LAUTERBACH
DEVELOPMENT TOOLS

Integration with LabView

Release 09.2024

MANUAL

Integration with LabView

Integration with LabView

Version 05-Oct-2024

This document describes the installation process and the basic use of the LabVIEW Virtual Instruments (version 8.3 and later) for TRACE32.

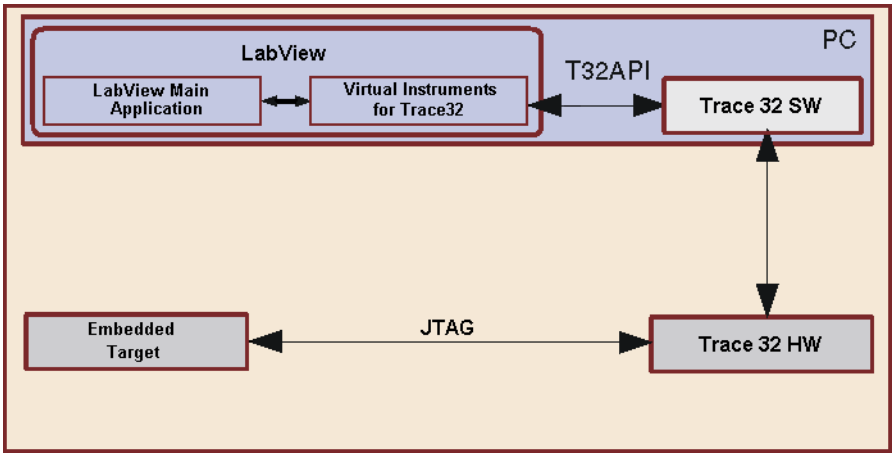
Overview

The LabVIEW Virtual Instruments for TRACE32 contains an interface that gives the LabVIEW internal VIs access to external hardware debuggers. Virtual Instruments for TRACE32 allows the LabVIEW application to interact with the program running on the embedded target.

The TRACE32 Software contains an interface for external control of TRACE32. Virtual Instruments for TRACE32 give the LabVIEW application possibility to control the hardware debugger and program run by the debugger.

The LabVIEW application makes use of the Virtual Instruments for TRACE32 through the **T32_LabView.dll** providing the TRACE32 API interface. The DLL contains the TRACE32 function calls via UDP/IP socket interface to the specified TRACE32 instance, after processing them, returns them using the same way.

Standard architecture for TRACE32 Virtual Instruments:



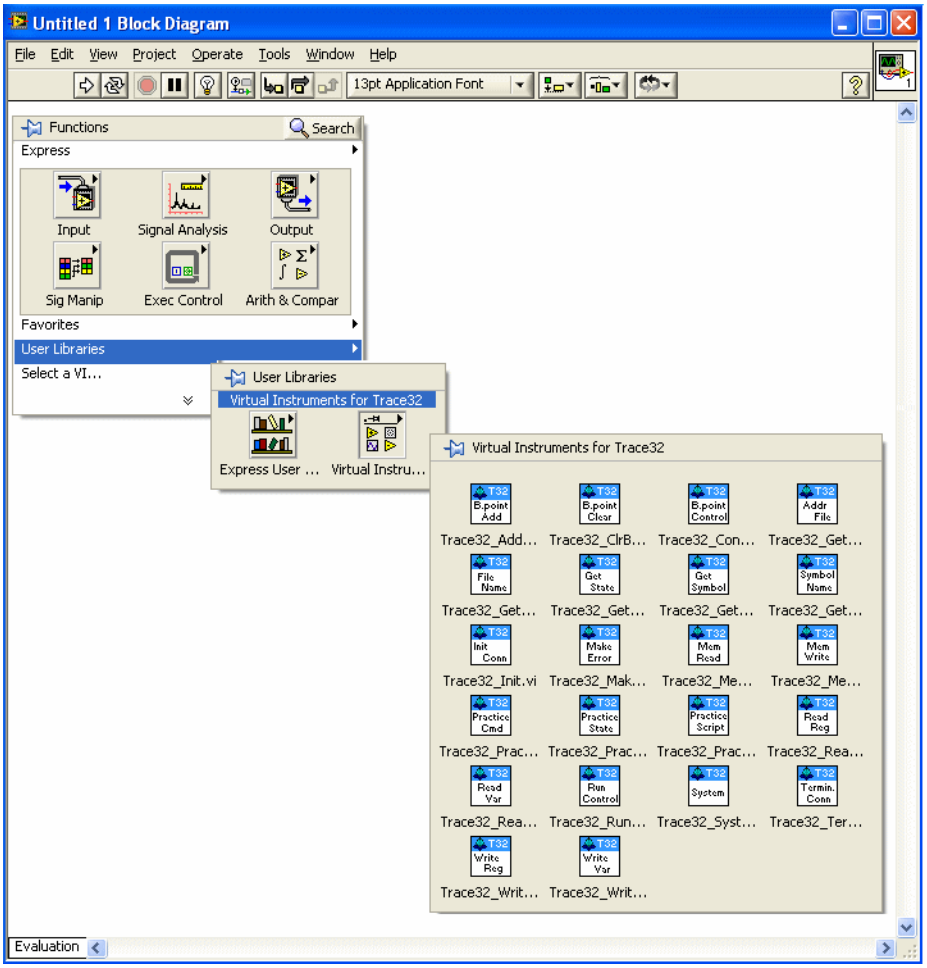
The library with Virtual Instruments for TRACE32 contains several block diagrams to communicate directly with TRACE32 and several controls simplifying the use of the Virtual Instruments.

