LYNX
SOFTWARE TECHNOLOGIES

LYNX MOSA.ic.Spyker-TZ™

In safety-critical and real-time applications like those in Aviation & Defense (A&D), timing-related bugs are not just a technical challenge—they're a cost multiplier, responsible for 20-30% of debugging efforts. When these issues are discovered late—especially during integration testing—development costs can skyrocket by 100x. Worse, late-stage detection threatens project deadlines, delays certification, and increases risk to mission success. The key to avoiding this is early detection and resolution.

By leveraging detailed trace analysis, SpyKer-TZ can reduce your certification efforts by up to 30%, giving your teams clearer insights into real-time system behavior. With up to 40% faster debugging times, development cycles are shortened dramatically, allowing teams to resolve timing issues with speed and precision. This means fewer delays, fewer certification bottlenecks, and greater confidence in meeting tight project deadlines—without sacrificing quality. (VDC Research, 2024), (Frost & Sullivan, 2024), (MarketsandMarkets, 2024)

SpyKer-TZ helps you debug faster, optimize more intelligently, and eliminate the most challenging bottlenecks in complex real-time systems.

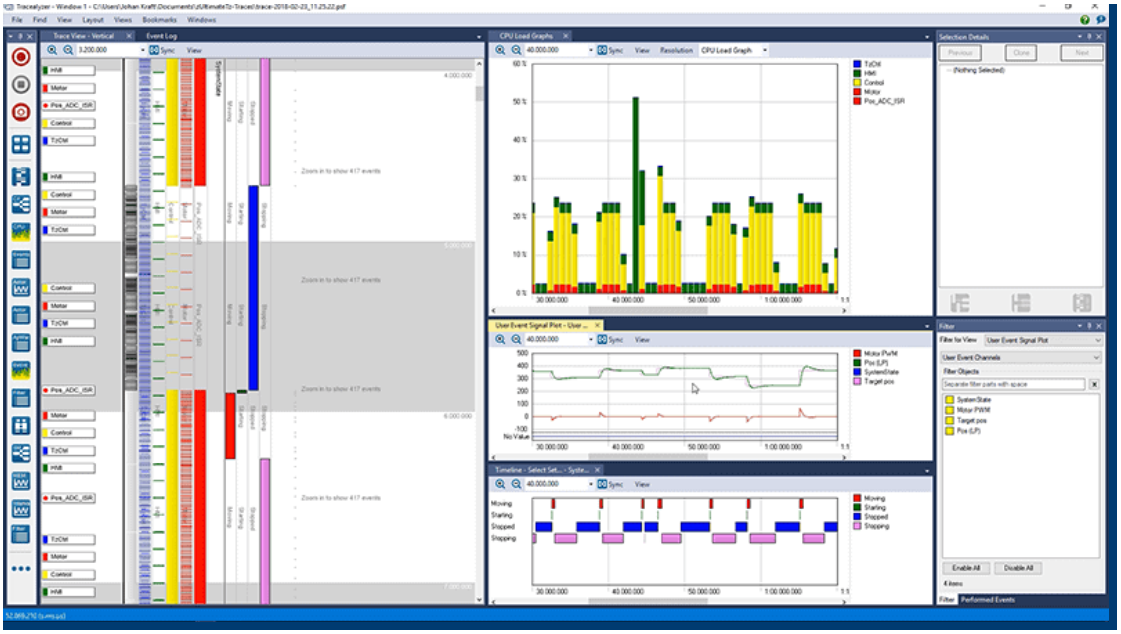
With live diagnostics and dynamic trace analysis, you gain unmatched visibility into software performance—uncovering hidden timing issues that can derail projects. With over 30 advanced visualizations, SpyKer-TZ gives you the tools to drive peak system efficiency and meet tight project deadlines with confidence.

