Renesas Technology & Solution Portfolio
**Microcontroller and Microprocessor Line-up**

<table>
<thead>
<tr>
<th>2010</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1200 DMIPS, Superscalar</strong></td>
<td><strong>1200 DMIPS, Performance</strong></td>
</tr>
<tr>
<td>- Automotive &amp; Industrial, 65nm</td>
<td>- Automotive, 40nm</td>
</tr>
<tr>
<td>600μA/MHz, 1.5μA standby</td>
<td>500μA/MHz, 35μA deep standby</td>
</tr>
<tr>
<td><strong>500 DMIPS, Low Power</strong></td>
<td><strong>165 DMIPS, FPU, DSC</strong></td>
</tr>
<tr>
<td>- Automotive &amp; Industrial, 90nm</td>
<td>- Industrial, 40nm</td>
</tr>
<tr>
<td>600μA/MHz, 1.5μA standby</td>
<td>200μA/MHz, 0.3μA deep standby</td>
</tr>
<tr>
<td><strong>165 DMIPS, FPU, DSC</strong></td>
<td><strong>Embedded Security, ASSP</strong></td>
</tr>
<tr>
<td>- Industrial, 90nm</td>
<td>- Industrial, 90nm</td>
</tr>
<tr>
<td>200μA/MHz, 1.6μA deep standby</td>
<td>1mA/MHz, 100μA standby</td>
</tr>
<tr>
<td><strong>25 DMIPS, Low Power</strong></td>
<td><strong>44 DMIPS, True Low Power</strong></td>
</tr>
<tr>
<td>- Industrial &amp; Automotive, 150nm</td>
<td>- Industrial &amp; Automotive, 130nm</td>
</tr>
<tr>
<td>190μA/MHz, 0.3μA standby</td>
<td>144μA/MHz, 0.2μA standby</td>
</tr>
<tr>
<td><strong>10 DMIPS, Capacitive Touch</strong></td>
<td></td>
</tr>
<tr>
<td>- Industrial &amp; Automotive, 130nm</td>
<td></td>
</tr>
<tr>
<td>350μA/MHz, 1μA standby</td>
<td></td>
</tr>
</tbody>
</table>

*Note: All devices are low power with specified current consumption and standby power.*
‘Enabling The Smart Society’

- **Challenge:**
  “Automotive electronic complexity is increasing exponentially. As cars become smarter with more feature content, as well as new drive-train technology and safety systems, development requires smarter tools and methods.”

- **Solution:**
  “A solution is to standardize software design processes, tools and software. Renesas has a long history of involvement, and offers a rich portfolio of solutions to facilitate this effort. This class will introduce the concepts, processes, and challenges of implementing AUTOSAR.”
Agenda

- Basics- Who is AUTOSAR?
- Basics- What’s the problem?
- Basics- When is AUTOSAR coming?
- Basics- What is defined?
- Basics- How does it work?
- ECU Configuration
- Challenges
- Wrap-up
- Takeaways
Basics- Who is AUTOSAR?
AUTomotive Open System ARchitecture
“Cooperate on standards, compete on implementation”

- Initial discussions in 2002
- Official since 2003
- Started with 5 German companies
- NEC Electronics Corp. and Renesas Technology Corp. joined the effort in 2004
- Approx. 175 Partners now (still growing)
- www.autosar.org
AUTOSAR – Partnership Structure

Core Partner (OEM & Tier 1 Supplier)
- Organizational control
- Technical contributions
- Administrative control
- Definition of external information (web release, clearance, etc.)
- Leadership of working groups
- Involvement in working groups

Premium Members (incl. Tool Manufacturers)
- Leadership of working groups
- Involvement in working groups
- Technical contributions
- Access to current information

Associate Members
- Access to finalized documents
- Utilization of standards
AUTOSAR – Partnership Structure

Core Partners

Premium Members

Associate Members

Growing Community
Please find updated info on www.autosar.org
Basics- What’s the Problem?
Example - Automotive Integrated Center Stack
Example - Automotive Integrated Center Stack
How much code?

