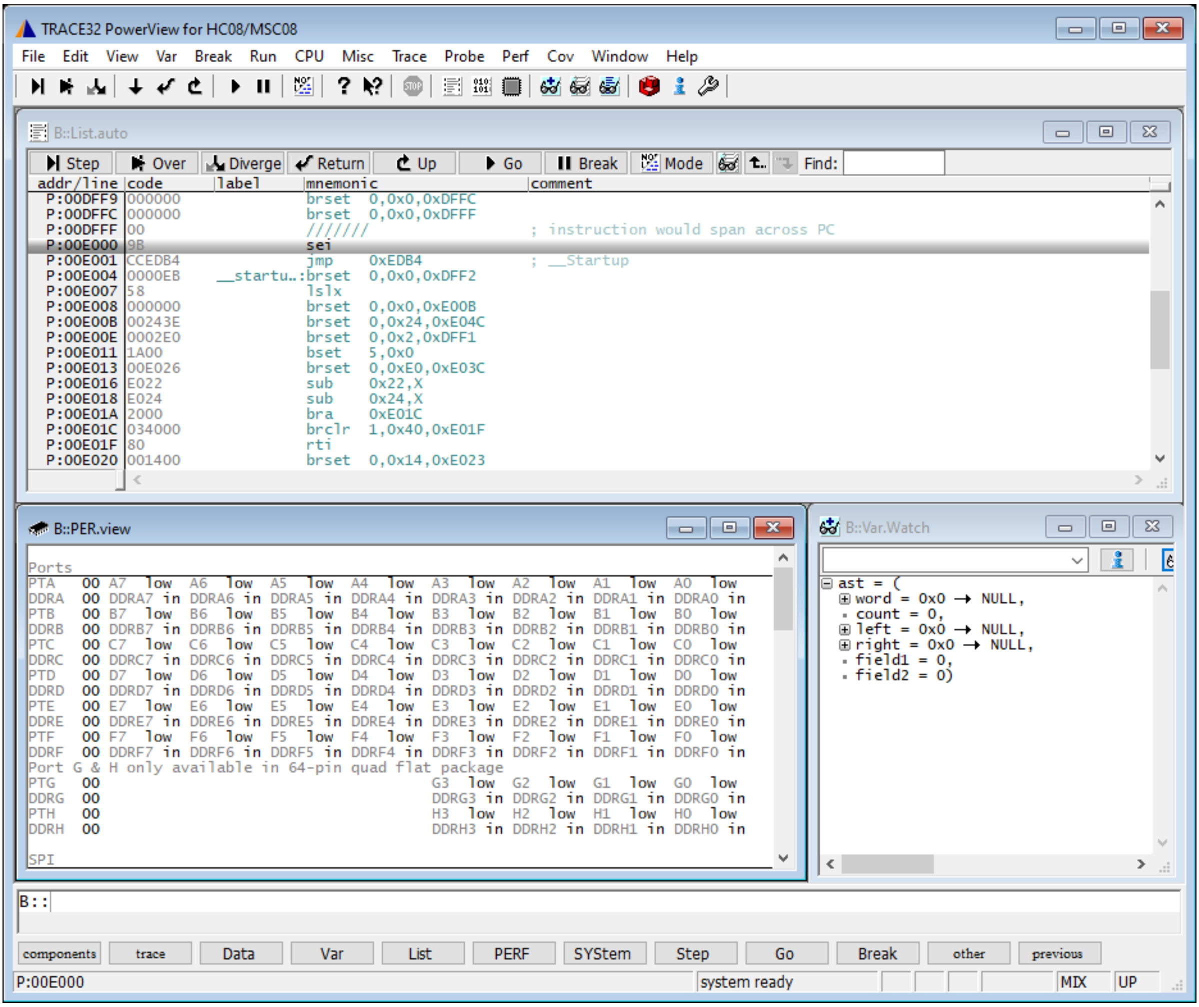
LAUTERBACH
DEVELOPMENT TOOLS

Simulator for HC08/MSC08

MANUAL

Simulator for HC08/MSC08

Version 06-Jun-2024



TRACE32 PowerView for HC08/MSC08
File Edit View Var Break Run CPU Misc Trace Probe Perf Cov Window Help
B::List.auto
Step Over Diverge Return Up Go Break Mode Find:
B::PER.view
B::Var.Watch
components trace Data Var List PERF SYStem Step Go Break other previous
system ready
MIX UP

```
addr/line code      label      mnemonic                  comment
  P:00DFF9 000000              brset   0,0x0,0xDFFC
  P:00DFFC 000000              brset   0,0x0,0xDFFF
  P:00DFFF 00                  ///////                   ; instruction would span across PC
  P:00E000 9B                  sei
  P:00E001 CCEDB4              jmp     0xEDB4            ; __Startup
  P:00E004 0000EB   __startu..:brset   0,0x0,0xDFF2
  P:00E007 58                  lslx
  P:00E008 000000              brset   0,0x0,0xE00B
  P:00E00B 00243E              brset   0,0x24,0xE04C
  P:00E00E 0002E0              brset   0,0x2,0xDFF1
  P:00E011 1A00                bset    5,0x0
  P:00E013 00E026              brset   0,0xE0,0xE03C
  P:00E016 E022                sub     0x22,X
  P:00E018 E024                sub     0x24,X
  P:00E01A 2000                bra     0xE01C
  P:00E01C 034000              brclr   1,0x40,0xE01F
  P:00E01F 80                  rti
  P:00E020 001400              brset   0,0x14,0xE023
```

```
Ports
PTA    00 A7    low  A6    low  A5    low  A4    low  A3    low  A2    low  A1    low  A0    low
DDRA   00 DDRA7 in   DDRA6 in   DDRA5 in   DDRA4 in   DDRA3 in   DDRA2 in   DDRA1 in   DDRA0 in
