LAUTERBACH
DEVELOPMENT TOOLS

Simulator for C166/ST10

MANUAL

Simulator for C166/ST10

TRACE32 Online Help

TRACE32 Directory

TRACE32 Index

TRACE32 Documents ..... 🗁

TRACE32 Instruction Set Simulators ..... 🗁

Simulator for C166/ST10 ..... **1**

TRACE32 Simulator License **4**

Quick Start of the Simulator **5**

Peripheral Simulation **7**

Troubleshooting **8**

FAQ **8**

CPU specific SYStem Settings and Restrictions **9**

SYStem.CONFIG — Configure debugger according to target topology — 9
SYStem.CONFIG.DAP — Describe the target configuration — 9
SYStem.CPU — CPU type — 9
SYStem.LOCK — Lock and tristate the debug port — 9
SYStem.MemAccess — Select run-time memory access method — 10
SYStem.Mode — Establish the communication with the simulator — 10
SYStem.Option.DUALPORT — Run-time memory access for all windows — 11
SYStem.Option.IMASKASM — Disable interrupts while single stepping — 11
SYStem.Option.IMASKHLL — Disable interrupts while HLL single stepping — 12
Special Functions — 13

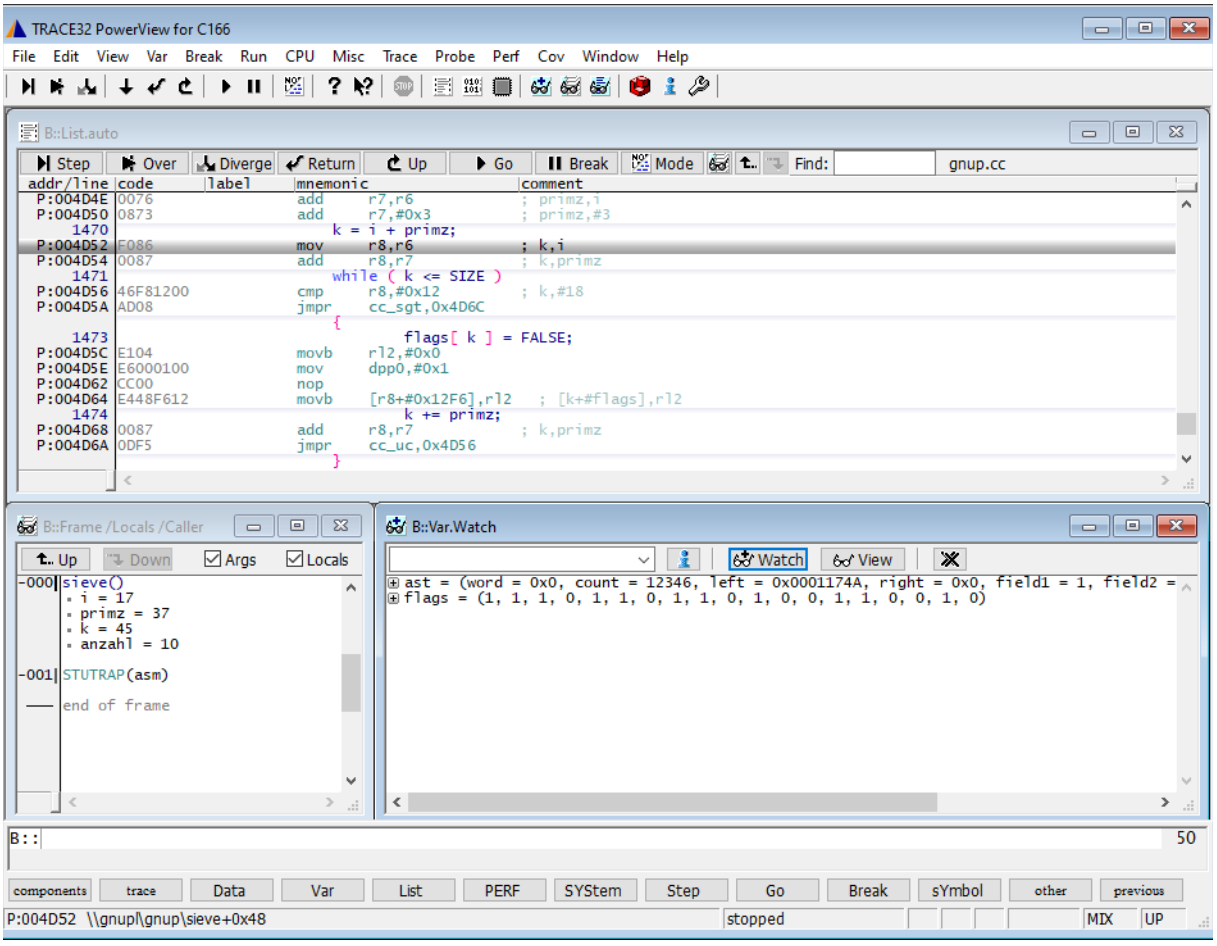
TrOnchip Commands **14**

TrOnchip.state — Display on-chip trigger window — 14
TrOnchip.RESet — Set on-chip trigger to default state — 14

