

# TIMING SUITE FOR REAL-TIME SYSTEMS



## 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	Preempton time (AUTOSAR CP only)
DL	Deadline ("max. RT")
NST	Net slack time

## **ON-THE-FLY INSTRUMENTATION** WHILE THE SOFTWARE IS EXECUTING

		-	
1			
est Explo	ilear and a		ų.
Soli / Sou	NOR PRES		1.2
11.5	sten: VSCU		
= <b>0</b>	Source Nes		
	a ver		
0	Synthesia		
	Sundione (code)	N LIMT	2
	02_M_AND	N LIMIT	E
		N.LIMIT	E
	02 M AND	NUMIT	Ēv
•	Carl D. Constanting		2
Cunteer VEC	·u		
System VE	:u		•
System VEC Symbol smarch	10	* (al)	• •
System VEC Symbol smarch hume_mol_hwe	.us	(ie) *	
System VEC Symbol search here_mol_hwe Symbol	:u _10	* (al) = 5+:	2 × 1 ×
System VEC Symbol smarch here_mol_hwa Symbol HWE_ddPL_) GWC_ddPL_)	10 _10 MVE_100ms_begn_jst	* (al) = Sec a acco	• • • •
Syntem VEC Synted weach News_mpl_hvie Synted HWE_MPL_J HWE_MPL_J	10 _10 MVE_100ms_begin [st. MVE_100ms_end [star MVE 10ms_1st. between	* (sil) = Sec a acco t acco t acco y test	• > 4 4.4.4
Syntees VEC Symbol search Symbol HWE_MPL_) HWE_MPL_) HWE_MPL_)	10 _10 HWE_100ms_begin [star HWE_100ms_mt] [star HWE_10ms_lim_[segree HWE_10ms_lim_pegree	* (pl) = Sec a acco t acco r sec r sec r sec r sec	• • •
Syntees VEC Symbol search Symbol Hart[_CMPL_] HWE_MPL_] HWE_MPL_] HWE_GMPL_]	U _10 WE_100ms_beam (st WE_100ms_end (star WE_10ms_ts_perce WE_10ms_ts_ender	* (al) = Sec a. Acco t. Acco r. Sec = Sec	× > [2 · · ·
Synbol Imach Synbol Imach Synbol HWE_IMPLJ HWE_IMPLJ HWE_IMPLJ HWE_IMPLJ HWE_IMPLJ	U 10 10 10/5_100ms_beprist 10/5_100ms_tet_bepro 10/5_100ms_tet_endor 10/5_100ms_tet_endor 10/5_100ms_tet_endor	* (pl) = Sec a. ecc r. 3ed = 3ed = 3ed	• • • • •
Synbol Imach Synbol Imach Synbol HWE_IMPL_J HWE_IMPL_J HWE_IMPL_J HWE_IMPL_J HWE_IMPL_J	U 10 100 and page (at 10% 100m and page 10% 100m and page 10% 10m and and 10% 10m and and 10% 10m and and 10% 10m and and 10% 10m and and	* [si] = Sec a. acco r. 3cd c. 3cd c. 3cd r. 3cd	2 2 4 1 × 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2 ×
Syntex VIX Syntbol search here_prol_hwe Syntbol HWE_MAPL_) HWE_MAPL_) HWE_MAPL_) HWE_MAPL_) HWE_MAPL_) Search HWE_MAPL_)	10 10 10 10 10 10 10 10 10 10	* [st] = Sec a. eco r. led c. led r. led	4 3 4 4
Syntees VEX Symbol search share_net_inve share_net_inve share_net_inve share_net_inve share_net_inve share_net_inve share_net_inve share_net_inve share_net_inve share_net_inve	10 10 10 10 10 10 10 10 10 10	* (pl) = Sec a. eco r. led c. led y. led y. led r. led y. led r. led	¢ • • • • • • •
System VIX System VIX System such sweps sw	10 10 10 10 10 10 10 10 10 10	* (pl) = Sec t = 500 r = 5000 r = 500 r = 500	( <
Synder VEX Synded insearch hine_red Unive Synded wive_red Unive wive_red Unive wive_red Unive wive_red Unive wive_red Unive wive_red Unive Wive_red Unive Wive_red Unive Wive_red Unive Wive_red Unive C	10 10 10 10 10 10 10 10 10 10	* (pl) = Sec a acco c sec y Sec	• • • • • • • •
Synders VEX Synded awarch here_red_lives Synded wext_notel_ wext_notel_ wext_notel_ wext_notel_ wext_notel_ wext_notel_ wext_notel_ wext_notel_ wext_notel_ executions execution	10 10 10 10 10 10 10 10 10 10	* [st] = Sec a. ecc t. ecc r. ide c. ide r. ide r. ide r. ide	· · · · · · · · · · · · · · · · · · ·
Synbol seed in the collection of the collection	10 10 10 10 10 10 10 10 10 10	* [si] = Sec a. eco t. eco r. ided r. ided r. ided r. ided r. ided	
System VES Symbol asset Symbol Hive, and Jiwe Symbol HWE, IAPL, J HWE,	10 10 10 10 10 10 10 10 10 10	* (el) - Sec -	

Key benefits include:

On-the-fly instrumentation while the software is executing

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

Suitable also for dataaccesses (providing min/ max data age, access frequency)

#### PERMANENT TIMING ANALYSIS AND TIMING SUPERVISION

## 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.



GLIWA GmbH & Co. KG Pollinger Str. 1 82362 Weilheim i.OB. | Germany

gliwa.com

fon +49 - 881 - 13 85 22 - 0 fax +49 - 881 - 13 85 22 - 99 mail: info@gliwa.com