

Lauterbach Debugger for SAFERTOS® : TRACE32®

The TRACE32® System includes a configurable RTOS debugger to provide debugging in real time operating systems. The package contains a ready to run configuration for SAFERTOS®.

Functional Overview

The SAFERTOS awareness provides easy access to task lists and other kernel information, generates performance analysis data and enables effective debugging with the TRACE32 systems on targets running SAFERTOS. The functions available to the developer include:

- A display of all SAFERTOS System Resources;
- Task Related Breakpoints;
- Task Stack Coverage;
- Dynamic Thread Performance Measurement;
- SAFERTOS Specific Display of Trace Listing;
- Statistic Evaluation and Graphic Display of Task Run Times;
- Statistic Evaluation and Graphic Display of Function Run Times;
- PRACTICE Functions for OS Data;
- SAFERTOS related Pull-Down Menu.

SAFERTOS Awareness on TRACE32:

The TRACE32 provides display functions to show the system resources of SAFERTOS. All objects, such as Tasks and Queues can be displayed and inspected. The processor is capable of reading the target memory while the application is running, or by using the emulator's dual port memory (if available). The display of all these regions can be viewed non-intrusively in real time. The tables are updated permanently, without affecting the application at all.

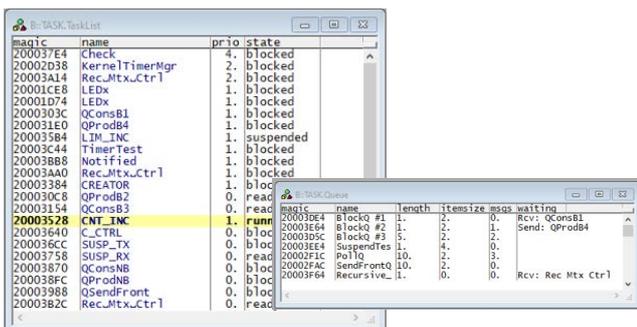


Figure Task list window and detailed queue window

SAFERTOS Specific Features

- Display of all SAFERTOS System Resources
- SAFERTOS Specific Display of Trace Listing
- SAFERTOS related Pull-Down Menu

Task Related Breakpoints

Any breakpoint set in the TRACE32 debugger can be set dependent on a specific task. This means, that the breakpoint will then only halt if hit by the specified task. This allows task related debugging even in shared code.

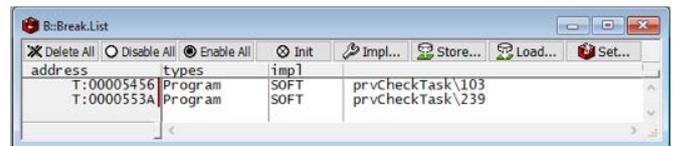


Figure Breakpoint list with task dependent breakpoints

Task Stack Coverage

In real time systems it is quite important to know how much stack space each Task consumes. For this purpose a special window shows the current and the maximum usage of the stack of each separate Task.

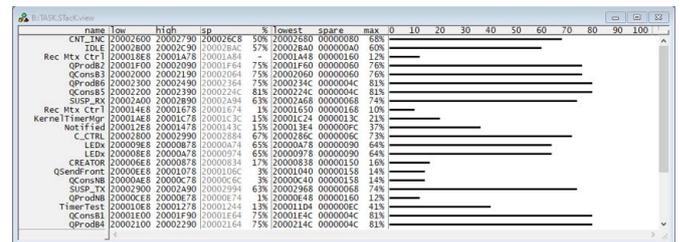


Figure Task stack coverage window

Dynamic Task Performance Measurement

While SAFERTOS is running, TRACE32 is capable of repeatedly evaluating the current running task and collecting the results. The percentage each task consumes of the system is then displayed graphically and updated permanently. The longer the measurement is run, the more exact the results. This feature is available without any hardware trace.

