



TIMING SUITE FOR REAL-TIME SYSTEMS

T1.timing

State of the art timing analysis with industry-hardened methods and tools. T1.timing empowers and enables and is the most frequently deployed timing tool in the automotive industry, being used for many years in more than a thousand mass-production projects.

As a worldwide premiere, the ISO 26262 ASIL D certified T1-TARGET-SW allows safe instrumentation-based timing analysis and timing supervision. On the desk, at the HIL and in the car – even in mass-production.



PC

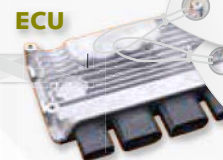
Host-Interface
e.g. USB or Ethernet

T1-HOST-SW

Interface HW
(if required)
e.g. GLIWA U2C,
any Vector HW,
media converter,
customer Gateway
or selected Debugger

Target Interface

typically an existing bus such as
CAN, CAN FD, Ethernet, FlexRay
or even debug-interfaces or INCA.
Supported protocols: T1, UDP on
ETH or Diagnosis ISO 14229 (UDS),
ISO 14230



ECU

T1-TARGET-SW

Little bandwidth required
by T1 → minimal impact
on existing communication

Minimal overhead
for tracing:
0.2% to 0.4% CPU-load

Pure SW solution:
no HW modification
or special HW required!

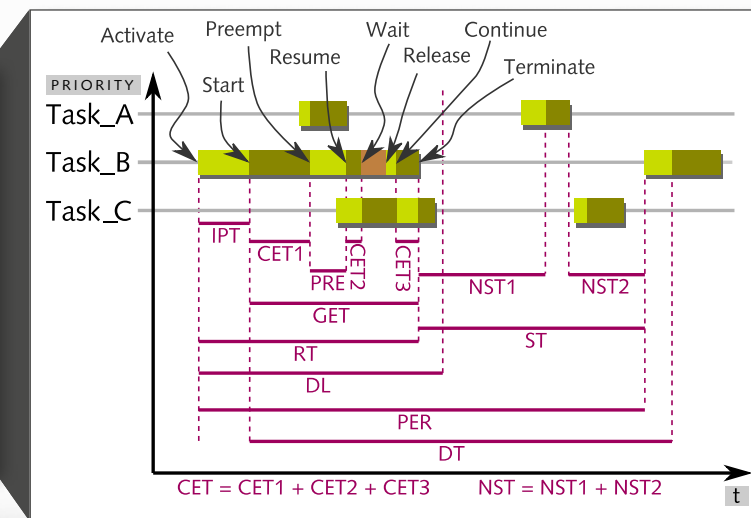
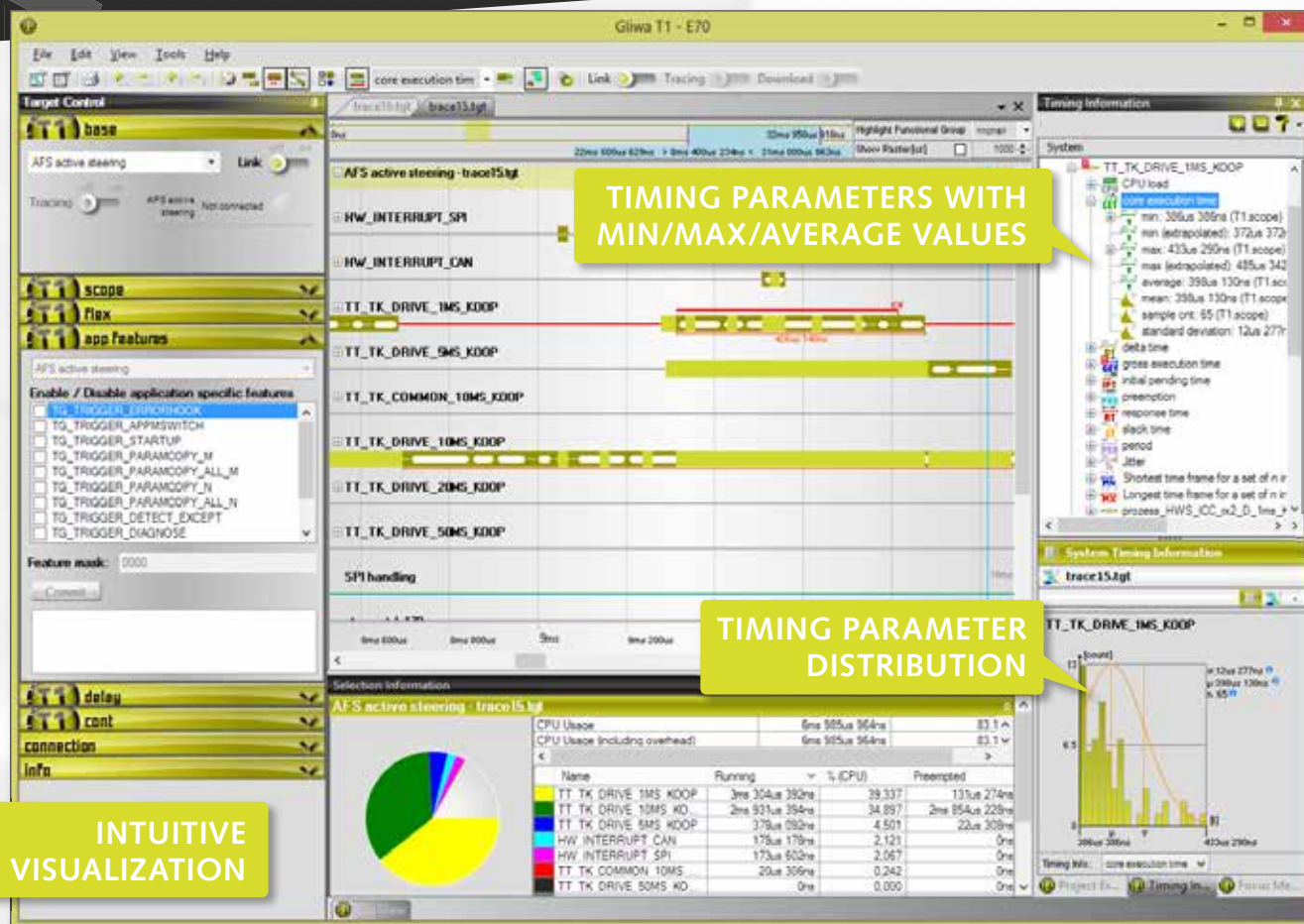
T1.scope

