Nov 6, 2008

Getting Started with Freescale's AUTOSAR OS and Microcontroller Abstraction Layer (MCAL) Software Packages

PA110

Dr. Andreas Both / Zhang Enqin
Automotive Runtime Software
Abstract

► The automotive software standard AUTOSAR defines a new software architecture for use in automotive applications.

► Continuing the general introduction to AUTOSAR given at FTF 2006 and 2007, this session will explain what AUTOSAR software Freescale develops and what is needed to get started with it.
Structure of this Session

► Freescale AUTOSAR Software

  • Microcontroller Abstraction Layer
  • Operating System
  • Configuration Methodology
  • Roadmap

► Live Demo

  • Autosar MCAL and OS Configuration with EB tresos™ Studio
Standards driving E/E Architecture Design

- Technology partnerships and open standards encouraging “plug-and-play” approach
  - FlexRay Protocol
  - Automotive Open System Architecture (AUTOSAR)
  - Japan Automotive Software Platform Architecture (JasPar)
AUTOSAR – An Industry Standard

- Specification R2.0
- Improvements R2.1
- Release 2.0
- Release 2.1

2005 2006

Phase II (2007 – 2009) Basic Software & RTE
- Specification R3.0
- Concepts R4.0
- Specification R4.0
- Improvements R4.0
- Release 3.0
- Release 3.1
- Release 4.0

2007 2008 2009
AUTOSAR – What is it?
AUTOSAR BSW Architecture—Basic Layers

- Application Layer
- AUTOSAR Runtime Environment
- Services Layer
- ECU Abstraction Layer
- Microcontroller Abstraction Layer
- Basic Software
- Complex Driver
- Microcontroller

Application Software Component
Application Software Component
Application Software Component
Application Software Component
Freescale’s offering is based on AUTOSAR Basic Software

- AUTOSAR software „product“ packages from Freescale
  - MCAL
  - Operating System
Microcontroller Abstraction Layer (MCAL)
Microcontroller Abstraction Layer

Microcontroller Drivers
- MCU driver providing PLL initialization, RAM initialization, Reset, Wakeup
- Drivers for internal peripherals, e.g. on-chip watchdog, general purpose timer
Microcontroller Abstraction Layer

Memory Drivers
- Drivers for on-chip memory devices (e.g., internal flash, internal EEPROM)
- Functions with direct μC access (e.g., RAM test)
Microcontroller Abstraction Layer

Communication Drivers
• Drivers for ECU onboard communication: SPI, I2C
• Drivers for in-vehicle communication: LIN, CAN, FlexRay™ networking
Microcontroller Abstraction Layer

I/O Drivers
- Drivers for analog and digital I/O: e.g., analog to digital controller (ADC), pulse width modulation (PWM), digital input output (DIO)
Freescale Microcontroller Abstraction Layer

- MCAL drivers \(^{(1)}\) + Flash EEPROM Emulation
- Delivered with AUTOSAR conformant configuration tool.

(1) V2.1 releases do not contain RAM Test module.
BSW Configuration Classes

- **Pre-compile configuration**
  - Configuration parameters can not be changed after compilation

- **Link-time configuration**
  - Configuration is determined by linker scripts
  - Configuration parameters can not be changed after link process

- **Post-build configuration**
  - Configuration parameters can be changed after build process without complete re-flash of ECU
  - Configuration parameters are stored at a known memory location
  - Post-build configuration class BSW modules might also contain pre-compile or link-time parameters (not all parameters have to be post-build)
AUTOSAR Operating System
Configurable in AUTOSAR-conformant configuration tool

Available in Scalability Classes 1, 2, 3, 4 to fit the needs of different applications

- SC1 – deterministic RTOS baseline (tasks, events, counters, alarms, messages)
- SC2 – timing based task determinism (low-latency, precise timing for periodic tasks)
- SC3 – protected memory (MMU/MPU) for tasks avoids memory collisions for safety systems
- SC4 – timing and memory protected tasks, utilizes the full capabilities of the silicon for secure and protected RTOS designed specifically for the automobile.
AUTOSAR OS overview

