

**TRACE32®**

# DEBUG & TRACE

## TriCore™ AURIX™



```
/** the ...  
int TC4 mixed[SIZE];  
** result of the classification */  
CLSResult TC4_result;  
  
int core4_main(void)  
void TC4_mix(SInput* in) {  
    int i;  
    for(i=0;i<SIZE;++i) nIxProcess;  
    {  
        nProcessNext = 0;  
        for(int i = 0; i < 6; i++)  
            nNowProcessing[i] = -1;  
    }  
int core4_main() {  
    int nIxProcess;  
    while(1) {  
        nIxProcess = nProcessNext;  
        nNowProcessing[4] = nIxProcess;  
        TC4_mix(&inputBufs[nIxProcess]);  
        CLS_classify(&TC4_result, TC4_mixed);  
        TC0_fill(inputBufs[0].a, k*3, 0x24*1Base);  
        TC0_fill(inputBufs[0].b, k*5, 0x24*1Base);  
        nProcessNext = 0;  
        while(1) {  
            nIxProcess = nProcessNext;  
            nNowProcessing[0] = nIxPro  
            TC0_fill(inputBufs[nIxPro  
            CLS_classify(&TC0_result,  
            /** prepare for filling new buffer  
            /** determine by how many fill
```

# TriCore™

## TriCore™ AURIX™ at a glance

For more than 20 years Lauterbach has been supporting the latest TriCore™ AUDO™ and AURIX™ microcontrollers. Our tool-chain offers:

- Debugging of all TriCore CPUs and auxiliary cores.
- Tracing of TriCore CPUs and some auxiliary cores via on-chip trace, high-speed serial trace or DAP streaming even for production devices with miniMCDS.
- Code Coverage according to ISO 26262 (Tool Qualification Support Kit)
- AUTOSAR-aware debugging & profiling
- Support for Rust includes source code level debugging.



TriCore™ AURIX™

## TriCore™ AURIX™ Debug Solutions

target system	core type	core state
Single debug solution for all cores		
1: Chip		
1: SMP Sub System	TriCore	
0: TriCore0	TriCore	stopped
1: TriCore1	TriCore	stopped
2: TriCore2	TriCore	stopped
3: TriCore3	TriCore	stopped
4: TriCore4	TriCore	stopped
5: TriCore5	TriCore	stopped
2: Core	HSM\ARM	stopped
3: SMP Sub System	GTM	
0: Core	GTM	stopped
1: Core	GTM	stopped
2: Core	GTM	stopped
3: Core	GTM	stopped
4: Core	GTM	stopped
5: Core	GTM	stopped
6: Core	GTM	stopped
7: Core	GTM	stopped
4: TC3X_SCR	i8051	running



TRACE32® enables concurrent debugging of all TriCore™ CPUs as well as all auxiliary cores of an AURIX™ SoC.

This allows to examine their interaction in depth.

The above includes:

- TC4x: up to 6 TriCore™ CPUs, CSR (TriCore™), SCR (XC800), PPU (ARC), GTM and cDSP
- TC2xx/3xx: up to 6 TriCore™ CPUs, HSM (Cortex®-M), SCR (XC800), GTM

Beside standard features like Step/Go/Break and flash programming TRACE32® offers:

- AUTOSAR-aware debugging
- Debugging of multi-OS configurations
- Hypervisor-aware debugging for Memory Protection Unit (MPU) hypervisors

DISCOVER MORE: [www.lauterbach.com/supported-platforms/architectures/tricore](http://www.lauterbach.com/supported-platforms/architectures/tricore)

## TriCore™ AURIX™ Trace Solutions

- Records information, generated by the MCDS module, about instruction execution and data accesses of multiple TriCores™ CPUs, GTM, cDSP and PPU as well as transfers on the on-chip buses and peripheral state transitions.
- Provides triggers and filters to limit recorded data and to use trace buffers effectively.
- Enables long-term recoding by streaming trace data to the host (TRACE32® trace streaming).
- Allows to generate timestamps for correlating the program flow with external signals
- Trigger programming language (CTL) allows to use precise trace filters and exact triggers without detailed MCDS knowledge.