Introducing Lynx SpyKer-TZ Powered by Tracealyzer from Percepio

SpyKer-TZ™ – the first dynamically instrumented system trace analyzer, designed to give developers complete visibility into system behavior over time. By automatically instrumenting the kernel at runtime—without the need for rebooting—SpyKer-TZ empowers teams to track down elusive bugs, optimize performance, and resolve complex timing issues across multithreaded, multi-OS, and bare metal environments. With powerful visualizations like trace views,

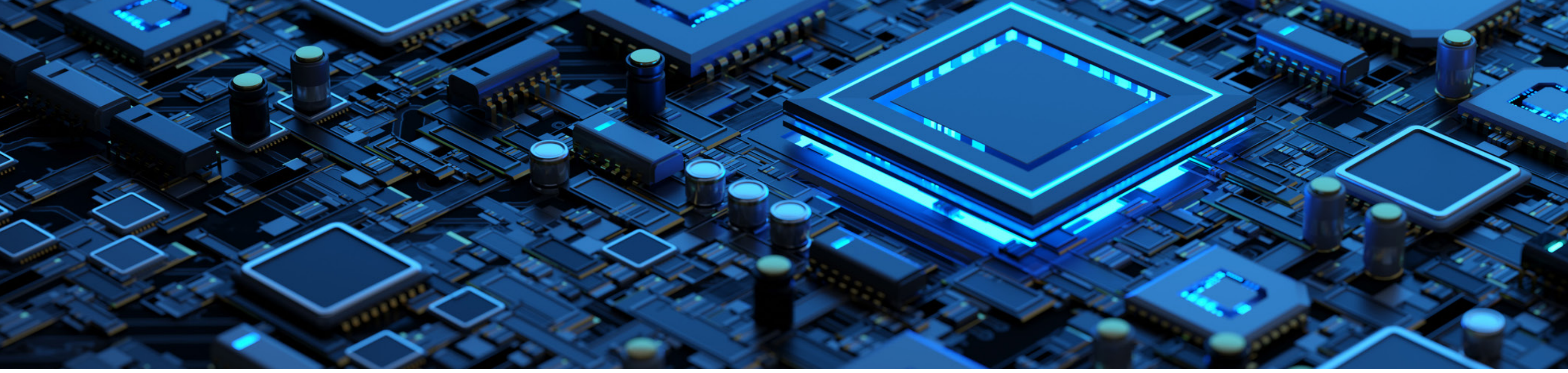
CPU load graphs, and event logs, it streamlines debugging, allowing for the quick identification and resolution of anomalies like CPU load spikes, all while reducing debugging time by up to 40%. Whether you're working on LynxOS-178 or LynxSecure, SpyKer-TZ accelerates your workflow and improves system reliability by eliminating bottlenecks before they derail your project.

Key Features and Capabilities



SpyKer-TZ

- Dynamically instrumented trace analysis with low overhead
- Auto-instrumentation capabilities for LynxOS-178 and LynxSecure
- Powerful front-end visualization tools for straightforward data interpretation
- Fast, non-intrusive event capture with a focus on minimizing development costs
- Remote operation and target system restoration to its original state after analysis
- Low Overhead Trace Patch: SpyKer-TZ's trace patch imposes minimal overhead, ensuring objective measurements and eliminating issues with timing invasiveness, which allows effective tracking of even the most elusive bugs
- Detailed Event Visualization: The front-end GUI, written in Java, supports multiple concurrent visualizations, pop-up event details, and event filtering. Zoom and jump features allow precise exploration of data
- Local and Remote Operation: The front end can run locally or remotely via TCP/IP, making SpyKer-TZ ideal for real-world testing scenarios
- Target System Restoration: After data collection, the target system is automatically restored to its original state, minimizing impact
- Visual trace diagnostics with more than 30 insightful views
- Compatibility with multiple operating systems, including Lynx Operating System and Linux variants
- Both streaming and snapshot modes allow flexible data capture for trace
- Advanced analysis tools, including CPU Load Graph, User Event Signal Plot, and Communication Flow Graph
- Unique insights into multi-threaded software behavior for faster troubleshooting and performance improvements



