Embedded Success dSPACE

Production Code Generation & System Architecture

Frank Hein

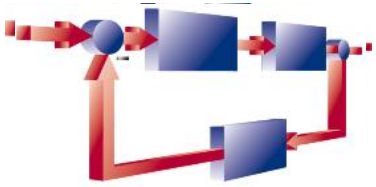
dSPACE GmbH

Technologiepark 25 33100 Paderborn

Tel. +49 5251 1638-545 Fax +49 5251 66529

Email: FHein@dspace.de

Today's Development Process - The V Cycle



Control Design

Calibration

TargetLink

C code generation from MATLAB/Simulink

Generated C code is as efficient as hand code

High reliability, readability and reproducibility of generated C code

Rapid Control Prototyping

Hardware-in-the-Loop Simulation

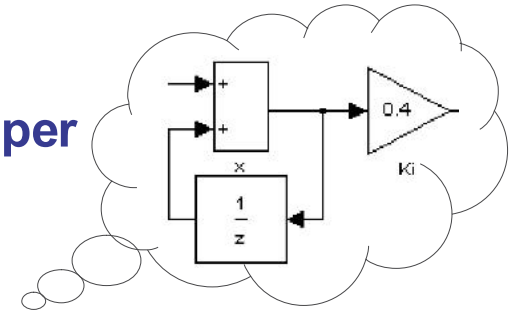
Automatic Production Code Generation

Traditional Approach

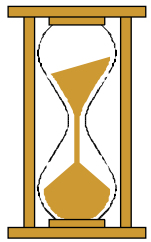
dSPACE

Function Developer

⇨ algorithm knowledge

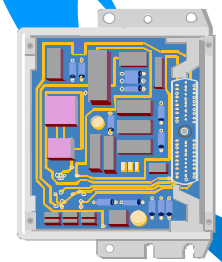


Specs



Time

```
#include <math.h>
if (a > 0)
 ki = 0.4*x+z1;
```



Production Code

Problems:

