AUTOSAR Technology at dSPACE
# Contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR Technology</td>
<td>3</td>
</tr>
<tr>
<td>dSPACE Products for AUTOSAR Applications</td>
<td>4</td>
</tr>
<tr>
<td>SystemDesk®</td>
<td>5</td>
</tr>
<tr>
<td>TargetLink®</td>
<td>8</td>
</tr>
<tr>
<td>TargetLink and SystemDesk</td>
<td>13</td>
</tr>
<tr>
<td>RTI AUTOSAR Blockset</td>
<td>14</td>
</tr>
<tr>
<td>VEOS®</td>
<td>17</td>
</tr>
<tr>
<td>Bus Implementation Tools</td>
<td>18</td>
</tr>
<tr>
<td>Support of AUTOSAR Releases</td>
<td>19</td>
</tr>
<tr>
<td>Cooperation Between dSPACE and EB Automotive</td>
<td>21</td>
</tr>
</tbody>
</table>
AUTOSAR Technology

About AUTOSAR

AUTOSAR Standard
The AUTOSAR (AUTomotive Open System ARchitecture) partnership is an alliance in which the majority of OEMs, suppliers, tool providers and semiconductor companies work together to develop and establish an open de facto standard for automotive electric/electronic (E/E) architectures. Managing the growing E/E complexity is another important aspect of their work. The goal of AUTOSAR is to fulfill future vehicle requirements, such as availability and safety, software upgrades/updates and maintainability, as well as increased scalability and flexibility to integrate and transfer functions.

For more information about the AUTOSAR partnership, please see www.autosar.org

Benefits of AUTOSAR
As a de facto standard for automotive E/E architecture, AUTOSAR contains specifications for communication interfaces between application functions and basic software functions. These specifications support the separate development of different functions as software components (SWC). They simplify the tasks of combining functions into one electronic control unit (ECU) and of integrating multiple ECUs. Exchanging and reusing software components is also easier, as functions that were already tested can be used again, which in turn reduces the development and testing effort.

Using AUTOSAR
Single and networked ECUs have grown more complex in recent years, and OEMs often spread function development across several suppliers. They provide the suppliers with the AUTOSAR SWC description XML file, which contains specifications for the application, software components of the actuators and sensors, and their communication interfaces. These interfaces simplify the integration of the functions later on in the development process. By exporting and exchanging the system description, software component description and ECU configuration files, all the developers involved – on both the OEM and the supplier side – can share their knowledge of the system structure. This enhances their understanding of the software in the ECUs and is useful throughout the development process.

AUTOSAR Activities at dSPACE
dSPACE joined the AUTOSAR partnership as a Premium Member in April 2004 and is active in defining and developing parts of the architecture and its specifications, which many dSPACE products support. These products help you adapt your development process to the AUTOSAR standard smoothly.
dSPACE Products for AUTOSAR Applications
Tools for AUTOSAR-compliant development

- Comprehensive AUTOSAR support throughout the development process
- Tight interaction between dSPACE tools
- Up-to-date support for AUTOSAR releases
- Cooperation with third-party tools

### dSPACE Tools for Developing AUTOSAR-Compliant ECU Software

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemDesk</td>
<td>Planning, implementing and integrating complex system architectures and generating virtual ECU for validation</td>
</tr>
<tr>
<td>TargetLink AUTOSAR Module</td>
<td>Modeling functional behavior and generating AUTOSAR-compliant production code</td>
</tr>
<tr>
<td>RTI AUTOSAR Blockset</td>
<td>Use of AUTOSAR-compliant production software components and modules on the dSPACE MicroAutoBox II real-time development platform</td>
</tr>
<tr>
<td>VEOS</td>
<td>Platform for PC-based simulation of virtual ECU for early validation of ECU software and functions</td>
</tr>
<tr>
<td>RTI CAN MultiMessage Blockset</td>
<td>Simulating CAN communication based on AUTOSAR network descriptions</td>
</tr>
<tr>
<td>RTI LIN MultiMessage Blockset</td>
<td>Simulating LIN communication based on AUTOSAR network descriptions</td>
</tr>
<tr>
<td>dSPACE FlexRay Configuration Package</td>
<td>Simulating FlexRay communication based on AUTOSAR network descriptions</td>
</tr>
</tbody>
</table>
SystemDesk®
Modeling system architectures and generating virtual ECUs

- ECU software architecture and system development for AUTOSAR R3 and R4
- Easy and efficient modeling with diagrams and editors
- Generation of virtual ECUs for early validation with VEOS® or SCALEXIO®
- Coupling with TargetLink® for function development
- Complete and documented API for automation
- Powerful validation of AUTOSAR models

Application Areas
SystemDesk is a system architecture tool that provides sophisticated and extensive support for modeling AUTOSAR architectures and systems for application software. Comprehensive graphical support facilitates this support and also enables efficient and error-reduced working during large-scale projects.