Memory Classes **15**

Simulator for C166/ST10

Version 06-Jun-2024



All general commands are described in the **"PowerView Command Reference"** (ide_ref.pdf) and **"General Commands Reference"**.

TRACE32 Simulator License

[build 68859 - DVD 02/2016]

The extensive use of the TRACE32 Instruction Set Simulator requires a *TRACE32 Simulator License*.

For more information, see **www.lauterbach.com/sim_license.html**.

To start the simulator, proceed as follows:

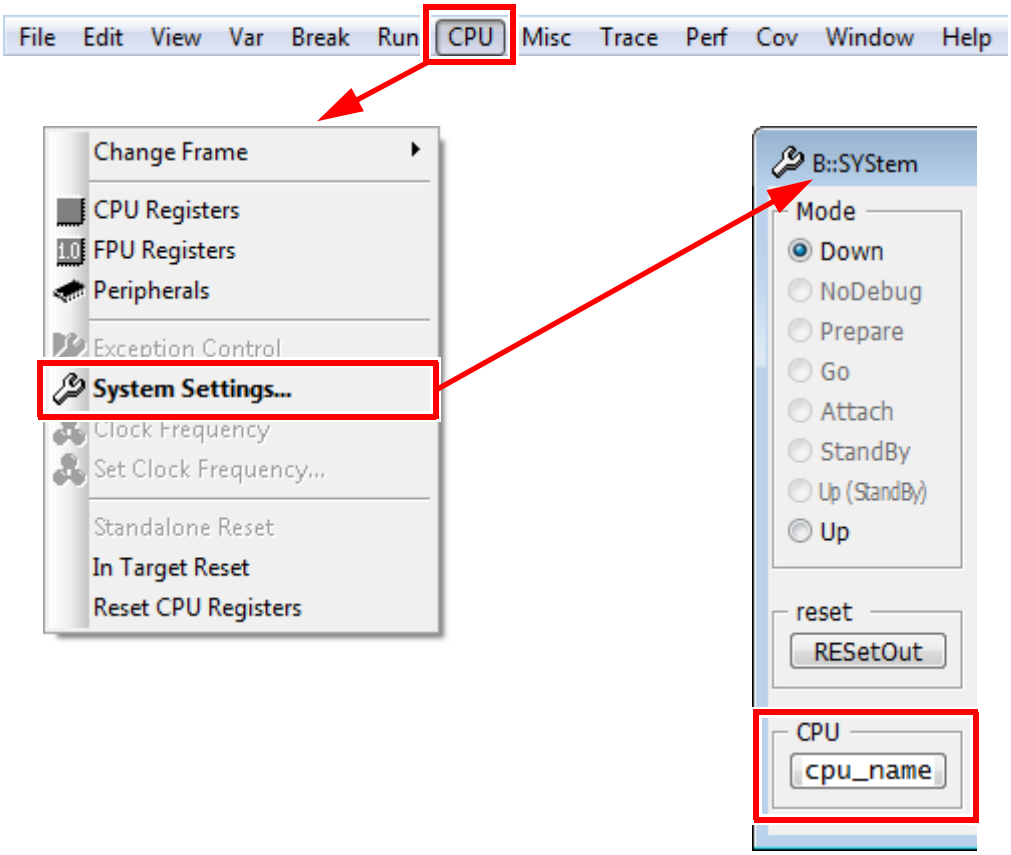
1. Select the device prompt for the Simulator and reset the system.

```
B::

RESet
```

The device prompt `B::` is normally already selected in the TRACE32 command line. If this is not the case, enter `B::` to set the correct device prompt. The **RESet** command is only necessary if you do not start directly after booting TRACE32.

2. Specify the CPU specific settings.




```
SYStem.CPU <cpu_name>
```

The default values of all other options are set in such a way that it should be possible to work without modification. Please consider that this is probably not the best configuration for your target.

3. Enter debug mode.

```
SYStem.Up
```

This command resets the CPU and enters debug mode. After this command is executed it is possible to access memory and registers.