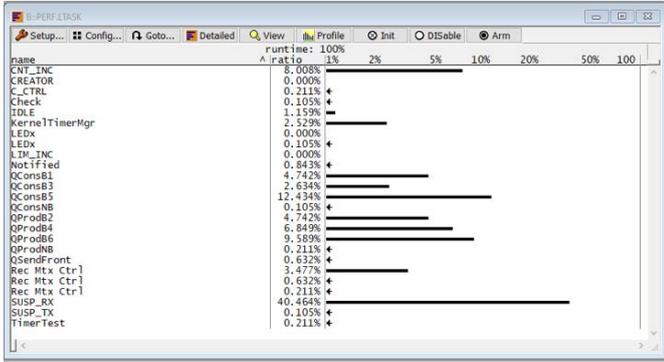


Figure Performance measurement list

Task Related Evaluation of Function Run Times

The statistical and graphical evaluation of function calls and function run times (if available) can be done for the currently running task. This is necessary if different tasks call one single function at the same time, or if a Task Switch occurs in between the function.

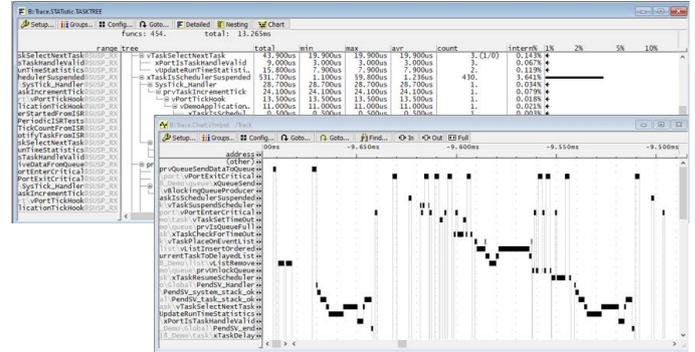


Figure Statistics on function hierarchy, task related (above)
Flow of functions, related to the calling tasks (below)

SAFERTOS Specific Display of Trace Listing

The data recorded in the real time trace buffer (if available) can be displayed and interpreted in relation to the operating system. The task switches are displayed graphically, either exclusively or in addition to other information.

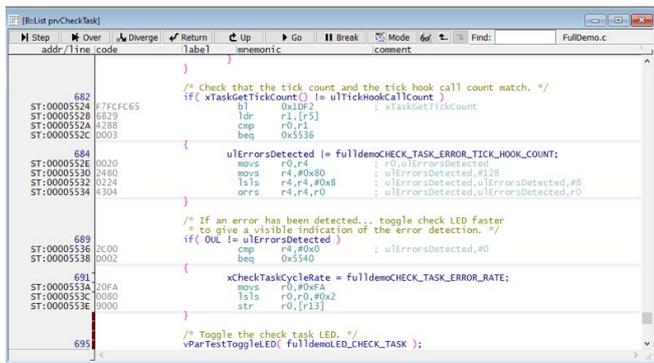


Figure Trace listing, task switch, memory accesses, assembler and HLL information

PRACTICE Functions for OS Data

The support includes extended PRACTICE functions for SAFERTOS specific data, to be used in setup and test scripts. E.g. the function "task.config(magic)" returns the address of the so called magic value, which corresponds to the id of the current executing task.

SAFERTOS Related Pull-Down Menu

The menu bar of the TRACE32 user interface can be fully customized, allowing the developer to create a new pull down menu, including operating system specific commands. This provides fast access to the SAFERTOS awareness features.

Statistic Evaluation and Graphic Display of Task RunTimes

Out of the real time trace buffer (if available), the analyzer can calculate statistic tables of task run times and task switches. A diagram shows which task was active at a specific time, giving a clear view of the behaviour of the system.

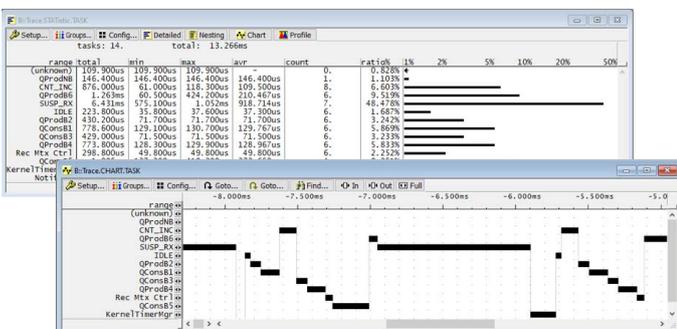


Figure Statistics and flow of tasks