NOTE:	This integration uses internally the **TRACE32 Remote API**. The Remote API has **restrictions** if TRACE32 runs in demo mode. Please see there for further details.

Installation and Configuration of VIs for TRACE32

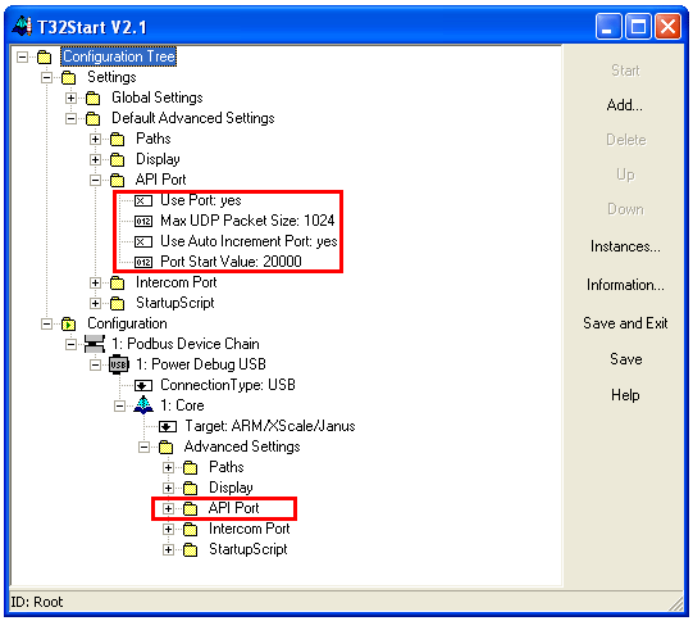
Follow these steps to install the VIs for TRACE32 and configure TRACE32:

1. Close TRACE32.
2. Close LabVIEW.
3. In the TRACE32 installation directory, browse to the subdirectory demo\env\labview. Unpack the contents of the **virtual_instruments_for_trace32_labview<version>.zip** file to the directory with user library files in your LabVIEW installation. The user library files are located in the following directory: **C:\Program Files\National Instruments\LabVIEW\user.lib**. Localization of the user.lib directory can be different, depending on the LabVIEW installation directory.
4. Start LabVIEW.
5. Open the new blank VI file and select the **Window -> Show Block Diagram** from menu. Click the right mouse button to see the functions palette and select **User Libraries** and **Virtual Instruments for TRACE32**.



Now, all Virtual Instruments for TRACE32 are available in LabView.

6. The VIs for TRACE32 communicate with TRACE32 through T32API over a local UDP port. Choose a free port (for example 20000 - default for T32_Init.vi) and set it as below.

