Freescale Software for Automotive
Body, Chassis, Powertrain, Cluster
and Ethernet Applications

FTF-AUT-F0191

Dr. Andreas Both | Auto MCU Software Manager

A P R . 2 0 1 4
Session Introduction

• Increasing automotive system and software complexity requires a microcontroller supplier to also become a software supplier

• Freescale software products and services helps to offload your software teams workload

• Freescale‘s software products and services enable faster time to market and higher maturity products

• Presenter: Andreas Both
  Manager, Automotive Software
  Automotive Microcontrollers

• Length: 60 minutes
Session Objectives

• After completing this session you will be able to:
  - Name Freescale‘s major areas of software development
  - Understand Freescale‘s competence to develop production-level software for automotive
  - Describe where and how Freescale‘s software offering can help your ECU designs
Agenda

• Introduction and Agenda Review
• Software in the Automotive Industry
• Software Product Review
• Solutions Enabled by Freescale Software
• Session Review and Wrap-up
Software in the Automotive Industry
Software becomes a battleground for our customers

- Battle for innovation
- Battle for resources
- Battle for skills

Source: „Accelerating Automotive Software Delivery“, http://www.electric-cloud.com
Freescale provides software products where in-depth hardware knowledge is crucial – value-add software products such as AUTOSAR MCAL and OS, Core Self Test, application-specific libraries to address unique hardware features.
Freescale Automotive Software - Global Support

Detroit
Customer Support

Austin
SW Quality Management

Munich
SW Product Management
SW Architecture Team
Customer Support

Roznov
V&V Team
MCAL Development
Ethernet Development
Motor Control CoE
Customer Engineering

Bucharest
MCAL Development
OS Development
SW Quality Management
Customer Engineering

Paris
Customer Support

Tokyo
Customer Support

Shanghai
Customer Support
Auto SW Development Process Evolution

Enabling ISO 26262 compliancy

Products in development

- **2008**
  - Customer Assessed
  - Automotive SPICE Level 1

- **2009**
  - Customer Assessed
  - Automotive SPICE Level 2

- **2010**
  - Customer Assessed
  - Automotive SPICE Level 3

- **2011**
  - Automotive SPICE Level 3
  - ISO26262 Prototypes

- **2012+**
  - Automotive SPICE Level 3
  - ISO 26262

Products in development
Areas of Activities

- Autosar Operating System
- Autosar MCAL Drivers
- Core Self Test
- Device Self Test
- Math and Motor Control Lib
- Ethernet AVB
- Ethernet Camera Application
- Motor Control Toolbox (Matlab/Simulink)
- Init Tools
- Value Add Services
Freescale Support for AUTOSAR
AUTOSAR –
THE WORLDWIDE AUTOMOTIVE STANDARD FOR E/E SYSTEMS

AUTOSAR (AUTomotive Open System ARchitecture) develops a standardised open software architecture for automotive electronic control units. The partnership of automotive manufacturers, suppliers, and tool and semiconductor vendors started in 2003. Ten years later, AUTOSAR masters the growing complexity of automotive electric/electronic (E/E) architecture. It introduced and enabled new technologies, improved development efficiency and is going to be the architecture to satisfy future requirements.
AUTOSAR Basic Software

- **Freescale Standard Products** (shaded blue below) - MCAL (source code), OS (source code) and Config Tool (executable) for MCAL and OS.

- **Partner Products** (Elektrobit, Vector, KPIT, etc.) – The rest of AUTOSAR basic software as needed & Integration Services (FSL IP + Partner IP + Customer IP)

- **Complex Drivers** (shaded green below) – custom software offered by Freescale Consulting & Professional Engineering Services
AUTOSAR MCAL Product Content

- MCAL drivers for each MCU peripheral, compliant to Autosar 2.1 / 3.x / 4.0
- AUTOSAR 3.0/4.0 MCAL: no RamTst, CoreTst, FlashTst modules
- All components configurable in any AUTOSAR-compliant configuration tool
- Configuration Tool EB tresos Studio™ and plugins are part of the product
AUTOSAR Operating System

• Configurable in AUTOSAR 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.

