OS and RTE Profiling for TriCore™ AURIX™

AUTOSAR Classic basic software trace hooks and a filtered trace recording with TRACE32® allow the collection of OS and RTE runtime data for all CPUs of a TriCore™ AURIX™. This article introduces the solution jointly developed by Vector and Lauterbach.

Andrea Martin, Technical Marketing

Customer Demands

Multicore TriCore™ AURIX™ microcontrollers are a popular choice in the automotive industry for the development of Electronic Control Units (ECU). A typical AURIX™ second generation processor has up to six TriCore CPUs. The MCDS trace and trigger logic for the TriCore™ AURIX™ generates trace data for the program execution only for a limited number of CPUs, but never for all of them simultaneously.

Many customers, however, want trace data that includes all TriCore CPUs, especially when it comes to OS and RTE runtime measurements. In addition, many TRACE32 users demand that the trace data can be forwarded directly to a timing tool, to verify them against the system requirements.

Joint Solution

In order to meet these requirements, Vector and Lauterbach have jointly developed a solution for Vector's AUTOSAR classic basic software MICROSAR. The main idea is to configure trace hooks in such a way that trace data with timestamps are generated for all task switches, all task state changes, all ISR events, and all Runnable starts/ends.

The joint solution consists of the following steps:

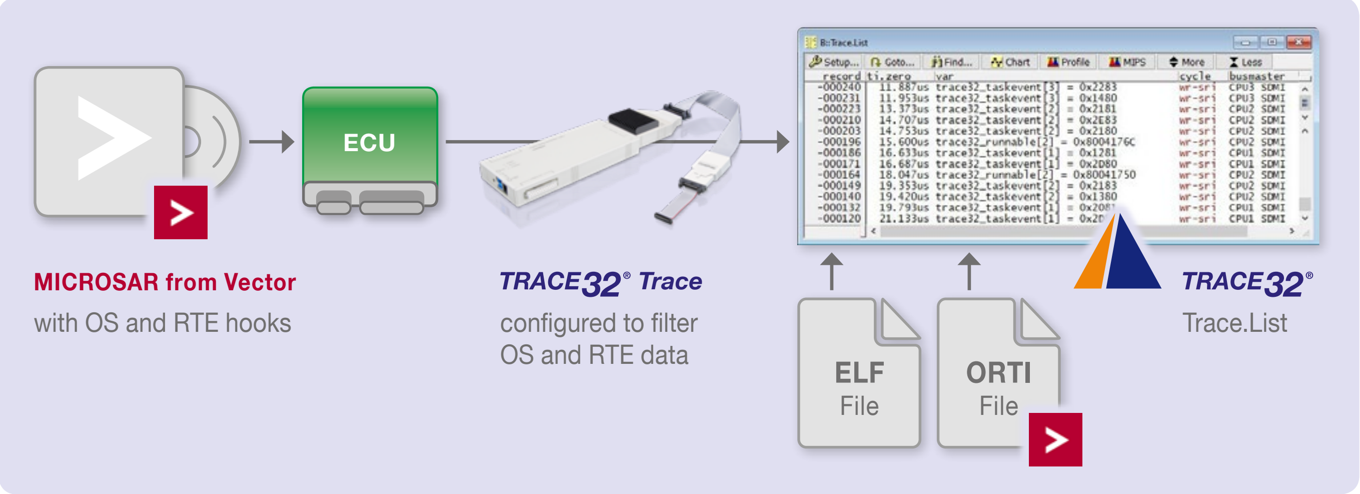
1) **MICROSAR from Vector**
   Configure trace hooks to generate trace data for OS and RTE events.

2) **TRACE32**
   a) Configure the MCDS trace and trigger logic to generate trace data for OS and RTE events only.
   b) Start and stop the program execution to record the trace data.
   c) Export the trace recording in a format readable by Vector's TA.Inspection.

3) **TA.Inspection from Vector's TA Tool Suite**
   Import TRACE32 trace measurements to analyze them in terms of response times, utilization, and other metrics *(see picture on the back)*.

The trace measurements can, alternatively, also be analyzed and displayed by TRACE32.