1.7M

5.7M

10M

100M

Source: http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code/0
Reasons for the effort

- Rising Automotive electronic complexity
  - Quantity of software increasing
  - ECU counts increasing

- Large number of disparate processors used
  - Software portability limited
  - High effort for reuse of software features

- Customized solutions increase:
  - Maintenance cost
  - Testing effort
  - OEM integration effort
  - Risk
Overall Objectives of AUTOSAR

- Standardization
  - Workflows, software interfaces, tools

- Increase modularity and transferability of features
  - Process to manage feature allocation
  - Clear division of hardware dependent and standard software versus the higher level features

- Facilitate collaboration
  - Draw on experience from domain experts

- Increase dependability and quality
  - Reuse standard solutions among Tier 1’s and across OEMs

- Reduce effort/time to market
Basics - What is Defined?
System Configuration

VFB view

Tool supporting deployment of SW components

MAPPING

Source: http://www.autosar.org
ECU Configuration

Source: http://www.autosar.org
Basics- When is AUTOSAR Coming?
AUTOSAR is Here

Volume of ECUs with AUTOSAR

Source: http://www.autosar.org
R4.0 Additions

- Multi Core
  - Main impact will be on OS
  - System impact (power saving, memory sharing)
  - Software reuse from low end to high end

- Cryptography
  - Hardware accelerations (ICU, SHE)
  - AUTOSAR software libraries

- Functional safety
  - Development processes
  - Safety-related features (Core test, RAM test, ROM test, etc.)

- Ethernet
  - Legal requirement for OBD-II
  - Investigation to use Ethernet as network backbone
Basics- How does it Work?
3rd Party Software

- Goal is to create a complete AUTOSAR solution

- Renesas supplies MicroController Abstraction Layer (MCAL – hardware-dependent software) drivers for standard peripherals and communication interfaces.
MCAL Components

Microcontroller Drivers
- GPT Driver
- Watchdog Driver
- MCU Driver

Memory Drivers
- FLS Driver
- FEE Driver

Communication Drivers
- SPI Driver
- LIN Driver
- CAN Driver
- FlexRay Driver
- Ethernet Driver

I/O Drivers
- ICU Driver
- PWM Driver
- ADC Driver
- DIO Driver
- PORT Driver

Source: http://www.autosar.org
### MCAL Development Roadmap

<table>
<thead>
<tr>
<th>Autosar Release</th>
<th>Body</th>
<th>Safety / Chassis</th>
<th>Powertrain</th>
<th>Airbag</th>
<th>Instrumentation</th>
<th>ADAS</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AS4.0</strong></td>
<td>RH850F1X</td>
<td>RH850/P1X</td>
<td>RH850/E1X</td>
<td>RH850/R1X</td>
<td>RH850/D1X</td>
<td>RH850/V1X</td>
<td>2013</td>
</tr>
<tr>
<td></td>
<td>V850/Fx4L</td>
<td>V850/Px4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2012</td>
</tr>
<tr>
<td></td>
<td>V850/Fx4</td>
<td></td>
<td></td>
<td></td>
<td>V850/Dx4</td>
<td></td>
<td>2012</td>
</tr>
<tr>
<td><strong>AS 3.x</strong></td>
<td>RH850F1X</td>
<td>RH850/P1X</td>
<td>RH850/E1X</td>
<td>RH850/R1X</td>
<td>RH850/D1X</td>
<td>RH850/V1X</td>
<td>2013</td>
</tr>
<tr>
<td></td>
<td>V850/Fx4L</td>
<td>V850/Px4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>available</td>
</tr>
<tr>
<td></td>
<td>V850/Fx4</td>
<td>SH2A</td>
<td></td>
<td></td>
<td>V850/Dx4</td>
<td></td>
<td>available</td>
</tr>
<tr>
<td></td>
<td>V850/Fx3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>available</td>
</tr>
<tr>
<td></td>
<td>R32C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>available</td>
</tr>
<tr>
<td><strong>AS 2.1</strong></td>
<td>V850/Fx4</td>
<td>V850/Px4</td>
<td>SH2A</td>
<td></td>
<td></td>
<td></td>
<td>available</td>
</tr>
<tr>
<td></td>
<td>V850/Fx3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>available</td>
</tr>
<tr>
<td></td>
<td>R32C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>available</td>
</tr>
</tbody>
</table>
3rd Party Software

- What about non-standard peripherals?