• Availability of SC2,3,4 depends on MCU family / presence of MPU
EB tresos Studio

- EB tresos Studio is an easy-to-use tool for ECU standard software configuration, validation and code generation
- Full support for the AUTOSAR standard and the Freescale AUTOSAR software

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

Freescale’s AUTOSAR Integration Partners receive Freescale MCAL and OS releases for pre-integration into their proprietary Autosar BSW products.
# AUTOSAR MCAL Availability Matrix

<table>
<thead>
<tr>
<th>QM</th>
<th>ISO26262 ASIL A-D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASR 4.0.3</strong></td>
<td></td>
</tr>
<tr>
<td>MPC560xB/C/D</td>
<td>Calypso</td>
</tr>
<tr>
<td>MPC564xB/C</td>
<td>Panther</td>
</tr>
<tr>
<td>MPC564xA</td>
<td>McKinley</td>
</tr>
<tr>
<td>MPC564xL</td>
<td>Racerunner</td>
</tr>
<tr>
<td>MPC567xK</td>
<td>Materhorn</td>
</tr>
<tr>
<td>Calypso</td>
<td>Rainier</td>
</tr>
<tr>
<td>MPC567xK</td>
<td>Cobra55</td>
</tr>
<tr>
<td>VMbrid-M4</td>
<td></td>
</tr>
<tr>
<td>Halo-M4</td>
<td></td>
</tr>
<tr>
<td>Rayleigh-M4</td>
<td></td>
</tr>
<tr>
<td>Treerrunner-M4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ASR 3.0 / 3.1</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC560xB/C/D</td>
<td>Calypso</td>
</tr>
<tr>
<td>MPC564xB/C</td>
<td>Panther</td>
</tr>
<tr>
<td>MPC560P</td>
<td>McKinley</td>
</tr>
<tr>
<td>MPC564xL</td>
<td>Racerunner</td>
</tr>
<tr>
<td>MPC567xK</td>
<td>Materhorn</td>
</tr>
<tr>
<td>MPC567xK</td>
<td>Rainier</td>
</tr>
<tr>
<td>MPC567xF</td>
<td>Cobra55</td>
</tr>
</tbody>
</table>

| S12X                    | MPC5604E                 |
| S12P/G                  | MPC563xM                 |
| MPC5668                 |                          |

Also ASR 3.2 available

| S12X                    | MPC5604E                 |
| S12P/G                  | MPC563xM                 |
| MPC5668                 |                          |
# AUTOSAR OS Availability Matrix

<table>
<thead>
<tr>
<th>ASR 4.0.3</th>
<th>QM</th>
<th>ASR 3.0 / 3.1</th>
<th>QM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC560xB/C/D</td>
<td>Vybrid-M4</td>
<td>MPC560xB/C/D</td>
<td>Panther</td>
</tr>
<tr>
<td>MPC564xB/C</td>
<td>Halo-M4</td>
<td>MPC564xP</td>
<td>McKinley</td>
</tr>
<tr>
<td>MPC560xP</td>
<td>Rayleigh-M4</td>
<td>MPC564xL</td>
<td>Racerunner</td>
</tr>
<tr>
<td>MPC564xL</td>
<td>Trerunner-M4</td>
<td>MPC567xK</td>
<td>Matterhorn</td>
</tr>
<tr>
<td>Calypso</td>
<td></td>
<td>MPC567xK</td>
<td>Rainier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPC567xF</td>
<td>Cobra55</td>
</tr>
</tbody>
</table>

Available: Pixelated
In Development / Planned: Yellow
Supported (or Planned) through Autosar BSW Vendors: Gray

<table>
<thead>
<tr>
<th>MPC560xB/C/D</th>
<th>MPC5604E</th>
<th>MPC563xM</th>
</tr>
</thead>
<tbody>
<tr>
<td>S12X</td>
<td>MPC564xB/C</td>
<td>MPC560xP</td>
</tr>
<tr>
<td>S12P/G</td>
<td>MPC564xL</td>
<td>MPC567xK</td>
</tr>
<tr>
<td>MPC5668</td>
<td>MPC567xF</td>
<td></td>
</tr>
</tbody>
</table>

Supported through Autosar BSW Vendors:

- Panther
- McKinley
- Racerunner
- Matterhorn
- Rainier
- Cobra55
EB Tresos Safety OS for ISO 26262

- AUTOSAR OS with protected Microkernel
- AUTOSAR 3 and 4 compatible
- Up to ASIL-D
- Already in project use

Safety Features

Data Protection
- Write protection between OS applications

Stack Protection
- Only corresponding context (Task, ISR, kernel,…) has write access
- Stack overflow/underflow prevention

Context Protection
- Interrupted context is stored in protected memory area

OS Protection
- System Call interface separates application and kernel
- Private stack for the OS kernel

Hardware Error Protection
- OS provides exception handling including reporting to application

Source: Elektrobit, 2012
Freescale Software for Control of Electrical Motors
Automotive Math and Motor Control Library Set

- Set of basic trigonometric, general math, filter & motor control functions
- Customer Algorithms can be developed independent of target platform peripherals
- Enable easy migration between platforms with minimized effort
# Auto Math and Motor Control Library Set – Contents