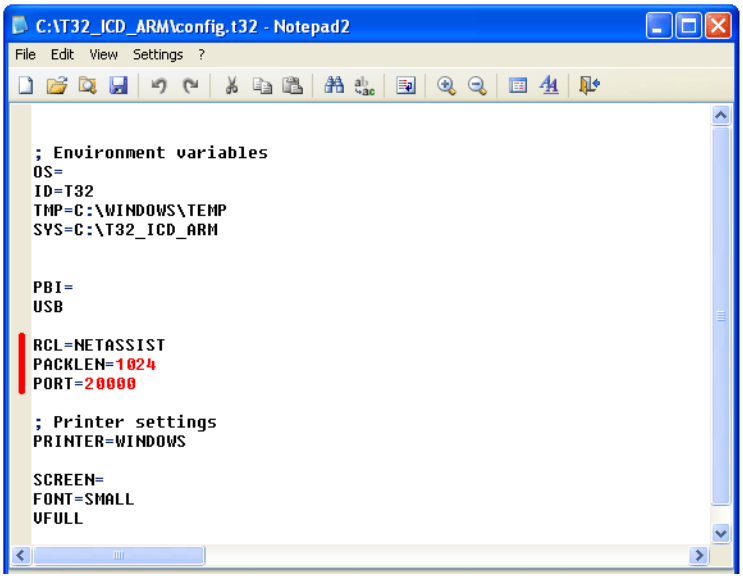


In case of using more than one TRACE32 simultaneously, please remember to set different API port numbers for each TRACE32 configuration.

If, for some reason, T32Start is not used, the API port number can be set in the configuration file “config.t32“, which usually can be found in the TRACE32 installation directory. Please add the following lines to this file:

```
RCL=NETASSIST
PACKLEN=1024
PORT=20000
```

Due to internals issues, please place an empty line in front and after these lines.



7. Start TRACE32.

   Now TRACE32 and LabView are properly configured.

Virtual Instruments for TRACE32 Description

The following section contains detailed descriptions of all elements included in Virtual Instruments for TRACE32.

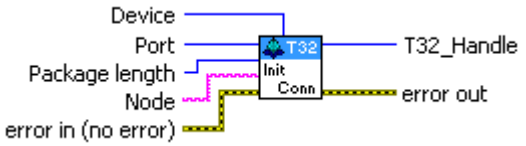
Trace32_Init.vi connects LabVIEW to TRACE32 and creates a “T32_Handle”. Ensure that all subsequent TRACE32 VIs are connected to this handle; especially take care in conditional executions, that the handle connections are continuous. TRACE32 VIs for LabVIEW older than 2011 may break, if no handle is connected. VIs for LabVIEW version 2011 and above check the handle and throw an error if the handle is missing.

Trace32_Terminate.vi closes the connection between TRACE32 and LabVIEW. Call this in any case; an attempt to call Trace32_Init.vi without having the connection closed before will throw an error.

Trace32_Init.vi

Creates a connection between LabVIEW and TRACE32. Must be called before any other VI will try to connect to TRACE32.

Connector Pane



Controls and Indicators

u32 Device - Specifies TRACE32 device. Currently there are only two values defined:

- **0x0** - Basic operating system of the TRACE32 (“::“); disables all device specific commands.
- **0x1** - Debugger (“B::“), including basic OS command (default).

u32 Port - Defines the UDP port to use. If omitted, the default value (20000) is set. Be sure that these settings fit to the RCL settings in the “`config.t32`“ file.

u32 Package length - Specifies the maximum data package length and can not be bigger than 1024 and must fit to the value defined in the “`config.t32`“ file. Default value is 1024.

Node - Defines the UDP host name, on which the TRACE32 display driver runs. Default is “localhost” (TRACE32 and LabVIEW are running on the same machine).

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

T32_Handle - TRACE32 handle. Created by T32_Init and required by all VIs for TRACE32. In 64-bit environments, the handle is actually of U64 type.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_Terminate.vi

Deletes connection between LabView and TRACE32. It is called when connection to TRACE32 should be terminated. After this operation T32_Handle is no longer valid.

Connector Pane

Controls and Indicators

T32_Handle - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32. After this operation is no longer valid.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

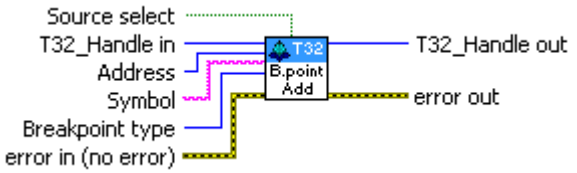
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_AddBreakpoint.vi

Sets breakpoints on specified absolute address or symbol name.

