Introduction to Adaptive AUTOSAR

Dheeraj Sharma
July 27, 2017
Overview

- Software Platform and scope of Adaptive AUTOSAR
- Adaptive AUTOSAR architecture and roadmap
- EB Adaptive Platform and Prototyping solution
Requirements for a future car infrastructure

• Main drivers
  – Automated Driving
  – Car-2-X applications

• Requirements
  – High computing power
  – High data rates
  – High availability, fail-operational systems
  – Update over the air
Consolidated E/E architecture
Future architecture of car infrastructure

• Split up ECUs in low performance IO Controller and high performance controller
• Establish a service-oriented architecture (SOA)

• **Performance Controller**
  – High computation power with heterogeneous computing
  – Widespread, POSIX-like Operating System (e.g. Linux), Adaptive AUTOSAR
  – Extensive update capabilities
  – Safe & Secure

• **IO Controller**
  – Provide Sensor and Actuator Services
  – Deeply embedded, real-time Operating System (e.g. Classic AUTOSAR)
  – Limited (non-standardized) support for partial updates
Major Types of Adaptive Platforms

Type 1: µC Partitioning

Type 2: Core Partitioning

Type 3: Virtual Resource Partitioning

Type 4: Performance, MultiCore
Overview

• Software Platform and scope of Adaptive AUTOSAR

• Adaptive AUTOSAR architecture and roadmap

• EB Adaptive Platform and Prototyping solution
Adaptive AUTOSAR (ARA)

- **Execution Management**
  - Lifecycle management of platform (machine) and application (process) incl. privileges of access control and machine states

- **Persistency**
  - Load data from persistent storage and store data over boot and ignition cycle

- **Communication Management**
  - SOME/IP based including serialization and service discovery
  - Publish/subscribe mechanism for intra- and inter-ECU communication

- **Platform Health Management**
  - Alive Supervision

- **Diagnostics**
  - Event memory management and diagnostic service handling

- Developed in C++ (C14)
- OS will provide POSIX PSE51 interfaces to the application but Adaptive Platform will require further features
Process Model

Each application lives in its own address space

Access to platform functionality via libraries

Communication via implementation specific IPC

Adaptive AUTOSAR Foundation

Adaptive AUTOSAR Services

(Virtual) Machine / Hardware
Functional Clusters

AUTOSAR Runtime for Adaptive Applications

Adaptive AUTOSAR Services

Time Management
Execution Management
Software Configuration Management
Security Management
Diagnostics
Persistency
Platform Health Management
Logging and Tracing
Hardware Acceleration
Communication Management

Adaptive AUTOSAR Foundation

(Virtual) Machine / Hardware

APIs and services exposed to applications by functional clusters

Services provided via Communication ARA API

Language specific APIs as part of ARA
Service Interface

Adaptive Application

- Public Interface
  Part of the adaptive AUTOSAR API and specified in the SWS.

- Protected Interface
  Interaction between functional clusters. Not normative, intended to make specification more readable and to support integration of SW into demonstrator.

- Private Interface
  Interaction between elements within a functional cluster. Not used in specifications.

Network

C++11 Language Binding
Dispatching and Discovery
SOME/IP Transport
IPC Transport
TCP/IP
IPC
Ethernet Driver

|normative interface| informative interface| private interface|
• Continuous AUTOSAR releases twice a year (March, October)
• Distribution of AUTOSAR Code Base for concept validation
• Adaptive AUTOSAR specifications will be in status “development” until R18-10 (no backward compatibility granted)
Overview

• Software Platform and scope of Adaptive AUTOSAR

• Adaptive AUTOSAR architecture and roadmap

• **EB Adaptive Platform and Prototyping solution**
EB tresos AdaptiveCore - System

**Host**
- EB tresos Studio for AdaptiveCore
- EB tresos AdaptiveCore SDK
- EB Build Environment

**Target**
- ARA Components/Adaptive Platform
- Operating System (incl. BSP)
- 3rd Party Modules
EB tresos Studio for AdaptiveCore

System Configuration Tool

Development
- IDE for construction and qualification of application software components

Integration & Configuration
- Configuration of system manifest and binding of application software components to adaptive platform on machine and application level
EB tresos AdaptiveCore

Host System
- Studio
- SDK
  - Key Features: Configurator, Generators, Manifest Builder, CDT for C++
  - Key Features: CDT, QEMU
- DEBUG
  - Key Features: Deploy, Run, Debug

Target System
- Application SWC
- Application SWC
- Application SWC
- Application SWC

Runtime Environment for Adaptive (ARA)

Foundation
- Time Management
- Execution Management
- Logging and Tracing
- Communication Management
- Platform Health Management
- Persistence
- HW Acceleration

Services
- Diagnostics
- Security Management
- SW Configuration Management

Execution Management
- Management
- Management
- Management
- Management

Diagnostics
- Management
- Management
- Management
- Management

Security Management
- Management
- Management
- Management
- Management

OS
- Linux
- BSP

Adaptive Platform
- Bootloader
- Hypervisor

All bold names are licensable items.
Target support

• Lead Target platforms
  – Renesas R-Car H3

• Further support planned for
  – Nvidia DrivePX
  – Intel Denverton

• Further portings on request
  – Please inquire

• Development based on Yocto Linux
  – Development of ASIL-B Linux distribution

• Porting to other operating systems upon request
  – Please inquire
We take AUTOSAR to the road!