In addition, SystemDesk makes validating AUTOSAR-based system architectures possible at an early development stage. For example, virtual ECUs (V-ECUs) are generated out of the application software and used as units under test with dSPACE simulation platforms.

Key Benefits
- Clear system overview due to graphical modeling with diagrams
- Flawless data input due to sophisticated dialogs and editors
- Easy generation of virtual ECUs for validation purposes
- Reliable project quality through comprehensive rule-based validation of consistency and correctness
- Convenient development process due to automation of recurring tasks or remote control by third-party tools via the open API

Integration into the Development Workflow
SystemDesk is closely integrated in the dSPACE product portfolio to assist the development of ECU software. With SystemDesk you can also generate V-ECUs that offer the same software components as those that will run on the final ECU. Using V-ECUs you can perform PC-based simulation runs with dSPACE VEOS® for virtual validation or start your hardware-in-the-loop (HIL) tests with dSPACE SCALEXIO® even before all ECU hardware prototypes are available.
SystemDesk Modeling Module

Application Areas
The SystemDesk Modeling Module lets you model AUTOSAR software architectures and systems that contain one or more networked AUTOSAR ECUs. It supports the software architecture development process from modeling the software components (SWCs) to specifying system aspects, including the hardware topology and network communication.

Modeling the Software Architecture
The first step is to define the software components (SWCs), which each represent a functionality, and the data they exchange with each other. Since the components can be developed individually, development work can be split between several developers. For example, an OEM can develop an overall system design and then extract the specifications for suppliers to provide the SWCs. Additionally existing SWCs or other parts of the architecture, such as data types, can be imported for reusing existing software. SystemDesk supports the modeling process by a graphical representation of the SWCs and their interconnections. New SWCs, their ports and their connections are modeled in this diagram, providing developers an intuitive working method and offering a well-defined overview of even the most complex systems.

Designing Software Components
The next step is to define and configure the internal behavior of the software components. Among others, this involves runnables, which are C functions inside of software components, and the means of communication between them and with the outside of the software component. SystemDesk supports the configuration with sophisticated dialogs. These dialogs guide you through so you can enter data easily and avoid potential errors. In addition, several integrated validation procedures make sure the modeled elements and references are AUTOSAR-compliant and consistent.

Modeling the system
In the last process step, a system consisting of a software architecture, a hardware topology, and a network communication description is created. The software architecture was developed in the first step when the software components were designed. The hardware topology of an E/E system can be specified independently of any software architecture. It describes the available ECUs and their interconnections via buses. Each software component from the software architecture has to be mapped to one ECU. The communication between atomic software components that reside on different ECUs (inter-ECU communication) has to be mapped to the network communication. This means that data elements have to be mapped to network signals. A predefined communication matrix, such as a DBC, FIBEX, or LDF file, can be imported into SystemDesk to configure the network communication. In SystemDesk you can create a custom view of the system to concentrate on single parts or tasks. Convenient mapping editors present the data in a clear order and offer easy means for working on specific tasks. Once the system is modeled, it can be integrated either in a basic software configuration tool to create the final ECU or in SystemDesk as a virtual ECU for simulation on a dSPACE simulation platform.
**SystemDesk V-ECU Generation Module**

**Application Areas**
For virtual validation applications, the SystemDesk V-ECU Generation Module lets you configure and generate virtual ECUs (V-ECUs). V-ECUs enable you to test the system’s overall behavior with either the PC-based simulation platform VEOS, the hardware-in-the-loop (HIL) simulator SCALEXIO® or the rapid control prototyping (RCP) platform MicroAutoBox II as soon as the implementation with C code is available.

**Feature Overview**
V-ECUs created with SystemDesk contain components from the application and the basic software, and provide functionalities comparable to those of real ECUs. For this, the SystemDesk V-ECU Generation Module makes it possible to configure and generate basic software (BSW) modules: e.g., the RTE, OS, ECU State Manager or NVRAM Manager. For a real ECU, configuring the basic software is often a tedious task. Not so with SystemDesk, as you can choose to let SystemDesk automatically configure the BSW modules, which is often sufficient for simulation purposes. If you want to configure a module specifically to your needs, SystemDesk also offers convenient editors, for example for mapping RTE events to tasks.

