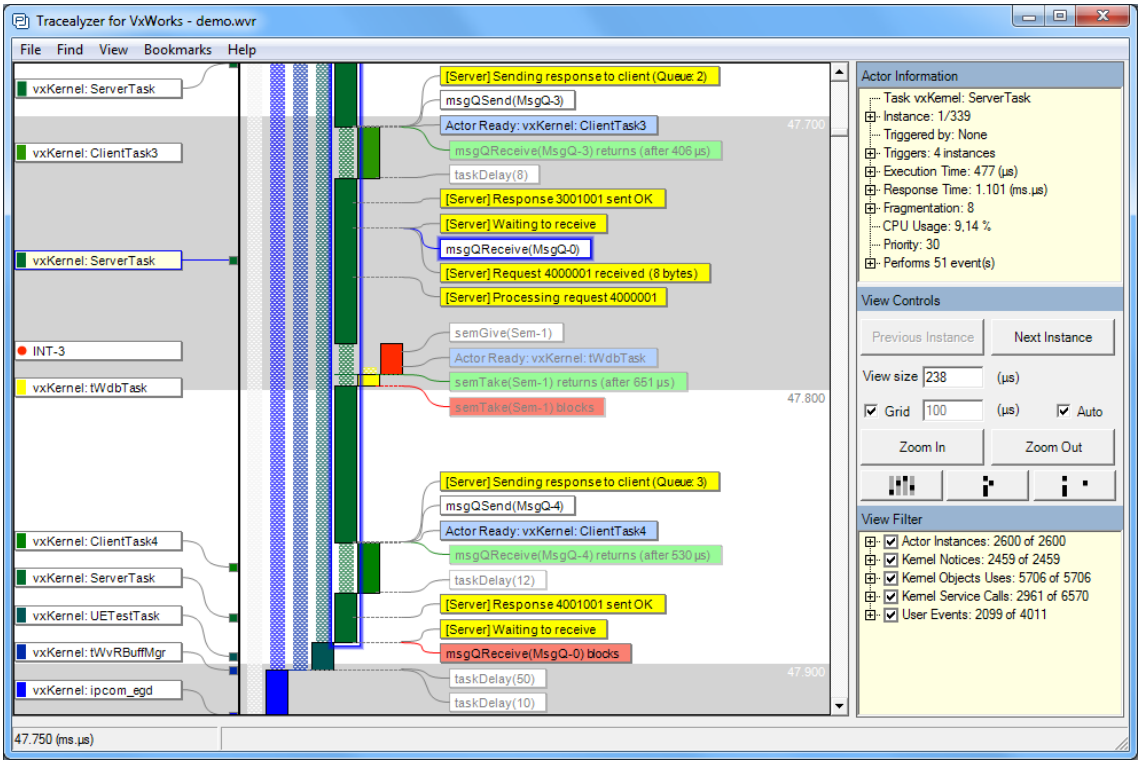
Percepio Tracealyzer™

Tracealyzer provides an unprecedented level of insight into the run-time world of your RTOS or Linux-based software system. Tracealyzer allows you to solve complex software problems in a fraction of the time otherwise needed, develop more robust designs to prevent future problems and find new ways of improving your software performance.

When developing advanced multi-threaded software systems, a traditional debugger is often insufficient for understanding the behavior of the integrated system, especially regarding timing issues. **Tracealyzer** visualizes the run-time behavior through more than 20 innovative views that complement the debugger perspective. The Tracealyzer views are interconnected in intuitive ways which makes them very powerful and easy to navigate.



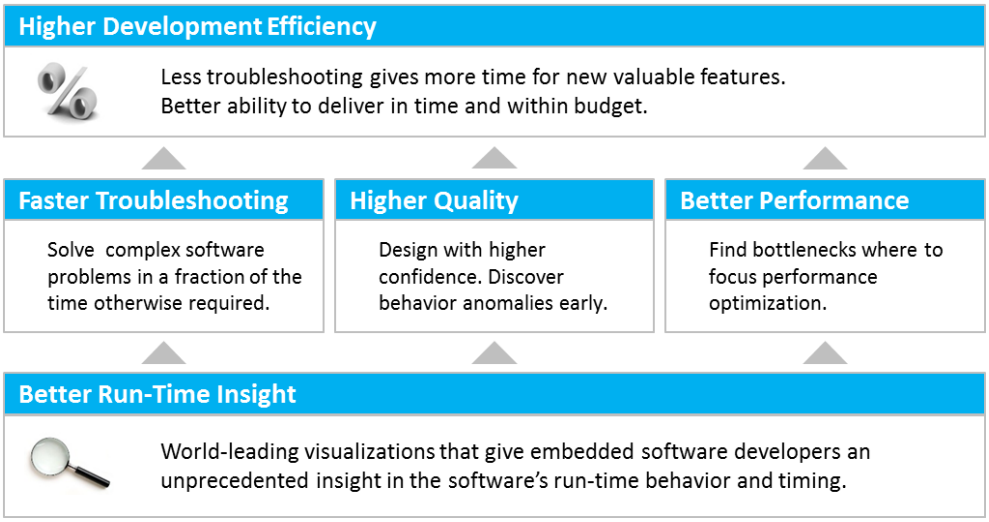
The Tracealyzer tools runs on Windows or Linux PCs and are available for target systems running Linux, FreeRTOS, Wittenstein OpenRTOS and SafeRTOS, Wind River VxWorks, Micrium µC/OS-III and SEGGER embOS.