Whether during software development, integration or verification, without the visualization of the real system there is no insight into what is actually happening on the processor. T1.scope enables this insight and puts developers, integrators and testers in the position to be able to verify the timing of their embedded software.

Key benefits include:

- Intuitive **visualization of what is actually going on in the software** – it has never been easier to track down the cause of timing issues.
- Pure software instrumentation-based approach: **no hardware modification or special hardware required**
- Minimal **overhead of 0.2% to 0.4% CPU-load** for tracing all tasks and all interrupts in a typical set-up of an ECU for engine management or chassis control
- Detailed profiling: capturing of CPU-load and all kinds of **timing parameters with min/max/average/distribution** information
- Constraints: verification of **timing requirements**
- T1 offers the best insight for any target interface bandwidth:
 - Environments with **low bandwidth** (e. g. CAN): snapshots of a few hundred milliseconds
 - Environments with **high bandwidth** (e. g. Ethernet): streaming with real-time visualization and analysis for seconds, minutes, hours and days through add-on product T1.streaming

**OVERHEAD OF ONLY
0,2% TO 0,4% CPU-LOAD**

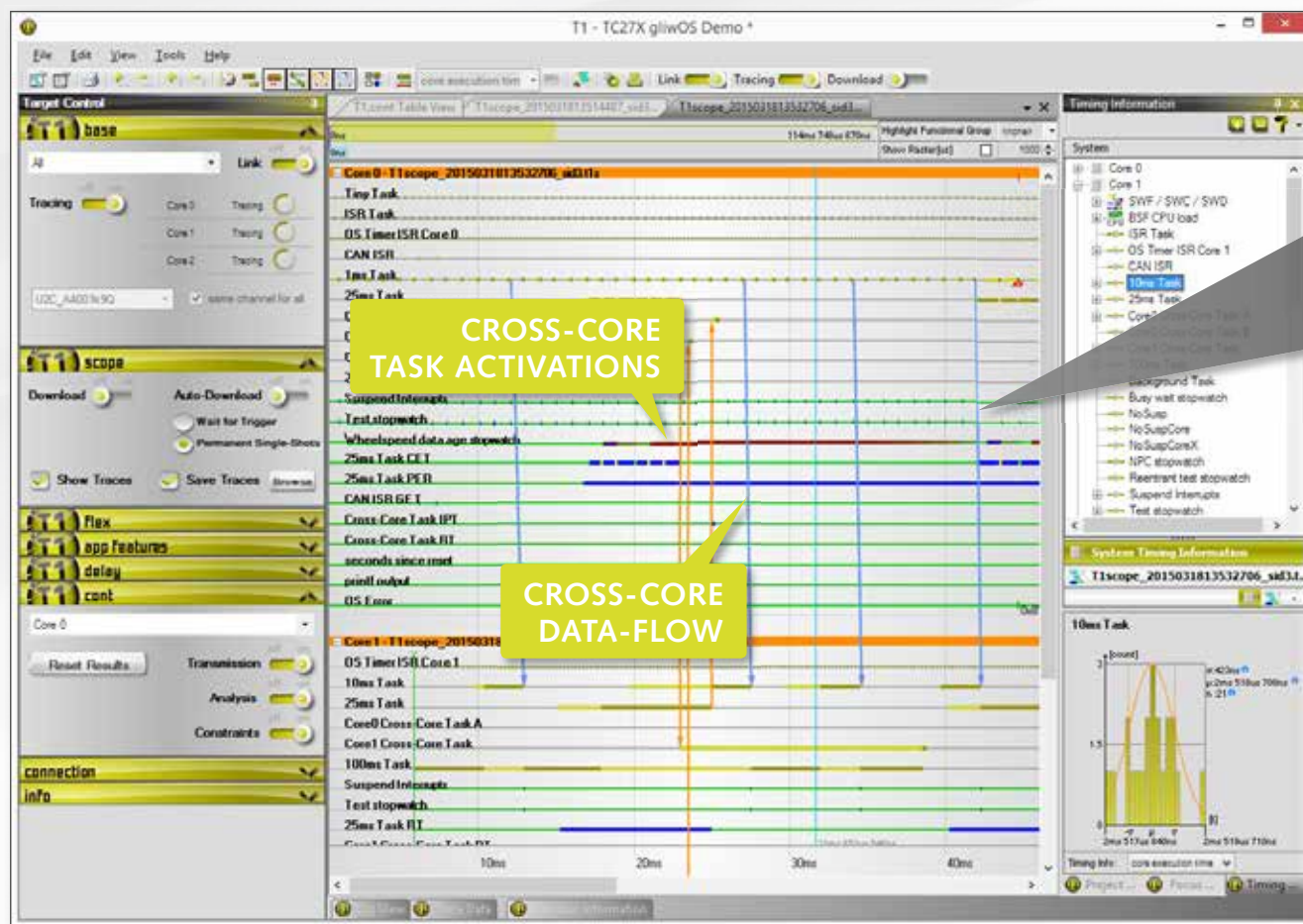


ABR.	EXPLANATION
IPT	initial pending time
CET	core execution time
GET	gross execution time
RT	response time
DT	delta time
PER	period
ST	slack time
PRE	Preemption time (AUTOSAR CP only)
DL	Deadline ("max. RT")
NST	Net slack time

T1.flex

ON-THE-FLY INSTRUMENTATION WHILE THE SOFTWARE IS EXECUTING

Who said instrumentation-based tracing is not flexible and requires software compilation when moving instrumentation points around? Be prepared to see some magic when using T1.flex, the flexible on-target and at-run-time instrumentation. Select any function or even some lines of code and **immediately see the corresponding timing parameters** such as min/max/average core-execution-time (CET), delta-time (DT) and CPU-load.



