

A Time-Triggered Middleware for Safety-Critical Automotive Applications

Ayhan Mehmed and Wilfried Steiner

TTTech Computertechnik AG, Vienna, Austria
{ayhan.mehmed,wilfried.steiner}@tttech.com

Advanced driver assistance systems (ADAS) are one of the fastest growing sectors in the automotive industry. Initially developed as add-on comfort features, ADAS now target highly intelligent, fully autonomous, vehicle control systems. Towards this goal various technical challenges have to be addressed to guarantee dependable and deterministic vehicle behavior.

A clear challenge is the growing number of ECUs, which drives complexity, weight, power, space consumption and ultimately the cost. For this a higher degree of integration of function per ECU is needed. At the same time, the integration of functions from different domains and with differing requirements, will require interference-free coexistence of mixed-criticality functions. The demand for further ECU consolidation, requires virtualization like techniques, which allow the coexistence of multiple operating systems on the same ECU. Last but not least, the vehicle systems of today have to execute their functions with respect to real-time, meet the highest safety requirements (up to ASIL-D) and must accelerate their development cycles.

To address these challenges we outline a novel time-triggered middleware for vehicle cyber-physical systems (CPS), namely TTIntegration. Complementary to the TTIntegration middleware, the AUTOSAR software architecture and concepts have been followed as a reference. Starting from top, on application level the software components communicate according to the AUTOSAR standardised application interfaces. One level down, the TTIntegration middleware is placed for a clear separation between the integrated applications and the basic software and hardware. Apart from serving as abstraction layer the middleware enables the execution of tasks according to a time-triggered paradigm - a set of concepts and principles, which provide (i) a predictable timing behavior of each application, by ensuring sufficient CPU-time and memory, and (ii) a guaranteed freedom of interference between the applications. Furthermore, for high degree of function integration, the middleware enables multi-core system-on-chips (SoCs) to run in parallel with different operating systems (e.g. AUTOSAR, VxWorks) and collaborate with each other. By extending the AUTOSAR environment for all possible operating systems, all applications can then be moved between different SoCs with ease. Finally, to facilitate the process of software integration, the middleware provides parallel, multi-vendor development and integration paths for individual software components.