

TRACE32®



Linux Debugging

- ARM®/Cortex™
- Intel® Atom™
- MIPS® Architecture
- Power Architecture®
- and others



- Linux-aware debugging of single core systems
- Linux-aware debugging of SMP systems

- The only requirement for Stop-mode debugging is a functioning JTAG interface. This enables debugging from the reset vector.
- Debugging of the kernel and beyond process boundaries is possible because the TRACE32 debugger offers both Linux and MMU support.
- If the software ceases to react, the processor can be stopped to find out the point in the code where the processor crashed.
- If the processor is stopped, neither the kernel nor any other process can interfere with your analysis.

Run-mode Debugging

Debugging via GDB protocol is Run-mode debugging. This means only the selected process is stopped, while the kernel and all other processes are not influenced.

List of Linux processes

name	id	space	sel	stop
OMAP_UART3:	403.	403.	0x0193	
kpsmoused:	489.	489.	0x01E9	
irq/363-rtc0:	495.	495.	0x01EF	
kworker/u2:	513.	513.	0x0201	
sieve:	545.	545.	0x0221	•
gdbserver:	546.	546.	0x0222	
hello:	549.	549.	0x0225	✓ •

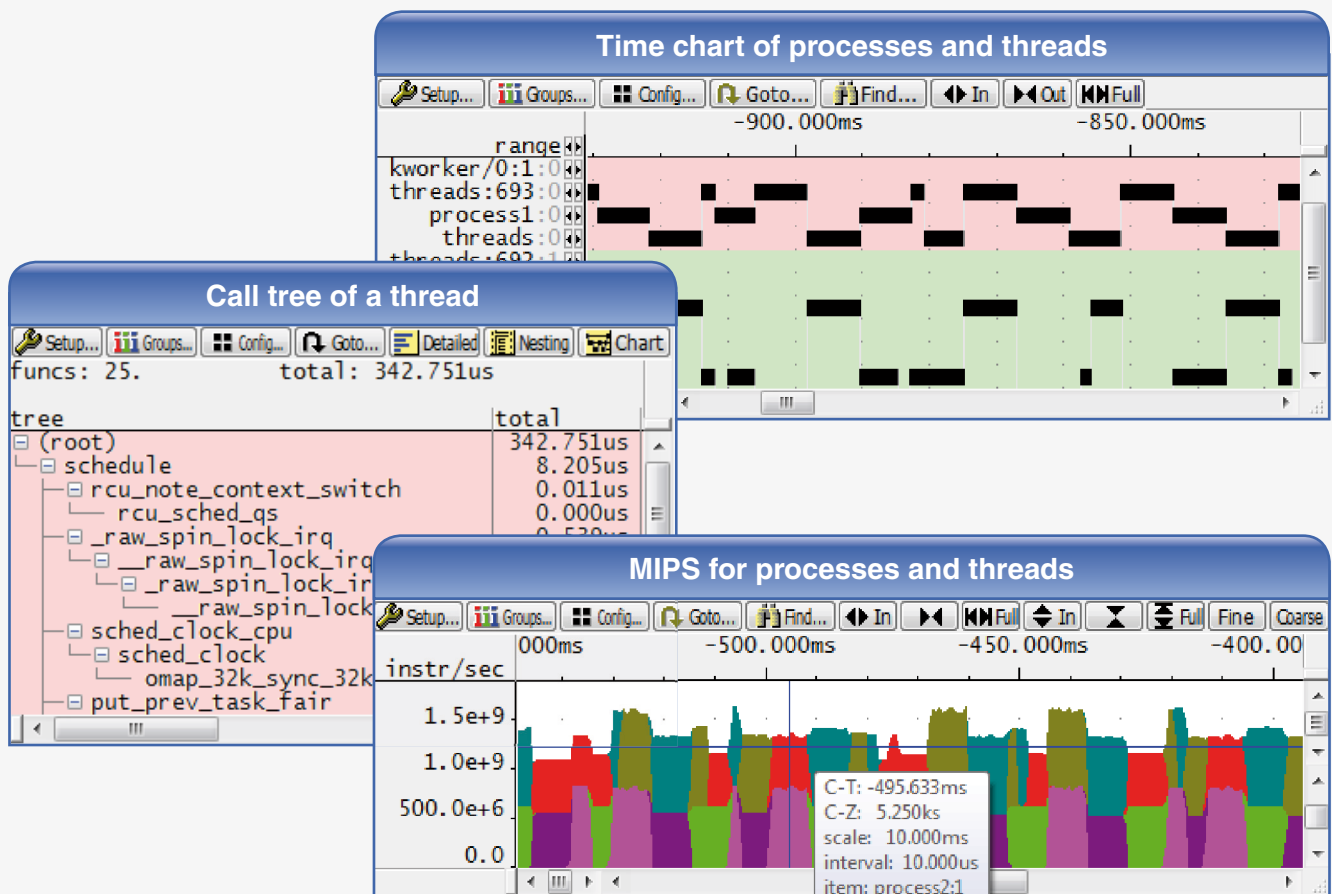
Benefits

- Ideal for pure application process debugging
- Communication interfaces remain active (e.g. Ethernet, RS232)
- As a special feature, TRACE32 allows the user to concurrently use Run- and Stop-mode debugging.

Linux-aware Tracing

Trace technologies allow an in-depth analysis of the behavior and the timing characteristics of the embedded system. Core trace modules generate information on the instruction execution and running processes for their related core.

Off-chip parallel or serial trace ports allow a TRACE32 trace tool to record this information for Linux-aware trace analysis, providing effective troubleshooting, comprehensive profiling and quality assurance.

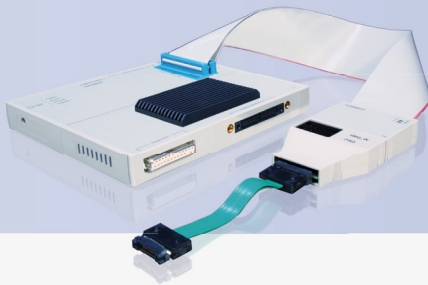


Hardware-based Products

Power Debug

Debugger via JTAG

Mainly Stop-mode debugging
Combined Run- and Stop-mode debugging possible



Real-Time Trace

Linux-aware single core and multicore tracing
Stop-mode debugging only

Power Trace

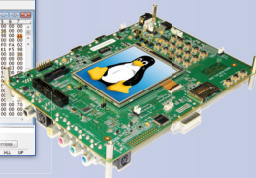
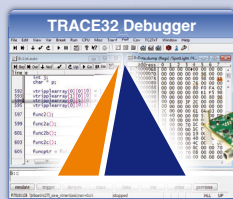
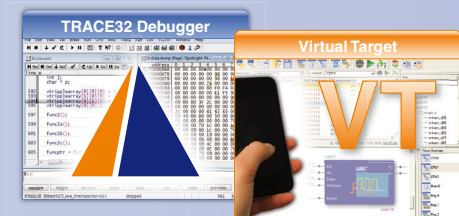
Software-only Products

Benefit from the same user interface on all products.

Front-End

Debugger for Virtual Targets

Mainly Stop-mode debugging
Run-mode debugging possible



Debugger via GDB Protocol

Run-mode debugging
GDB and KGDB

Front-End