AUTOSAR OS is OSEK/VDX™ OS plus:

- New core features
  - Software and hardware counters
  - Schedule tables with time synchronisation
  - Stack monitoring

- Protection features
  - Timing protection, memory protection and service protection
  - OS applications, trusted and non-trusted code
OS Application and Trusted and Non-Trusted Code

► Integrity level: trusted and non-trusted code

► OS application
  • A block of software including tasks, ISRs, hooks and trusted functions
  • **Trusted**: An OS application that has unrestricted access
  • **Non-trusted**: An OS application that has restricted access

► Trusted function
  • A service function with unrestricted access
  • Provided by a trusted OS application
AUTOSAR OS Memory Protection

► Protection boundary
  • Task/category 2 ISR
  • OS application (all its Task/Category 2 ISRs)

► Protected memory
  • Stack RAM
  • Private data RAM
  • Code flash

► Focus on write protection, read and execute protection are optional
Usage of Memory Protection

► A Non-trusted OS application task
  • Can only access the memory it needs
  • Therefore not this OS application task is protected, but all other tasks are protected from it

► Memory protection can be used, e.g.,
  • To separate different applications on one MCU
  • For isolating controller functionality from independent sub-suppliers
  • To fulfill safety constraints
  • As a debug feature (faulty memory access is prevented, stack overflow is prevented, protection hook is called)

► Memory protection must be supported by on-chip MPU
Usage of Service Protection

Service Protection

• Protection against faulty/corrupted OS service calls by an OS Application
• Examples
  ▪ OS Application calls ShutDownOS()
  ▪ OS Application tries to execute ActivateTask() on a task belonging to another OS Application
• Protection Hook is called upon detection of a service protection error
Usage of Timing Protection & Global Time

**Timing Protection**
- Execution time enforcement
  - Bounds the execution of ISRs, resource locks and interrupt disabled sections at runtime to a statically configured value ("time budget")
- Arrival rate enforcement
  - Bounds the number of times that an ISR can execute in a given timeframe to a statically configured limit
- Protection Hook is called upon detection of a timing protection error

**Global Time / Synchronization Support**
- Requires a global time source, e.g. the FlexRay network time
- This feature allows schedule tables to be synchronized with a global time through special OS service calls
## AUTOSAR OS Scalability Classes 1–4

<table>
<thead>
<tr>
<th>Feature</th>
<th>Scalability Class 1</th>
<th>Scalability Class 2</th>
<th>Scalability Class 3</th>
<th>Scalability Class 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSEK OS (all conformance classes)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Counter Interface</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Schedule Tables</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stack Monitoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Protection Hook</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Timing Protection</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Global Time/Synchronization Support</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Memory Protection</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OS Applications</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Service Protection</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CallTrustedFunction</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Software Configuration
EB tresos Studio is a easy-to-use tool for ECU standard software configuration, validation and code generation

- Full support for the AUTOSAR standard
- Full support for the Freescale AUTOSAR software and the EB tresos AutoCore

- Integrated, graphical user interface
- Based upon Eclipse and open standards
- Online-help and parameter-specific help

Source: Elektrobit
Main Window

Project Browser

Node Outline

Editor

Parameter Information

Error & Problem Messages

Source: Elektrobit
Errors & Warnings

User corrects the problem

Interactive problem resolution

Source: Elektrobit
Parameter Definition

Jump to link

Source: Elektrobit

Parameter "OsCounterType"

... and its corresponding entry in the description file (*.EPD)
Parameter Description Files – EPD/EPC

