Safe and Seamless Integration of Tegra into the In-Vehicle Network

Stefaan Sonck Thiebaut, OpenSynergy
OpenSynergy is a global provider of software solutions for embedded automotive systems.

OpenSynergy’s core product portfolio consists of the key software components necessary to create efficient automotive solutions in the areas of:

- Infotainment
- Connectivity
- Driver Information
- Driver Assistance
# OpenSynergy Product Portfolio

## Products

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COQOS SDK</td>
<td>• The standards-based COQOS software platform is the basic technology underpinning all OpenSynergy solutions</td>
</tr>
<tr>
<td>Blue SDK</td>
<td>• Leading independent Bluetooth stack</td>
</tr>
<tr>
<td>Voice SDK</td>
<td>• Voice band audio processing</td>
</tr>
<tr>
<td>Update SDK</td>
<td>• Versatile software update mechanism for embedded devices</td>
</tr>
</tbody>
</table>

## Engineering

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car2Infrastructure</td>
<td>• Model-based telematics stack</td>
</tr>
<tr>
<td></td>
<td>• Backend system simulation</td>
</tr>
<tr>
<td></td>
<td>• End-to-end security solutions</td>
</tr>
<tr>
<td>Development and</td>
<td>• AUTOSAR software systems</td>
</tr>
<tr>
<td>Integration</td>
<td>• Architecture and software design</td>
</tr>
<tr>
<td></td>
<td>• Implementation and integration</td>
</tr>
<tr>
<td></td>
<td>• Board support packages</td>
</tr>
</tbody>
</table>

## Support

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Standard support for OpenSynergy’s products</td>
</tr>
<tr>
<td></td>
<td>• Access to product updates</td>
</tr>
</tbody>
</table>
Agenda

• Background on automotive trends & challenges

• Background on AUTOSAR

• Targets of the project

• Architecture

• Process & tools

• Demonstration

• Summary
Automotive Trends & Challenges

Convergence

Connected Car

Safety & Security

Standardization

Towards autonomous driving
Possible Software Architecture of Future Automotive ECUs

- **Infotainment Applications**
  - Android
  - GENIVI

- **OEM-controlled Applications, e.g. Instrument Cluster**
  - GENIVI
  - QNX

- **Driver Assistance System**
  - Shared drivers

- **Resource Management**

- **Real-time Applications**

- **Virtualization / Hypervisor**

- **Hardware**
  - Typically: Multi-Core ARM CORTEX with powerful GPU

- **Other Vehicle Systems**

- **SW Update**
  - System State
Agenda

• Background on automotive trends & challenges

• Background on AUTOSAR

• Targets of the project

• Architecture

• Process & tools

• Demonstration

• Summary
Development Partnership „AUTOSAR“
(www.autosar.org)

Defines the software architecture and development methodology for automotive ECUs.

Up-to-date status see: http://www.autosar.org

AUTOSAR Partners (March 2014)
AUTOSAR Guided Tour (Part 1)
Motivation and Principles
2013-08

from “Global Automotive E/E Standard”, Rick Flores, General Motors, presented at Open Architecture Summit, Nov 4, 2014
1. software architecture of automotive devices

2. methodology to configure automotive devices

3. application interfaces.
AUTOSAR Software Architecture
System per ECU

- Component API Generator
- Component API e.g. app.h
- SW-C Implementation
- AUTOSAR RTE Generator
- OS, COM, ... Generator
- Other Basic SW Generator
- MCAL-Generator

Information/Database (no files)
Generation step: complex algorithm or engineering work

Agenda

• Background on automotive trends & challenges

• Background on AUTOSAR

• Targets of the project

• Architecture

• Process & tools

• Demonstration

• Summary
Targets of the project

• Facilitate integration of Tegra into the in-vehicle CAN network (through MCU)

• Allow use of AUTOSAR methodology to configure the integration in the vehicle bus

• Control information flow between non-AUTOSAR and AUTOSAR partitions

• Integrate the configuration & build process into the Vibrante SDK

• Make it possible to run AUTOSAR Software-Components (applications) on Tegra

• Allow the integration of OEM-specific AUTOSAR variants on Tegra

• Take advantage of virtualization/hypervisor technology
Agenda

- Background on automotive trends & challenges
- Background on AUTOSAR
- Targets of the project
  - Architecture
- Process & tools
- Demonstration
- Summary
High-Level Concept

- ACF = Automotive Communication Framework
- BSW = Basic-Software (AUTOSAR Infrastructure Software)
- SW-C = Software Component (AUTOSAR Application Software)
- MCAL = Microcontroller Abstraction Layer (Drivers)
- OS = Operating System
Use Case 1: Communication to Vehicle Bus