Preparing the V-ECU for connection to plant models is easy with SystemDesk. You can even include intervention points at which errors for stimulating the RTE are inserted, which is a lot easier than in a real ECU. With that you can easily test the application software in various error scenarios. Besides the task of configuring and generating the V-ECU, SystemDesk supports you during the whole process. A powerful validation checks the AUTOSAR architecture and reports problems in the input ARXML, enabling you to fix them instead of running into them later during the RTE generation. And, thanks to the complete automation API, SystemDesk can also be used in automated processes for generating the V-ECUs.

**Integration Into the Development Tool Chain**
V-ECUs can be simulated with the dSPACE PC-based simulation platform VEOS for early verification of the ECU software. VEOS offline simulation can be used for simulating single V-ECUs and ECU networks including bus simulation, and also for software-in-the-loop (SIL) and processor-in-the-loop (PIL) simulation. V-ECUs can also be used for HIL simulation with dSPACE SCALEXIO, where ECU hardware prototypes which are not yet available are substituted by V-ECUs. For RCP purposes, the RTI AUTOSAR Blockset allows V-ECUs to be deployed on dSPACE’s RCP system MicroAutoBox II. They can be simulated together with the physical plant, and be evaluated in the laboratory or in the vehicle.

No matter of which simulation platform you use, you can use ControlDesk® Next Generation for visualizing and experimenting the simulation, AutomationDesk for test automation, and ModelDesk for model and environment parameterization. The advantage is that you can create and test layouts and automated tests for the HIL simulation on VEOS and reuse them on SCALEXIO.
TargetLink®
Automatic production code generator

Highlights
- High-quality production code generation directly from Simulink®/Stateflow®
- Powerful software design and testing features
- High-performance AUTOSAR support
- Certified for IEC 61508 and ISO 26262

Application Area
Model-based design has become the established development method across many industries, and production code generation is the logical step for turning models into efficient, production-ready code. TargetLink generates production code (C code) straight from the MATLAB®/Simulink/Stateflow graphical development environment. The C code generation options range from plain ANSI C code to optimized fixed- or floating-point code for AUTOSAR platforms. Versatile code configuration options ensure that the production code copes with processor constraints.

Key Benefits
Converting graphical models directly into production code ensures perfect consistency between model and code. With deterministic code generation by TargetLink, the same model always results in the same proven code to guarantee the highest software quality. Every step can be tested against the specification via the built-in simulation features. This allows early verification and translates directly into cost savings, for example, by avoiding expensive ECU software errors.

Efficient Coding
Efficiency is the key to production-quality code. Efficient code requires a minimum of execution time and resources to run on a cost-efficient embedded processor. Code generated by TargetLink is proven to be as efficient as handwritten code. Other factors also make TargetLink such a useful tool: code readability, traceable model/code dependency, and last but not least, the ability to configure the code generation to produce exactly the kind of code that is required.

Seamless Tool Chain
TargetLink seamlessly connects function development and code generation for the control unit or prototyping hardware. It also automatically closes the gap between the design and verification phases. The result: transparent, well-defined development processes for conventional ECUs and AUTOSAR ECUs. Calibration files, AUTOSAR software component descriptions, and comprehensive documentation are generated in addition to the production code.
Model-Based Design for AUTOSAR Software Components

The TargetLink AUTOSAR Module
The optional TargetLink AUTOSAR Module makes TargetLink's modeling, simulation and code generation features available for designing AUTOSAR software components (SWCs). Developers can specify AUTOSAR structure elements, such as runnables, ports, and communication interfaces, simply at model level. Hence, model-based designs are directly implemented in the form of AUTOSAR-compliant production code.

Functionality
- Model-based design for AUTOSAR software components (SWCs)
- Code generation for AUTOSAR SWCs
- Importing, exporting and merging software component descriptions
- Simulating and testing SWCs
- AUTOSAR frame model generation
- Round trips with other AUTOSAR tools based on SWC description files
- Convenient round trips with SystemDesk based on SWC container exchange

Benefits
- Efficient and easy modeling using proven AUTOSAR workflows
- AUTOSAR-compliant code generation
- Easy migration of existing TargetLink models to AUTOSAR
- Testing and verifying SWCs in early design phases
- Eliminates tedious manual creation of software component description files
- Easy integration of TargetLink in an AUTOSAR tool chain
In brief: You go from model to AUTOSAR-compliant code faster.