*"FreeRTOS+Trace has **doubled our development speed.** Problems that otherwise would take days to solve are obvious with this tool and just a quick fix. We use it all the time."*
Alex Pabouctisids, Lead Firmware Engineer, Flyability.

Percepio AB
Köpmangatan 1A
72215 Västerås
SWEDEN

www.percepio.com

Since Tracealyzer does not require special debug hardware, it can be used both as a lab tool and during field use. Some customers even keep the recording active in their release builds using a "flight recorder" configuration, which gives them valuable "post mortem" diagnostics on real-world issues.



Faster Troubleshooting: Tracealyzer allows for capturing rare, sporadic errors which otherwise can be very hard to reproduce and analyze. Many problems can be solved in a fraction of the time otherwise required.

Higher Quality: Tracealyzer is not only a "fire extinguisher" to use on specific hard problems. Discover and avoid potential future problems, such as blocking system calls that are close to a timeout. When designing new features, you can avoid unsuitable designs that could cause problems related to timing, CPU usage, scheduling or other task interactions.

Better Performance: Find new ways of improving the software performance. There might be "hot-spots" in your software's behavior where small changes in timing may give substantial performance improvements. Tracealyzer provides several ways of finding such hot-spots. Get a more responsive software system, or fit more software functionality into the same hardware platform.

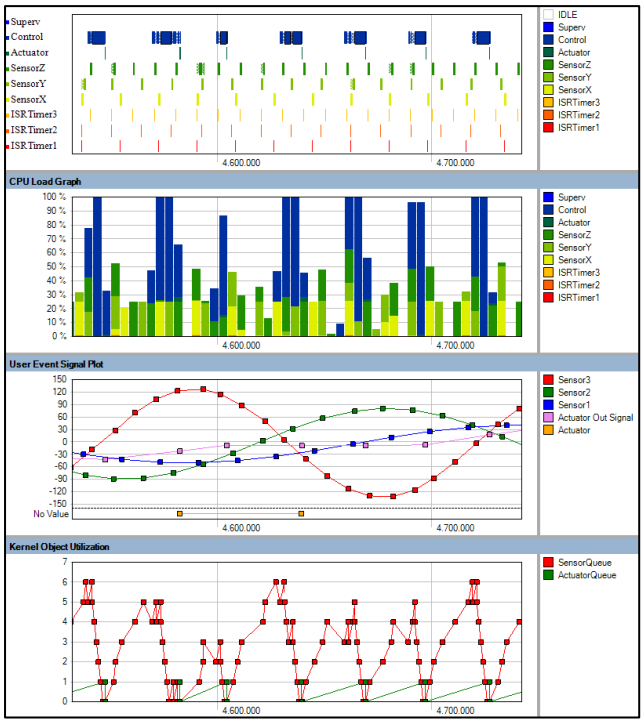
Control system tuning: Control system developers can benefit from the support for plotting custom application data. Plot your inputs and outputs, and correlated with the task scheduling to better understand how your software timing affects control performance.

Other benefits: Tracealyzer can help getting new developers productive faster and allow you to evaluate the performance of third-party software, such as embedded databases, touch screen drivers or communication stacks. And since we support several common operating systems for embedded software, you can probably keep the Tracealyzer support even if changing operating system.

Tracealyzer allows developers to spend less time troubleshooting and more time on creating valuable software features. **Deliver quality software *on time* and *within budget*!**

Percepio AB
Köpmangatan 1A
72215 Västerås
SWEDEN
www.percepio.com

Tracealyzer provides **over 20 graphical views** of the runtime behavior of your application, an arsenal of perspectives that allows you to quickly find relevant parts of the trace. The main trace view shown on the first page uses a vertical time-line where events like kernel calls are shown using text labels.



There are several supporting views with horizontal time-lines, which can be shown in a common window with synchronized scrolling. The screenshot on the left shows an example with four synchronized views:

Horizontal Trace View: Shows the scheduling on a horizontal time-line to facilitate correlation with other views.