- Renesas and partner companies can supply complex device drivers.
3rd Party Software

- Tier 1 and/or 3rd party software vendor(s) contribute the Basic Software (BSW - hardware-independent software)

- Tier 1 or 3rd party software vendor(s) contribute the RunTime Environment (RTE – top-level abstraction software)
3rd Party Software

- Integration is a joint activity with Renesas and the 3rd party vendor
  - Joint project planning
  - Issue tracking tools
  - Regular meetings

- Renesas is open to working with any BSW provider.
ECU Configuration
Software Design Process

application idea → HW User Manual → SW implementation → HW/SW integration
Sw development flow will change: Configuration Tool replaces hand-coded configuration

Integration of top-level application & low-level software

HW/SW integration
XML as an exchange format

- Input and output formats are **standardized** by AUTOSAR
- This ensures that **any** AUTOSAR compatible configuration editor can be used
“ECU Spectrum” Editor Tool

- A simple configuration editor is included in the Renesas MCAL package free of charge

- Generic tool for “quick start” and testing that offers:
  - AUTOSAR compatible xml file read / write
  - Basic validation checks
  - Read-back of existing configuration descriptions
ECU Configuration (Overview)

This is a precise description with information about:
- Number of instances (e.g. channel no.)
- Parameter value definitions

The Generator translates the configuration information into source code that is then used by the driver module.
Generators

- Implementation specific tool to generate code that contains the configuration data from the AUTOSAR configuration description file(s)
- Renesas delivers Generators as **command line** executables (one for each software module)
  - Command line interface that take the ECU configuration description file as **input**
  - **Generate** .h files (for pre-compile configuration) and .c files (for link-time and post-build configuration)
  - Are written in **perl**
- Renesas Generators provide **plug-in capability** for configuration editors and can be used in **makefile** environment
Challenges
Questions

What are some possible downsides to hardware abstraction?

What are some possible downsides to commonization and standardization?
Layered approach provides great flexibility, but

- it increases configuration complexity and the number of chances to "get it wrong"
  - Tools have to balance ease of use against the restriction of this flexibility
Challenges – Standardization

- **Standard API**
  - Designed for the “Least Common Denominator”
  - Decreases the advantage of innovation
    - Non-standard features may not be available
    - Special features must be either
      - made transparent to BSW by MCAL, or
      - handled outside of MCAL by “complex device drivers”

- **Software Supply Chain**
  - BSW from multiple vendors must work together
  - Integration and runtime issue resolution requires cooperation, potentially among competitors
Challenges – Commonization

- Specifications still under development
  - Released versions available, but not all changes/updates are backward-compatible, even within minor revisions

- OEMs adopt at different times and for different reasons.
  - Support of multiple releases necessary to support legacy and future development

- Historically, OEMs have different interpretations or desired features which are not agreed upon
  - OEM-specific AUTOSAR implementations increase complexity
  - More is standardized in later revisions to avoid this.
Wrap-Up
Pitfalls to Avoid - Possible Misconceptions

- “AUTOSAR-compliance is a precise concept”
  - Full process
  - Black box behavior
  - Everything in-between

- “Common API” means the processor no longer matters
  - Analogy – Windows and x86 -> Intel’s HTT
  - Processor architectures, instruction sets, pipelines, and special peripherals and features can make all the difference.

- “Common API” means “cheaper and faster to market”
  - True, but only after reuse is factored in. Change costs money.

- “Standard” means “Easy”
Optimizations

- Herstellerinitiative Software (HIS) recommended optimization AUTOSAR
  - Subset specification
  - Allows application to 16-bit and smaller/less powerful 32-bit microcontrollers

- Similar initiative from JASPAR

- Black box AUTOSAR-compliance

- Complex Device Drivers
  - Implement leaner hardware access for time-critical features
  - Control unspecified hardware peripherals (e.g. RDC, EMU)
Takeaways

- Implement the subset that makes sense
  - AUTOSAR is not an end in itself
  - “Don’t sacrifice usability for the sake of reusability.”

- Get help from the experts
  - Save money in the long run by getting it right from the start
  - Reap the benefits of lessons learned by others
Questions?
‘Enabling The Smart Society’ in Review...

■ **Challenge:**
  “Automotive electronic complexity is increasing exponentially. As cars become smarter with more feature content, as well as new drive-train technology and safety systems, development requires smarter tools and methods.”

■ “*One solution is to standardize software design processes, tools and software. Renesas has a long history of involvement, and offers a rich portfolio of solutions to facilitate this effort. This class will introduce the concepts, processes, and challenges of implementing AUTOSAR.*”

■ Do you agree that we accomplished the above statement?