Connector Pane



Controls and Indicators

Source select - Specifies breakpoint source.
- **False** - Absolute address (default).
- **True** - Symbol name.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Address - Absolute address value.

Symbol - Symbol name.

u32 Breakpoint type - Specifies type of breakpoint.
- **0x0** - Default TRACE32 type of breakpoint (default).
- **0x1** - Onchip type of breakpoint.
- **0x2** - Hardware type of breakpoint.
- **0x3** - Software type of breakpoint

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

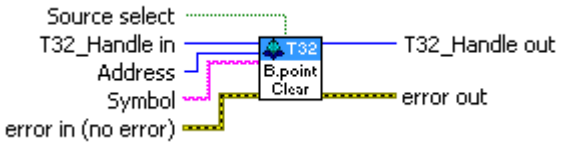
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_ClrBreakpoint.vi

Clears breakpoint on given absolute address or symbol name.

Connector Pane



Controls and Indicators

Source select - Specifies breakpoint source.
- **False** - Absolute address (default).
- **True** - Symbol name.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Address - Absolute address value.

Symbol - Symbol name.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

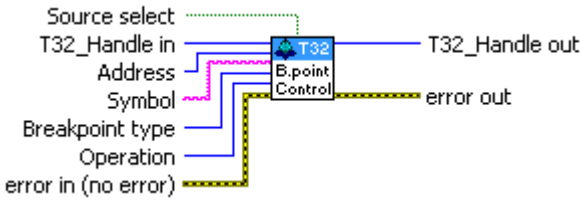
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_ControlBreakpoint.vi

Controls behavior of the breakpoint given by absolute address or symbol name.

Connector Pane



Controls and Indicators

Source select - Specifies breakpoint source.
- **False** - Absolute address (default).
- **True** - Symbol name.

u32 T32_Handle in - TRACE32 handle. Created by Trace32_Init and required by all VIs for TRACE32.

u32 Address - Absolute address value.

Symbol - Symbol name.

u32 Breakpoint type - Specifies type of breakpoint.
- **0x0** - Default TRACE32 type of breakpoint (default).
- **0x1** - Onchip type of breakpoint.
- **0x2** - Hardware type of breakpoint.
- **0x3** - Software type of breakpoint

u32 Operation - Specifies operation to execute.

- **0x0** - Add breakpoint.
- **0x1** - Delete breakpoint.
- **0x2** - Enable breakpoint.
- **0x3** - Disable breakpoint.
- **0x4** - Delete all breakpoints.
- **0x5** - Disable all breakpoints.
- **0x6** - Enable all breakpoints

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

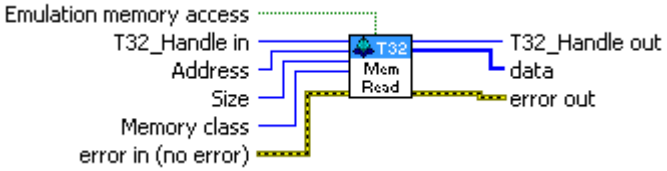
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_MemoryRead.vi (deprecated)

Reads data from target memory. The size of the data block is not limited.The access parameter defines the memory access class and access method.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (E:, dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Address - Target memory address to start read.

u32 Size - Number of bytes to read.

u32 Memory class - Memory access class, values as defined below.

Generically used memory access class values (independent of CPU architecture):	
0	Data access, D:
1	Program access, P:
12	AD
13	AP
15	USR

Additional memory access class values for ARM CPUs:	
2	CP0
3	ICEbreaker
4	ETM
5	CP14
6	CP15
7	ARM logical
8	THUMB logical
9	ARM physical
10	THUMB physical
11	ETB
14	DAP
Additional memory access class values for PowerPC CPUs:	
2	SPR
3	DCR
4	TLB
5	PMR
6	P: real mode address
7	P: virtual mode address

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u8 data - Block of output data.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

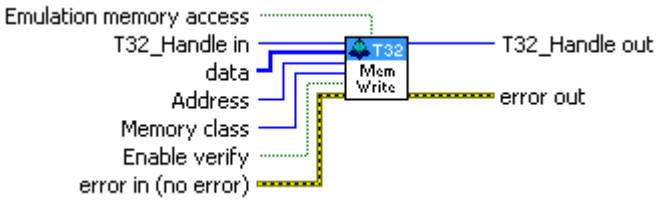
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_MemoryWrite.vi (deprecated)

Writes data to target memory. The size of the data block is not limited. This function should be used to access variables and make other not time critical memory writes. The access flags define the memory access class and access method.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (E:, dual port access).
- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u8 data - Data area to write.

u32 Address - Target memory address to start read.

u32 Memory class - Memory access class, see **'Trace32_MemoryRead.vi'**.