Value Proposition

Benefit to User	Value Metrics	Feature
Instantly visualizes real-time program behavior without slowing down development.	Reduces debugging time by up to 40%	Dynamic Trace Analysis
Provides immediate insights into application performance and timing issues.	Shortens time-to-market by 3-6 months	Live Diagnostics
Simplifies setup without needing a specially instrumented kernel.	Cuts setup and debugging costs by up to 50%	Auto-Instrumentation
Powerful graphical views to analyze system behavior and bottlenecks.	Speeds up root cause analysis up to 5 times faster	Over 30 Visualization Tools
Flexibility to trace long sessions or quick snapshots for detailed analysis.	Enhances system reliability with 60% fewer defects	Streaming and Snapshot Modes
Detects and resolves elusive bugs and timing issues effectively.	Reduces certification efforts by up to 30%	Real-Time Insights
Identifies CPU load, task scheduling, and performance hotspots.	Optimizes software performance, leading to up to 25% cost savings	System Bottleneck Analysis

Use Cases and Benefits

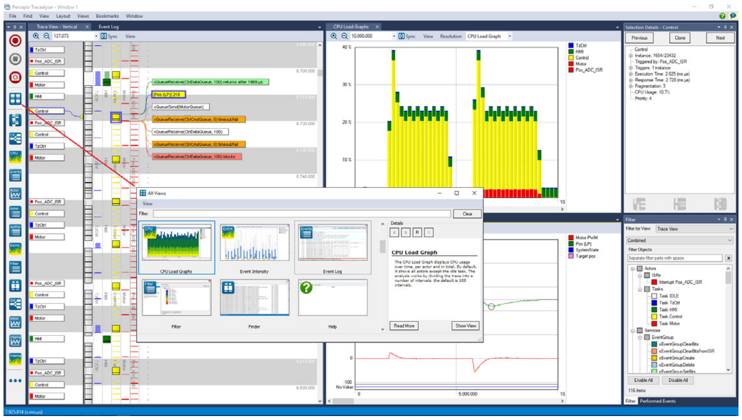
- Reduced Development Costs - Automated data collection and analysis, reducing manual debugging and lowering costs
- Faster Troubleshooting and Debugging - Identifies performance bottlenecks and bugs quickly, speeding up fixes
- Improved Software Performance - Pinpoints hotspots and areas of improvement to optimize system efficiency
- Enhanced Product Quality - Ensure robustness and reliability through detailed software diagnostics
- Support for Certification and Compliance - Facilitates system analysis to meet industry standards and certifications

Technical Specifications

- Spyker-TZ supports LynxOS-178, LynxSecure, and multiple Linux variants
- Spyker-TZ offers a variety of detailed views for examining the system's collected data, enabling deeper insights and analysis

Trace Main View

The Trace Main View in Spyker-TZ provides an intuitive, high-level visualization of your system's behavior, offering real-time insights into application performance at every stage of development. This tool goes beyond debugging—it allows you to spot performance issues early, even in complex third-party or "black box" code, and gain a deeper understanding of execution flows without needing access to source code. Whether you're troubleshooting or reverse-engineering software stacks, the Trace Main View helps you make informed decisions faster, reducing development time and costs while boosting overall software quality and reliability.



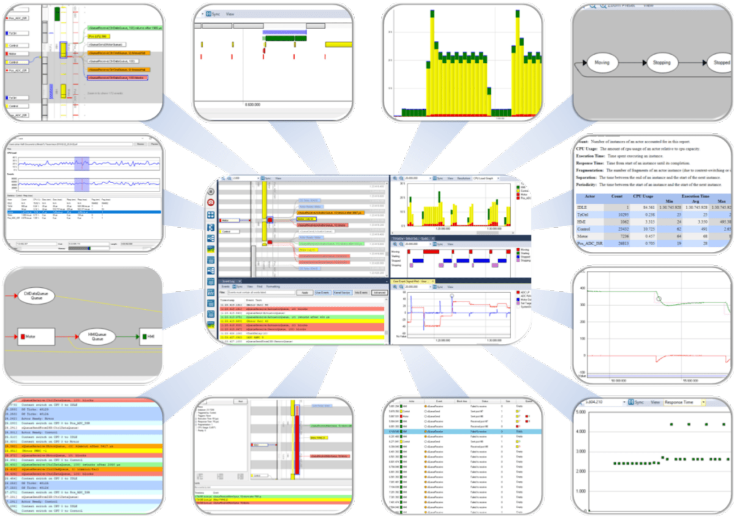
Control
Instance: 1654/23432
Triggered by: Pos_ADC_ISR
Triggers: 1 instance
Execution Time: 2.625 (ms.µs)
Response Time: 2.728 (ms.µs)
Fragmentation: 3
CPU Usage: 10,7%
Priority: 4

