



TIMING 1st CLASS

NEW 2.0
Version

Version 2.0 - State of the art timing suite

The T1-TARGET-SW has undergone a **fundamental reworking** and is now **more efficient** (saving up to 30% RAM), up to **50% faster** and yet delivers even **more accurate** results. The T1-HOST-SW too has been improved, comes with a choice of two newly developed skins and offers increased usability.



FEATURE	RELATED T1 COMPONENT	AVAILABLE IN T1 VERSION		
		1.5	1.6	2.0
Highly efficient scheduling tracing	T1.scope	✓	✓	✓
User defined stopwatches/events	T1.scope/T1.cont	✓	✓	✓
Offline timing analysis	T1.scope	✓	✓	✓
On-target timing analysis	T1.cont	✓	✓	✓
Full automation (e.g. for HIL tests)	T1.test	✓	✓	✓
Supervision of timing constraints	T1.scope/T1.cont	✓	✓	✓
No target HW modification required	all	✓	✓	✓
Online load injection	T1.delay	✓	✓	✓
Basic online code instrumentation	T1.flex	✓	✓	✓
Extended online code instrumentation	T1.flex		✓	✓
Focussed measurements	T1.cont		✓	✓
Improved online CPU-load analysis	T1.cont		✓	✓
Basic overhead consideration	T1.cont		✓	✓
Multicore support	all			✓
Timing comparison between versions	T1.diff			✓
AUTOSAR 4.0 Timing Extensions	T1.scope/T1.cont			✓
Functional groups	T1.scope/T1.cont			✓
Extended overhead consideration	T1.cont			✓
Online runnable instrumentation	T1.flex			✓
Extended T1.flex features	T1.flex			✓
Data Access Inspector	T1.flex			✓
Statistical analysis	T1.scope			✓
User data in traces	T1.scope			✓
Safe OS support	all			✓

NEW
Features



Version 2.0 - State of the art timing suite

Understand multicore timing behavior, visualize the history of your project's timing properties, integrate AUTOSAR timing features and more. Use V2.0 of T1 and benefit from a broad spectrum of new features, each of which derives from our direct experience in numerous, mass-production projects.

The number of **multicore** projects is steadily growing. Whereas a solid timing verification of a single-core system was just about achievable without tool support, this is, with multicore, unrealistic. T1 offers simultaneous tracing of different cores using just one interface (e.g. CAN) to the PC with little bandwidth. T1 plays a deciding role in taking control of multicore complexity and thus saves development time.

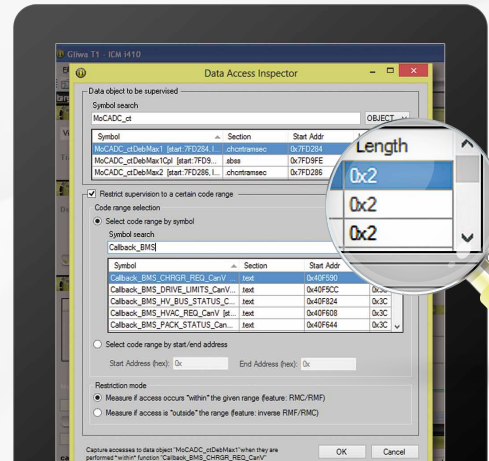
The new feature **T1 diff** allows **Timing comparison between versions** during project development. Automatable warnings show immediately that, for example, the computation time of a SWC has dramatically increased from one software version to the next. This automation uncovers problems early, leading to additional time savings.

T1 V2.0 supports the **AUTOSAR 4.0 Timing Extensions**. T1 interprets and automatically verifies AUTOSAR-specified timing requirements or event chains, for example from an OEM. An additional AUTOSAR feature is the possibility to collect together code fragments into **Functional groups**. For example, this allows all parts of a given SWC to be highlighted in the trace.

The previous version of T1 already supported the correction of measurement errors. V2.0 now offers **extended overhead consideration**, taking into account the different kinds of trace points and thus providing more accurate timing information.

Static instrumentation of all runnables of a system requires source code changes (not always possible), adds an inflexible execution time overhead and permanently occupies a considerable portion of the valuable trace buffer. The new **online runnable instrumentation** allows the user to decide at run-time exactly which runnables shall be traced.

V2.0 of T1 comes with an **extended set of T1.flex features** that allow very specific code and data measurements without the necessity to instrument the software at compile time.



A horror for any developer are data corruption or system crashes caused by an unintended write somewhere in the software. It can take weeks to track down these kinds of problems. The T1 **Data Access Inspector** allows data accesses to be visualized, with intended accesses being filtered out. This leaves only the unintended accesses, isolating the cause of the problem.

For all timing parameters, T1 now displays **statistical analysis** including histograms with configurable class widths.

The two worlds of „Scheduling/Timing“ and application data can be united in a single view with the feature **User data in traces**. T1 allows the tracing of user data of type uint8, sint8, uint16, sint16, uint32, sint32, uint64, sint64 and float as well as strings and arbitrary binary data blocks. This way, T1 can record and display the timing relationship to the scheduling. Many timing problems are connected with application data.

With its **Safe OS support**, T1 can now be easily integrated into a safety-OS environment, specifically where application code cannot directly disable interrupts.

WE DEVELOP PRECISION



www.gliwa.com

