

Winview Reference Guide

Second Edition

The Viewer Reference Guide was prepared by the technical staff of Innovative Integration, November 1997.

For further assistance contact

Innovative Integration
5785 Lindero Canyon Rd.
Westlake Village, CA 91362

PHONE: (818) 865-6150
FAX: (818) 879-1770
email: techsprt@innovative-dsp.com
FTP: ftp://ftp.innovative-dsp.com/pub/techsprt/public_html/ftp
WWW : http://www.innovative-dsp.com

This document is Copyright 1997 by Innovative Integration. All rights are reserved.

Document: VSS\ documents\viewer.doc

Introduction

Viewer is a software debugging utility supplied with Innovative Integration DSP boards. Viewer supports interactive execution of each of the DLL functions supplied in the Zuma toolset, interactive display of all DLL allocated and addressable memory structures in a variety of formats and other forms of low-level DSP board control.

Viewer is useful during the DSP code development cycle, before a Host application program has been written and debugged to deal with data flow between the DSP and the Host PC.

Viewer is based on a public domain Windows Forth package, Win32For. Viewer supports the full extensibility of Forth scripts may be written in the Forth language to assist in the Host/DSP debugging effort.

Starting the Program

Viewer may be executed by changing into the II_BOARD directory and executing the VIEWER.EXE program file. When invoked, the program will open a single window, shown below.

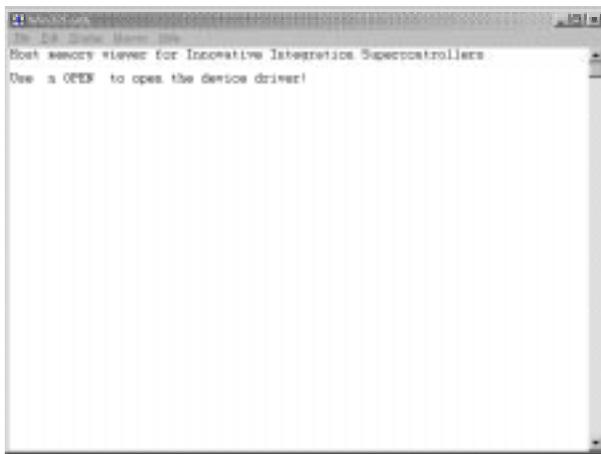


Figure 1: Viewer main window

Opening the Target

Before attempting to communicate with the target processor, you must first “open” the target device and driver. This is accomplished using the `open` command.



```

Host memory viewer for Innovative Integration Supercontrollers
Use a OPEN to open the device driver!

! open
      Device:      C
      Quiet:       0
      Target:      0
      cell_size:   4
      I/O address: 200
      Op Reg address: 200
      Dualport virt address: C240000
      Dualport size: 1000
      Busmaster virt address: C240000
      Busmaster size: 1000
      Busmaster phys address: 0000
      IRQ #:       7
      Vector:      0
      Mail: C240FC0
      DLL Version: 2.01
      VxD Version: 2.20
      Talker Version: 0.00
      Num_processors: 1 ok
  
```

Figure 2: Opening the target DSP

When the target is opened, strategic entries in the DLL cardinfo structure are read and displayed. While Viewer provides access to all of the cardinfo elements, only the most common ones are displayed during open.

Accessing Shared Memory

All of Innovatives bus-based DSP products provide some sort of “shared memory”. On ISA bus boards, this is a fixed block of dual ported memory, accessible by both the Host PC processor and the DSP processor.

On newer, PCI-based boards, the shared memory is actually Host PC memory allocated by the board’s device driver as contiguous, page-locked memory suitable for use as a “bus-mastering” target or source.

Regardless of the type of shared memory, Viewer provides a means of accessing it. This can be very helpful during the development process, before your custom Host program has been developed. Ultimately you must generate custom code for both the Host and the target to provide the umbilical communications layer between the DSP program running on the target board and your Host application program. But in the interim, Viewer can be used to provide basic access to the shared memory pool and limited diagnostics capabilities.