record	run	address	cycle	data	symbol	ti.back	busmaster
	0			/* Enabling core3 : tc3.. */			
	0			void TC0_StartCore3()			
	0			{			
	0			extern void __noinline__ __noreturn__ __jump__ _start_tc3(void);			
	0			CPU3_PC.U = (unsigned int)_start_tc3;			
50	0			movh.a a15,#0xF888			
	0			st.w [a15]-0x1F8,d15			
+00004136	0	D:F887FE08		wr-data	70101348		0.080us
+00004148	1	P:70100EA6		ptrace	.._intmem\cstart_tc1\_start+0x86		2.080us
	1			isync			
+00004154	1			st16.w [a15],d15			
	1	D:F0036258		wr-data	FFFC00F3		0.000us
	1			ld16.w d15,[a15]			
+00004166	1	D:F00362A8		rd-spb	FFFC000F		0.000us
+00004175	1	D:F0036258		rd-data	FFFC000F		0.000us
	1			movh.a a15,#0x0			

(feature availability depends on used device)

DISCOVER MORE: [www.lauterbach.com/supported-platforms/architectures/tricore](http://www.lauterbach.com/supported-platforms/architectures/tricore)

TriCore™

## Code Coverage for Functional Safety

- The code coverage functionality supports all metrics required for ISO26262 and other safety standards: statement, branch, MC/DC, function and call coverage.
- The TRACE32® Tool Qualification Support-Kit streamlines TRACE32® tool qualification effort and costs for the TRACE32® Instruction Set Simulator and all TriCore™ trace solutions.

**Multi-level code coverage overview**

address	tree	coverage	decision	0%	50%	100%	lines
P:C00004BE--C0000515	\main	incomplete	60.000%	[Progress Bar]			10.
P:C00004BE--C00004E9	\main	incomplete	42.857%	[Progress Bar]			7.
P:C00004EA--C0000515	\Delay	stmt+dc	100.000%	[Progress Bar]			3.
P:C0000516--C0001317	\carV2	incomplete	47.651%	[Progress Bar]			149.
P:C0000516--C00008E5	\CarDemoInit	incomplete	0.000%	[Progress Bar]			41.
P:C00008E6--C00010A9	\CarDemoCalc	incomplete	65.853%	[Progress Bar]			41.
P:C00010AA--C000110B	\IncreaseGear	incomplete	72.727%	[Progress Bar]			11.
P:C000110C--C0001171	\DecreaseGear	incomplete	0.000%	[Progress Bar]			11.
P:C0001172--C00011F9	\RevolutionsToDownshift	incomplete	80.000%	[Progress Bar]			15.
P:C00011FA--C0001281	\RevolutionsToUpshift	incomplete	80.000%	[Progress Bar]			15.
P:C0001282--C0001317	\Ratio	incomplete	80.000%	[Progress Bar]			15.
P:C0001318--C0001349	\fp_bit	incomplete	57.865%	[Progress Bar]			356.
P:C0001318--C0001349	\__extendsfdf2	stmt+dc	100.000%	[Progress Bar]			6.

**Source and object code measurement**

Id	Dec/Cond	True	False	Coverage	Addr/Line	Code	Label	Mnemonic
				ok	P:C0000DDA	2F 60	mov16.a	a15, d2
				ok	P:C0000DDC	D2 40	ld16.w	d2, [a15]0x34
				ok	P:C0000DBB	12 C2	add16	d2, #0x1
7	1.	.		taken	P:C0000DB0	00 0B E2 7F	jge	d2, d15, 0xc0000DP6
				incomplete	101	DecreaseGear (Car.Gear) ;		// change Gear--
				never	P:C0000DE4	FD 00 00 7B	movh	d15, #0xD000
				never	P:C0000DE8	F0 01 4F 1B	addi	d15, d15, #0x14
				never	P:C0000DRC	FF 60	mov16.a	a15, d15

