Percepio Tracealyzer v4.8.1.13174 for Win x64

2025-02-14 23:38:36





Percepio Tracealyzer is the best solution for visual hint diagnostics, providing embedded applications programmers fantastic insight into their runtime systems. This allows for easier debugging of system-level problems, discovering application design defects, and quantifying the application's time and resource utilization. Make sure your code is dependable, responsive, and efficient. Otherwise, find out why.

Percepio Tracealyzer provides state-of-the-art tracing and follows visualization applications developed since 2004. The visualization created traces that are simpler to review and comprehend. This makes it much easier to identify and understand bugs, find answers, and confirm them. Attempt Tracealyzer now and boost your development.

Percepio Tracealyzer may be utilized side-by-side using conventional debuggers, like IAR, Keil, or even Eclipse tools, and matches the in-depth debugger perspective with different additional viewpoints on the platform level, perfect for comprehending real-time issues in which a timeless debugger isn't adequate. For example, in real-time systems like motor control, it's frequently not feasible to halt the method for debugging. You may record your embedded applications and other interest factors in real time as the system executes by employing program tracing.

Tracealyzer supplies over 30 perspectives of the runtime behavior, such as job scheduling and interrupts handler/ISR implementation, task time (implementation time and other metrics), action priority, CPU load, and memory utilization, in addition to interactions between activities and disrupt handlers/ISRs through, e.g., message queue, semaphore, and mutex objects. In this manner, you can see if your job priorities, implementation times, and timeouts are appropriate to prevent timeout errors from job preemption or obstruction. For instance, maybe the USB job has to be permitted to preempt the TCP/IP job to decrease its response time and prevent a semaphore timeout. Maybe some mutex is utilized by several tasks and thus often blocks the implementation and leads to delays?

Percepio Tracealyzer also allows custom logging of"User Occasions" from the application code to find countries and factors with time, parallel with all the other viewpoints. This may be utilized to test and debug the real-time behavior of your calculations, such as engine control loops, the application's logic, and performance. For example, just how much implementation time is necessary? Can I conduct my control loop at a more incredible speed? Is the job timing stable?

The profiling views permit you to determine precisely what regions of the system utilize most funds (e.g., chip time and heap

所有软件资源 100%安全 已检测 无病毒 无篡改
去预览

标签

Applications Tools

memory). Maybe some parts utilize more CPU cycles than are moved due to inefficient code. The memory utilization views also let you find potential memory leaks. You can filter out all"malloc" events that have fitting"free" events with the"Show staying only" alternative, revealing all events that might be a memory leak.

All viewpoints are interconnected intelligently, meaning you can click a data point in 1 perspective and see the corresponding place in another related perspective. Moreover, all timeline viewpoints could be synchronized, meaning you may scroll and scroll them side-by-side. In this manner, you may benefit from several viewpoints at precisely the same time and better comprehend the trace.

The trace may be streamed in real-time, either with a supported debug probe or a custom target-host port, like a USB or TCP/IP. Thus far, this is encouraged for FreeRTOS, Micrium, and SAFERTOS, but an added streaming service is intended.

No exceptional trace hardware is necessary, as the is based on practical application instrumentation with minimal overhead. In this manner, it functions on any chip or microcontroller, presuming you have a couple of kilobytes in your trace recorder module. It's designed for 32-bit target chips and upward, such as ARM Cortex-M MCUs, for example, STMicroelectronics STM32, NXP LPC, Kinetic Collection, Renesas Synergy, Silicon labs EFM32, Cypress PSoC, Atmel SAM, TI MSP432, TI TMC4, and Infineon XMC4. We also encourage Renesas RX, Renesas RZ, Microchip PIC32, Atmel AVR32, ARM Cortex-R, ARM Cortex-A, Xilinx Zynq, Xilinx Microblaze, Altera Nios II, along with Synopsys ARC chips. And even if your chip family isn't yet wholly supported, it is easy to create your interface. You must specify a few macros, chiefly for studying the present moment.

Key Features of **Percepio Tracealyzer**:

1. **Faster Troubleshooting:** Tracealyzer permits capturing infrequent, intermittent mistakes, which can be extremely tough to replicate and examine. Many embedded application issues can be solved at a fraction of the time required. Tracealyzer recording may be utilized in parallel with existing debuggers from the laboratory or deployed in area systems to catch rare, sporadic mistakes.
2. **Better Software Quality:** Tracealyzer isn't merely a"fire extinguisher" for particularly complex issues. You might even find and prevent possible future problems, like blocking system calls that are near some timeout. Additionally, Tracealyzer provides you with a better knowledge of the present system runtime dynamics; therefore, when designing new attributes, you can avoid unsuitable designs that might result in issues associated with time, CPU utilization, scheduling, or other job interactions.
3. **Better Performance:** The enhanced insight also lets you discover new methods of enhancing the functionality of the embedded application. There could be"low-hanging fruits," in which little changes in activity timing may provide significant performance improvements. Tracealyzer provides several methods of discovering"hot spots" where jobs are delayed by more excellent prioritized jobs that may run somewhat sooner or later to decrease resource conflicts. Without proper visualization, such optimizations are extremely tough to discover. In this manner, you can find a more reactive software program or match more application functionality in precisely the same hardware platform.
4. **Control program tuning:** Control system programmers may benefit from plotting custom program information, which we call user occasions. Plots of, e.g. control signs (sensor inputs and actuator outputs) may be connected with the application's timing to comprehend anomalies in the management functionality better. Additionally, plots of job time (e.g., periodicity) may be utilized to examine software timing variants that impact control functionality.
5. **Other attributes:** The enhanced penetration level also allows for the quicker introduction of new programmers and improved chances to assess the operation of new third-party applications, including embedded databases, touch display drivers, or communicating stacks. Ultimately, because we encourage many standard operating systems for embedded applications, you can often maintain the Tracealyzer support if switching to a different operating system.

版本列表

Percepio Tracealyzer v4.4.2.12977 for Win x64

Percepio Tracealyzer v4.5.0.13035 for Win x64

Percepio Tracealyzer v4.8.1.13174 for Win x64

苏ICP备19032038号 Powered by getShopSys