LEVERAGING SLEEVE FOR REMOTE DEVELOPMENT WITH VECTOR TA TOOL SUITE AND INFINEON AURIX TC397

VECTOR INFORMATIK GMBH
SLEEVE GMBH

ABSTRACT

This white paper showcases the seamless integration of Sleeve, a platform designed for embedded remote development, testing, and deployment, with Vector TA Tool Suite —a user-friendly suite of software products tailored for the timing aware design, simulation, and verification of embedded real-time systems. The specific hardware setup employed involves the Infineon Aurix TC397 in conjunction with the Lauterbach TRACE32 debugger.

The primary focus of this white paper is to demonstrate the efficiency and simplicity of conducting timing analysis on the remote target with minimal installation efforts. The setup offers the added advantage of instant scalability to accommodate multiple users concurrently. Furthermore, it facilitates the effortless transfer of large trace data to the cloud for in-depth analysis.

A distinctive feature highlighted in this document is Sleeve's capability to support centralized tooling updates. This functionality ensures that any tooling updates are promptly delivered to all users across the network, enhancing collaboration and ensuring a uniform and up-to-date development environment for the entire team.

This white paper serves as a practical guide, illustrating how the integration of Sleeve with Vector TA Tool Suite and Infineon Aurix TC397 optimizes remote development processes, streamlines timing analysis, and supports collaborative efforts in a distributed development environment.

WHAT IS SLEEVE

Introduction

In the dynamic landscape of modern and distributed environments, efficient resource utilization remains pivotal for scalable embedded software development. Providing every developer within an organization with unhindered access to hardware resources is a critical milestone, as failure to achieve this can result in unforeseen expenses and production delays, disrupting product delivery.

The goal of Sleeve is to eradicate constraints imposed by inefficient hardware resource utilization. It aims to usher in a new era of fully automated, constraint-free hardware access, aiming to reduce hardware costs, cut down software development expenditures, and eliminate production delays.

Sleeve a cutting-edge fusion of cloud and edge technology. Sleeve comprises two integral components: Sleeve Hub, a web-based interface providing users with full access, and Sleeve Dock, an edge component ensuring swift and reliable communication with embedded devices. This innovative combination is meticulously designed to reduce development time, enhance availability, and ensure scalability across development and operations, regardless of project or company size.

Sleeve Hub

Sleeve Hub stands as the forefront entry point for users, offering an intuitive web interface that transforms the whole development experience. This chapter explores the multifaceted features of Sleeve Hub, emphasizing its pivotal role in initiating virtual workspaces directly connected to remote hardware.