- Communication
- Ambiguity of specs
- Resource conflicts

⇨ **Large turnaround time!**

Software Spezialist

⇨ implementation + coding knowledge

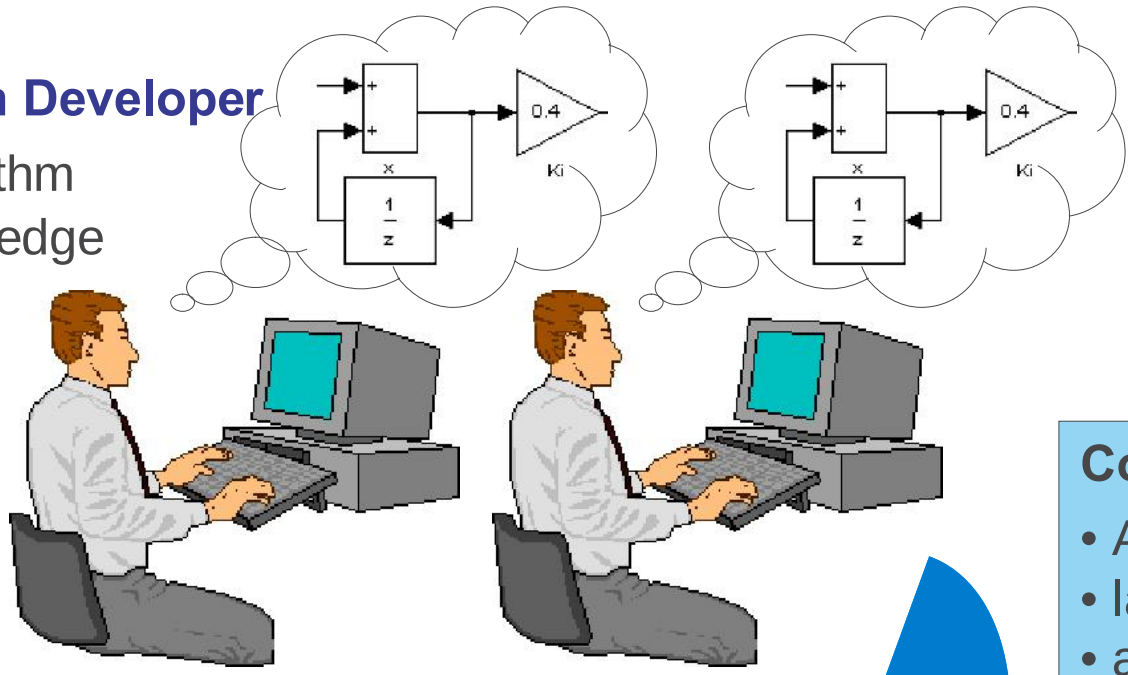
Approach with Code Generation

Function Developer

⇨ algorithm knowledge

Software Specialist

⇨ implementation knowledge



TargetLink

Code Generator

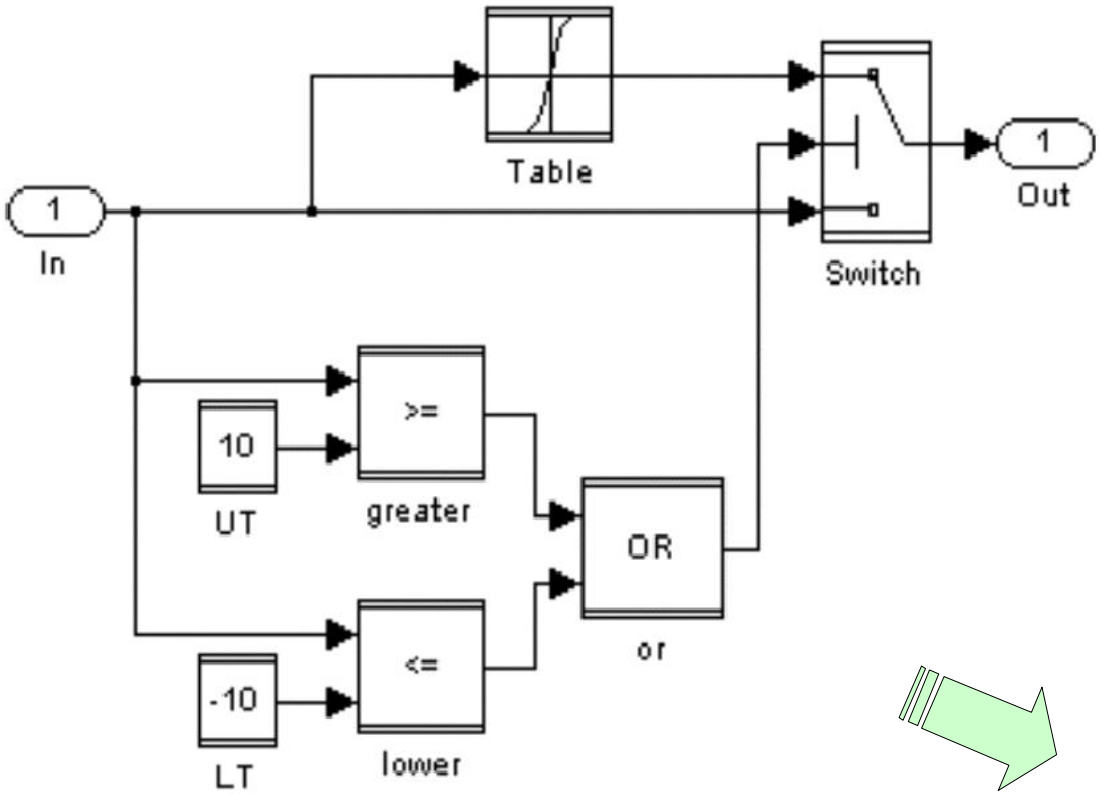
⇨ coding knowledge

Coding knowledge:

- ANSI-C
- language extensions
- assembly language
- processor architecture

↳ **… and how to optimally use it!**

Example of Interblock Optimization



RCP Code:

```
bool1 = (In >= 10);
bool2 = (In <= -10);
bool3 = bool1 || bool2;

tmp1 = table_lookup(Table, In);
tmp2 = in;

if (bool3)
    out = tmp1;
else
    out = tmp2;
```

TargetLink Code:

```
if ((in >= UT) || (in <= LT))
    out = table_lookup(Table, in);
else
    out = in;
```

- no unnecessary code overhead
- no waste of execution time
- no temporary variables

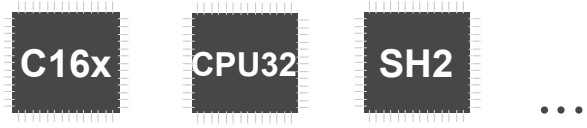
Block Optimization

Problem

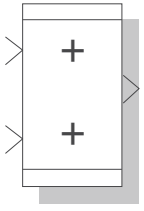
- Many different implementations for one algorithm exist
- Optimal implementations differ between compiler/processors

Solution: Carefully designed code patterns

- Code pattern library
- Contains best code pattern for each compiler/processor combination
- Optional for:
  - ANSI C only
  - Assembly language and compiler specific C



Example:



16 bit Addition with saturation

TargetLink

- Production Code Generation directly from Simulink® / Stateflow®
  - ANSI C Code with the efficiency of hand-written code
  - Target Simulation Modules for testing the generated code on evaluation boards
  - Data Management with the dSPACE Data Dictionary
  - Special modules for AUTOSAR and OSEK Support
  - Target optimized code for different compiler-processor combinations

TargetLink Code Properties

- TargetLink generates ANSI C code and optionally non-ANSI C or even assembly code when the benefit is substantial.
- Speed and memory efficiency correspond to handwritten code.
- Fixed-point and floating point code generation is supported.
- The generated code is readable, well commented and easily inspectable.
- The generated code is easy to integrate, and it is easy to integrate legacy code
- Efficiency achieved by TargetLink through
  - Sophisticated interblock optimization.
  - Carefully selected code patterns.
  - Compiler/processor specific block optimization through the use of language extensions and assembler macros (Target Optimization Modules).

Code Efficiency

- Delphi, 2002

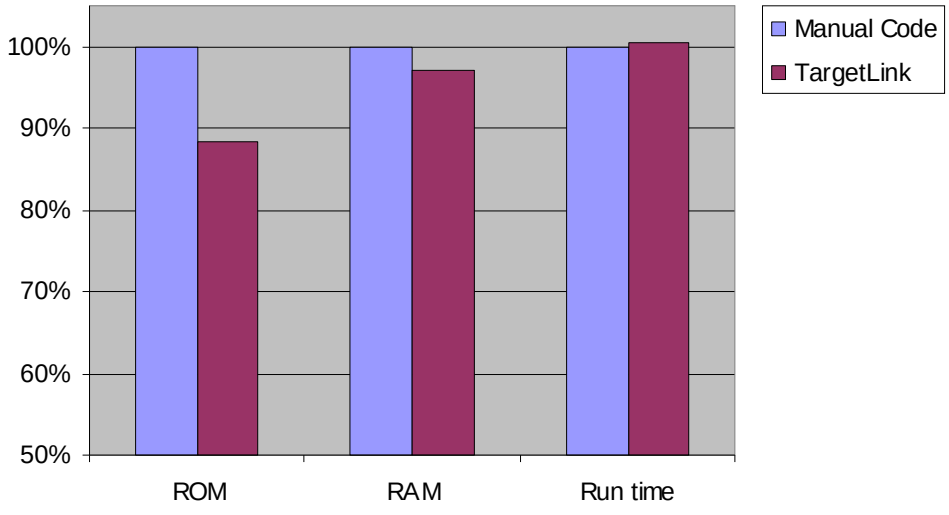
ROM	RAM	Stack	Speed
0.96 – 1.1	0.97 – 1.2	1.2 – 1.25	0,75 – 1.2