PTB    00 B7    low  B6    low  B5    low  B4    low  B3    low  B2    low  B1    low  B0    low
DDRB   00 DDRB7 in   DDRB6 in   DDRB5 in   DDRB4 in   DDRB3 in   DDRB2 in   DDRB1 in   DDRB0 in
PTC    00 C7    low  C6    low  C5    low  C4    low  C3    low  C2    low  C1    low  C0    low
DDRC   00 DDRC7 in   DDRC6 in   DDRC5 in   DDRC4 in   DDRC3 in   DDRC2 in   DDRC1 in   DDRC0 in
PTD    00 D7    low  D6    low  D5    low  D4    low  D3    low  D2    low  D1    low  D0    low
DDRD   00 DDRD7 in   DDRD6 in   DDRD5 in   DDRD4 in   DDRD3 in   DDRD2 in   DDRD1 in   DDRD0 in
PTE    00 E7    low  E6    low  E5    low  E4    low  E3    low  E2    low  E1    low  E0    low
DDRE   00 DDRE7 in   DDRE6 in   DDRE5 in   DDRE4 in   DDRE3 in   DDRE2 in   DDRE1 in   DDRE0 in
PTF    00 F7    low  F6    low  F5    low  F4    low  F3    low  F2    low  F1    low  F0    low
DDRF   00 DDRF7 in   DDRF6 in   DDRF5 in   DDRF4 in   DDRF3 in   DDRF2 in   DDRF1 in   DDRF0 in
Port G & H only available in 64-pin quad flat package
PTG    00                                          G3    low  G2    low  G1    low  G0    low
DDRG   00                                          DDRG3 in   DDRG2 in   DDRG1 in   DDRG0 in
PTH    00                                          H3    low  H2    low  H1    low  H0    low
DDRH   00                                          DDRH3 in   DDRH2 in   DDRH1 in   DDRH0 in

SPI
```

```
⊟ ast = (
  ⊞ word = 0x0 → NULL,
  · count = 0,
  ⊞ left = 0x0 → NULL,
  ⊞ right = 0x0 → NULL,
  · field1 = 0,
  · field2 = 0)
```

```
B::
```

```
P:00E000
```

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**.