<table>
<thead>
<tr>
<th>Target Platform</th>
<th>CodeWarrior</th>
<th>GreenHills Multi</th>
<th>WindRiver Diab</th>
<th>Cosmic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qorivva</td>
<td>RTM Rev 1.0</td>
<td>RTM Rev 1.0</td>
<td>RTM Rev 1.0</td>
<td>RTM Rev 1.0</td>
</tr>
<tr>
<td>S12ZVM</td>
<td>RTM Rev 1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MLIB
- Absolute Value, Negative Value
  - MLIB_Abs, MLIB_AbsSat
  - MLIB_Neg, MLIB_NegSat
- Add/Subtract Functions
  - MLIB_Add, MLIB_AddSat
  - MLIB_Sub, MLIB_SubSat
- Multiply/Divide/Add-Subtract Functions
  - MLIB_Mul, MLIB_MulSat
  - MLIB_Div, MLIB_DivSat
  - MLIB_Mac, MLIB_MacSat
  - MLIB_VMac
- Shifting
  - MLIB_ShL, MLIB_ShLSat
  - MLIB_ShR
  - MLIB_ShBi, MLIB_ShBiSat
- Normalisation, Round Functions
  - MLIB_Norm, MLIB_Round
- Conversion Functions
  - MLIB_ConvertPU, MLIB_Convert

### GFLIB
- Trigonometric Functions
  - GFLIB_Sin, GFLIB_Cos, GFLIB_Tan
  - GFLIB_Asin, GFLIB_Acos, GFLIB_Atan, GFLIB_AtanXY
- Limitation Functions
  - GFLIB_Limit, GFLIB_VectorLimit
  - GFLIB_LowerLimit, GFLIB_UpperLimit
- PI Controller Functions
  - GFLIB_ControllerPr, GFLIB_ControllerPrAW
  - GFLIB_ControllerPrp, GFLIB_ControllerPrpAW
- Interpolation
  - GFLIB_Lut1D, GFLIB_Lut2D
- Hysteresis Function
  - GFLIB_Hyst
- Signal Integration Function
  - GFLIB_IntegratorTR
- Sign Function
  - GFLIB_Sign
- Signal Ramp Function
  - GFLIB_Ramp
- Square Root Function
  - GFLIB_Sqrt
- Finite Impulse Filter
  - GDFLIB_FilterFIR
- Moving Average Filter
  - GDFLIB_FilterMA
- 1st Order Infinite Impulse Filter
  - GDFLIB_Filter1Init
  - GDFLIB_Filter1
- 2nd Order Infinite Impulse Filter
  - GDFLIB_Filter2Init
  - GDFLIB_Filter2

### GDFLIB
- Clark Transformation
  - GMCLIB_Clark
  - GMCLIB_ClarkInv
- Park Transformation
  - GMCLIB_Park
  - GMCLIB_ParkInv
- Duty Cycle Calculation
  - GMCLIB_SymStd
- Elimination of DC Ripples
  - GMCLIB_ElimDcBusRip
- Decoupling of PMSM Motors
  - GMCLIB_DecouplingPMSM
- Angle Tracking Observer
  - Tracking Observer
- PMSM BEMF Observer in Alpha/Beta
- PMSM BEMF Observer in D/Q
- Content To Be Defined

### Delivery Content
- Matlab/Simulink Bit Accurate Models
- User Manuals
- Header files
- Compiled Library File
- License File (to be accepted at install time)
Model Based Design Support with MC-Toolbox

- Model Based Design is very common in the Motor control development
- Many companies model their controller algorithm and the target motor and use the simulation environment to accelerate their algorithm development.
- Reduced time to market and higher product quality by having an executable that can be simulated and implemented directly from the model.
- The Motor Control Development Toolbox includes Simulink™ plug-in libraries for an embedded Freescale target
- An integrated environment and tool chain supports configuring and generating the necessary software, including initialization routines, device drivers, and a real-time scheduler to execute algorithms specifically for controlling motors.
- For the final stage of this type of the integration of the control algorithm software with target MCU hardware motor Control Development Toolbox autogenerates 100% of the code from the model.

<table>
<thead>
<tr>
<th>MCUs Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC5643L/7xK</td>
</tr>
<tr>
<td>MPC574xP</td>
</tr>
<tr>
<td>S12ZVM</td>
</tr>
<tr>
<td>KV10Z</td>
</tr>
</tbody>
</table>
First Production Gateway with Ethernet Diagnostic – SOP 2008 (Freescale MPC5567)
One Pair Ethernet : Emerging Auto Standard

• OPEN Alliance is designed to encourage wide scale adoption of **Ethernet-based, single pair unshielded networks** as the standard in automotive applications. The jointly developed OPEN Alliance SIG will address industry requirements for improving in-vehicle safety, comfort, and infotainment, while significantly reducing network complexity and cabling costs.