Source: Lev Vitkin, Delphi, USA

Generated code better than hand code

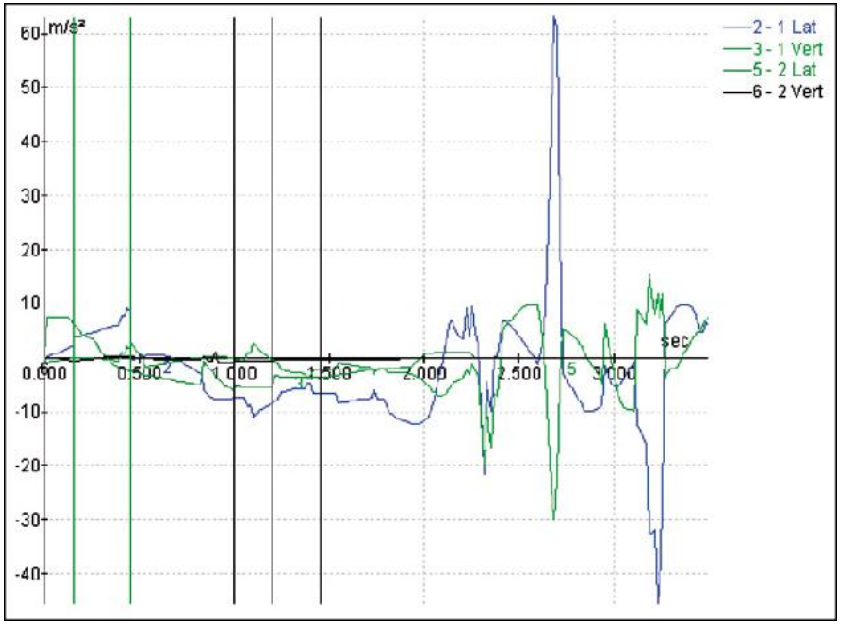
- German Tier One Supplier, February 2006:

	Manual Code	TargetLink
ROM [bytes]	2182 (100%)	**88,5%**
RAM [bytes]	104 (100%)	**97,1%**
Run time [msec]	**2,27 (100%)**	100,4%



Delphi: Surviving a Vehicle Rollover

- Delphi's developments for passenger safety systems
- A rollover detection algorithm was implemented and is now in production
- Significant code improvements were achieved using TargetLink's code profiling techniques



Simulated sensor signals show the lateral and vertical accelerations to be analyzed by a rollover detection algorithm

Delphi: Surviving a Vehicle Rollover

Autocode Success Story

- Code-Profiling led to handcode-efficiency

  RAM and throughput (execution time) down by 75%
- Integrated in 1.5 days
- In production
- Statements from software engineers:

  “The generated code was easy to understand. Every comment and variable name was a great help.”

  “In my opinion it saved a lot of time. It is a good base for developing target C code. The main backbone of code was almost unchanged.”

TargetLink was used to generate efficient fixed-point C code from Simulink and Stateflow models

Delphi: Surviving a Vehicle Rollover

dSPACE

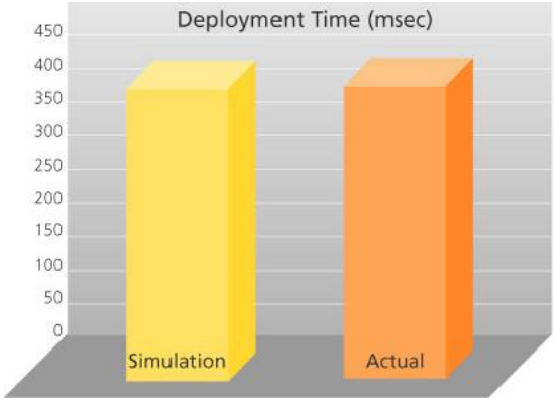
Equipment and Methods

- TargetLink
- Target Optimization Module
- Motorola HC12 Evaluation Board for code profiling
  - Throughput (execution time)
  - RAM (including stack)
  - ROM
- Back-to-Back-Tests (MIL, SIL and PIL simulations) at earliest stage

Delphi: Surviving a Vehicle Rollover

Autocode reduces risks

- No transcription errors
- No spec misinterpretation
- Match with model performance

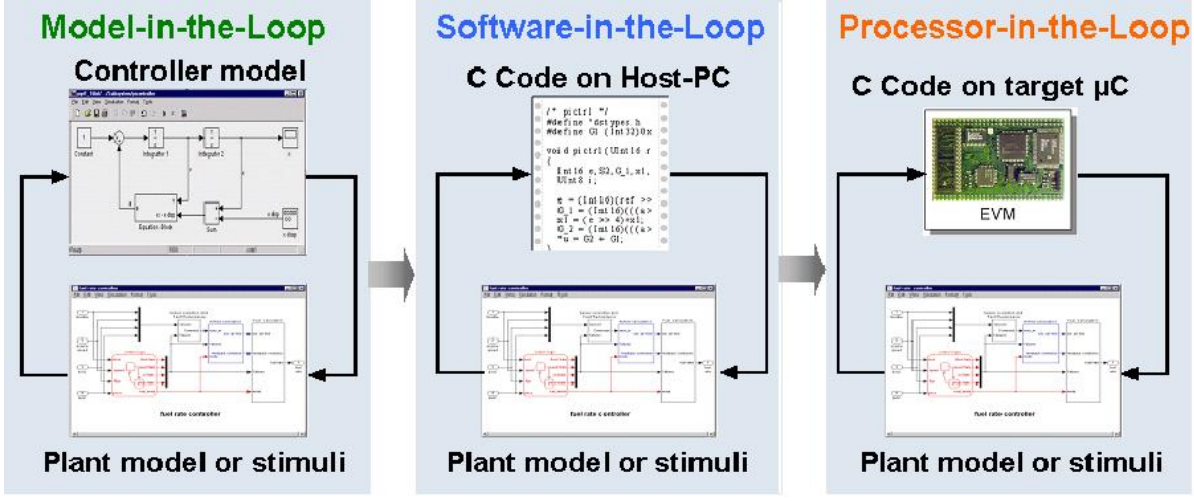


The deployment time measured for the autocoded algorithm matched the simulation

Insights into the algorithm gained from using TargetLink's code analysis tools and an evaluation board led to huge improvements in RAM consumption and throughput (execution time)