CPU Load Graph

The CPU Load Graph displays CPU usage over time, per actor and in total. By default, it shows all actors except the idle task. The analysis works by dividing the trace into a number of intervals, the default is 100 intervals.

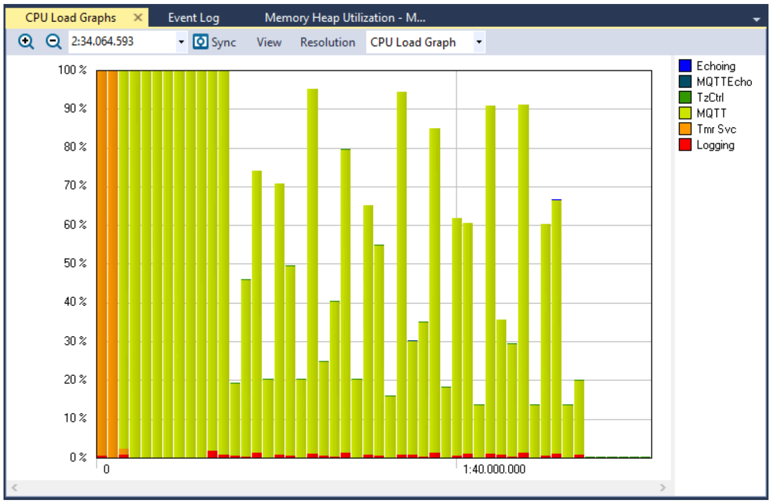
Many Views to Focus on Your Data

SpyKer-TZ gives you access to over 30 specialized views, each designed to offer deep insights into your system's behavior. From CPU load and memory usage to custom user events and detailed logs, each view provides a unique perspective on performance, helping you identify and resolve potential issues quickly. These views work together to give you a comprehensive understanding of every critical element of your application, ensuring you can optimize performance, improve reliability, and meet real-time deadlines with confidence. SpyKer-TZ equips you with the tools you need to focus on the data that matters most and make informed decisions at every stage of development.



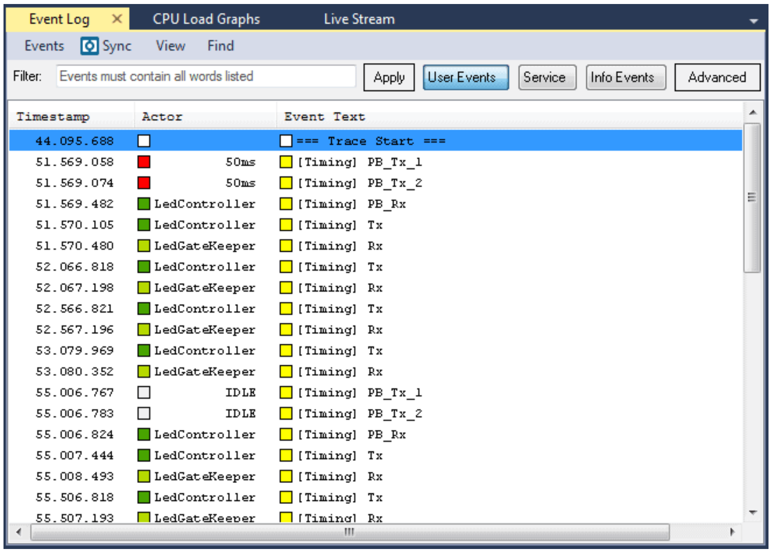
CPU Load Graph View

The CPU Load Graph View in Spyker-TZ provides a clear visualization of CPU usage over time, allowing you to quickly identify high-load periods that could lead to scheduling bottlenecks or performance degradation. By pinpointing spikes in CPU usage, you can assess whether your system has the capacity to handle additional workloads while still meeting critical real-time deadlines. This makes it an indispensable tool for optimizing performance and ensuring your system remains reliable under varying operational conditions.