TargetLink AUTOSAR Block Library
To design AUTOSAR SWCs, TargetLink offers the TargetLink AUTOSAR Block Library:
- AUTOSAR runnables and operation calls modeled via the TargetLink Function/Runnable block.
- AUTOSAR communication mechanisms (e.g., sender-receiver or client-server communication) specified in TargetLink Inport and Outport blocks.
- Specific AUTOSAR features (e.g. signal acknowledgement, signal invalidation and status signals) are modeled using special ComSpec blocks.
- Optional SWC sender/receiver blocks to represent AUTOSAR ports in a Simulink/TargetLink model.

TargetLink blocks for modeling AUTOSAR SWCs.
Designing AUTOSAR SWCs

To define an AUTOSAR runnable, the Function/Runnable block is applied to a modeled subsystem just as for normal TargetLink functions. This can be done either for new models designed from scratch or for legacy models with established control functions. To define how data is exchanged between the runnables of one or more SWCs, TargetLink Inports and Outports are used just as in non-AUTOSAR applications. TargetLink supports sender-receiver, interrunnable and synchronous client-server communication. Property specifications are made both on block level and via the TargetLink Data Dictionary. The same model can be used to generate both AUTOSAR-compliant code and standard TargetLink code.
AUTOSAR-Compliant Code Generation
TargetLink generates production code for AUTOSAR software components and provides all the code generation options for optimization. Modeled runnables are implemented as C functions and AUTOSAR communication mechanisms specified in TargetLink Imports/Outports are implemented as run-time environment (RTE) macros according to the AUTOSAR standard.

```c
void controller_runnable(void)
{
    /* call of function: controller/Controller_Runnable/RteApiFunction */
    ref = Rte_InRead_PosController_port1_ref();
    ...
    /* Sub: controller/Controller_Runnable/output */
    S12_e = |Int16|((0|Int16| ref) = {{0|Int16| Rte_InRead_PosController_LinPos}}); ...
    /* TargetLink output: controller/Controller_Runnable/output */
    call of function: controller/Controller_Runnable/RteApiFunction1 */
    Rte_OutWrite_PosController_pport1_upi|((Int15) (S13_mP1 << 2));
}
```

The generated C code for a runnable with three included RTE macro calls.

Simulating and Testing AUTOSAR SWCs
With TargetLink, SWCs can be simulated in three simulation modes:
- Model-in-the-loop (MIL)
- Software-in-the-loop (SIL)
- Processor-in-the-loop (PIL)

Multiple SWCs can be simulated in one simulation run. Communication between SWCs is simulated to the extent supported by the Simulink design environment; for example, there is no asynchronous client-server communication.
Importing and Exporting SWC Descriptions
To integrate the code generated for SWCs in the overall AUTOSAR software architecture, software component description files are required. These describe the structural elements such as runnables or ports that are used in an SWC. To save function designers tedious manual work, TargetLink creates the component descriptions and exports the descriptions in AUTOSAR XML format. The TargetLink Data Dictionary also lets you import and merge existing component descriptions to achieve a seamless AUTOSAR development process with a tool like SystemDesk.

AUTOSAR Frame Model Generation
Using a software component description, TargetLink automatically generates a frame model containing the relevant AUTOSAR ports and runnables. The developer can then insert the control algorithm into this model frame to obtain a complete AUTOSAR software component. This procedure makes it much easier to migrate existing models to AUTOSAR or to start modeling from scratch.

Migrating Standard TargetLink Models to AUTOSAR
Existing non-AUTOSAR TargetLink models can easily be migrated to AUTOSAR using the TargetLink AUTOSAR Migration Tool. It converts individual subsystems to AUTOSAR runnables and supports flexible specification of AUTOSAR properties by means of hook functions. You can configure the tool to produce precisely the kind of AUTOSAR-compliant code you require. The TargetLink AUTOSAR Migration Tool is available for download free of charge from www.dspace.com/goto?tl_ar_migration.