Source: Elektrobit
Parameter Description Files – EPD/EPC

Source: Elektrobit
Parameter Description Files – XDM

EB tresos Studio Configurator

BSW Module Description

XDM

EB tresos Studio Generator

BSW Module Configuration

EPC

c, h templates

c, h

Generated Code

Legend

- AUTOSAR Files
- Elektrobit Files
- Generated Files

Source: Elektrobit
Parameter Description Files – Beyond MCAL

Legend
- AUTOSAR Files
- Elektrobit Files
- Generated Files

Source: Elektrobit

1) currently in development
Configuration Checking

Hard-to-find errors detected instantly

Error: ID does not start with zero

Source: Elektrobit
EB tresos Studio - Summary

► User-Friendly GUI
  ➔ helps to avoid errors
  ➔ easy navigation
  ➔ one integrated tool environment

► Open Interfaces
  ➔ integration of user-specific BSW modules (1)
  ➔ integration of user-specific importers and exporters (1)
  ➔ customizable through Java plug-ins (1)

► Integrated Environment
  ➔ allows FSL MCAL and OS configuration, or complete AUTOSAR
    BSW configuration
  ➔ planned to be extended with more EB tools by Elektrobit
    (SW-C Editor, Inspector, etc.)

Source: Elektrobit

(1) requires full EB tresos Studio version from EB
Folder Structure of MCAL Package

- Parameter description files (.epd/.epc + .xdm)
- Generators for Pre-Compile, Post-Build, Link-Time
- EB tresos Studio Plugin

Complete source code
Makefile
User Manual
Integration Manual
AUTOSAR Software Roadmap
Freescale AUTOSAR Software Release Framework

► Beta Release (BETA)
  • All functionality included.
  • All testing targets achieved.
  • Complete documentation set.
  • For use in production projects. Not recommended for production freeze.

► Release to Market (RTM)
  • Beta Period completed
  • Identified issues addressed
  • Target level for Quality and Maturity achieved
  • Ready for production freeze
How to read the SW Roadmap

- Beta Release Date
- RTM Release Date
- Development Period
- MPC5561 MCAL 2.1 /GHS
- MCU
- Supported Compiler
- Type of SW

Color Coding:
- Available
- In Design
- Planned
- Proposed
- Last Proposed Revision

Compiler Coding:
- /GHS = Green Hills MULTI
- /WRS = Windriver DIAB
- /CW = Freescale CodeWarrior
- /COS = Cosmic
32-bit AUTOSAR MCAL Roadmap

Available
In Design
Planned
Proposed
Last Proposed Revision

MPC5567 MCAL 2.1 /WRS/GHS/CW
MPC5565 MCAL 2.1 /WRS/GHS/CW
MPC5561 MCAL 2.1 /WRS/GHS/CW
MPC5634M/33M MCAL 3.0
MPC5632R MCAL 3.0
MPC5633M MCAL 2.1 /GHS
MPC551x(*) MCAL 2.1 /GHS
MPC5604B/3B MCAL 2.1 /GHS
MPC5604P MCAL 3.0
MPC5604S MCAL 3.0
MPC5607B/6B/5B MCAL 2.1 or 3.0 (**)
MPC551x MCAL 3.0 or 3.1 (**)
MPC5608S MCAL 3.0
MPC5607B/6B/5B MCAL 2.1 or 3.0 (**)
MPC5602B MCAL 3.0 or 3.1 (**)

* Beta for MPC5517G/16G only
** AUTOSAR version dependent on market demand
32-bit AUTOSAR OS Roadmap

- **Available: MPC5516 OS 2.1/GHS EAR**
- MPC5567 OS 2.1/WRS/GHS/CW
- MPC5565 OS 2.1/WRS/GHS/CW
- MPC5561 OS 2.1/WRS/GHS/CW

- MPC5634M/33M OS 3.0
- MPC5633M OS 2.1/GHS
- MPC5632R OS 3.0
- MPC551x OS 2.1/GHS
- MPC551x OS 3.0 or 3.1 (**)