Dumping Shared Memory

One of the most common requirements is for a means of viewing data transferred from the DSP to the Host via the bus during DSP target code development. This can be particularly handy when dealing with bus-mastering cards because even when using a JTAG-based debugger (such as Code Composer, included in our Developers Packages) it may not be possible to see the Host memory targeted by the transfer.

For example, assume that you are attempting to bus master a packet of floating point data from the target DSP to the host shared memory. You may wish to verify that the data has been converted to IEEE format by the DSP properly and that the full packet has been transferred. Viewer supports a wide variety of dumping commands to make this easy.

```

Viewer
File Edit Display Memory Help
ok
signed ok
# ID dump
0 0 19211D5 8000 3E844 0000000000000000
10 70000 35258 8000 7048 0000000000000000
20 400150 400000 0 -7F1D7F11 0000000000000000
30 FE4 0 0 0 0000000000000000
ok
unsigned ok
# ID dump
0 0 19211D5 8000 3E844 0000000000000000
10 70000 35258 8000 7048 0000000000000000
20 400150 400000 0 00020000 0000000000000000
30 FE4 0 0 0 0000000000000000
ok
floating ok
# ID dump
0 1.0000000E+000 -3.7176616E+000 1.0000000E+000 1.0000000E+000 0000000000000000
10 1.0000000E+000 1.0000000E+000 1.0000000E+000 1.0000000E+000 0000000000000000
20 1.5000000E+000 1.5000000E+000 1.0000000E+000 0.0000000E+000 0000000000000000
30 1.0000000E+000 1.0000000E+000 1.0000000E+000 1.0000000E+000 0000000000000000
ok
same ok
# ID dump
0 0.0000000E-001 536.59517E-040 10.10190E-040 3.584622E-040 0000000000000000
10 6.4204047E-040 3.050451E-040 10.10190E-040 402.76923E-040 0000000000000000
20 54.535589E-040 54.79471E-040 0.0000000E-001 -248.81076E-040 0000000000000000
30 57.00402E-040 0.0000000E-001 0.0000000E-001 0.0000000E-001 0000000000000000
ok

```

Figure 3: Variants of Viewers dump command

Once data has been moved from the target DSP to the Host shared memory, you may use one of Viewers dump commands to see the data in various formats. The variants are

bmdump	To display a range of bus master memory
dpdump	To display a range of dual ported memory
dump	To display a range of host memory (Viewer application local space)

Viewer may also be used to view resources located in the I/O space of the PC. The variants are:

idump	To display a range of the DSP board's I/O space
odump	To display a range of the DSP board's operations register space (PCI only)
ioidump	To display a range of host I/O space (Win95 only)

If the target Talker is running (after a Reset), Viewer allows you to view target memory without moving its contents to the Host first. The commands are:

pdump	To display target DSP program memory (Talker monitor must be running)
ddump	To display target DSP data memory (Talker monitor must be running)

Each of these commands is *modal* and selects the default memory region to be accessed in all subsequent dump and plot commands. These spaces may be explicitly made active using the space mode commands:

program-space	Selects target program memory region as “default” memory space
data-space	Selects target data memory region as “default” memory space
bm-space	Selects bus master memory region as “default” memory space
dp-space	Selects dual port memory region as “default” memory space
memory-space	Selects host PC memory region as “default” memory space
i-space	Selects DSP board I/O block as “default” memory space
op-space	Selects target operation registers region as “default” memory space
io-space	Selects Host PC I/O region as “default” memory space

Dumping and plotting commands respect the current display format type, set by one of the format mode commands below.

signed	Sets display format to signed integer
unsigned	Sets the display format to unsigned integer
floating	Sets the display format to TI 32-bit floating-point format
ieee	Sets the display format to IEEE-754 32-bit floating point format.

These mode commands remain in effect until explicitly changed. The default mode is unsigned.