TargetLink in an AUTOSAR Tool Chain
For AUTOSAR software development, TargetLink is typically combined with an AUTOSAR architecture tool like SystemDesk. A software architecture with multiple components is specified in the architecture tool, while TargetLink is used to “fill” and implement the individual SWCs. The two tools exchange data on the basis of AUTOSAR XML files. Both a top-down approach (starting with the software architecture) and a bottom-up approach (starting with a TargetLink function model) can be used for AUTOSAR round trips.
**TargetLink and SystemDesk**

TargetLink and SystemDesk are the ideal combination for developing AUTOSAR-compliant software. The two tools exchange SWC containers that hold not only ARXML files, but also source files for implementation, A2L files, and other helpful meta information. This is a reliable, transparent way to perform AUTOSAR round trips with minimum user intervention.

In addition, TargetLink users have convenient, direct access to SystemDesk’s V-ECU generation and VEOS’ simulation capabilities. This lets them evaluate and test the behavior of TargetLink components as parts of complex systems in early development stages with constant interfaces during the different development stages.

Developing AUTOSAR software with SystemDesk and TargetLink based on the exchange of SWC containers.
RTI AUTOSAR Blockset
MATLAB/Simulink- and AUTOSAR-based co-development, validation and prototyping on MicroAutoBox II

Highlights
- Combination of AUTOSAR software development and model-based controller design in MATLAB/Simulink
- Support of virtual ECUs (V-ECUs) for software validation and rapid control prototyping
- Dedicated AUTOSAR OS with a comprehensive feature set
- Seamless workflow from offline simulation, to rapid control prototyping, to production software development for target ECUs

Application Areas
The RTI AUTOSAR Blockset is an extension to Real-Time Interface (RTI) with a focus on rapid control prototyping and ECU software testing and validation. You can use it to execute AUTOSAR software from individual AUTOSAR SWCs to entire virtual AUTOSAR ECUs on MicroAutoBox II, dSPACE’s compact, in-vehicle, real-time-capable system. Here are some typical application scenarios for the blockset:
- Prototyping and evaluating new Simulink control algorithms that include existing AUTOSAR software components (SWCs)
- Benchmarking new AUTOSAR SWCs on a standardized reference hardware
- Testing and validating AUTOSAR software under real-time conditions before the target ECU becomes available
- Testing and validating AUTOSAR software together with the physical plant, in-lab or in-vehicle

Key Benefits
The RTI AUTOSAR Blockset lets you import AUTOSAR software from SystemDesk into MATLAB/Simulink in the form of virtual ECUs (V-ECUs) and use it on MicroAutoBox II. V-ECUs are realistic representations of future ECU software. By using the RTI I/O libraries, the V-ECU can use the powerful and flexible I/O of MicroAutoBox II. The model-based configuration of the I/O blocks makes setting up a V-ECU with physical I/Os quick and easy. Because of its comprehensive I/O range, the same MicroAutoBox II can be equipped with different V-ECUs to stand in for a multitude of AUTOSAR ECUs.

The RTI AUTOSAR Blockset includes a dedicated AUTOSAR OS for MicroAutoBox II. This OS supports AUTOSAR versions 3.x and 4.x. It conforms to AUTOSAR OS Conformance Class 1, ensuring a high level of compatibility. By supporting advanced OS concepts, such as extended tasks, the OS enables a powerful real-time execution of AUTOSAR software that closely resembles the behavior of production-grade target ECUs. The OS lets you test real-time aspects of the software even before prototypes of a target ECU become available.
Functionality Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
</table>
| AUTOSAR OS on MicroAutoBox II        | - Real-time-capable, preemptive AUTOSAR OS  
- Automatic configuration of OS and BSW services  
- Conforms to AUTOSAR Scalability Class 1 and comprises:  
  - Full OSEK OS conformance (all conformance classes)  
  - Support of all OSEK APIs  
  - AUTOSAR Counter Interface  
  - AUTOSAR Software Free Running Timer Interface (SWFRT)  
  - AUTOSAR Schedule Tables  
  - AUTOSAR Stack Monitoring |
| Measurement and calibration          | - Synchronized measurement of Simulink signals and global V-ECU variables in parallel  
- Standard ASAM A2L file support  
- Extended measurement, calibration and debugging possibilities compared to RTEs on production-grade target ECUs to allow tracking RTE communication |
| Basic Software (BSW) Services        | - ECU State Manager (EcuM)  
- NVRAM Manager (NvM)  
- AUTOSAR CAN stack |
| Supported AUTOSAR versions           | - Versions 3.1, 3.2  
- Versions 4.0.3, 4.1.1, 4.1.2 |
| Support for bypassing                | AUTOSAR software is prepared for virtual ECU bypassing with the RTI Bypass Blockset |

**Working with AUTOSAR on MicroAutoBox II**

The RTI AUTOSAR Blockset lets you import AUTOSAR software from SystemDesk into MATLAB/Simulink in the form of virtual ECUs (V-ECUs) and use it on MicroAutoBox II. V-ECUs contain AUTOSAR application software, the AUTOSAR Runtime Environment (RTE), AUTOSAR Basic Software (BSW), and OS configurations.