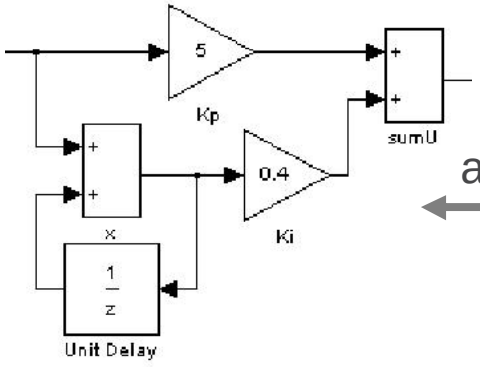
Unmatched Simulation Support

- TargetLink offers MIL, SIL, PIL simulation at a mouse-click
  - No need for separate test model
  - No need for manual insertion into test harness model
- Integrated Data Logging for ***all*** simulation modes
- Integrated Result Plotting
  - Direct visualization of simulation results
  - Direct comparison of MIL/SIL/PIL results
  - No need for writing your own plotting scripts
- SIL/PIL simulation enhanced by Code Coverage measurement



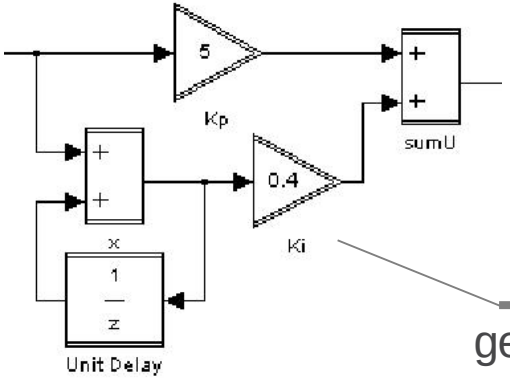
Basic Workflow with TargetLink

Simulink blockset



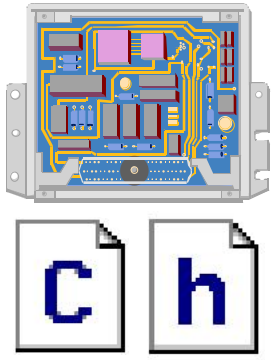
automatic

TargetLink blockset



Code generation

ECU



Code generation

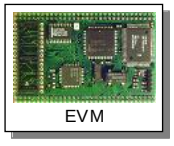
Model-in-the-Loop Simulation on the Host PC

algorithm design
- behavior validity checks
- scaling (auto or manual)
- overflow detection
- reference traces

Software-in-the-Loop Simulation on the Host PC

fixed-point effects
- quantization errors
- saturation and overflow
- implementation option

Processor-in-the-Loop Simulation on evaluation board



code validation
- profile execution time
- measure stack size
- measure RAM / ROM
- final verifications