Event Log

The user event feature in Spyker-TZ allows you to log and visualize custom application data, providing deep insights into key system behaviors, such as sensor data or state machine operations. By synchronizing these events with other views, you gain a comprehensive understanding of your system's performance in real time. Additionally, you can define intervals to measure the time between events, offering critical timing insights for processes that must meet strict operational deadlines, ensuring precision and reliability in your system's execution.

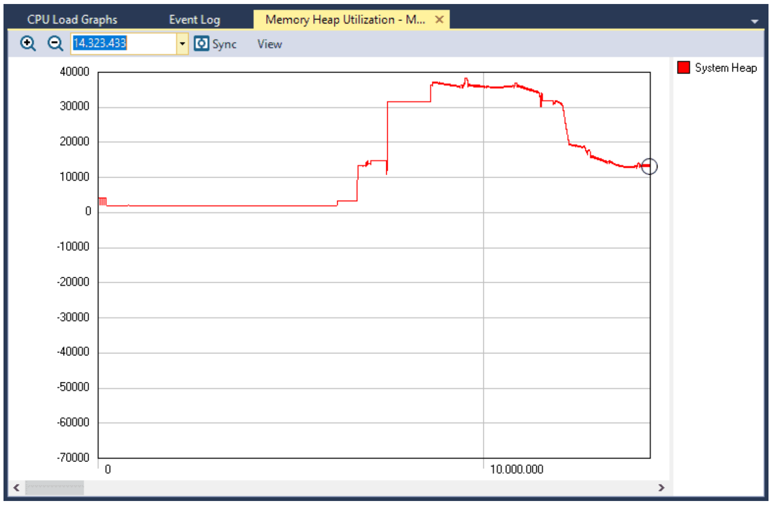


Filter: Events must contain all words listed

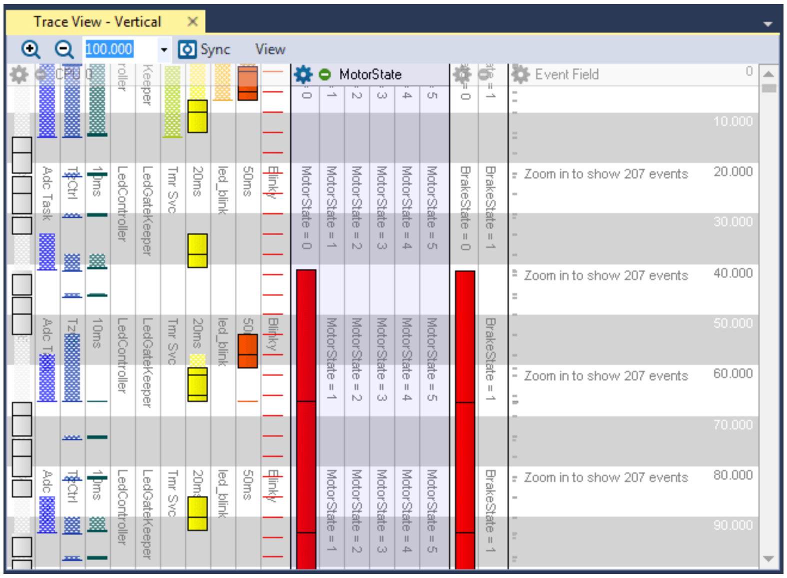
```
Timestamp    Actor               Event Text
44.095.688   □                   □ === Trace Start ===
51.569.058   ■              50ms ■ [Timing] PB_Tx_1
51.569.074   ■              50ms ■ [Timing] PB_Tx_2
51.569.482   ■ LedController     ■ [Timing] PB_Rx
51.570.105   ■ LedController     ■ [Timing] Tx
51.570.480   ■ LedGateKeeper     ■ [Timing] Rx
52.066.818   ■ LedController     ■ [Timing] Tx
52.067.198   ■ LedGateKeeper     ■ [Timing] Rx
52.566.821   ■ LedController     ■ [Timing] Tx
52.567.196   ■ LedGateKeeper     ■ [Timing] Rx
53.079.969   ■ LedController     ■ [Timing] Tx
53.080.352   ■ LedGateKeeper     ■ [Timing] Rx
55.006.767   □              IDLE ■ [Timing] PB_Tx_1
55.006.783   □              IDLE ■ [Timing] PB_Tx_2
55.006.824   ■ LedController     ■ [Timing] PB_Rx
55.007.444   ■ LedController     ■ [Timing] Tx
55.008.493   ■ LedGateKeeper     ■ [Timing] Rx
55.506.818   ■ LedController     ■ [Timing] Tx
55.507.193   ■ LedGateKeeper     ■ [Timing] Rx
```