The RTI AUTOSAR Blockset automatically configures the AUTOSAR OS and the BSW services on MicroAutoBox II according to the configuration settings. For instance, the AUTOSAR CAN stack is configured automatically based on the configuration you made in SystemDesk. Configuring I/O such as digital or analog input and output ports with production tools is often a very complex and error-prone task. The model-based I/O configuration with the dSPACE RTI I/O blocksets makes setting up a V-ECU with physical I/Os quick and easy. Its comprehensive range of I/O lets you adapt MicroAutoBox II to a wide range of ECUs.

In combination with the RTI AUTOSAR Blockset, MicroAutoBox II can therefore flexibly stand in for a multitude of different AUTOSAR ECUs – long before physical prototypes of an ECU become available (see figure).
A Seamless Development Tool Chain

dSPACE offers a seamless tool chain for working with AUTOSAR software. This tool chain consists of the dSPACE tools TargetLink, SystemDesk®, VEOS®, and the RTI AUTOSAR Blockset for MicroAutoBox II. AUTOSAR SWCs are the building blocks of AUTOSAR software and can be developed in various ways. They can, for instance, be created manually as C source code. Or they can be developed with a model-based production code generator such as dSPACE TargetLink.

With SystemDesk, SWCs can be imported and combined into an AUTOSAR application software. In addition, SystemDesk lets you model entire software architectures, configure the corresponding AUTOSAR OS and BSW services, generate an RTE and export these as V-ECUs. Such V-ECUs can be run and tested on a PC in the offline simulator dSPACE VEOS to validate functional properties of the software. The same V-ECU can also be executed without modifications on MicroAutoBox II by using the RTI AUTOSAR Blockset. Real-time-specific parts of the software behavior can thus be tested and the V-ECU can be validated under realistic operating conditions. With this approach, the behavior of AUTOSAR software can be prototyped together with a physical plant in the laboratory, and experienced directly in the vehicle.

In order to optimize or enhance certain aspects of a V-ECU, the RTI Bypass Blockset allows individual functions and runnables to be replaced with new Simulink models. Hence, you can evaluate and test the behavior of the new models in combination with the existing ECU software. This tool chain covers all phases of function and software development, from control design to software architecture definition and integration and from offline simulation to in-vehicle RCP (see figure). The continuous use of ControlDesk® Next Generation leverages synergies, because the same measurement and calibration data and experiment layouts can be used across all platforms, and even in the production ECU.

![Diagram of the tool chain](image-url)
VEOS®
Platform for PC-based simulation of models and ECU network communication

Highlights
- Early validation of ECU software by PC-based simulation
- Comprehensive, realistic simulation of ECU network communication for CAN and LIN buses
- Seamless integration with RCP and HIL tool chains
- Openness through support of significant standards like AUTOSAR and Functional Mock-up Interface
- Support of multi-model scenarios

Application Areas
dSPACE VEOS is a PC-based simulation platform that promotes the use of virtual validation in the development of electronic control units (ECUs). VEOS makes it possible to simulate a wide range of different models – function models, Functional Mock-up Units (FMUs), virtual ECUs (V-ECUs), and vehicle models – independently of any specific simulation hardware in early development stages. For multi-model scenarios, VEOS supports importing, connecting and running any number of functions and plant models based on Simulink or Functional Mock-up Interface (FMI) 2.0, thereby extending the scope of your applications.

Key Benefits
Running on a standard PC, VEOS gives function developers, software architects and ECU testers numerous new options for virtual validation in an early project phase.
- New functions can be integrated with the overall ECU software to test how they interact.
- A virtual test bench with powerful engine and vehicle dynamics models is permanently available for designing complex controller strategies.
- Complex vehicle and environment models can be integrated with V-ECUs to simulate and test an entire virtual vehicle.
- In preparation for hardware-in-the-loop simulation, models and tests can be created, set up and run on a PC independently of the hardware-in-the-loop (HIL) system.