Modifying Shared Memory

The currently active memory region may be modified using a number of Viewer commands, listed below. These commands support clearing or filling a region of memory or altering a single cell of memory.

t!	Stores a value into specified target memory cell
t@	Retrieves a value from specified target memory cell
tf!	Stores a floating point number into specified target memory cell.
tf@	Retrieves a floating point number from specified target memory cell.
tfill	Fills a region of target memory with a seed value
terase	Clears a region of target memory to zeros.
tdump	To display a region of memory in the “current” memory space.

Each of the se target memory operators deals with the current default *memory space* and *data type*.

Plotting Shared Memory

Data may be plotted rather than dumped in numeric format. Plotting is also modal and is controlled by the same display formatting commands listed above. To plot a data range, use the plot command, as shown below.

0 100 plot To plot the data range starting at offset 0 in the current memory region using the current display format .

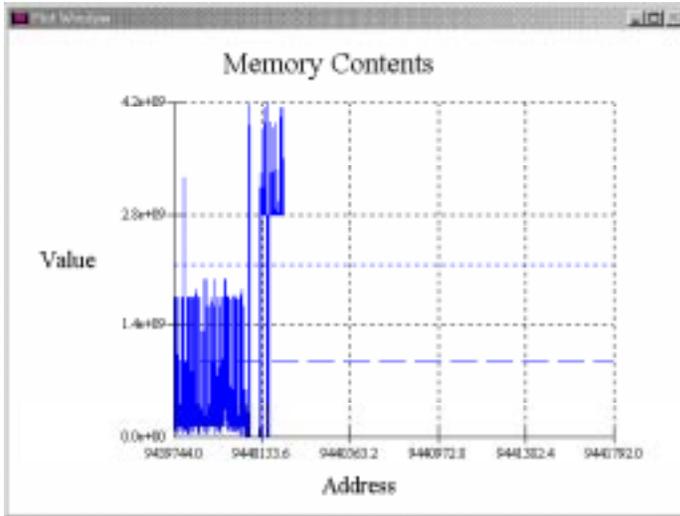


Figure 4: Viewers plot window

Plots, like dumps are automatically performed out of the currently selected memory region using the current data type.

Generic View DLL functions

The functions tabularized below may be executed interactively within Viewer. The parameters to each function must be pushed onto the stack in the order shown prior to invoking the function. The parameters column below lists the required parameters for each function. The dash in the parameter description denotes where the function name should be substituted when executing the command.