SoC

AUTOSAR

Hypervisor

NVIDIA Tegra K1

MCU

- CAN Communication
- Additional I/O
- Power Management

Nano AUTOSAR Framework

Inter-Processor Communication

CAN Gateway

MCAL

OS

ACF

Application

• CAN Communication
• Additional I/O
• Power Management
Use Case 2: Integration of OEM Applications

- AUTOSAR SW-C
- AUTOSAR BSW with OEM Specifics
- MCAL
- RTE
- OS
- Application
- Hypervisor
- NVIDIA Tegra K1

MCU

- CAN communication
- Additional I/O
- Power Management

Nano AUTOSAR Framework

Inter-Processor Communication

CAN

I/O
Agenda

• Background on automotive trends & challenges

• Background on AUTOSAR

• Targets of the project

• Architecture

• Process & tools

• Demonstration

• Summary
Overall Process

1. **Configure**
   - AUTOSAR System Information

2. **Configuration**
   - Generate Code

3. **Build**
   - Flash
   - Other Applications
   - Other Vibrante SDK Elements

4. **Run**
   - TEGRA image
   - MCU image
Overall Process

- AUTOSAR System Information

  - Configure
  - Configuration
  - Generate Code

  - Other Applications
  - Other Vibrante SDK Elements

  - Build

  - MCU image

  - TEGRA image

  - Flash

  - Run
Configuration and Code Generation

Configure CAN
Configure OS
Configure ACF
Configure CAN Gateway

AUTOSAR Editor

ARXML

ECU Configuration

Generate CAN
Generate OS
Generate ACF
Generate CAN Gateway

OpenSynergy Code Generators

C
CAN Code (SoC and MCU)
OS Code (SoC)
ACF Code (AUTOSAR and Linux)
CAN Gateway Code
Examples of configuration items

- **Configure CAN**
  - CAN system (including MCU)
    - CAN speed
    - hardware filtering

- **Configure OS**
  - Scheduling behavior on the TEGRA

- **Configure ACF**
  - Availability of signals in other partitions (like Linux)
    - Conversion of CAN frames to/from logical signals
    - Timing behavior of outgoing CAN frames

- **Configure CAN Gateway**
  - CAN Gateway Code

OpenSynergy Code Generators

- CAN Code (SoC and MCU)
- OS Code (SoC)
- ACF Code (AUTOSAR and Linux)
- CAN Gateway Code
Agenda

• Background on automotive trends & challenges
• Background on AUTOSAR
• Targets of the project
• Architecture
• Process & Tools
• Demonstration
• Summary
Development Scenario

Example AR ARXML

Add new CAN MSG and Signal

Updated Configurations

Generate (command Line)

C/H Files

Build

Flash

Image

MOVIE
Agenda

• Background on automotive trends & challenges

• Background on AUTOSAR

• Targets of the project

• Architecture

• Process & tools

• Demonstration

• Summary
Summary

• Future versions of Vibrante SDK will include support for AUTOSAR and CAN-integration.

• Solution consists of:
  • MCU running nano-AUTOSAR handling CAN, IO, power management
  • AUTOSAR partition on Tegra handling CAN processing, AUTOSAR basic-software functionality and applications
  • Automotive Communication Framework for communication to non-AUTOSAR operating systems
  • Tooling integrated in Vibrante.
Contact

OpenSynergy GmbH
Rotherstraße 20
D-10245 Berlin
Germany
Phone +49 30 60 98 54 0-0
E-Mail info@opensynergy.com
Web www.opensynergy.com

OpenSynergy GmbH
Starnberger Str. 22
D-82131 Gauting / Munich
Germany
Phone + 49 89 8934 13-33
E-Mail bluetooth@opensynergy.com

OpenSynergy, Inc. (USA)
765 East 340 South
Suite 106
American Fork, Utah 84003
Phone +1 619 96 21 725
E-Mail bluetooth@opensynergy.com

OpenSynergy, COQOS SDK, Blue SDK, IrDA SDK, Voice SDK, Update SDK, Qonformat, and other OpenSynergy products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of OpenSynergy GmbH in Germany and in other countries around the world. All other product and service names in this document are the trademarks of their respective companies. These materials are subject to change without notice. These materials are provided by OpenSynergy GmbH for informational purposes only, without representation or warranty of any kind and OpenSynergy GmbH shall not be liable for errors or omissions with respect to the materials. © OpenSynergy GmbH 2015