The MICROSAR trace hooks write the details of the OS and RTE events either to the Online Data Acquisition (OLDA) address range or to an address range within the Local Memory Unit (LMU). Each 32-bit write access includes the core identifier and the encoded event data.



```
 record ti.zero     var                                   cycle    busmaster
-000240  11.887us trace32_taskevent[3] = 0x2283           wr-sri   CPU3 SDMI
-000231  11.953us trace32_taskevent[3] = 0x1480           wr-sri   CPU3 SDMI
-000223  13.373us trace32_taskevent[2] = 0x2181           wr-sri   CPU2 SDMI
-000210  14.707us trace32_taskevent[2] = 0x2E83           wr-sri   CPU2 SDMI
-000203  14.753us trace32_taskevent[2] = 0x2180           wr-sri   CPU2 SDMI
-000196  15.600us trace32_runnable[2] = 0x8004176C        wr-sri   CPU2 SDMI
-000186  16.633us trace32_taskevent[1] = 0x1281           wr-sri   CPU1 SDMI
-000171  16.687us trace32_taskevent[1] = 0x2D80           wr-sri   CPU1 SDMI
-000164  18.047us trace32_runnable[2] = 0x80041750        wr-sri   CPU2 SDMI
-000149  19.353us trace32_taskevent[2] = 0x2183           wr-sri   CPU2 SDMI
-000140  19.420us trace32_taskevent[2] = 0x1380           wr-sri   CPU2 SDMI
-000132  19.793us trace32_taskevent[1] = 0x208            wr-sri   CPU1 SDMI
-000120  21.133us trace32_taskevent[1] = 0x2D             wr-sri   CPU1 SDMI
```

MICROSAR from Vector
with OS and RTE hooks

TRACE32® Trace
configured to filter
OS and RTE data

TRACE32®
Trace.List

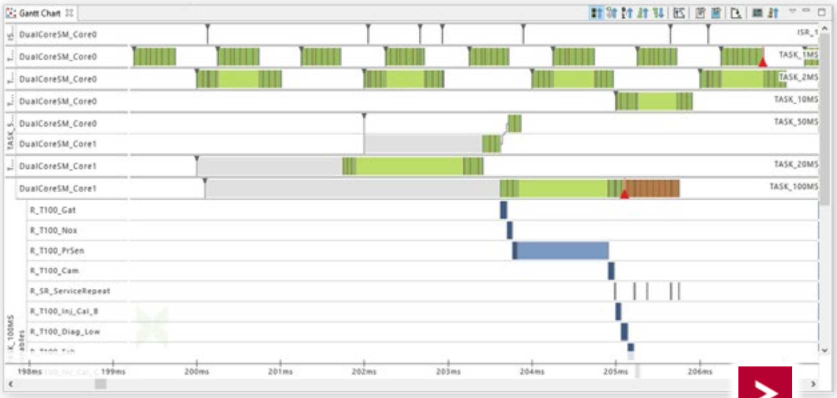
In order to advise MCDS, MCDSlight or miniMCDS to generate trace data only for these write accesses, the trigger logic has to be programmed accordingly. To ease this programming a ready-to-use trigger program is provided by Lauterbach. The table on the right shows which trace recording methods are supported by the AURIX™ families TC2xx and TC3xx.

	Onchip Trace	*DAP Streaming AGBT Trace*
MCDS	TC2xx, TC3xx	TC2xx, TC3xx
MCDSlight	TC3xx	TC3xx
miniMCDS	TC3xx	–

The trace data for the OS and RTE events only requires a medium bandwidth. TRACE32 can be configured so that the trace data is streamed into a file on the host computer at recording time when DAP streaming (TRACE32 CombiProbe 2) or AGBT tracing (TRACE32 PowerTrace Serial) is used. This can significantly extend the recording time.