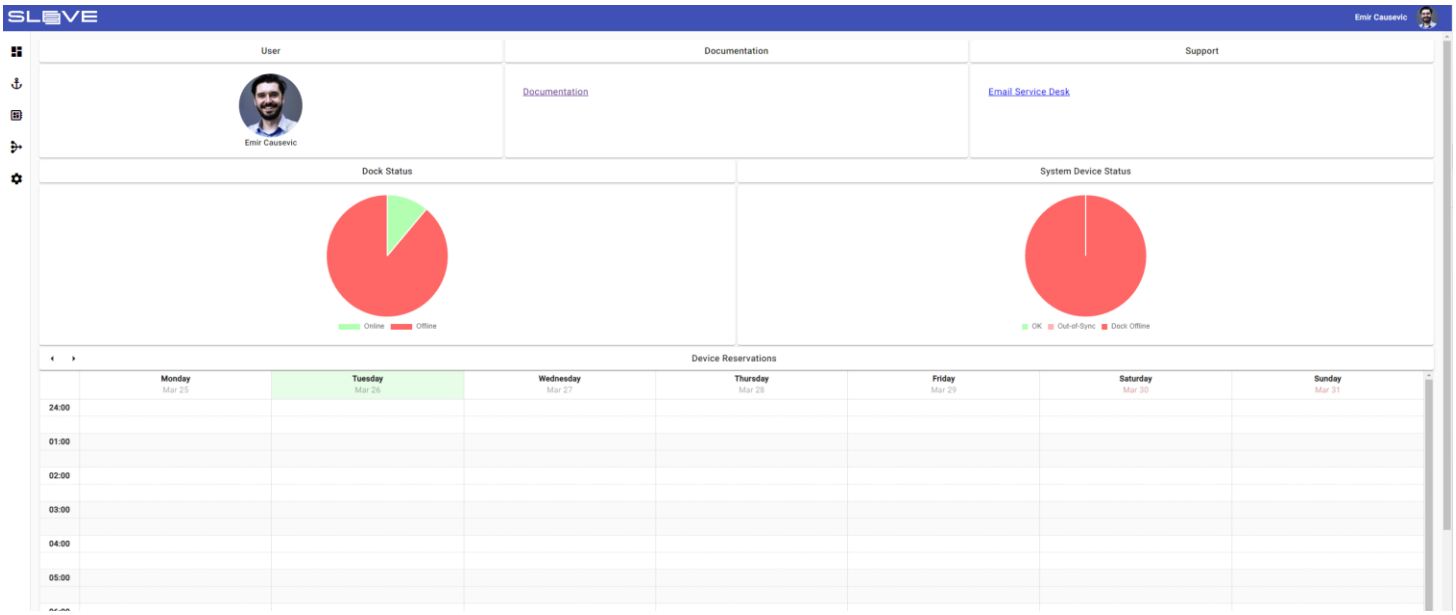

Figure 1 Sleeve Hub initial dashboard shows status of the connected devices

Sleeve Hub empowers users to effortlessly initiate virtual workspaces, bridging the gap between the local environment and vast possibilities of remote hardware. Virtual workspaces serve as dynamic

environments where tools seamlessly run on both the cloud and edge, ensuring a fluid and efficient development experience.

Facilitating a streamlined workflow, Sleeve Hub introduces a comprehensive device booking mechanism. This feature ensures that users have exclusive and secure access to the hardware resources they require, minimizing conflicts and optimizing resource utilization.

Sleeve Hub introduces a flexible user management system with two distinct user types:

- administrator
- regular users

Additionally, users can be efficiently sorted into groups based on their roles or projects, enhancing collaborative efforts and organizational efficiency.

Ensuring the utmost security, Sleeve Hub incorporates a robust OAuth2.0-based user registration system. This not only streamlines the onboarding process but also guarantees the confidentiality of user credentials, fostering a secure working environment.

To keep users informed and in control, Sleeve Hub boasts a detailed dashboard. This feature provides real-time insights into the health of the system, ensuring transparency and proactive management of resources.

Sleeve Hub is not just a web interface; it's your portal to a seamless, collaborative and efficient development journey. From virtual workspaces and secure user registration to complete hardware booking and system health monitoring, Sleeve Hub is designed to elevate your development experience, putting the power in your hands.

Sleeve Dock

In the dynamic landscape of remote development, Sleeve Dock emerges as a crucial edge component seamlessly integrating embedded hardware. This chapter dives into the transformative capabilities of Sleeve Dock, explaining its role as the orchestrator of hardware connections, the connection between Sleeve Hub, and the guardian of secure communication channels.

Sleeve Dock positions itself in proximity to embedded hardware, acting as the orchestrator of connections. It is meticulously designed to streamline and optimize the connection between the virtual workspace and the physical hardware. This pivotal role ensures a harmonious link between the developer's virtual environment and the real-world embedded systems.

Sleeve Dock serves as the vital link connecting Sleeve Hub to the embedded hardware. This synergy creates a holistic ecosystem, enabling users to seamlessly manage, control, and execute applications within the virtual workspace, transcending the barriers of physical distance.

Security is paramount in the Sleeve ecosystem. Sleeve Dock together with Sleeve Hub establishes a secure VPN tunnel.

Sleeve Dock is not just an edge component; it's the guardian of connectivity, execution, and security at the heart of embedded hardware. Through its orchestration of connections, seamless integration with Sleeve Hub, and the establishment of secure VPN tunnels, Sleeve Dock is a crucial part of Sleeve..

SLEEVE TOOLING REQUIRENMENTS

In the dynamic realm of embedded software development, Sleeve not only facilitates remote connectivity but also streamlines the deployment of tools crucial for the development process. This chapter focuses on the specific requirements for tools within the Sleeve system, with a spotlight on Linux-based tooling.

Sleeve is built with versatility in mind, supporting both Windows and Linux-based tools. While Windows tools leverage virtual machine technology, Linux tools operate within latest container technology. For the purpose of this white paper, our emphasis is on the unique features and capabilities of Sleeve's Linux tooling support.

Sleeve Linux Tooling

Sleeve simplifies tool deployment through containerization. It packages tools, along with their dependencies, into a Sleeve container template. This approach ensures consistency and eliminates compatibility issues, allowing for a seamless and reliable execution environment.

Within the virtual workspaces, Sleeve introduces the concept of actions. Actions represent single executions of tools, providing a granular and modular approach to development tasks. These actions operate within an underlying shared folder structure, enabling efficient file transfer between consecutive actions.

Sleeve's containerization model allows for the indefinite reuse of containers across multiple virtual workspaces. This means that once a container is created, it can be utilized in any number of virtual workspaces and in conjunction with other tools, irrespective of whether they are Windows or Linux-based.