TargetLink – Successfully used by …

dSPACE

Audi

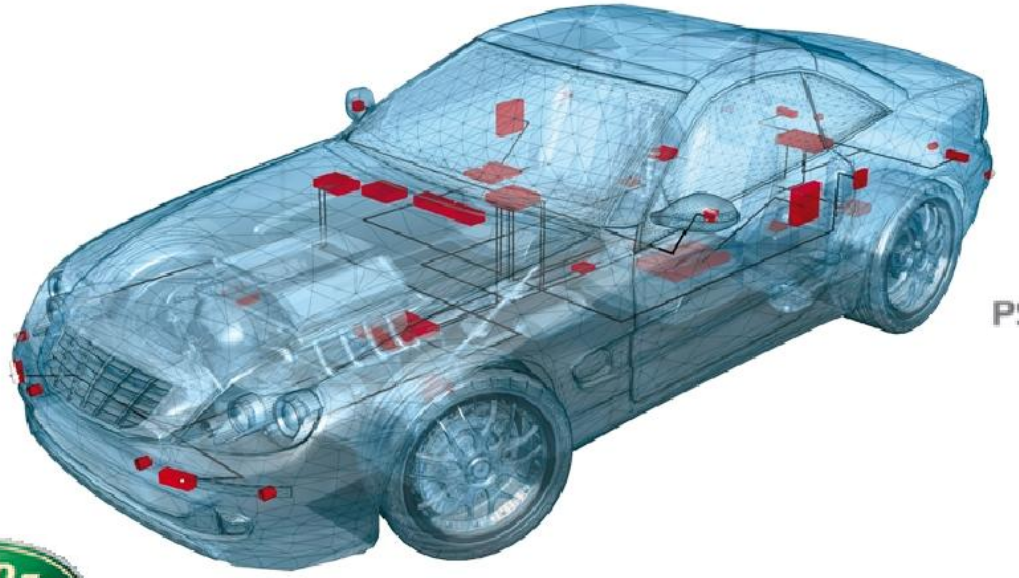
BMW

DAIMLER

NISSAN

PORSCHE

CHRYSLER



RENAULT

FIAT

PSA PEUGEOT CITROËN

Ford

JAGUAR

SAAB

LAND ROVER

HYUNDAI

HONDA

MAN

VOLVO

Volvo Car Corporation

VOLVO

VW

TargetLink – Successfully used by …

dSPACE

BHTC COMFORT IN MOTION

BOSCH

MAGNETI MARELLI

MARQUARDT

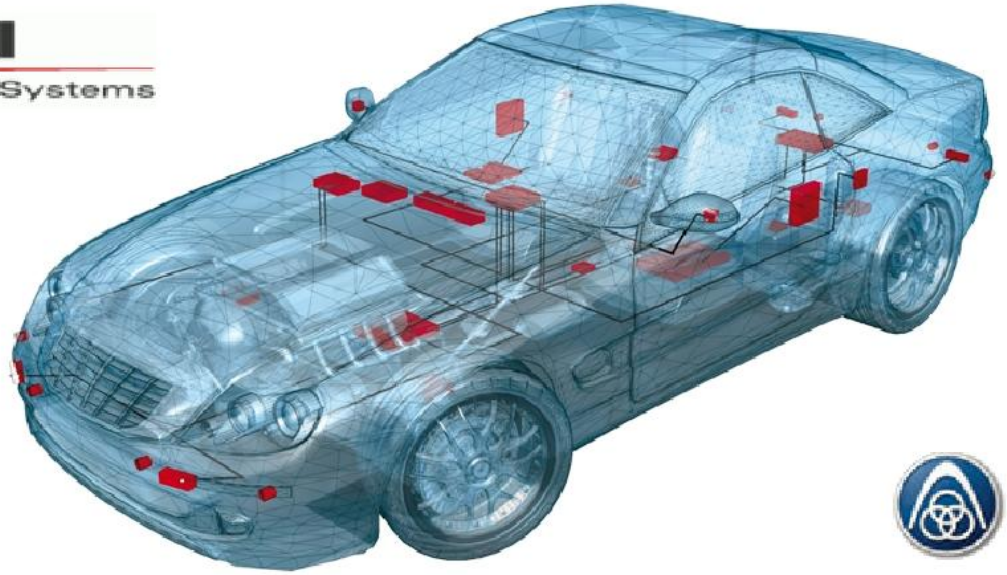
MITSUBISHI ELECTRIC

BorgWarner

Continental

MAGNA STEYR
more value•more car

DELPHI
Automotive Systems



MAGNA
MAGNA DRIVETRAIN

HELLA

RÜCKER

HITACHI
Inspire the Next

TRW
Automotive

Jatco

ThyssenKrupp
Automotive

ZF Lenksysteme

KEIHIN

KOSTAL

Valeo

Visteon

WABCO

ZF

EADS: Unmanned aerial vehicle (UAV) „Barracuda“

- Barracuda: demonstrator and design platform for future UAVs
- TargetLink for all algorithms (flight control, autopilot, flight management, calculation of flight data, navigation, signal consolidation in the triplex redundant system)
- 45% of the source code of the flight control computer was generated automatically using TargetLink

ZF Steering Systems: Steer-By-Wire

"TargetLink is the connecting link between prototyping and the target system.

TargetLink guarantees seamless software design, which is frequently demanded by our customers"

Matthias Haußmann, ZF Lenksysteme

- Standard architecture is replaced by electrical or electrohydraulic actuators
- Simulink model with 355 blocks used to design the controller
- TargetLink fits ideally into the development environment
- Reduction of development time using TargetLink's Code Generator by 40%