4. Load the program.

```
Data.LOAD.<file_format> <file>      ; load program and symbols
```

See the **Data.LOAD** command reference for a list of supported file formats. If uncertain about the required format, try **Data.LOAD.auto**.

A detailed description of the **Data.LOAD** command and all available options is given in the reference guide.

5. Start-up example

A typical start sequence is shown below. This sequence can be written to a PRACTICE script file (*.cmm, ASCII format) and executed with the command **DO** *<file>*.

```
B::                                 ; Select the ICD device prompt

WinCLEAR                            ; Clear all windows

SYStem.CPU <cpu_name>               ; Select CPU type

SYStem.Up                           ; Reset the target and enter
                                    ; debug mode

Data.LOAD.<file_format> <file>      ; Load the application

Register.Set pc main                ; Set the PC to function main

PER.view                            ; Show clearly arranged
                                    ; peripherals in window      *)

List.Mix                            ; Open source code window    *)

Register.view /SpotLight            ; Open register window       *)

Frame.view /Locals /Caller          ; Open the stack frame with
                                    ; local variables            *)

Var.Watch %Spotlight flags ast      ; Open watch window for
                                    ; variables                  *)
```

*) These commands open windows on the screen. The window position can be specified with the **WinPOS** command.

For more information, see **"API for TRACE32 Instruction Set Simulator"** (simulator_api.pdf).

Troubleshooting

FAQ

Please refer to https://support.lauterbach.com/kb.

SYStem.CONFIG Configure debugger according to target topology

The **SYStem.CONFIG** commands have no effect on the simulator. They are only provided to allow the user to run PRACTICE scripts written for the debugger within the simulator without modifications.

SYStem.CONFIG.DAP Describe the target configuration

The **SYStem.CONFIG** commands have no effect on the simulator. They are only provided to allow the user to run PRACTICE scripts written for the debugger within the simulator without modifications.

SYStem.CPU CPU type

Format:	**SYStem.CPU** <*mode*>
<*mode*>:	**161** | **163** | **164** | **165** | **166** | **167**

Selects the processor type. The ROM debugger requires also a modification in the debug monitor for different processor types.

SYStem.LOCK Lock and tristate the debug port

Format:	**SYStem.LOCK** [**ON** | **OFF**]

The command has no effect for the simulator.

Format: **SYStem.MemAccess Enable | StopAndGo | Denied**
SYStem.ACCESS (deprecated)

Enable Memory access during program execution to target is enabled.
CPU (deprecated)

Denied Memory access during program execution to target is disabled.

StopAndGo Temporarily halts the core(s) to perform the memory access. Each stop takes some time depending on the speed of the JTAG port, the number of the assigned cores, and the operations that should be performed.

SYStem.Mode Establish the communication with the simulator

Format: **SYStem.Mode** *<mode>*

SYStem.Down (alias for SYStem.Mode Down)
SYStem.Up (alias for SYStem.Mode Up)

<mode>: **Down**
Up

Default: Down.

Selects the target operating mode.

Down The CPU is in reset. Debug mode is not active. Default state and state after fatal errors.

Up The CPU is not in reset but halted. Debug mode is active. In this mode the CPU can be started and stopped. This is the most typical way to activate debugging.

Format: **SYStem.Option.DUALPORT** [**ON** | **OFF**]

Default: OFF.

Dualport access of memory while simulation in running.

SYStem.Option.IMASKASM Disable interrupts while single stepping

Format: **SYStem.Option.IMASKASM** [**ON** | **OFF**]

Default: OFF.

If enabled, the interrupt mask bits of the CPU will be set during assembler single-step operations. The interrupt routine is not executed during single-step operations. After single step the interrupt mask bits are restored to the value before the step.

Format: **SYStem.Option.IMASKHLL** [**ON** | **OFF**]

Default: OFF.

If enabled, the interrupt mask bits of the CPU will be set during HLL single-step operations. The interrupt routine is not executed during single-step operations. After single step the interrupt mask bits are restored to the value before the step.

DPP(<*offset*>**)**

Returns the memory addressed by the short pointer argument. The lower 14 bits of the argument hold the offset, the two upper bits the DPP selector.

TrOnchip.state — Display on-chip trigger window

Format: **TrOnchip.state**

Opens the **TrOnchip.state** window.

TrOnchip.RESet — Set on-chip trigger to default state

Format: **TrOnchip.RESet**

Sets the TrOnchip settings and trigger module to the default settings.

Memory Class	Description
D	Data
P	Program
B	Bit

C	Memory access by CPU
E	Emulation memory access
A	Absolute (physical) memory access