Deliver

Sleeve enables precise tooling delivery for every users need.

Collaborate

Sharing virtual workspaces and files between users enables highly efficient working.

Scale

Scalable delivery to a large number of users along large-scale tooling updates.

One of Sleeve's key strengths lies in its ability to deliver a scalable rollout of any tool to large numbers of users in the instance. This setup ensures a consistent and efficient deployment process, catering to the diverse tooling needs of developers across different projects.

Sleeve extends its capabilities to facilitate the sharing of workspaces, allowing for the deployment of full virtual workspaces to an unlimited number of users in the instance. This feature, combined with large-scale tooling updates, ensures that the entire team operates in a synchronized and updated development environment.

With Sleeve, disaster recovery becomes a streamlined process accomplished with a single click. The system is engineered to swiftly recover workspaces and tools, minimizing downtime and ensuring continuous development progress even in the face of unforeseen challenge.

Sleeve's approach to tooling requirements and scalable deployment not only simplifies the development process but also sets the stage for efficient collaboration, large-scale updates, and disaster recovery.

USE CASE

In this chapter, we explore a comprehensive use case setup that demonstrates the seamless integration of Vector`s TA Tool Suite, Lauterbach TRACE32, and Infineon AURIX TC397 within the Sleeve system. This use case exemplifies the efficient execution of software, flash operations, and timing analysis in a collaborative and remote development environment.

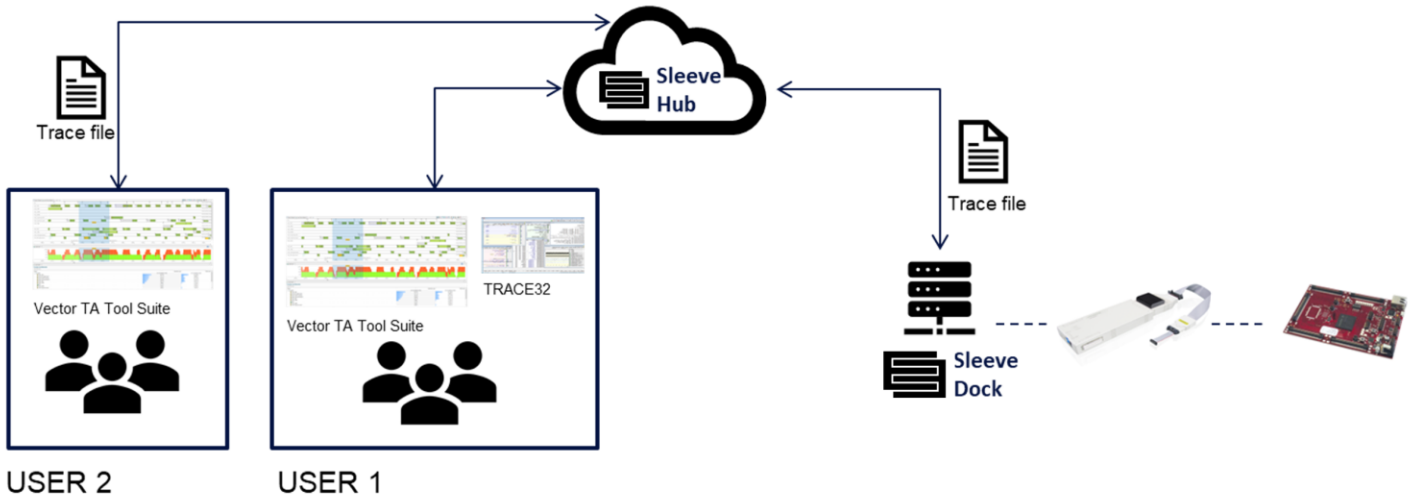

Figure 2 Use Case Setup with Vector TA Tool Suite, Infineon AURIX and Lauterbach TRACE32

Components of the Use Case

- Hardware Setup: Infineon AURIX TC397
  - The Infineon AURIX TC397 serves as the target hardware, providing the real-world context for software execution and analysis.
- Software Development and Debugging Tools: Vector TA Tool Suite and Lauterbach TRACE32
  - Vector TA Tool Suite: A user-friendly collection of software products for the design, simulation and verification of embedded real-time systems.
  - Lauterbach TRACE32: A powerful debugger and trace tool, essential for recording traces and facilitating software flashing on the AURIX board.

Workflow Execution within Sleeve System

Step 1: Software Flashing with TRACE32

The software in form of .elf file is flashed onto the Infineon AURIX TC397 board using Lauterbach TRACE32. This step ensures the deployment of the developed software onto the physical hardware.