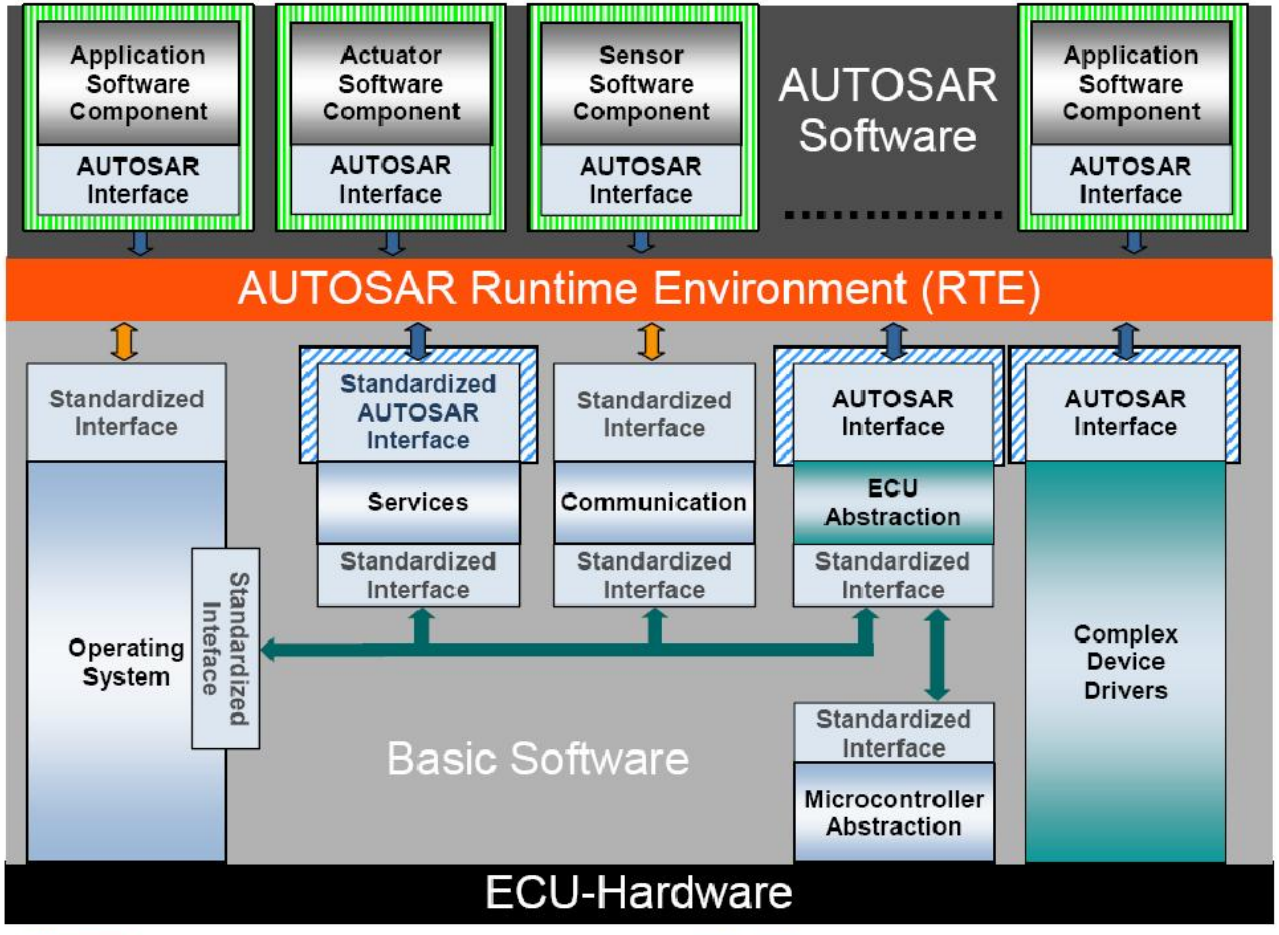
AUTOSAR Support

- Product level support since end of 2006
  - dSPACE strategy: early and competent support of automotive standards
  - AUTOSAR might not yet be interesting for you, but at the time it will, TargetLink will offer a mature and proven-in-practice solution!
- First production projects, e.g. at
  - Daimler AG, Body,
    refer to

    AUTOSAR in the development process – procedure for introducing model-based AUTOSAR function development into production projects, Christian Dziobek, Dr. Florian Wohlgemuth, Dr. Thomas Ringler, Daimler AG, dSPACE Magazine, 01/2008
  - AUDI AG, Chassis,
    refer to

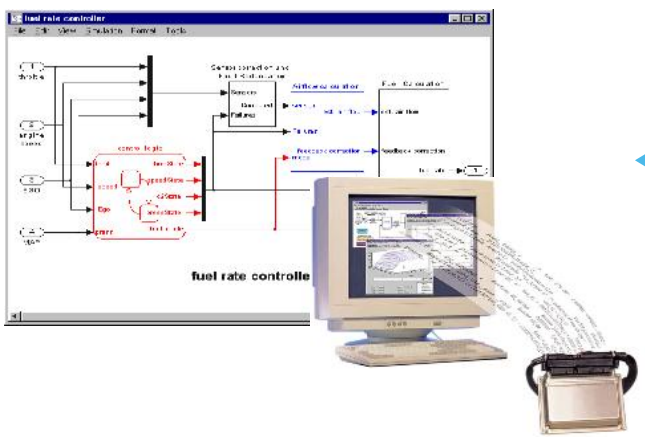
    Systematic AUTOSAR-Migration, Frank Gesele, Dr. Karsten Schmidt, AUDI AG, dSPACE NEWS 2008/1, Feb 2008

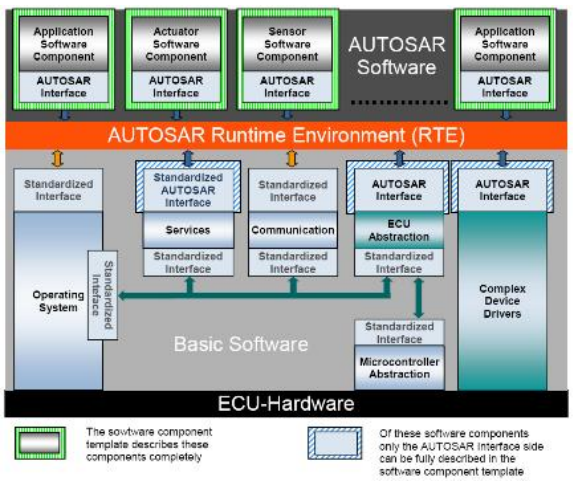
Why Customers Need AUTOSAR



AUTOSAR Software Component Support

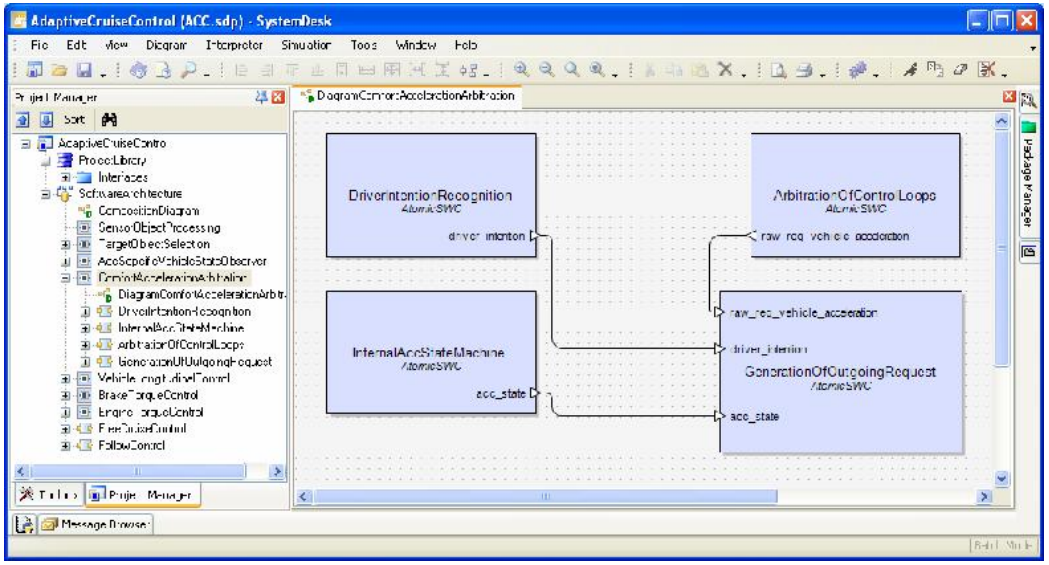
- Support for AUTOSAR Software Components includes
  - Modeling: Model AUTOSAR Software Components by means of special AUTOSAR Blocks to define AUTOSAR Ports, Runnables etc.
  - Code Generation: Generate AUTOSAR compatible code with corresponding RTE function calls
  - AUTOSAR SW-C description support:
    - Generate xml-File according to AUTOSAR SW-Component Template
    - Import SW-C description into the DD for reference from within the TargetLink model





