ELISA Enabling Linux in Safety Applications

APERTIS

Reproducible System Composition Combining Linux, Xen & Zephyr on One Embedded Hardware

Apertis as system element in the ELISA project's example system

Case Study

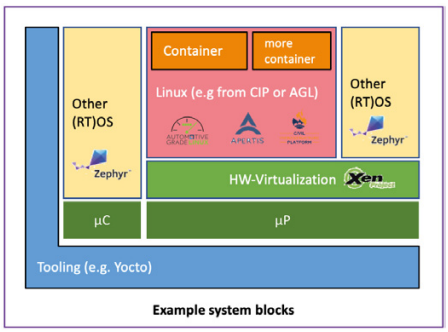
This case study describes the evolution path of new software-defined industries towards modular and flexible system architectures.

Introduction

Industries across the board are witnessing a shift towards software-defined systems, where previously distributed functions are now consolidated into single hardware systems. This transition is made possible by the continuous advancements in system-on-chip (SoC) compute performance, enabling the realization of heterogeneous mixed-criticality systems. The work focuses on the Automation, Automotive, and Aerospace industries, which have specific requirements in terms of safety and real-time capabilities.

System Setup and Similarities

Setting up these software-defined systems is a complex and time-consuming task, involving infrastructure setup, base system creation, and ongoing maintenance. Despite the diversity of use cases, the overall system architectures in these industries exhibit similarities.



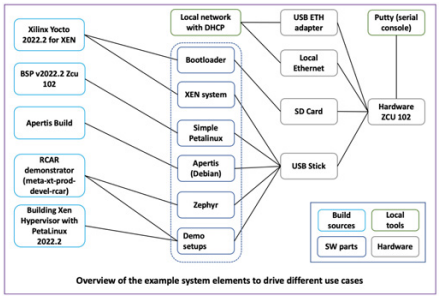
Example system blocks

Typically, these architectures include a Real-Time Operating System (RTOS), a performance rich operating system like Linux or Android, and a hypervisor solution. The RTOS can be implemented on either a microcontroller (µC) or a microprocessor (µP).

Typical Demo System Limitations

Each use case imposes unique requirements on the system. To assess feasibility, demo systems and proof-of-concepts (PoCs) are developed. While these systems are showcased at fairs, exhibitions, and conferences, they often lack comprehensive documentation and reproducibility. The long-term maintenance, adaptability, and component replacement aspects are often overlooked.

ELISA Approach to Systems



Overview of the example system elements to drive different use cases

The ELISA Systems Working Group aims to break the pattern of ad hoc system architectures by focusing on reproducibility and flexibility. Instead of solely adapting existing architectures to specific use cases, the group emphasizes understanding the system elements and their interdependencies. By establishing a reproducible setup, encompassing documentation, continuous integration (CI), and automated testing, the system can be subjected to specific use cases.

Use cases reveal dependencies on system elements and transform the example system into a fully functional demo system. The GitHub documentation already includes simple examples, providing further insights. For more information, please follow the QR code.

Outlook

The ELISA Systems Working Group is actively expanding its base system setup to incorporate a combination of hardware and software variants. This extension enables the visualization of variant-specific interfaces and dependencies. It addresses the question of how changes in the RTOS, hardware, Linux distribution, or hypervisor impact the migration from one product to another. Furthermore, this approach simplifies tailoring the system to specific use cases to let system providers focus on differentiating elements.

> "The existing guidelines for prototyping systems are limited, making it difficult and time-consuming to reproduce demos. The ELISA systems working group aims to address this issue by offering infrastructure and comprehensive documentation, covering everything from architecture to use cases. This initiative empowers others to develop their own systems."

Philipp Ahmann,
Technical Steering Committee Chair of the ELISA Project and Product Manager for Embedded Open Source at Bosch

APERTIS Zephyr Xen Project THE LINUX FOUNDATION BOSCH