

Percepio Tracealyzer®

Tracealyzer provides Visual Trace Diagnostics for an unprecedented level of insight into the runtime world of your embedded 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 full software system. Tracealyzer visualizes the runtime behavior through more than 30 innovative views that complement the debugger perspective. The views are interconnected in intuitive ways which makes them very powerful and easy to navigate.



Tracealyzer runs on Windows and Linux hosts and is available for target systems running Linux, FreeRTOS, Zephyr, Arm Keil RTX5, Micrium µC/OS-III, Wittenstein OpenRTOS and SafeRTOS, Azure RTOS ThreadX, Wind River VxWorks, and On Time RTOS-32.

LISTEN TO OUR PARTNERS

“The combined best-in-class visualization assets of Percepio’s Tracealyzer and our industry leading ThreadX RTOS will greatly help embedded developers confidently get their IoT products to market quicker.”

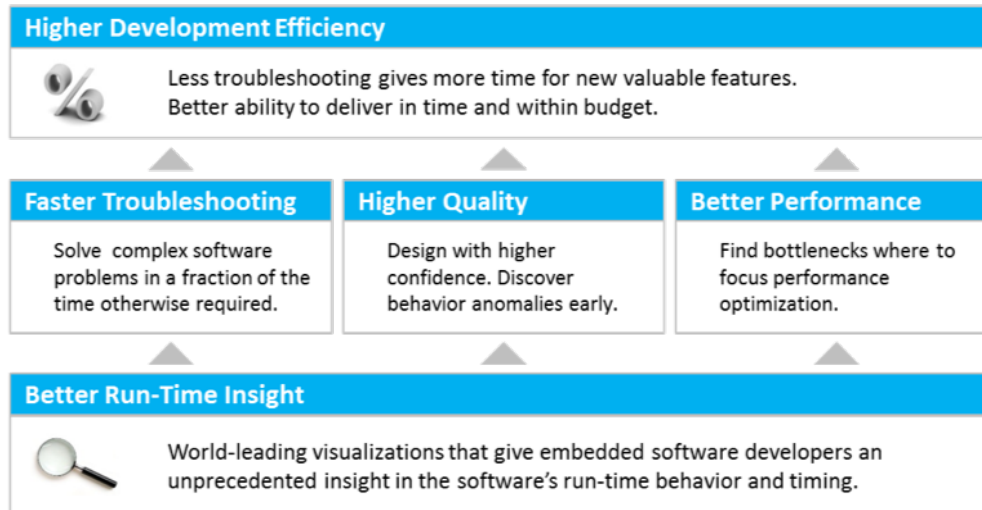
*William E. Lamie
CEO, Express Logic*

“Tracealyzer has always offered exceptional value to our global community of users by giving engineers direct insight into how their applications are executing. This information is invaluable when both optimizing and debugging FreeRTOS applications.”

*Richard Barry
Founder of FreeRTOS*

Tracealyzer can work in both streaming and snapshot mode. **Streaming mode** allows for recording long sessions by transferring the data continuously to the host system, e.g. using a supported debug probe or a network connection. Streaming traces can span hours, days or even weeks.

In **snapshot mode**, trace data is stored in a RAM buffer in the target system, from which a trace snapshot can be uploaded and viewed at any time. Snapshot traces are limited by the amount of RAM available for the trace buffer but can be used on essentially any system. In snapshot mode, Tracealyzer does not require special debug hardware and can therefore be used in deployed operation. Some customers even keep the recording active in their release builds, which gives them valuable post-mortem diagnostics.



Faster Troubleshooting: Tracealyzer allows the developer to capture rare, sporadic errors which 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 problems. Discover and avoid potential future problems, such as inefficient design or 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, memory allocation, scheduling, or other task interactions.

Better Performance: Find new ways to improve software performance. There might be hot-spots in your software's behavior where small changes may give substantial performance improvements.

Control System Tuning: Control system developers can benefit from the support for plotting custom application data. Plot your inputs, outputs, and interval data 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 drivers or communication stacks. And since we support several common operating systems for embedded software, you can probably keep using Tracealyzer even when changing operating system.