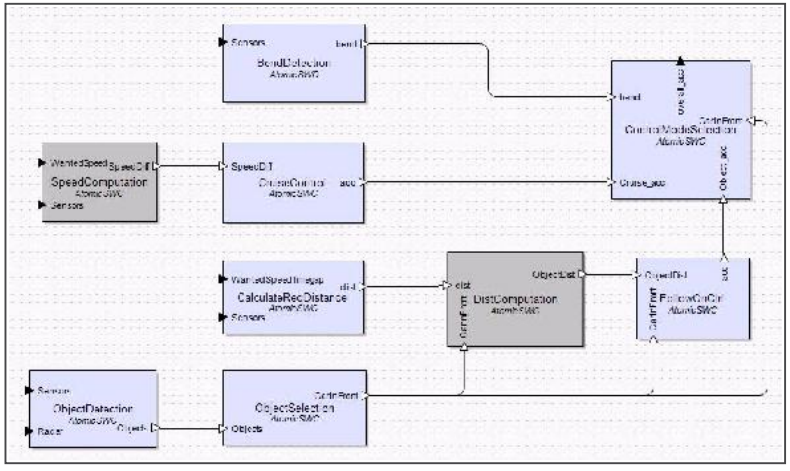
dSPACE SystemDesk 1.0 – Main Features

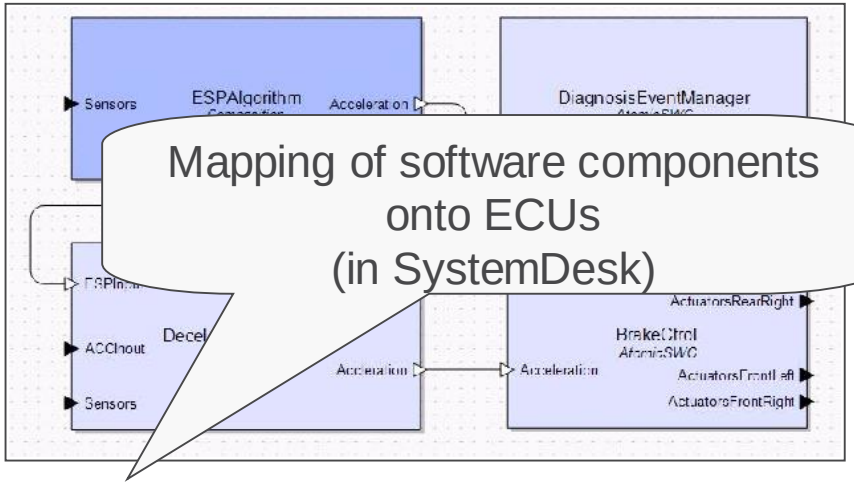
- Support of AUTOSAR 2.0
- Modeling of
  - Software architectures
  - Hardware topologies
  - System descriptions
- Import of DBC files for CAN bus
- RTE generation



- Connection to dSPACE TargetLink for modeling control algorithms
- Connection to AUTOSAR R2.0 compliant BSW configuration tools, espc. EB tresos 2007b
- Process support, e.g. tool automation and scripting

From Software Architectures onto ECUs





Mapping of software components onto ECUs (in SystemDesk)

CAN-Bus

Mapping of Logical Communication onto Bus Signals

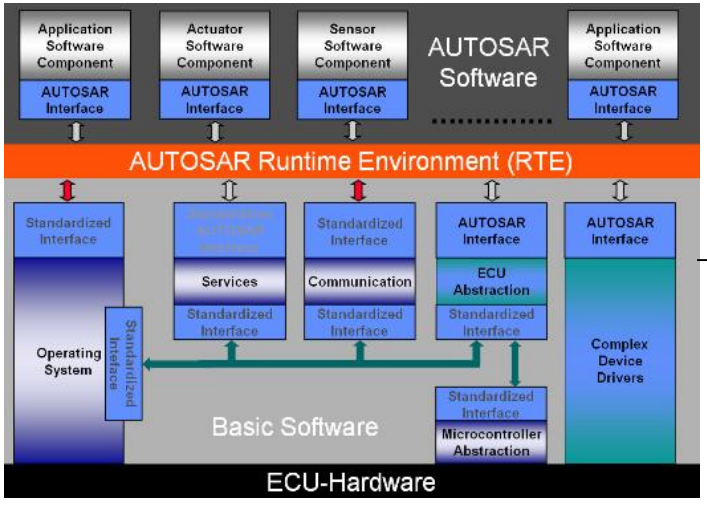
Mapping onto Bus Signals in SystemDesk

Logical Communciation

Bus Signals (e.g. from dbc)

