Virtual ECUs for Developing Automotive Transmission Software

Dr. Thomas Liebezeit, Jakob Bräuer, Roland Serway (IAV)
Dr. Andreas Junghanns (QTronic)
Innovative Fahrzeug-Getriebe und Hybrid & Elektro-Antriebe, Dezember 2011
Virtual ECUs for Automotive Software

Agenda

- Motivation
- Software-in-the-Loop setup
- Debugging
- Experience
- Conclusion
Virtual ECUs for Automotive Software
Motivation and objective

• **Motivation**
  – Series Transmission Software development
    • Different software variants
  – Functional behaviour testing
    • dSpace Hardware-in-the-Loop (HiL) systems and test vehicles
    • Limited possibilities for troubleshooting and analysis of software
    • Fully utilized HiL systems

• **Objective**
  – Debugging of series transmission function software
Virtual ECUs for Automotive Software
IAV’s boundary conditions

• Usage of Software-in-the-Loop
  – Runs completely on Developer PC
  – Enables convenient debugging

• Full-featured debugging
  – Break points (fix, conditional)
  – Reading and changing of run-time variables

• No code changes allowed

• Full process control by IAV

• All-time deployable by developer

• Reuse standard data sources (A2L, PAR, DBC)

• Consistency over X-in-the-Loop (SiL, HiL)
Virtual ECUs for Automotive Software
Silver

- Silver from QTronic GmbH
  - Software-in-the-Loop (SiL) simulation environment
  - All relevant automotive standard formats supported
  - Allows debugging via Microsoft Visual Studio
  - IAV has already experience with Silver

![Diagram of Silver features]

- **Configurable GUI**
- **Debugging**
  - MS Visual Studio
  - PDB
- **Data Handling**
  - MDF
  - CSV
  - CANape/INCA
- **Test and Adaptation**
- **Python Scripting**

**Virtual ECU**
- A2L
- PAR
- DCM
- HEX
- Control software tasks
- Fix-point C code

**Environment Model**
- As DLL, S-Function, ...
- MATLAB/Simulink
- Modelica with Dymola/SimulationX

**Rapid Prototyping**
- Via CAN
Virtual ECUs for Automotive Software

Agenda

- Motivation
- SiL setup
- Debugging
- Experience
- Conclusion
Virtual ECUs for Automotive Software
SiL Setup: Transmission Software

- **Transmission software**
  - C-Code (Hand coded, auto code from TargetLink)
  - Mostly accessible as code, some as LIB
  - Interface to Virtual ECU
    - ECU BIOS calls
    - get/set functions for sensor, actuator, CAN data
  - SiL task
    - Compile for PC processor (x86, Microsoft C Compiler)
    - Current developer code
Virtual ECUs for Automotive Software
SiL Setup: Virtual ECU

• **Virtual ECU**
  – Hardware and software
    • IO interface to Silver-API
      – Input and output signals
      – Virtual CAN
    • Timing
      – Task slices
    • BIOS functionality
    • Non-volatile memory
  – SiL task
    • Write C-Code using Silver Basis Software (SBS)
    • New: Virtual CAN
Virtual ECUs for Automotive Software
SiL Setup: Environment Model

- **Environment model**
  - Longitudinal vehicle dynamics and CAN rest bus
  - SiL task
    - Reuse existing HiL model
      - Implemented in Simulink
      - Adjust Timing
      - Switch block set to Silver block library (IO, CAN)
    - Compile for PC processor (x86)
      - Silver simbuild tool
      - Real Time Workshop
      - Microsoft C Compiler
  - Rollout via version control system
Virtual ECUs for Automotive Software

SiL Setup: Silver Configuration

- **Silver configuration**
  - Graphical user interface
    - Start/stop simulation
    - Interact with simulation (gear lever, accelerator and brake pedal)
    - Display and change elementary information
      - Software: A2L measurement signals and parameters
      - Model: Status information
    - PAR file flashing
    - Access A2L
    - SiL task
      - Setup new experiment
Virtual ECUs for Automotive Software

Agenda

- Motivation
- SiL setup
- Debugging
- Experience
- Conclusion
Virtual ECUs for Automotive Software Debugging

- **Characteristics**
  - SiL setup (as described before) with current developer software
  - Stops whole simulation (incl. environment model)
  - Configuration is easily adaptable

- **Debugging focus**
  - Situations that are difficult to produce
  - Timing errors
  - Re-simulation of vehicle measurements
  - Fault simulation
  - „Living code“
Virtual ECUs for Automotive Software
Debugging

- **Silver Signal debugging**
  - Stop Silver simulation at arbitrary time
  - Analyse signals (with history even new added)

- **Code debugging**
  - Open QTronic Silver with correct experiment
  - Open Microsoft Visual Studio
    - Open C file
    - Set break point(-s)
    - Attach to Silver process
  - Start Simulation
  - Debug
Virtual ECUs for Automotive Software

Agenda

- Motivation
- SiL setup
- Debugging
- Experience
- Conclusion
Virtual ECUs for Automotive Software
Experience: Build up and maintenance

- IAV had already experience with Silver usage
- First build up of SiL Setup
  - Work of IAV (with help from QTronic)
  - Effort: 6 MM
- Maintaining/ updating effort
  - Keep running since 1 year
  - Tasks
    - Adapting to function software changes (frequently)
    - Model updating (less frequently)
    - Change to new Silver API 2.4:
      - Better access to model data
      - Use build in CAN
    - Process improvements
  - Effort: ca. ½ developer
Virtual ECUs for Automotive Software Experience

• Added value from debugging
  – New quality of debugging
    • Step through code
    • Full access to all variables
    • Full history of signals in Silver
  – Analysis times reduced
  – Faster change-analysis-change cycles

• User acceptance
  – Growing acceptance
  – Advantages are fully accepted
  – High availability requested
Virtual ECUs for Automotive Software

Agenda

- Motivation
- SiL setup
- Debugging
- Experience
- Conclusion
Virtual ECUs for Automotive Software

Conclusion

• Silver enables build up of automotive SiL simulations easily
• Debugging at SiL level is successful
• SiL will be established soon
• Additional use cases planned for SiL simulation
Thank you!

Dr. Thomas Liebezeit
IAV GmbH
Ingenieurgesellschaft Auto und Verkehr
Carnotstraße 1, 10587 Berlin
Telefon: +49 30 39978-9021
thomas.liebezeit@iav.de