Table 1: Generic DLL Function List

Function	Parameters	Description
target_open	target - f	Opens driver for specified target DSP board. Returns boolean.
target_close	target - f	Closes driver for specified target DSP board. Returns boolean
target_cardinfo	target - a	Returns address of cardinfo structure for target.
iicoffld	string target handle - f	Loads a COFF executable file onto target DSP
host_interrupt_enable	target - f	Enables a previously installed virtual interrupt handler.
host_interrupt_disable	target - f	Disables a previously enabled virtual interrupt handler
host_interrupt_install	target fcn -	Installs a virtual interrupt handler
host_interrupt_deinstall	target -	Removes a virtual interrupt handler.
target_reset	target -	Physically asserts reset on the target DSP board.
target_run	target -	Deasserts reset on the target DSP board
target_outport	target port value -	Outputs a value to specified DSP board I/O port address
target_inport	target port - n	Inputs a value from specified DSP board I/O port
target_opreg_outport	target port value -	Outputs a value to specified DSP board operation port address
target_opreg_inport	target port - value	Inputs a value from specified DSP board operation port
target_control	target bit state -	Modifies a bit in the control register of the target DSP board
read_mailbox	target box - value	Reads the specified mailbox of the target DSP board
write_mailbox	target box value -	Writes to the specified mailbox of the target DSP board.
check_outbox	target box - f	Interrogates the specified output mailbox status
check_inbox	target box - f	Interrogates the specified input mailbox status
read_mb_terminate	target box key mode - f	Reads the specified input mailbox, if full
write_mb_terminate	target box value mode - f	Writes to the specified output mailbox, if empty
clear_mailboxes	target	Clears all mailboxes to empty state
mailbox_interrupt	target value -	Interrupts the target DSP after writing value to special mailbox
mailbox_interrupt_ack	target -	Acknowledges target to Host interrupt, returns special mailbox contents
target_key	target - key	Reads terminal mailbox, returns an 8-bit contents
target_emit	value target -	Writes 8-bit value to terminal mailbox
target_Tx	value target -	Writes 32-bit value to terminal mailbox
target_Rx_target	target - y	Reads 32-bit value from terminal mailbox
target_get_semaphore	semaphore target -	Gains ownership of specified target semaphore
target_interrupt	target -	Interrupts target DSP board
target_request_semaphore	semaphore target -	Requests ownership of specified target semaphore
target_own_semaphore	semaphore target -	Interrogates ownership status of specified semaphore
target_release_semaphore	semaphore target -	Relinquishes control of specified semaphore
target_check	target - f	Interrogates for Talker running on target
start_app	target -	Starts a previously downloaded target application program
start_talker	target - f	Starts the target Talker executing.
target_revision	target - f	Returns the revision of the target Talker
talker_fetch	target addr - n	Uses the Talker to fetch contents of specified target memory address
talker_store	target addr value -	Uses Talker to store value to specified target memory address
talker_download	target addr cnt -	Downloads a block of data or code to target DSP
talker_launch	target addr -	Launches downloaded application at boot vector address
talker_resume	target -	Resumes execution after suspended by Talker (not available all targets)
talker_registers	target -	Returns Talker register save address on target
target_slow	target -	Changes bus control to permit safe FLASH ROM access
target_fast	target -	Changes bus control to support fast target code execution
talker_flash_sector_erase	sector target -	Erases specified sector in FLASH ROM on target
talker_flash_init	target -	Initializes FLASH ROM on target.
talker_flash_offset	offset target -	Specifies memory offset of base of FLASH ROM on target

Appendix I: Command Reference

Summary of Viewer features and commands

Viewer is capable of operating on a variety of memory regions. The commands below may be used to enable display or modification of specific memory regions using the target memory operator commands. These commands are modal and remain in effect until explicitly changed.

Table 2: Viewer “target” memory selection commands

Function	Parameters	Description
io-space	--	Makes host I/O space the “current” target memory space
memory-space	--	Makes host memory space the “current” target memory space. All addresses are relative to the base of Viewers executable image in memory
bm-space	--	Makes host bus master memory space the “current” target memory space. All addresses are relative to the base of the page-locked bus master memory block.
dpram-space	--	Makes shared DSP/Host dual-port memory the “current” target memory space. All addresses are relative to the base of the shared memory pool.
o-space	--	Makes DSP card operations register space the “current” target memory space. All target memory addresses are specified relative to the beginning of this region.
i-space	--	Makes DSP card I/O space the “current” target memory space. All target memory addresses are specified relative to the beginning of this region.
program-space	--	Makes DSP card program memory the “current” target memory space. All target memory addresses are specified relative to the beginning of this region.
data-space	--	Makes DSP card data memory space the “current” target memory space. All target memory addresses are specified relative to the beginning of this region.
i-space	--	Makes DSP card I/O space the “current” target memory space. All target memory addresses are specified relative to the beginning of this region.

Viewer supports storing and fetching from the “currently-selected” target memory region. The commands below may be used to modify the currently memory region. These commands are subject to the current target memory region mode, selected above.