Quick Start of the Simulator

To start the simulator, proceed as follows:

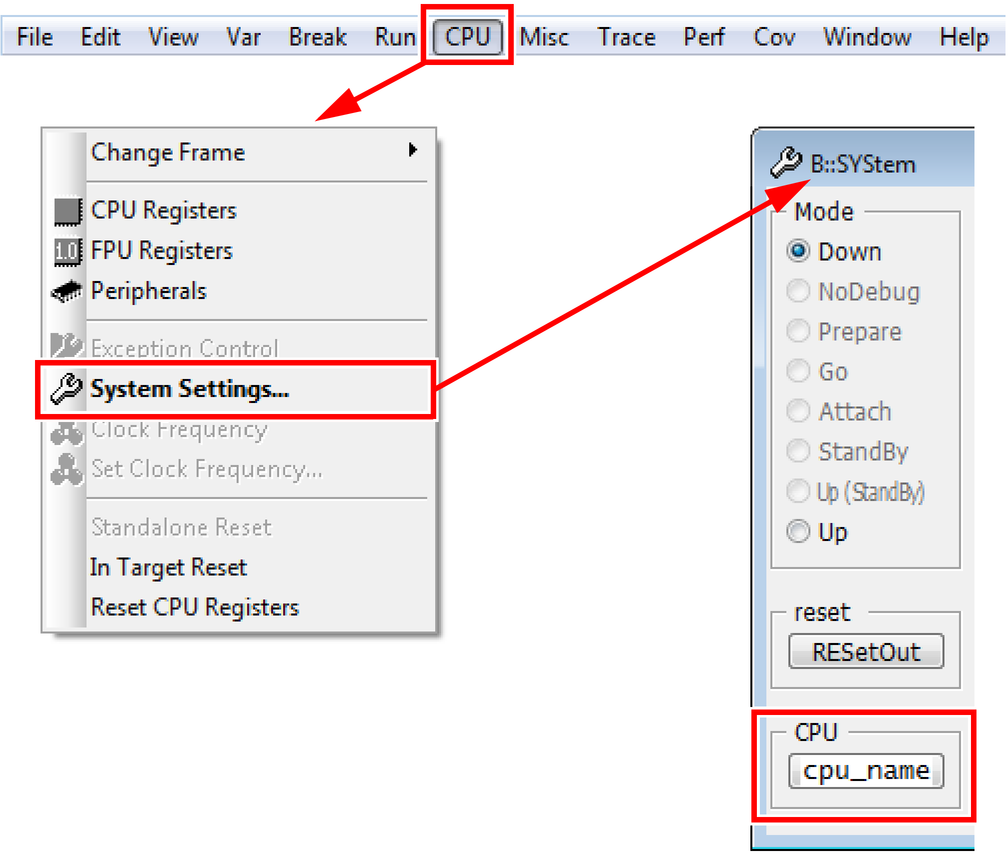
1. Select the device prompt for the Simulator and reset the system.

```
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.

Peripheral Simulation

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

Troubleshooting

No information available.

FAQ

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

CPU specific SYStem Settings and Restrictions

SYStem.CPU — Select CPU type

Format:	**SYStem.CPU** *<type>*
<type>:	**AUTO** **MC9S08GB32** | **MC9S08GB60** | **MC9S08GT32** | **MC9S08GT60** |

With this command the processor type is selected. The MC9S08 has internal device and mask specific registers which allow the debugger to select the CPU type automatically. This is done for all known versions if the pull down menu in the field ***CPU*** shows AUTO. For new mask revisions or devices you can select the CPU manually with this command.

NOTE: If you type `SYStem.CPU` at the command line followed by a blank, the softkeys (below the command line) provide you with all supported derivatives.

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** **NoDebug** **Go** **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.

NoDebug — The CPU is running. Debug mode is not active. Debug port is tristate. In this mode the target should behave as if the debugger is not connected.

Go The CPU is running. Debug mode is active. After this command the CPU can be stopped with the break command or if any break condition occurs.

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.

If the mode **Go** is selected, this mode will be entered, but the control button in the **SYStem.state** window jumps to the mode **Up**.

SYStem.MemAccess Select run-time memory access method

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

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

Denied (default) 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.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.

SYStem.Option.IMASKHLL Disable interrupts while HLL single stepping

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.

TrOnchip Commands

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 Classes

Memory Class	Description
P:	Program
D:	Data
E:	Hidden access to CPU memory while foreground emulation is active (“Emulation memory access” is the original of this name)