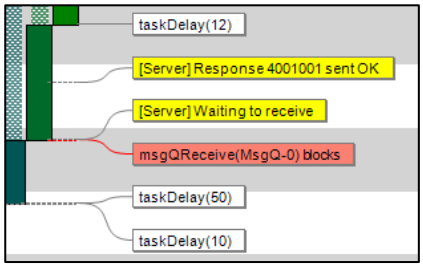
CPU Load Graph: Shows the amount of CPU time used by each task and interrupt handler, and the total CPU usage.

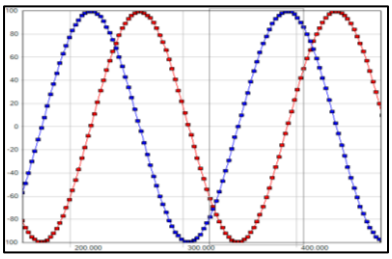
User Event Signal Plot: Shows a plot of application data logged in *User Events* (read more below).

Kernel Object Utilization: Shows the utilization of buffered kernel objects, such as message queues.

By double-clicking on a data point or interval, you focus the main trace view on the selected point in time.

Tracealyzer supports **User Events**, allowing you to log any event or data in your application. They appear as yellow labels in the trace view. User Events can be used as an alternative (or complement) to classic debug "printf" calls and for plotting of user data. Since User Events are stored very quickly, they can be used also in time-critical code. And since you get the User Events into the Tracealyzer views, it is easy to correlate these with overall system behavior.





The **User Event Signal Plot** view allows you to plot any data logged as User Events. This is highly useful for analysis of control loops and other time-dependent algorithms as it allows you to correlate the data points with the other Tracealyzer views to find the cause behind anomalies in the plot. For instance, incorrect scheduling priorities or CPU overload might have delayed the computation. Double-click on a data point to find the location in the trace where the data was logged.

Tracealyzer performs **advanced analyses** of the trace data to facilitate analysis of complex behavior. For instance, when selecting a specific "msgQReceive" call, the corresponding a

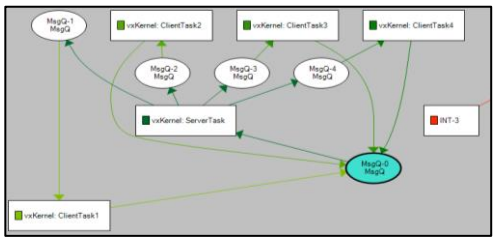
Percepio AB
Köpmangatan 1A
72215 Västerås
SWEDEN

www.percepio.com

"msgQSend" call can be highlighted. This allows you to follow the data flow between tasks and analyze chains of related tasks.

47.242	vxKernel: ClientT	msgQSend		Sent post #1	1	1			
47.269	vxKernel: ClientT	msgQSend		Sent post #2	2	1	2		
47.297	vxKernel: ClientT	msgQSend		Sent post #3	3	1	2	3	
47.325	vxKernel: ClientT	msgQSend		Sent post #4	4	1	2	3	4
47.325	vxKernel: Server	msgQReceive		Received post #1	3	1	2	3	4
47.453	vxKernel: Server	msgQReceive		Received post #2	2	2	3	4	
47.601	vxKernel: Server	msgQReceive		Received post #3	1	3	4		
47.740	vxKernel: Server	msgQReceive		Received post #4	0	4			
47.892	vxKernel: Server	msgQReceive	2.731	Trying to receive...	0	Empty			
50.622	vxKernel: ClientT	msgQSend		Sent post #5	1	5			

The **Communication Flow** graph visualizes dependencies with respect to communication and synchronization between tasks, interrupt handlers, and other kernel objects such as semaphores and message queues. This gives a visual feedback of the runtime architecture with respect the executed code, providing the "big picture" and allows you to quickly spot any unexpected dependencies.



Download Tracealyzer today and start exploring the features directly. Tracealyzer comes with a 30-day evaluation period (fully functional) and a pre-recorded demo trace. For further information, visit **www.percepio.com**.

"The many system views of the Tracealyzer from Percepio made it easy to quickly identify issues in our system that we have not noticed using (Wind River) System Viewer. The visualization has several advantages over the System Viewer and makes it much easier to understand the system behavior."
Johan Fredriksson, Software Architect, SAAB AB.

"ABB Robotics is using the first generation Tracealyzer in all of the IRC5 robot controllers shipped since 2005. The tool has proven its value many times in all corners of the world."
Roger Kulläng, Global System Architect, ABB Robotics.

"In today's tough competition with time-to-market pressure constantly increasing, visualization support is natural for software developers in order to produce software of higher quality, in shorter time and at a lower cost. We choose Tracealyzer from Percepio."
Jörgen Appelgren, R&D Manager, Atlas Copco Rock Drills.

Percepio AB
Köpmangatan 1A
72215 Västerås
SWEDEN
www.percepio.com