Enable verify - Set to enable verify after write.
- **False** - Disable verify.
- **True** - Enable verify.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

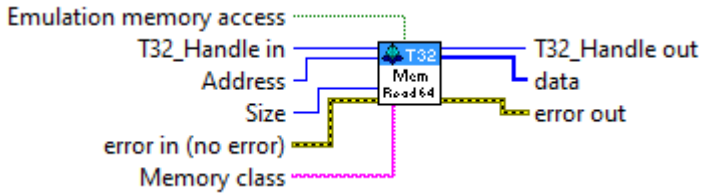
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_MemoryRead64.vi

Reads data from target memory. The size of the data block is not limited.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (E:, dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u64 Address - 64-bit target memory address to start read.

u32 Size - Number of bytes to read.

abc Memory class - Memory access class as string containing the access class specifier as listed in the Processor Architecture Manuals without the colon.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

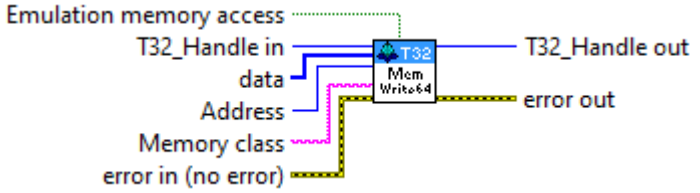
u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Trace32_MemoryWrite64.vi

Writes data to target memory. The size of the data block is not limited. This function should be used to access variales and make other not time critical memory writes.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (E:, dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u8 data - Data area to write.

u64 Address - 64-bit target memory address to start read.

u32 Memory class - Memory access class as string containing the access class specifier as listed in the Processor Architecture Manuals without the colon.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

Trace32_ReadCpuRegister.vi

Reads numerical value from the CPU register. Use the same register names as in TRACE32 CPU registers window.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Register name - Name of the CPU register. Use the same register names as in TRACE32 CPU registers window.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

- **status** - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- **source** - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Register value - Contains value read from CPU register.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_WriteCpuRegister.vi

Writes numerical value to the CPU register. Use the same register names as in TRACE32 CPU registers window.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Register name - Name of the CPU register. Use the same register names as in TRACE32 CPU registers window.

u32 Register value - Contains new value to write to the CPU register.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

- **status** - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- **source** - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

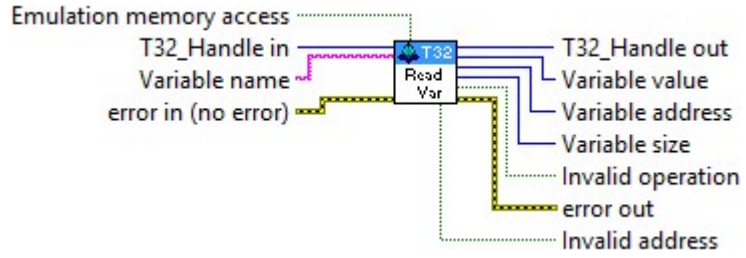
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_ReadVariable.vi

Returns the contents of the HLL expression as an integer. Use the same variable names as in application on TRACE32.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (access class E:, aka dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable name - Name of the variable. Use the same variable names as in application on TRACE32.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable value - Contains value read from variable.

Variable address - Contains address of the variable.

Variable size - Contains size of the variable in bytes.

Invalid operation - If true, indicates that the reading of the variable failed. E.g. when a PRACTICE script is active in TRACE32, this operation will fail.

Invalid address - If true, indicates that no address could be determined. E.g. if the given variable is a local variable out of scope, or if it is stored in a register.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_WriteVariable.vi

Writes numerical value to the variable. Use the same variable names as in application on TRACE32.