LISTEN TO OUR PARTNERS

“What Percepio has achieved with Tracealyzer is to remove the guesswork of designing an RTOS-based system. Its' ability to record and then analyze what's happening gives developers a view into their system that would be impossible to have otherwise.”

Jean Labrosse
CEO Micrium

“Percepio Tracealyzer enables unparalleled insight into the execution, timing and interactions of the RTX5 real-time operating system, the MDK middleware components and the user application.”

Reinhard Keil
Senior Director of Developer Tools, Arm

Tracealyzer provides **more than 30 graphical views** of the runtime behavior of your application, an arsenal of perspectives that allows you to quickly locate 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.

Views may be synchronized with respect to scrolling, zooming and filters, and it is possible to customize this in many ways.



The screenshot on the left shows an example with four stacked synchronized views. Note the blue selection of 50 ms shown in all views. The horizontal trace view is zoomed in to 200 ms, while the other views show 10 second overviews.

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

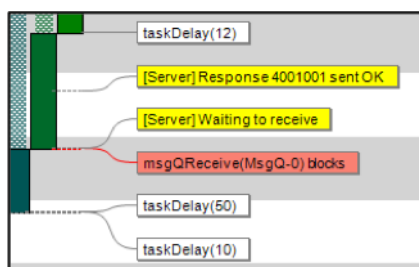
CPU Load Graph shows the amount of CPU time used by each task and interrupt handler, and the total CPU usage.

User Event Signal Plot shows a plot of application data logged in User Events (see below for more information about these events).

Interval Timeline shows user-defined intervals in the trace, i.e., the time between two specified events, such as logged state variables in the target system.

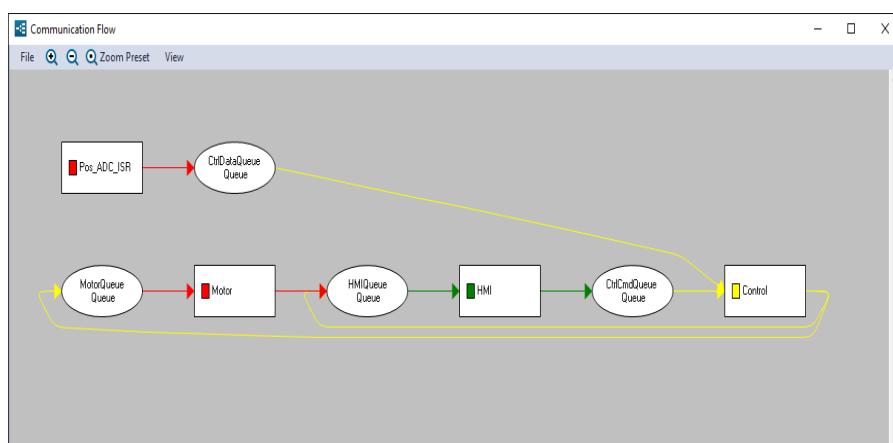
Live Visualization* lets you view a trace live while recording. You can even pause individual views to zoom in and inspect details while recording continues in the background.

Unlimited Tracing* lets you monitor your application over long test runs, spanning hours or days, and see analysis results, such as task execution times, immediately. Once recording has finished, you can watch a preview of the trace and select a smaller section to load for a full analysis.



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

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 you an overview of the runtime architecture and allows you to quickly spot any unexpected dependencies.

Custom Metrics: Define custom intervals that highlight the time between selected software events. Moreover, with user-defined state machines you can visualize state information logged in the trace, from software state variables or even hardware states.

*) Depending on target operating system.

WHAT OUR CUSTOMERS ARE SAYING

- “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 chose Tracealyzer from Percepio.”
Jörgen Appelgren, R&D Manager, Atlas Copco Rock Drills
- “Tracealyzer 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

Tracealyzer performs advanced analysis of the trace data to present a comprehensive view of software behavior. For instance, when selecting a specific “msgQReceive” call, the corresponding “msgQSend” call can be highlighted. This allows you to follow the data flow between tasks and analyze chains of related tasks.

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

Download Tracealyzer today and start exploring the features immediately. Tracealyzer comes with a time-limited, fully functional evaluation license and a pre-recorded demo trace. For further information, visit percepio.com.



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