Systematic Extension to the dSPACE Tool Chain
VEOS works hand in hand with other dSPACE products to provide a complete tool chain for the development and testing process. This means that tools and models which are commonly used in rapid control prototyping and hardware-in-the-loop simulation can also be used in the virtual world. Similarly, layouts from HIL simulation can be reused in PC-based simulation with VEOS and vice versa. VEOS also provides open interfaces to connect and utilize your existing tools.
- Simulink® and dSPACE Run-Time Target for generating C-code-based simulations
- TargetLink® for generating AUTOSAR and non-AUTOSAR simulations based on production code
- SystemDesk® for generating virtual ECUs
- RTI Bypass Blockset for extending V-ECUs with Simulink controller models
- Automotive Simulation Models for complex environment models
- ModelDesk for graphically configuring and parameterizing environment models
- ControlDesk® Next Generation for experimenting and visualizing simulations
- MotionDesk for visualizing simulation scenarios
- AutomationDesk for creating tests and automating simulation runs
Bus Implementation Tools
For combining dSPACE systems with communication networks

- Convenient configuration of restbus setups
- Ideal for handling complex communication setups
- Database file import (LDF, DBC, FIBEX, AUTOSAR)

Application Areas
The bus implementation tools are an extension for Real-Time Interface (RTI) and ConfigurationDesk®. They can be used for combining dSPACE systems with communication networks and configuring the networks. There is a dedicated tool for each bus system:
- RTI CAN MultiMessage Blockset
- RTI LIN MultiMessage Blockset
- dSPACE FlexRay Configuration Package

Key Benefits
The bus implementation tools are a time-efficient and cost-efficient solution for managing complex communication setups, especially in hardware-in-the-loop applications. The tools let you control, configure and manipulate a high number of PDUs, messages and signals with convenient graphical user interfaces. Network configurations can easily be read in from communication matrix description files such as LIN description files (LDF), database container files (DBC), ASAM MCD-2 D (FIBEX) and AUTOSAR System Template files.

Interface Configuration
Configurations are based on imported communication descriptions such as AUTOSAR System Templates. These describe the entire communication in a vehicle. You can select elements and configure them for simulation with the bus implementation tools, whose convenient user interfaces make programming unnecessary.

Parameters and Messages
After creating a configuration with one of the bus implementation tools, you can view it in ControlDesk® Next Generation. The Bus Navigator in ControlDesk provides an easy way to handle bus configurations, including layout generation, monitoring and logging. ControlDesk also makes it easy to create test cases with settings on PDU, message and signal levels.

dSPACE FlexRay Configuration Tool
## Support of AUTOSAR Releases

<table>
<thead>
<tr>
<th>Product</th>
<th>AUTOSAR Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.0</td>
</tr>
<tr>
<td><strong>SystemDesk®</strong></td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>–</td>
</tr>
<tr>
<td>4.4</td>
<td>–</td>
</tr>
<tr>
<td>4.3</td>
<td>–</td>
</tr>
<tr>
<td>4.2</td>
<td>–</td>
</tr>
<tr>
<td>4.1</td>
<td>–</td>
</tr>
<tr>
<td>4.0</td>
<td>–</td>
</tr>
<tr>
<td>3.2</td>
<td>–</td>
</tr>
<tr>
<td>3.1</td>
<td>–</td>
</tr>
<tr>
<td>3.0</td>
<td>–</td>
</tr>
<tr>
<td>2.1</td>
<td>–</td>
</tr>
<tr>
<td><strong>TargetLink®</strong></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>–</td>
</tr>
<tr>
<td>4.0</td>
<td>–</td>
</tr>
<tr>
<td>3.5</td>
<td>–</td>
</tr>
<tr>
<td>3.4</td>
<td>–</td>
</tr>
<tr>
<td>3.3</td>
<td>–</td>
</tr>
<tr>
<td>3.2</td>
<td>–</td>
</tr>
<tr>
<td>3.1</td>
<td>–</td>
</tr>
<tr>
<td>3.0.1</td>
<td>✓</td>
</tr>
<tr>
<td>3.0</td>
<td>✓</td>
</tr>
<tr>
<td>2.3.1</td>
<td>✓</td>
</tr>
<tr>
<td><strong>RTI AUTOSAR Blockset</strong></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>–</td>
</tr>
<tr>
<td>2.1</td>
<td>–</td>
</tr>
<tr>
<td>2.0</td>
<td>–</td>
</tr>
<tr>
<td><strong>CAN MultiMessage Blockset</strong></td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>–</td>
</tr>
<tr>
<td>4.1</td>
<td>–</td>
</tr>
<tr>
<td>4.0</td>
<td>–</td>
</tr>
<tr>
<td>3.0</td>
<td>–</td>
</tr>
<tr>
<td>2.9</td>
<td>–</td>
</tr>
<tr>
<td>2.8</td>
<td>–</td>
</tr>
<tr>
<td>2.7</td>
<td>–</td>
</tr>
<tr>
<td>2.6</td>
<td>–</td>
</tr>
<tr>
<td>2.5.4</td>
<td>–</td>
</tr>
<tr>
<td><strong>LIN MultiMessage Blockset</strong></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>–</td>
</tr>
<tr>
<td>2.4</td>
<td>–</td>
</tr>
<tr>
<td>2.3</td>
<td>–</td>
</tr>
<tr>
<td>2.2</td>
<td>–</td>
</tr>
<tr>
<td>2.1</td>
<td>–</td>
</tr>
<tr>
<td>2.0</td>
<td>–</td>
</tr>
<tr>
<td>1.9</td>
<td>–</td>
</tr>
<tr>
<td>1.8.2</td>
<td>–</td>
</tr>
<tr>
<td><strong>FlexRay Configuration Package</strong></td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>–</td>
</tr>
<tr>
<td>3.5</td>
<td>–</td>
</tr>
<tr>
<td>3.4</td>
<td>–</td>
</tr>
<tr>
<td>3.3</td>
<td>–</td>
</tr>
<tr>
<td>3.2</td>
<td>–</td>
</tr>
<tr>
<td>3.1</td>
<td>–</td>
</tr>
<tr>
<td>3.0</td>
<td>–</td>
</tr>
<tr>
<td>2.6</td>
<td>–</td>
</tr>
</tbody>
</table>
Configuring and generating virtual ECUs (V-ECUs) is the basis for virtual validation of ECU software. You can use SystemDesk or TargetLink for generating implementations of V-ECUs. These V-ECUs are further compiled and built for execution on the PC-based simulation platform VEOS, the hardware-in-the-loop (HIL) simulator SCALEXIO®, or the rapid control prototyping (RCP) platform MicroAutoBox II.