Connector Pane

Controls and Indicators,

Emulation memory access - Set for emulation memory access (access class E:, aka dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable name - Name of the variable. Use the same variable names as in application on TRACE32.

u32 Variable value - Contains new value to write to the variable.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

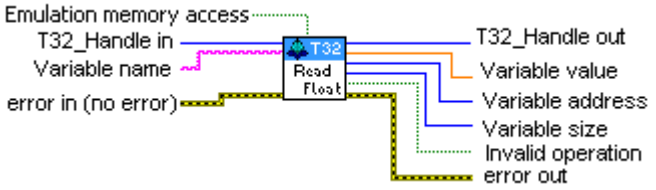
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_ReadVariableFloat.vi

Returns the contents of the HLL expression as a float. Use the same variable names as in application on TRACE32.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (access class E:, aka dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable name - Name of the variable. Use the same variable names as in application on TRACE32.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

SGL Variable value - Contains value read from variable.

u32 Variable address - Contains address of the variable.

u32 Variable size - Contains size of the variable in bytes.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_WriteVariableFloat.vi

Writes numerical float value to the variable. Use the same variable names as in application on TRACE32.

Connector Pane

Controls and Indicators,

Emulation memory access - Set for emulation memory access (access class E:, aka dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable name - Name of the variable. Use the same variable names as in application on TRACE32.

SGL Variable value - Contains new value to write to the variable.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

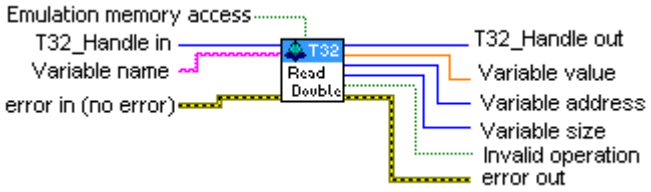
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_ReadVariableDouble.vi

Returns the contents of the HLL expression as a double. Use the same variable names as in application on TRACE32.

Connector Pane



Controls and Indicators

Emulation memory access - Set for emulation memory access (access class E:, aka dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable name - Name of the variable. Use the same variable names as in application on TRACE32.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

DBL Variable value - Contains value read from variable.

u32 Variable address - Contains address of the variable.

u32 Variable size - Contains size of the variable in bytes.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

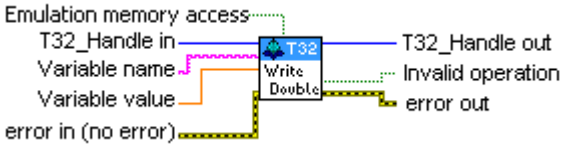
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_WriteVariableDouble.vi

Writes numerical double value to the variable. Use the same variable names as in application on TRACE32.

Connector Pane



Controls and Indicators,

Emulation memory access - Set for emulation memory access (access class E:, aka dual port access).

- **False** - Disable emulation memory access (default).
- **True** - Enable emulation memory access.

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Variable name - Name of the variable. Use the same variable names as in application on TRACE32.

DBL Variable value - Contains new value to write to the variable.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_GetState.vi

Use this VI to get the main state of the ICE. TRACE32 ICE can have four different states.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

- **status** - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- **source** - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 State - State can have four different values.

- **0x0** - General Information.
- **0x1** - System is halted, CPU makes no cycles.
- **0x2** - Emulation is stopped.
- **0x3** - Emulation is running.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

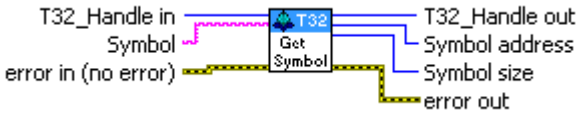
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_GetSymbol.vi

Evaluates HLL expression to address and symbol size.

Connector Pane



Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Symbol - Name of the symbol. Use the same symbol names as in application on TRACE32.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Symbol address - Contains address of the symbol.

u32 Symbol size - Contains size of the symbol in bytes.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

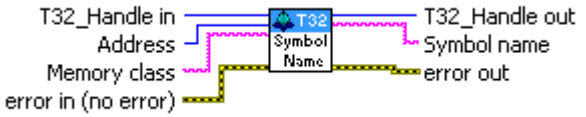
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_GetSymbolFromAddress.vi

Evaluates absolute address to symbol name.

Connector Pane



Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Address - Contains address of the symbol.

Memory class - Memory access class.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

- **status** - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- **source** - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 Symbol size - Contains size of the symbol in bytes.