Memory Heap Load

The Memory Heap Utilization View in Spyker-TZ provides clear visibility into memory allocation and deallocation, allowing you to quickly detect memory leaks and inefficiencies. By synchronizing it with other views, you can see how memory usage correlates with specific system tasks, offering deeper insight into performance bottlenecks. The Object History View further assists by tracking unmatched malloc and free events, helping you identify and resolve memory leaks efficiently, ensuring optimal resource usage and system stability.



Custom Trace View



The Custom Trace View in Spyker-TZ enables you to log and visualize custom application events, offering deeper insights beyond standard OS-level data. By creating user-defined event channels, you can track specific system actions, such as sensor data or critical system behaviors, and analyze them alongside OS events. This feature allows you to measure intervals between actions, like the time between a button press and release, giving you precise timing data to fine-tune performance. With the ability to define and monitor custom events, you can more effectively identify and resolve performance issues, ensuring your applications meet stringent system requirements.

Customer Testimonial

A major US-based aviation company had been chasing a multi-threaded timing issue in a third-party application library and device driver for 2-3 months with no success. After using SpyKer-TZ, they pinpointed the problem within a matter of days. SpyKer-TZ gave them the insights they needed to resolve the issue quickly and keep the project on track. The tool has been invaluable in helping diagnose complex timing problems, saving both time and significant costs.

“We spent months chasing a complex multi-threaded timing issue in a third-party application, with no progress. Once we deployed SpyKer-TZ, we pinpointed the root cause in just a few days. The tool gave us unprecedented visibility into system behavior, enabling us to resolve the issue quickly and keep the project on schedule. SpyKer-TZ didn’t just save us time–it helped avoid costly delays and ensured we met our critical deadlines. It’s now an essential part of our toolkit for diagnosing intricate timing challenges.”

- A Lead Systems Engineer at a US-Based Aerospace Program

Conclusion

By integrating SpyKer-TZ with MOSA.ic, developers can drastically reduce debugging time and mitigate the risk of late-stage timing issues that are common in complex, real-time embedded systems. SpyKer-TZ delivers unmatched visibility into system behavior, enabling teams to quickly identify performance bottlenecks and elusive bugs. With its advanced trace analysis, remote operation capabilities, and non-intrusive data collection, teams can optimize performance, reduce certification efforts by up to 30%, and accelerate time-to-market faster –all while improving system reliability and efficiency.

Experience the transformative power of SpyKer-TZ with MOSA.ic and revolutionize your approach to embedded software development. Contact us today to learn how SpyKer-TZ’s cutting-edge features can help you shorten development cycles, lower costs, and elevate product quality.

Visit our website: www.lynx.com or contact our team to schedule a demo and see how you can easily optimize your real-time systems.