Faster Development of AUTOSAR compliant ECUs through simulation

Andreas Junghanns, Jakob Mauss
QTronic GmbH

Michael Seibt
Dassault Systèmes Deutschland GmbH

ERTS 2014, Toulouse, France
February 5 - 7, 2014
Outline of the talk

Faster Development of AUTOSAR compliant ECUs through simulation

1. SW development process
2. some drawbacks
3. virtual ECUs
4. the AUTOSAR case
5. application examples
6. conclusion
Software Development Process

Function / Module Developers
Simulink / TargetLink

C code generation → .c ∪ .h
build for target → .hex
flash → ECU

feed-back to developers

system test
Software Development Process

Function / Module Developers
Simulink / TargetLink

C code generation
configuration

from suppliers
build for target

.build
.flash

ECU

HiL

after X days: late!

few & expensive: bottleneck!

feed-back to developers

system test
Software Development Process

Function / Module Developers
Simulink / TargetLink

C code generation

build for PC

from suppliers

virtual ECU

feed-back to developers

Faster Development of AUTOSAR compliant ECUs - ERTS 2014, Toulouse
real RTOS: at t=23ms, a 10 ms task is interrupted by an event triggered task. At that time, the context of the interrupted task needs to be saved. Execution of a task takes time.

Silver: Execution of a task takes zero time. Tasks run at scheduled execution time and execute 'infinitely fast': task execution can therefore not be interrupted: No need for context switch.
Virtual ECUs: Objectives

Run accurate closed-loop simulation of the complete system – On a PC

All engineers equipped with a virtual vehicle

System integration and feed-back within minutes

Virtual ECUs
Simulation of the ECU's application layer (tasks) as in the vehicle

Vehicle Simulation
High-fidelity, configurable
Virtual ECUs: the case of AUTOSAR

AUTOSAR Builder (Dassault Systèmes)

FMU for Co-Simulation 1.0
Standardized format for exchange of simulation models
Developed in EU project Modelisar 2008-2011
Running the Autosar FMU in Silver

Virtual ECU
- control software tasks
- fix-point C code
- one function per task

Silver Basic Software
- flash scale range check
- scheduler to run tasks
- IO driver to HW
- CAN

FMI
- Autosar Builder or other FMI enabled IDEs for control software
- measurement and calibration with CANape or INCA

TCP/IP
- XCP

Silver

configurable GUI
- attach to process
- scripting
- reader writer
- test

Microsoft Visual Studio
- PDB
- debug
- CTC++
- code coverage
- Python
- MDF DAT CSV

MATLAB Simulink
- rapid prototyping

plant model from Dymola, AMESim, SimulationX, MapleSim, ...

DBC
- A2L PAR DCM HEX
- HEX MAP A2L

Autosar Builder or other FMI enabled IDEs for control software
- TriCore and PowerPC emulation

measurement and calibration with CANape or INCA

Faster Development of AUTOSAR compliant ECUs - ERTS 2014, Toulouse
Does anybody use Virtual ECUs today?
IAV GmbH integrated Silver into the TCU development process

- run control software on Windows PC using Silver
- measure and calibrate using CANape and INCA
- simulate CAN using DBC file
- step and debug using Microsoft Visual Studio

IAV's development environment for TCUs

Automated validation of 200 transmission variants

- 200 different TCU variants
- run TCU code in Silver
- test with TestWeaver
- fully automated validation

details in:
ATZ elektronik 4/2013:
Stefan Gloss, Milan Slezák, Andreas Patzer:
Systematic validation of over 200 transmission variants.
Silver at transmission development at Mercedes-Benz

7G-DCT
Dual Clutch Transmission of Mercedes-Benz

- virtual TCU with Silver, SiL test with TestWeaver
- every software release: 24h test run in parallel on several PCs
- thousands of gearshifts generated and analyzed
- code coverage measured using CTC++ and reported using TestWeaver

details in:
Model-based Development of a Dual-Clutch Transmission using Rapid Prototyping and SiL
Virtual ECU for Gasoline Engines

- run ECU code in Silver via TriCore chip emulation
- no access to model or C sources required: run HEX file in Silver or Simulink
- selected functions run much faster than real time
- use e.g. to calibrate selected engine functions on PC

Summary

Virtual ECUs based on
- AUTOSAR builder via FMU
- MBD or hand coded C via code rehosting
- hex files via chip simulation

Virtual ECUs speed up development on three levels
- split development in parallel threads:
  on PC per developer
- rebuild ECU within 5 minutes:
  more integration points
- simulation faster than real time:
  testautomation: increased test throughput
  calibration: enables new (mathematical) processes

Meet us at the ERTS EXPO: Stand #50