Software-intensive embedded systems are characterized by a constant increasing number of features, which perform with different levels of criticality, implementing functionality with continuously growing complexity. This rise in software complexity drives a parallel growth in the processing demands to the underlying hardware, materializing in the increase of the number of processing nodes (e.g. Electronic Control Devices ECU s in the automotive domain) used, leading to an explosion of inter node communication requirements (either on chip or between chips), with all the challenges and problems it entails. One way to tackle these challenges in a cost effective way is the usage of multicore processing platforms.

The usage of multicore CPUs allows for very fast and reliable communication between the cores, but at the same time exponentially increasing the complexity of the verification process for the system.

To effectively manage this complexity, development processes in general, and model-based approaches in particular, support the development assuming an idealized (component-based) model of computation, abstracting away from implementation issues like interference aspects of the execution platform resulting from shared computation or memory resources. In order to make these (seamless) model-based approaches usable, proper tooling support is required.

AutoFOCUS3 (https://af3.fortiss.org/) is an open-source model-based development tool, including a number of different analysis and verification tools as well as design space exploration functionality, task scheduling dependent on a number of system requirements (timing, resource, energy, safety, etc.), and code generators targeting software (C) and hardware (VHDL).

The presented demonstrator illustrates both a SW tool demonstrator and a corresponding HW demonstrator setup to show how a seamless model-based system approach could look like, w.r.t. to mixed-critical applications integrated on a (COTS) MC-platform. We illustrate how the challenges of both the increasing complexity of such systems, as well as the demands for more integrated HW platforms, namely multicore platforms, can be targeted using an integrated model-based system design.

The SW part of the demonstrator illustrates how an open-source CASE tool - AutoFOCUS3 - can be used for a requirements to deployment development of such systems, by modeling applications of different criticalities, generating and verifying code and schedules and finally deploying it on the target platform running a Real Time Operating System (RTOS). These applications are deployed on the same processing node, sharing computational resources, creating a mixed criticality system. The RTOS targeted in our development chain is PikeOS, which guarantees separation of concerns and timeliness of tasks, running on a quad-core Freescale IMX6Q CPU.

That means that AF3 does not only generate code for the different functions running as tasks on different cores on the target platform, but also generates a schedule. This schedule takes care that the tasks do not interfere with each other, can run in parallel and thus addresses the separation of the different criticalities. Consequently, the AF3 code generator guarantees the spatial and temporal separation of functions (tasks) of different criticalities.

In order to make the challenge of this mixed-criticality multi-core system demonstrable, we provide a gamification part, which needs user interaction. A floating ball in a glass tube can be controlled by an observer by moving his hand over an ultra sound sensor, providing input to the control loop implemented in the high criticality part of the system.

The low criticality application part of the system which is running in parallel on the same CPU consists of the computation of the digits of PI and of the Fibonacci sequence, providing computationally intensive neighbors to the control loop, without interfering with the highly critical application. All relevant system information (sensor values and current result of the computations) is displayed on a touch screen.