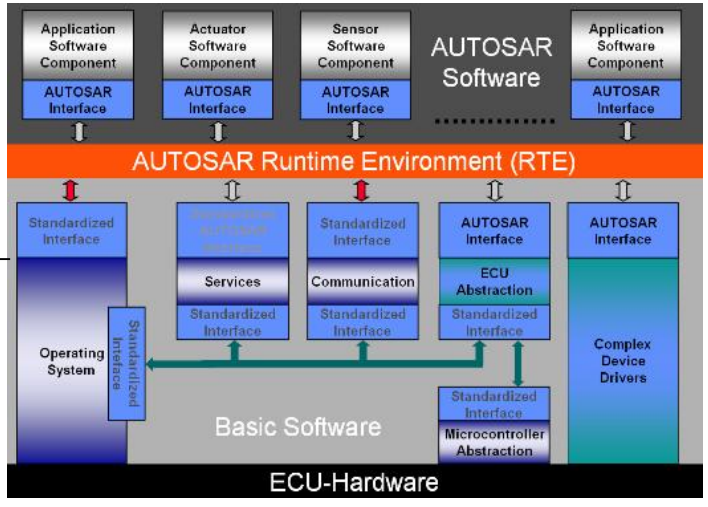
Node Mapping | Signal Mapping

Interface	Data Element	Bus	Message	Network Signal
if_percentage	value	BodyCAN	BrakeLampMsg	brake_lamp
if_bulb	value	BodyCAN	LeftRightIndicatorMsg	left
if_bulb	value	BodyCAN	LeftRightIndicatorMsg	right

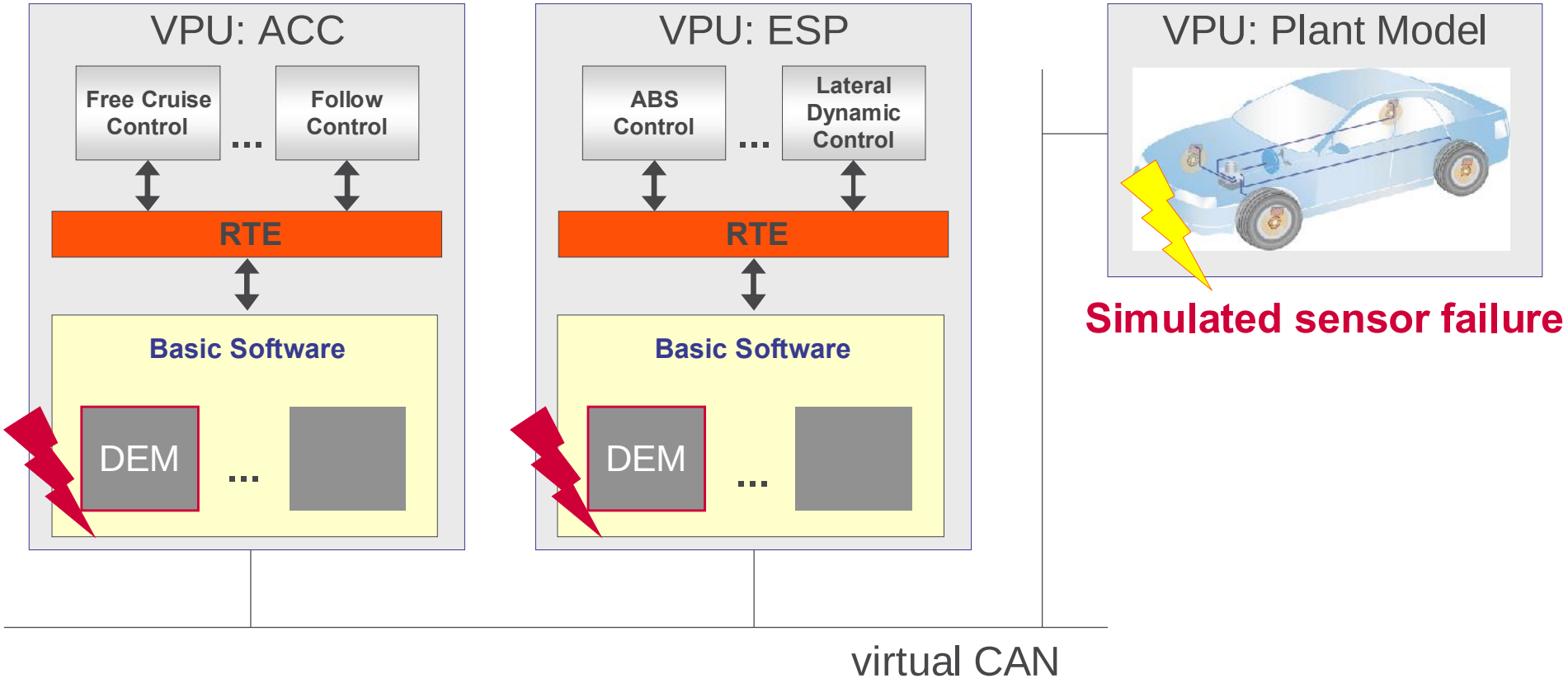


Supported Busses:

- **FlexRay**
- **CAN**
- **LIN**



Simulation of Models on PCs



- Early detection of errors
- Verification of diagnosis software

SystemDesk – Upcoming releases

dSPACE

SystemDesk 1.1

- AUTOSAR R2.1 and R2.0 (compatibility update for R2.1)
- LDF import
- Connection to requirements mgmt. tools like DOORS
- Connection to BSW config. tool EB tresos 2008-A

July 2008

SystemDesk 2.0

- AUTOSAR R3.0 and R2.1
- Simulation of software architectures
- Support for calibration & measurement
- Improved bus support espc. for FlexRay
- Connection to BSW config. tool EB tresos 2008-B

November 2008

Releases in 2009/2010

- AUTOSAR > R3.0
- Expansion of features provided by SystemDesk 2.0
- Connection to future BSW config. tool versions

2009

*) Due to ongoing specification work with AUTOSAR, features for future versions of SystemDesk are still under discussion.