DISCOVER MORE: [www.lauterbach.com/use-cases/tool-qualification](http://www.lauterbach.com/use-cases/tool-qualification)

## Profiling for AUTOSAR Classic Platform

- The support for the ARTI standard provides profiling of task state changes, runnables, and interrupts including their state changes. An export to the standardized ARTI format allows data exchange with 3rd-party timing tools.
- The support for the ORTI standard provides profiling of running tasks and (interrupt) service routines as well as a proprietary export.
- These use cases are supported by all TRACE32® trace solutions.

**Statistic of runnable runtimes**

range	count	min	max	avr
RteRunnable_ct_Calc_VehSpeed_ct_Calc_VehSpeed_Init:1	1.	162.513us	162.513us	162.513us
RteRunnable_ct_Calc_VehState_ct_Calc_VehState_Init:1	1.	172.575us	172.575us	172.575us
RteRunnable_ct_Calc_BrakeForce_ct_Calc_BrakeForce_Init:1	1.	78.050us	78.050us	78.050us
Runnable_ct_PlusCheck_VehAcc_ct_PlusCheck_VehAcc_Calc:0	433.	327.825us	365.013us	338.630us
Runnable_ct_PlusCheck_VehSpeed_ct_PlusCheck_VehSpeed_Calc:0	433.	332.638us	392.425us	346.737us
Angle_Calc:0	433.	309.313us	339.750us	319.480us
Shortterm:0	867.	240.325us	363.888us	272.254us
Angle_Calc:0	866.	288.750us	342.200us	315.724us
Longterm:0	87.	239.113us	268.900us	248.278us
Angle_Calc:0	87.	60.438us	93.638us	69.724us
VehAcc_Init:0	1.	175.738us	175.738us	175.738us
hSpeed_Init:0	1.	84.650us	84.650us	84.650us
Angle_Init:0	1.	128.488us	128.488us	128.488us
Isense_Init:0	1.	96.038us	96.038us	96.038us
Angle_Init:0	1.	87.275us	87.275us	87.275us
Angle_Init:0	1.	84.100us	84.100us	84.100us
BrakeForce:1	433.	52.475us	89.750us	65.776us
SetPoint_FL:1	433.	229.275us	261.300us	248.472us
SetPoint_FR:1	433.	244.838us	307.838us	269.156us
SetPoint_RL:1	433.	242.200us	294.738us	266.605us
SetPoint_RR:1	433.	190.888us	271.950us	228.252us
Action_Calc:1	433.	27.575us	31.525us	27.665us

**Chart of task states**

DISCOVER MORE: [www.lauterbach.com/os-awareness](http://www.lauterbach.com/os-awareness)

# DEBUG & TRACE SYSTEM

for any Infineon AURIX™ Microcontroller

## PowerDebug System



Powerful, modular, flexible debug system supporting TriCore™ and all auxiliary controllers

## PowerTrace System



Highest Performance Trace Recording of entire TriCore™-based microcontrollers

## CombiProbe 2



Streaming of on-chip trace via DAP debug port to external trace memory for AUTOSAR Profiling

DISCOVER MORE:  
[lauterbach.com/debugger](https://www.lauterbach.com/debugger)



DISCOVER MORE:  
[lauterbach.com/trace](https://www.lauterbach.com/trace)

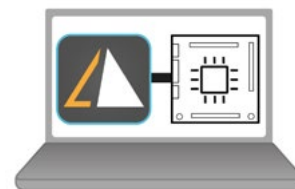


# SOFTWARE-Only Solutions

## DEBUG & TRACE VIA XCP



## DEBUG & TRACE OF SYNOPSYS VDK



DISCOVER MORE:  
[lauterbach.com/software](https://www.lauterbach.com/software)