Symbol name - Name of the symbol at address.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_GetAddressFromFile.vi

Evaluates source file name and line number to address.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

File - File name of the source file.

u32 Line number - Line number in the source file.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Address - Contains address of line.

u32 Size - Contains size of the line.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_GetSourceFileName.vi

Evaluates absolute address to the source file name with path and source file line number.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Address - Address from which source line should be found.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Symbol size - Contains size of the symbol in bytes.

File name - Name of the source file with path.

u32 Line number - Line number in source file.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_PracticeCmd.vi

With this VI a PRACTICE command is passed to TRACE32 for execution. Any valid PRACTICE command is allowed, including the start of a PRACTICE script (*.cmm) via the **DO** command.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Command - PRACTICE command to execute.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI , error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_PracticeScript.vi

Starts to execute PRACTICE script.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

Script name - PRACTICE script name with path.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

- **status** - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- **source** - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

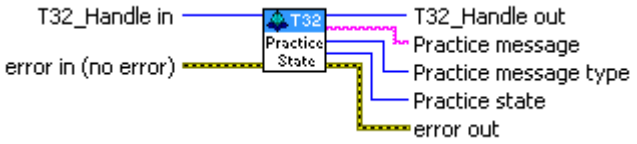
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_PracticeGetState.vi

Returns the run-state of PRACTICE. Use this command to poll for the end of a PRACTICE script started via'**Trace32_PracticeCmd.vi**' or'**Trace32_PracticeScript.vi**'.

Connector Pane



Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

PRACTICE message - Contents the last TRACE32 message from the message line.

u32 PRACTICE message type - The message types are currently defined as following and can be combined.

- **1** - General Information
- **2** - Error
- **8** - Status Information
- **16** - Error Information
- **32** - Temporary Display
- **64** - Temporary Information

u32 PRACTICE state- Returns the run-state of PRACTICE.

- **0** - Not running
- **1** - Running

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_RunControl.vi

Emulation control of the CPU.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Instruction- Emulation control command.

- **0** - Step
- **1** - Step over
- **2** - Go next
- **3** - Go return
- **4** - Go up
- **5** - Go
- **6** - Break
- **7** - Mode
- **8** - Reset

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

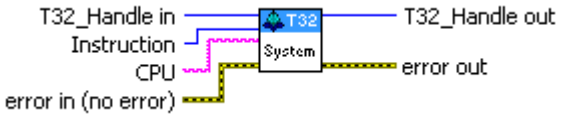
code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_System.vi

System control of the TRACE32.

Connector Pane



Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

u32 Instruction - System control command.

- **0** - System Down
- **1** - System Up
- **2** - System No Debug
- **3** - System Go
- **4** - System Attach
- **5** - System Stand By
- **6** - Set CPU

CPU - CPU name, require for “Set CPU” command.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

u32 T32_Handle out - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Trace32_Quit.vi

Quits Trace32. After this operation T32_Handle is no longer valid.

Connector Pane

Controls and Indicators

u32 T32_Handle in - TRACE32 handle. Created by **Trace32_Init** and required by all VIs for TRACE32. After this operation is no longer valid.

error in - Describes error conditions that occur before this VI or function runs. The default is `no error`. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out.

status - Is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

error out - Contains error information. If error in indicates that an error occurred before this VI, error out contains the same error information. Otherwise, it describes the error status that this VI produces.

status - Is TRUE (X) if an error occurred or FALSE.

code - Is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.

source - Specifies the origin of the error and is, in most cases, the name of the VI or function that produced the error. The default is an empty string.

Additional Controls for TRACE32

Controls can be used by the special VIs in order to simplify usage of instructions or commands, where they are required.

Trace32_RunCmd.ctl

Instructions for TRACE32_RunControl.vi.

Connector Pane

Controls and Indicators

u32 Instruction- Emulation control command.

- **0** - Step
- **1** - Step over
- **2** - Go next
- **3** - Go return
- **4** - Go up
- **5** - Go
- **6** - Break
- **7** - Mode
- **8** - Reset

Trace32_SystemCmd.ctl

Instructions for Trace32_System.vi.

Connector Pane

Controls and Indicators

u32 Instruction - System control command.

- **0** - System Down
- **1** - System Up
- **2** - System No Debug
- **3** - System Go
- **4** - System Attach
- **5** - System Stand By
- **6** - Set CPU

Trace32_OperationBreakpoint.ctl

Instructions for Trace32_ControlBreakpoint.vi.