Trace Evaluation in Vector's TA.Inspection

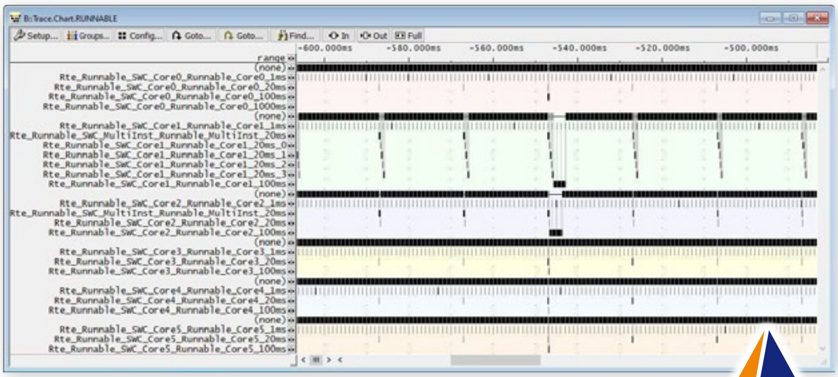
The option TA.Inspection from Vector's TA Tool Suite can be used to verify the timing behavior of the OS and RTE events against the system requirements. The command **Trace.EXPORT.TASKEVENTS <file>** decodes the TRACE32 trace recording with the help of the ELF and ORTI file and generates a csv file for the export to TA.Inspection. Vector will publish the description of this solution on their KnowledgeBase: **kb.vector.com**



TRACE32 Analysis Commands for OS Events and Runnables

Since the debut of trace capture tools, TRACE32 has always included commands to analyze the timing of tasks, task states and ISRs, only RTEs (Runnables) are new now. The following outputs are supported: numerical, timing chart and timing profile.

The screenshot on the right shows a Runnable time chart for all 6 CPUs of an Aurix™ TC397XA.


B::Trace.Chart.RUNNABLE
Setup...
Groups...
Config...
Goto...
Goto...
Find...
In
Out
Full
-600.000ms
-580.000ms
-560.000ms
-540.000ms
-520.000ms
-500.000ms
range
(none)
Rte_Runnable_SWC_Core0_Runnable_Core0_1ms
Rte_Runnable_SWC_Core0_Runnable_Core0_20ms
Rte_Runnable_SWC_Core0_Runnable_Core0_100ms
Rte_Runnable_SWC_Core0_Runnable_Core0_1000ms
(none)
Rte_Runnable_SWC_Core1_Runnable_Core1_1ms
Rte_Runnable_SWC_MultiInst_Runnable_MultiInst_20ms
Rte_Runnable_SWC_Core1_Runnable_Core1_20ms_0
Rte_Runnable_SWC_Core1_Runnable_Core1_20ms_1
Rte_Runnable_SWC_Core1_Runnable_Core1_20ms_2
Rte_Runnable_SWC_Core1_Runnable_Core1_20ms_3
Rte_Runnable_SWC_Core1_Runnable_Core1_100ms
(none)
Rte_Runnable_SWC_Core2_Runnable_Core2_1ms
Rte_Runnable_SWC_MultiInst_Runnable_MultiInst_20ms
Rte_Runnable_SWC_Core2_Runnable_Core2_20ms
Rte_Runnable_SWC_Core2_Runnable_Core2_100ms
(none)
Rte_Runnable_SWC_Core3_Runnable_Core3_1ms
Rte_Runnable_SWC_Core3_Runnable_Core3_20ms
Rte_Runnable_SWC_Core3_Runnable_Core3_100ms
(none)
Rte_Runnable_SWC_Core4_Runnable_Core4_1ms
Rte_Runnable_SWC_Core4_Runnable_Core4_20ms
Rte_Runnable_SWC_Core4_Runnable_Core4_100ms
(none)
Rte_Runnable_SWC_Core5_Runnable_Core5_1ms
Rte_Runnable_SWC_Core5_Runnable_Core5_20ms
Rte_Runnable_SWC_Core5_Runnable_Core5_100ms

Execution Details

If a task waits too long or an interrupt takes too much time, the MCDS trigger unit can be reset and reprogrammed in TRACE32 to provide program execution details for the critical code section on the affected CPU. This makes it easy to identify critical runtime issues.

Lauterbach will provide a support package for this solution on: **www.lauterbach.com/scripts.html**