**OPEN Alliance**
Promoting Members
• Broadcom (founding member)
• BMW (founding member)
• Bosch
• Continental
• Freescale (founding member)
• Harman
• Hyundai
• Jaguar-Landrover
• NXP (founding member)
• Renault
+ >30 registered Adopters
First Production Ethernet Surround Camera System – SOP 2013 (MPC5604E + Ethernet Camera Software)
Freescale Software Solution for Ethernet Surround Camera Nodes using MPC5604E / MPC5606E

- Purple: AUTOSAR OS 3.0, licensable from Freescale
- Green: Camera Application Software, licensable from Freescale
- Blue: Ethernet Streaming Software, licensable from Freescale
- Orange: Off the shelf Software, licensable from 3rd party (Elektrobit, IXXAT)
- Red: Hardware
Freescale Software Solution for Ethernet Surround Camera Nodes using MPC5604E / MPC5606E

**Ethernet Streaming Software**
- FEC Low Level Driver
  - IEEE802.3 (Ethernet)
  - IEEE802.1Q (VLAN)
- 1722 Stream Builder
  - IEEE1722 (AVB)
- UDP Stream Builder
  - RFC791(IP), RFC768(UDP)
  - RFC826 (ARP), required parts only
- Interface to TCP/IP Stack from Elektrobit
- Interface to PTP Stack from IXXAT

**Camera Application Software**
- Camera Application Data rate control
- Buffer overflow protection
- Stream Builder Interface support for UDP
- support for IEEE1722 (AVB)
- AUTOSAR OS Interface
- Video Encoder Driver
- I\(^2\)C Driver
- Imager Configuration Interface
- Imager Control Driver

**AUTOSAR OS**
- Operating System (OS) compliant to AUTOSAR 3.0 scalability class 1
- EB tresos Studio™ Configuration tool
Ethernet AVB Standard

- **Audio Video Bridging** (AVB) is a common name for the set of technical standards developed by the IEEE Audio Video Bridging Task Group of the IEEE 802.1 standards committee. The charter of this organization is to "provide the specifications that will allow time-synchronized low latency streaming services through IEEE 802 networks". These consist of:
  - **IEEE 802.1AS**: Timing and Synchronization for Time-Sensitive Applications (gPTP)
  - **IEEE 802.1Qat**: Stream Reservation Protocol (SRP)
  - **IEEE 802.1Qav**: Forwarding and Queuing for Time-Sensitive Streams (FQTSS)
  - **IEEE 802.1BA**: Audio Video Bridging Systems
Ethernet AVB for Automotive Sub Standard

- **Specifications absolutely required:**
  - IEEE 1722 Layer 2 Transport Protocol
  - IEEE 802.1AS Timing and Synchronization

- **Further specifications that can either be optional or will not be used:**
  - **IEEE 802.1Qat:** Stream Reservation Protocol (SRP)
    - Used to dynamically reserve bandwidth for specific streams
    - Distinguishes between class A, class B, class C and best effort traffic
    - Actual reservation is really implemented in the switch, endpoints just need to understand the protocol to initiate the reservation (pure software on the endpoint)
    - For automotive applications dynamic service discovery is prohibitive because of startup time
      - **Option I:** Static switch configuration
      - **Option II:** SRP with preconfigured startup
  - **IEEE 802.1Qav:** Forwarding and Queuing for Time-Sensitive Streams (FQTSS)
  - **IEEE 802.1BA:** Audio Video Bridging Systems
    - Currently work on automotive profile is ongoing
  - **IEEE 1722.1:** Device Discovery, Enumeration, Connection Management and Control Protocol (DECC)
    - For automotive applications this protocol is unlikely to be used due to the startup time required
Freescale Automotive Ethernet AVB

• Freescale‘s automotive Ethernet AVB software solution implements the following standards:
  - IEEE 802.3 - in the Ethernet AV Streaming Driver
  - IEEE 1722/1722a - in the 1722/1722a Stream Builder module
  - IEEE 802.1Qav - partly in the AVB Shaper module

• Freescale‘s automotive Ethernet AVB software solution interfaces to 3rd party or open source implementations of these standards:
  - IEEE 802.1AS - in the ptp interface module, for either
    ▪ ptpv2d stack (licensable under GPLv2, http://code.google.com/p/ptpv2d)
    ▪ IXXAT gPTP stack (commercially licensable from http://www.ixxat.de)
Automotive Ethernet AVB Solution for Audio Amplifiers
Example: AVB on Vybrid M4

