDSP PARAMETERS.

**[Utility] module**

---

**NAME: {modify file header}**

**OPERATION:**

This function allows to change the data file header to following types:

- HDR - NHD (EDSP header to No header)
- HDR - LPN (EDSP header to LOTUS printer file header format)
- NHD - HDR (No header to EDSP header)
- LPN - HDR (LOTUS header format to EDSP header)

NOTE While converting from NHD (No header) to HDR (EDSP header), it prompts for the data type of the source file. However, data is copied as it is from the source file regardless of the data type input.

No data conversion is carried out while modifying the file header. The only modified header information in the output file is:

1. The data set name (file name)
2. Number of samples
3. Format (data type)

**PARAMETERS:**

Modify File Header window displays the default parameter values which can be modified by selecting their select character at <option :> prompt.

<option :>

'1' <Input Filename> (refer to section FILENAME under EDSP PARAMETERS)

'2' <Header Type> valid inputs are:

- 'HDR' (EDSP header)
- 'NHD' (No header)
- 'LPN' (Lotus format)

'3' <Output Filename> (refer to section FILENAME under EDSP PARAMETERS)

'4' <Header Type> valid inputs are:

- 'HDR' (EDSP header)
- 'NHD' (No header)
- 'LPN' (Lotus format)

'm' <Modify File Header> perform file modification.

<Output File Datatype :>

While converting from NHD (No header) to HDR (EDSP header), it prompts for the data type of the output file. For valid inputs for data types, refer section DATATYPE under EDSP PARAMETERS.

**[Utility] module**

---

**NAME:** {modify file sequence}

**OPERATION:**

This function will modify EDSP standard file data sequence. Input files are expected to be ASCII data files with EDSP header in one of the following 3 sequences:

Adjacent (complex) Real and imaginary data (CPX)

re1 im1  
re2 im2

Staggered Real and imaginary data (STG)

re1  
im1  
re2  
im2

Stack Real and imaginary data (STK)

re1  
re2  
im1  
im2

**PARAMETERS:**

Modify File Sequence window displays the default parameter values which can be modified by selecting their select character at <option :> prompt.

<option :>

'1' <Input Filename> (refer to section FILENAME under EDSP PARAMETERS)

'2' <Sequence Type> valid inputs are:

'CPX' (Adjacent (complex) type)  
'STG' (staggered type)  
'STK' (stack type)

'3' <Output Filename> (refer to section FILENAME under EDSP PARAMETERS)

'4' <Sequence Type> valid inputs are:

'CPX' (Adjacent (complex) type)  
'STG' (staggered type)  
'STK' (stack type)

'm' <Modify File Sequence>  
performs file sequence modification.

**[Application] module**

---

**NAME: {User Application}**

**OPERATION:**

Choosing this option will prompt for user application name and will run the application with given path name.

**PARAMETERS:**

<Name: > User application program name.



## APPENDIX

### A-1 Data file header information

The standard header for all data files is the following:

```
-----  
DATASET file.dat  
VERSION 0  
SIGNAL  
DATE  
TIME Mon July 10 09:30:29 1989  
DATA_TYPE FLOAT  
NUM_SAMPS 10  
INTERVAL 0  
MAX_VALUE 0  
MIN_VALUE 0  
VERT_UNITS 0  
HORZ_UNITS 0  
COMMENTS  
DATA  
0.000000E+000  
5.887877E+002  
9.232038E+002  
6.890812E+003  
1.988828E+002  
4.181121E+005  
-5.980890E+004  
-9.828283E-002  
-9.992834E+009  
2.309208E-007  
-----
```

There are 14 lines of header information. EDSP, however uses only header lines 6 and 7. Their meanings are described below:

**DATA\_TYPE**

See section DATATYPE under EDSP PARAMETERS for various data types supported.

**NUM\_SAMPS**

The number of lines of data in this file after the last header, DATA.