Step 2: Tracing with Lauterbach TRACE32

Lauterbach TRACE32 is initiated to record traces during the software execution on the AURIX board. This step captures critical information about the program's execution.

Step 3: Timing Analysis with Vector TA Tool Suite

The recorded traces from TRACE32 can be seamlessly imported into the Vector TA Tool Suite within the Sleeve system. This enables users to conduct in-depth timing analysis, gaining insights into the temporal aspects of the embedded real-time system. TA Tool Suite provides this in-depth view by combining the trace information with an AUTOSAR configuration given by a Vector MICROSAR/DaVinci project.

After importing the trace (and optionally AUTOSAR project) files, features like constraint validation, timing analysis in GANTT Chart, histograms (e.g. jump to instances like outliers), constraint and metric tables support in detecting underlying issues.

Figure 3 shows an example, where a constraint (<=1ms) was set to verify the response time of “TASK_1MS”. The constraint table (1) shows that the constraint was violated 139 times in 2000 total occurrences. Now we can use the histogram (2) to see the violated instances and directly jump to the corresponding instances (3) in the GANTT chart (4) to understand why the 1ms constraint is not kept.

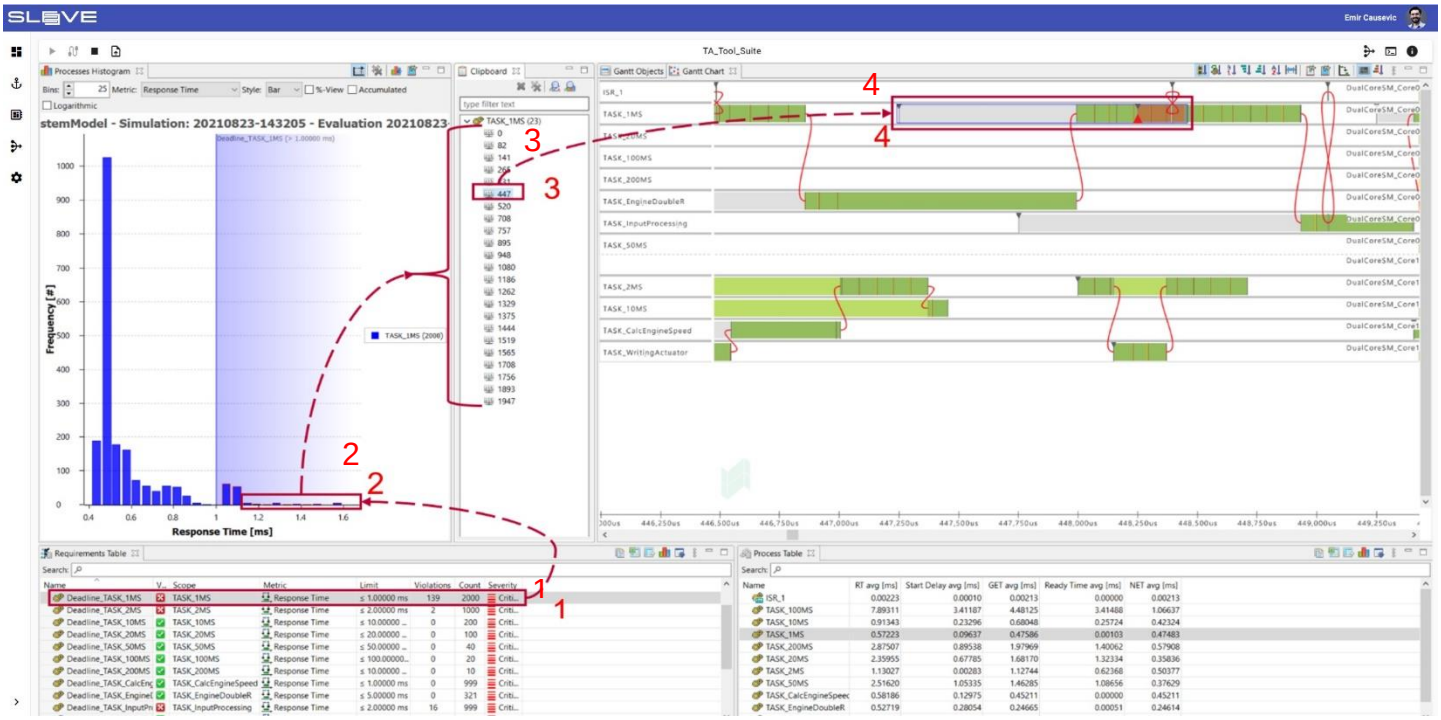

Figure 3: Analysis of recorded Trace in TA Tool Suite

There is a wide range of timing metrics that can be defined as constraints for automatic validation, for example defined TIMEX constraints like event chains or execution order constraints.