Freescale Autosar/OSEK OS

- PTP Stack 802.1AS timing and synchronize
- IEEE 1722a Transport Protocol
- Audio Framework
- Config Control Monitoring

Freescale Autosar/OSEK OS

- Ethernet PHY Driver (BroadReach)
- EAVB Ethernet Driver
- Non-AV Ethernet
- Media Clock Recovery (and frame sync generation)
- Audio Driver interface to ASRC and TDM/SAI
- Multi Core Communication

- 1588 / 802.1AS Ethernet Hardware (counter / time stamp)
- Interface to external PLL
- ASRC and TDM/SAI
- DMA, Shared Memory, semaphores

ARM Cortex M4

3rd Party RTOS / Autosar OS / RTE

- Configuration, Control, and Monitoring of AVB stack
- Application

- Multi Core Communication
- Virtual (MCAL) Driver for non-AV Ethernet traffic
- Autosar BSW (if required)

ARM Cortex A5

- DMA, Shared Memory, Semaphores
- Other non-AV related hardware

Hardware

Freescale Ethernet AV Streaming Driver Package (AVB)
Freescale Ethernet Audio Interface Package
Freescale Software, available off the shelf
Open Source or 3rd Party AVB Components
Tier1 software
Freescale Software for Compliance to ISO26262
What is a Software-Based Core Self Test (CST)

- It is a set of assembly instructions (software), which:
  - are executed by a core being under test
  - with the purpose to activate potentially faulty location within the core, and
  - in case of an error (*fault causes correct data become erroneous*) signal an error to the outside in a way not to be masked with the same fault.
### Types of CST Packages

<table>
<thead>
<tr>
<th>FCST</th>
<th>ICST</th>
<th>SCST</th>
</tr>
</thead>
<tbody>
<tr>
<td>CST with <strong>fault coverage</strong> metric</td>
<td>CST with <strong>instruction coverage</strong> metric</td>
<td><strong>Structural</strong> CST</td>
</tr>
<tr>
<td>• “Stuck-at” fault coverage is measured by using <strong>fault simulation method</strong> (also called <strong>fault-grading</strong>) and provided to the customers</td>
<td>• Customer gets a check-list with covered and not covered instructions</td>
<td>• Fault coverage analytically estimated (no fault-grading)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Instruction coverage is provided in addition</td>
</tr>
</tbody>
</table>
Types of CST Packages: What is Available

### Freescale Core Self Test Packages

<table>
<thead>
<tr>
<th>FCST</th>
<th>ICST</th>
<th>SCST</th>
</tr>
</thead>
<tbody>
<tr>
<td>• MPC560xP/B (z0h)</td>
<td>• Executed 95 – 98% of all instructions</td>
<td>• &gt;90% DC coverage (analytically estimated)</td>
</tr>
<tr>
<td>60% &amp; 80% stuck-at fault coverage <em>(EAR)</em></td>
<td><em>(this is valid mainly for older cores/devices, instructions like SPE2, decorated storage, etc. are not covered)</em></td>
<td>• 95 – 98% of all instructions tested</td>
</tr>
<tr>
<td>• S12Z (MagniV) 60% stuck-at fault coverage <em>(Beta)</em></td>
<td>• Executable on most of Power e200 cores <em>(RTMC)</em></td>
<td>• MPC577xK (z7) <em>(EAR)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MPC574xG/B (z2, z4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ARM based MCUs (M4, A5)</td>
</tr>
</tbody>
</table>

**Legend:**
- **Green** – available as EAR / Beta / RTMC release
- **Orange** – planned for development in 2014/15
CST: Operation Principles

• **Application**
  - Triggers test execution
  - Selects subset of tests to perform
  - Checks actual versus expected result
  - Should expect abnormal behavior

• **Self Test API**
  - Calls atomic tests
  - Checks result of every executed test
  - Compresses atomic test results into one 32-bit signature

• **Atomic test**
  - Short piece of assembly instructions grouped together according to some principles
  - Optimized to activate a fault and propagate error signaling
Summary and Wrap-up
Session Summary

• Freescale applies Automotive Spice Level 3 compliant processes to deliver software products, services and solutions for production use.

• Freescale’s automotive software portfolio includes AUTOSAR MCAL and OS, Math and Motor Control Libraries and Tools, Self Test software for support of ISO26262, and software stacks for Automotive AVB over Ethernet.

• Freescale’s software enables our customers to build scalable platforms for automotive body, powertrain, safety and chassis, and driver information systems.
Making the World a Smarter Place.