SELECT ANY VARIABLE AND GET READ AND WRITE ACCESSES VISUALIZED IN THE TRACE AS DATA-FLOW ARROWS

SELECT ANY FUNCTION OR FRAGMENT OF A FUNCTION...

...AND IMMEDIATELY RETRIEVE ESSENTIAL TIMING PARAMETERS

Key benefits include:

On-the-fly instrumentation **while the software is executing**

Suitable for **functions and code-fragments** (providing min/max/average CET, DT, CPU-load and more)

Suitable also for **data-accesses** (providing min/max data age, access frequency)

T1.cont

Who or what takes care of the timing when there is no PC connected to the ECU? T1.cont provides timing analysis on the target with minimal impact on the existing software. What's more T1.cont not only measures, it can also continuously supervise the timing – permanently making sure timing requirements are met.

Key benefits include:

- Permanent timing analysis and timing supervision
- **Ideal for profiling** the software (min/max timing parameters for selected tasks, runnables, data ages)
- Results can optionally be stored in non-volatile memory – **allowing profiling over weeks** with billions of executions

T1.delay

Test timing-related aspects of future versions of your software today! T1.delay makes it possible by providing scalable delay routines which consume a specified amount of net-run-time. These can be placed in the schedule at places where future functionality is to be added.

Other use-cases include determination of “head-room”: an empirical way to find out how much more CET can be added before the systems is overloaded.

T1.mod

A minor but very handy feature allowing reading from and writing to arbitrary memory. The intuitive symbol browser makes navigating to the data of interest much easier.