- MPC5604B/3B OS 2.1/GHS
- MPC5607B/6B/5B OS 2.1 or 3.0 (**)
- MPC5604B/3B OS 3.0
- MPC5608S OS 3.0
- MPC5604P OS 3.0
- MPC5606S OS 3.0
- MPC5604S OS 3.0

- Leopard OS 3.0
- MPC5668G OS 2.1 or 3.0 (**)
- MPC5668E OS 3.0
- MPC5604B/3B OS 3.0

**AUTOSAR version dependent on market demand**

- Available:
  - MPC5516 OS 2.1/GHS EAR
- Planned:
  - MPC5567 OS 2.1/WRS/GHS/CW
- Proposed:
  - MPC5565 OS 2.1/WRS/GHS/CW
- In Design:
  - MPC5561 OS 2.1/WRS/GHS/CW

- Last Proposed Revision:
  - MPC5634M/33M OS 3.0
  - MPC5633M OS 2.1/GHS
  - MPC5632R OS 3.0
  - MPC551x OS 2.1/GHS
  - MPC551x OS 3.0 or 3.1 (**)

- Available:
  - MPC5516 OS 2.1/GHS EAR
- Planned:
  - MPC5567 OS 2.1/WRS/GHS/CW
- Proposed:
  - MPC5565 OS 2.1/WRS/GHS/CW
- In Design:
  - MPC5561 OS 2.1/WRS/GHS/CW

- Last Proposed Revision:
  - MPC5634M/33M OS 3.0
  - MPC5633M OS 2.1/GHS
  - MPC5632R OS 3.0
  - MPC551x OS 2.1/GHS
  - MPC551x OS 3.0 or 3.1 (**)

2008
2009
2010
### 16-bit AUTOSAR MCAL Roadmap

- **Available**
- **In Design**
- **Planned**
- **Proposed**
- **Last Proposed Revision**

#### Roadmap Details:
- **.25um S12X are not supported**
- **.18um S12X are supported if min 256k**
- **.09um S12X will be supported if min 256k**

#### Product Line Options:
- **S12XE 1M MCAL 2.1 /CW**
  - S12XE 768 MCAL 2.1
  - S12XE 512 MCAL 2.1
  - S12XE 384 MCAL 2.1
  - S12XE 256 MCAL 2.1
- **S12XF 512 MCAL 2.1 /CW**
  - S12XF 384 MCAL 2.1
  - S12XF 256 MCAL 2.1
- **S12XS 256 MCAL 2.1 /CW**
  - S12XS 256 MCAL 3.0/3.1

#### Timeline:
- **2008**
- **2009**
- **2010**
16-bit AUTOSAR OS Roadmap

- .25um S12X are **not** supported
- .18um S12X are supported if **min 64k**
- .09um S12X will be supported if **min 64k**
Summary

► Freescale is investing into enablement of our automotive customers, beyond silicon delivery.

► AUTOSAR is a standardized automotive software architecture, globally gaining traction.

► As one example of customer enablement, Freescale develops AUTOSAR MCAL and OS for a wide range of 16- and 32-bit microcontrollers:

- **MPC5567/65/61**
  32-bit, Power Architecture e200z6 core
  - for high-end applications

- **MPC551x**
  32-bit, Power Architecture e200z1 + e200z0 cores
  - for body & gateway applications

- **MPC563x**
  32-bit, Power Architecture e200z3 core
  - for low-end powertrain applications

- **MPC560x**
  32-bit, Power Architecture e200z0 core
  - for body & instrument cluster applications

- **S12XE/F/S (min 256k)**
  16-bit, selected sub-families
  - for body applications
Now:

Live Demo of the AutoSAR configuration tool with the Freescale AutoSAR MCAL and OS software
## Related Session Resources

### Session Location – Online Literature Library

http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=052577903644CB

### Sessions

<table>
<thead>
<tr>
<th>Session ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Demos

<table>
<thead>
<tr>
<th>Pedestal ID</th>
<th>Demo Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2008.