Table 3: Viewer “target” memory operators

Function	Parameters	Description
t!	n a --	Pronounced “t store”. Stores integer n into address a in target memory. For example 0x100 0x1000 t! stores 100h into target memory address 1000h.
t@	a - n	Fetches integer n from address a in target memory. For example 0x1000 t@ returns contents of target memory address 1000h onto stack.
tf!	a - r -- (fp stack)	Stores floating pt r into address a in target memory. For example 1.23 0x1000 tf! stores floating point 1.23 into target memory address 1000h
tf@	a - -- r (fp stack)	Fetches floating pt r from address a in target memory. For example 0x1000 tf@ returns floating pt contents of target memory address 1000h.
tdump	a n --	Dumps n cells of the current memory region starting at address a according to current dump mode (ie signed). For example unsigned 0 100 ddump shows 100 cells of target memory starting at 0000h as unsigned integers.
tfill	a n c --	Fills n cells of target memory starting at a with integer c. For example: 0 100 0x1234 tfill fills 100 target memory cells starting at 0000h with value 1234h.
terase	a n --	Zeroes n cells of target memory starting at a. For example 0 1000 derase erases 1000 cells of target memory starting at offset n in the current memory space.

Viewer supports dumping ranges of target memory in text form and graphically. The commands below are used to display ranges of target memory.

Table 4: Target memory display operators

Function	Parameters	Description
tdump	a n --	Dumps n cells of target memory starting at offset a interpreted according to current dump mode. For example 0 100 DPDUMP shows 100 16-bit cells of dual port memory starting at D000:0000h
plot	a n --	Plots n cells of memory starting at address a according to current dump mode (ie signed) and active memory space. Requires EasyPlot in working directory. For example SIGNED 0 100 PLOT graphs 100 cells of memory starting at 0000h as unsigned integers.

The dump and plot commands operate on the current target memory region. The commands below modify how data is interpreted during the data display operation.

Table 5: TDUMP mode selector commands

Function	Parameters	Description
signed	--	Subsequent DUMPs/PLOTs show signed values.
unsigned	--	Subsequent DUMPs/PLOTs show unsigned values.
floating	--	Subsequent DUMPs/PLOTs show TI floating point values.
ieee	--	Subsequent DUMPs/PLOTs show IEEE floating point values.

The commands below are shorthand convenience forms of the dump and target memory access commands. They automatically select a target memory region and perform a target memory accesses using a single, short-form command. **Note:** All addresses are specified as *offsets* into the selected memory region.

Table 6: Shorthand memory dump commands

Function	Parameters	Description
iodump	a n --	Makes host I/O space current and dumps the specified range.
idump	a n --	Makes DSP card I/O space current and dumps the specified range.
odump	a n --	Makes DSP card operations space current and dumps the specified range.
bmdump	a n --	Makes host bus master memory space current and dumps the specified range.
dpdump	a n --	Makes dual port memory space current and dumps the specified range.
pdump	a n --	Makes target program memory space current and dumps the specified range.
ddump	a n --	Makes target data memory space current and dumps the specified range.
p@	a - n	Makes DSP program memory current and fetches from it.
p!	n a --	Makes DSP program memory current and stores into it.
d@	a - n	Makes DSP data memory current and fetches from it.
d!	n a --	Makes DSP data memory current and stores into it.
i@	a - n	Makes DSP board I/O space current and fetches from it.
i!	n a --	Makes DSP board I/O space current and stores into it.
o@	a - n	Makes DSP board operations register space current and fetches from it.
o!	n a --	Makes DSP board operations register space current and stores into it.
bm@	a - n	Makes bus master memory space current and fetches from it.
bm!	n a --	Makes bus master memory space current and stores into it.
dp@	a - n	Makes dual port memory space current and fetches from it.
dp!	n a --	Makes dual port memory space current and stores into it.
io@	a - n	Makes Host I/O space current and fetches from it. 32-bit form
io!	n a --	Makes Host I/O space current and stores into it.
ioh@	a - n	Makes Host I/O space current and fetches from it. 16-bit form
ioh!	n a --	Makes Host I/O space current and stores into it.
ioc@	a - n	Makes Host I/O space current and fetches from it. 8-bit form
ioc!	n a --	Makes Host I/O space current and stores into it.