Connector Pane

Controls and Indicators

u32 Operation - Specifies operation to execute.

- **0x0** - Add breakpoint.
- **0x1** - Delete breakpoint.
- **0x2** - Enable breakpoint.
- **0x3** - Disable breakpoint.
- **0x4** - Delete all breakpoints.
- **0x5** - Disable all breakpoints.
- **0x6** - Enable all breakpoints

Trace32_State.ctl

Indicators for Trace32_GetState.vi.

Connector Pane

Controls and Indicators

u32 State - TRACE32 State.

- **0x0** - General Information
- **0x1** - System is halted, CPU makes no cycles
- **0x2** - Emulation is stopped
- **0x3** - Emulation is running

Trace32_Error.ctl

Control used internally for identification of error.

Connector Pane

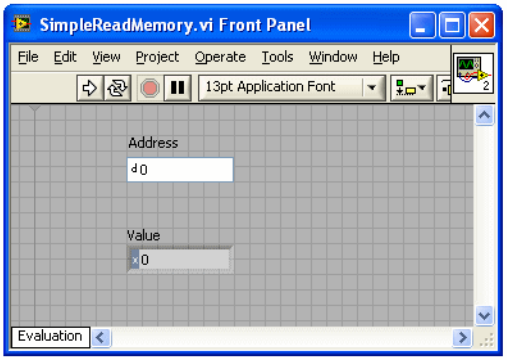
Controls and Indicators

u32 Error - Specifies type of error.

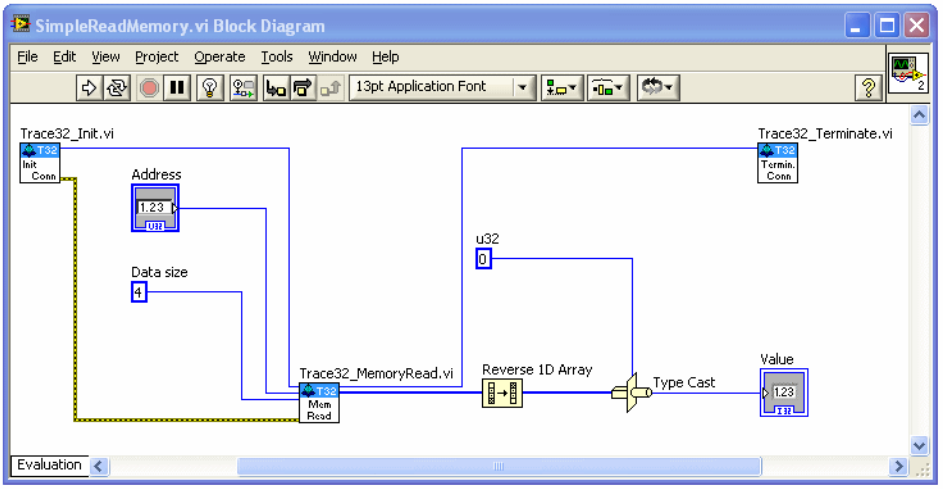
- **0** - Initialization failed.
- **1** - Termination failed.
- **2** - Memory read error.
- **3** - Memory write error.
- **4** - Add breakpoint error.
- **5** - Clear breakpoint error.
- **6** - Control breakpoint error.
- **7** - Get state failed.
- **8** - Get symbol failed.
- **9** - Write CPU register failed.
- **10** - Read CPU register failed.
- **11** - Write variable failed.
- **12** - Read variable failed.
- **13** - PRACTICE script error.
- **14** - PRACTICE command error.
- **15** - PRACTICE get state failed.
- **16** - Emulation control error.
- **17** - System error.
- **18** - Source file name failed.
- **19** - Symbol from address failed.
- **20** - Address from file failed.

In this quick start you will see how to create an application that will display content of memory from the embedded target.

1. Start LabView.
2. Create a new blank VI.
3. Create a front panel diagram similar to the one shown below.



4. Create a block diagram.



5. Start TRACE32.
6. Download and run the prepared application on embedded target (see **'Installation and Configuration VIs for TRACE32'**.
7. Run the SimpleReadMemory VI on LabView. LabView will read the four bytes from memory and will display the contents of memory in Value indicator on Front Panel.