The timing analysis can also be performed automated within a CI/CD Pipeline within the sleeve system. Actions like automated constraint validation can run in a pre-defined TA Tool Suite workflow, executed from the command line. TA Tool Suite can then export a comprehensive HTML report.

Cloud Backup for Collaborative Analysis:

The traces can be parallelly backed up in the cloud in the Sleeve System. This cloud repository serves as a centralized storage location accessible to multiple users. Other developers can pull the traces and conduct timing analysis in their own virtual workspaces without the need for direct hardware access.

Optimizing the System

The user can import the DaVinci project directly into the TA Tool Suite, using the TA Tool Suite Option Design to analyze dependencies, improve runnable to task mapping and compare different design architectures with the TA Tool Suite Simulation option. The changes can be made directly in TA Tool Suite. Finally, to use the optimized ECU configuration, TA Tool Suite can update the MICROSAR/DaVinci project files directly.

.

Update the tooling

Sleeve enables the centralized updating of the tooling that are executed in the virtual workspaces.

There are 3 steps in the process for updating the tooling:

a. Management of tool release versioning, offering a centralized and automated approach. Once a tool is released to Sleeve, it seamlessly transitions into the Sleeve CI/CD (Continuous Integration/Continuous Deployment) pipeline for rigorous testing of its functionality within the Sleeve environment.
b. Upon successful testing, a new tag for the release is generated, signifying its readiness for deployment. Users or project administrators seeking to update developers' software can effortlessly facilitate this process by providing the new tag to the designated action. With the next execution, the software is automatically updated across the designated environment.
c. For those desiring a more proactive approach, administrators have the option to configure the system to deliver new versions automatically as soon as they are supplied by the software vendor. By setting the tag in the action to "latest," software updates are seamlessly integrated into the environment upon release, ensuring that users benefit from the latest features and enhancements with minimal intervention.

In essence, Sleeve's centralized tool release versioning streamlines the update process, offering flexibility and automation to adapt to varying project requirements and software release cycles.

Benefits of the Use Case Setup

Collaboration and Accessibility:

Developers can collaborate seamlessly, leveraging cloud-backed traces for analysis. This enhances accessibility and ensures that insights gained from the timing analysis are shared across the development team.

Remote Analysis without Direct Hardware Access:

Sleeve's architecture enables users to conduct timing analysis in their individual workspaces without requiring direct hardware access. This decentralization of analysis enhances flexibility and scalability in a remote development environment.

The use case setup with Vector TA Tool Suite, Lauterbach TRACE32, and Infineon AURIX TC397 within the Sleeve system exemplifies the efficiency, collaboration, and accessibility achieved through remote development and analysis. This approach ensures that critical insights from timing analysis are not only captured efficiently but are also shared and utilized across the development team. The Sleeve system has a distinct advantage over typical cloud solutions like AWS and Azure, as it integrates physical hardware within the same system. This enables low-latency use cases, such as debugging and working with debugging tools, as well as handling more resource-demanding tasks on the same platform without offloading computation.

CONCLUSION

In conclusion, this whitepaper has unraveled the innovative capabilities of Sleeve, a platform meticulously crafted to revolutionize remote software development, testing, and deployment on embedded systems. The dynamic integration of Vector TA Tool Suite, Lauterbach TRACE32, and Infineon AURIX TC397 within the Sleeve system serves as a compelling testament to the platform's versatility and efficiency.

Sleeve's unique approach to tooling requirements, containerization, and scalable deployment ensures a seamless and secure environment for Linux-based tools. The marriage of cloud and edge technologies in Sleeve Hub and Sleeve Dock provides users with a cohesive and collaborative development experience, unbounded by physical constraints.

The detailed exploration of a use case involving software flashing, tracing, and timing analysis demonstrates the real-world application of Sleeve's capabilities. The ability to back up traces in the cloud, enabling collaborative analysis without direct hardware access, showcases the platform's commitment to accessibility, collaboration, and scalability.

As we conclude, Sleeve stands as a transformative force, empowering developers to break free from traditional constraints and embrace a future where remote development is not only efficient but also collaborative and secure. Whether facilitating large-scale tooling updates, disaster recovery mechanisms with a single click, or decentralized timing analysis, Sleeve paves the way for a new era in embedded software development.

Embark on this journey with Sleeve, where innovation meets practicality, and remote development evolves into a seamless, collaborative, and efficient experience. Together, let's shape the future of embedded systems development.