Viewer maintains two independent user-accessible stacks onto which parameters are placed for consumption by Viewer commands.

The *parameter stack* is a 32-bit wide stack used to contain addresses and integer parameters to and results from functions.

The *floating point stack* is used to hold used to hold floating point parameters to and results from Viewer functions. Floating point arithmetic takes place directly on the 8087 numeric stack. Viewer interprets numbers as reals when an 'e' is embedded in a literal number. Parameters on the fp stack are denoted below the parameter stack notation in the tables below.

Table 7: Viewer math and binary operators

Function	Parameters	Description
+	n1 n2 - n	Adds n2 to n1 leaving the result n. For example: 10 20 + . adds 10 and 20 and prints the result.
-	n1 n2 - n	Subtracts n2 from n1 leaving the result n. For example: 20 10 - . subtracts 10 from 20 and prints the result.
*	n1 n2 - n	Multiplies n1 by n2 leaving the result n.
/	n1 n2 - n	Divides n1 by n2 leaving the result n.
f+	-- r1 r2 - r	Adds r2 to r1 leaving the floating point result r. For example: 10 . 0 20 . 0 f+ f . adds 10 and 20 and prints the result.
f-	-- r1 r2 - n	Subtracts n2 from n1 leaving the result n. For example: 20 10 - . subtracts 10 from 20 and prints the result.
f*	-- r1 r2 - n	Multiplies r1 by r2 leaving the result r.
f/	-- r1 r2 - n	Divides r1 by r2 leaving the result r.
and	n1 n2 - n	Bitwise ANDs n1 and n2 leaving result n.
or	n1 n2 - n	Bitwise ORs n1 and n2 leaving result n.
xor	n1 n2 - n	Bitwise XORs n1 and n2 leaving result n.

The commands below support Viewer dictionary display and modification.

Table 8: Viewer dictionary commands

Function	Parameters	Description
words	--	Displays the names of all available Viewer commands.
empty	--	Empties the Viewer dictionary of all user-defined commands.
:	-- wordname	Begins definition of a new Viewer command (called a <i>word</i>).
;	--	Terminates definition of a new word.

The commands below affect global Viewer operation.

Table 9: Viewer system commands

Function	Parameters	Description
bye	--	Terminates Viewer, returns to the operating system.
z	-- filename	Invokes Codewright editor on specified filename
dir	-- dirspec	Displays the specified directory
chdir	-- dirspec	Changes to the specified directory

Table 10: Viewer system commands

Function	Parameters	Description
.s	--	Non-destructively prints entire parameter stack contents.
f.s	--	Non-destructively prints entire floating point stack contents.
.	n --	Prints the integer on top of the parameter stack.
f.	-- r --	Prints the real number on top of the floating point stack.
decimal	--	Changes default I/O conversion radix to decimal.
hex	--	Changes default I/O conversion radix to hexadecimal. Numeric literals prefixed with 0x are interpreted in hexadecimal, regardless of current radix.
bye	--	Terminates Viewer, returns to the operating system.
z	-- filename	Invokes Codewright editor on specified filename

The following commands are convenient Viewer command shortcuts to common DLL functions.

Table 11: Target DLL function shortcuts

Function	Parameters	Description
+reset	--	Places the target in the reset state
-reset	--	Removes the target from the reset state
reset	--	Reset-cycles the DSP board
run	-- filename	Downloads and runs specified COFF .OUT file
break	--	Fires a target interrupt
open	n --	Opens the specified target DSP device driver
close	--	Closes the currently open DSP device driver
inbox?	slot – f	Reports status of specified input mailbox
outbox?	slot – f	Reports status of specified output mailbox
?mailbox@	slot – n f	Reads input mailbox, if full. Returns value read and status.
?mailbox!	n slot --	Writes output mailbox, if empty. Returns status.
.boxes	--	Destructively dumps all input and output mailboxes