A simulation platform provided with a dSPACE Release can handle implementations of V-ECUs generated by product versions of SystemDesk or TargetLink of the same or prior dSPACE Releases, covering the AUTOSAR Releases supported by these tools. The handling of implementations of V-ECUs is designed to be backward compatible.
dSPACE GmbH, Paderborn, and Elektrobit (EB) Automotive GmbH, Erlangen, are cooperating in the area of producing AUTOSAR-compatible control unit software. In the AUTOSAR software architecture, Elektrobit’s EB tresos® covers the creation and configuration of the basic software, and dSPACE’s SystemDesk® the application layer and the system design. The closely coupled tools give customers a thoroughly tested tool chain for all AUTOSAR development steps, and far-reaching added value. The interplay between the two tools is based on standard AUTOSAR file formats.

**SystemDesk from dSPACE**
SystemDesk is a tool for developing software for both single and multiple electronic control unit (ECU) systems. Its purpose is to support the planning, implementation and integration of complex system architectures. It also simplifies process-oriented development in teams for both manufacturers and suppliers. Interaction with dSPACE’s AUTOSAR-compliant production code generator, TargetLink®, also improves integration with functional development.
**EB tresos®**
EB tresos® from Elektrobit is an integrally designed development and configuration platform for embedded standard software. In EB tresos® Studio, the requirements for the basic software can be configured graphically, and AUTOSAR formats can be read in. The tool then automatically makes the required operating system core and other AUTOSAR-compliant formats available. EB tresos® flexible plug-in design allows users to integrate their own tools and software modules, while ensuring that the individual components interact consistently.

**Interaction Between the Tools**
The starting point for designing the AUTOSAR software architecture of ECUs is SystemDesk. The graphical representation of components simplifies even the development of complex systems. The TargetLink production code generator allows developers to insert the ECU functionality into the application components of the software architecture modeled with SystemDesk. The AUTOSAR basic software core is generated with the tool EB tresos® Studio. The system information that the ECU requires for this is imported from SystemDesk into EB tresos® Studio, and the AUTOSAR basic software is preconfigured correspondingly. Data exchange between the tools is based on AUTOSAR XML files. The completion of the configuration and subsequent generation of source code in EB tresos® Studio leads to the production-ready ECU basic software core, EB tresos® AutoCore. The final AUTOSAR-compliant software can then be integrated into an ECU.

With this coordinated tool chain for software development, developers now benefit even more easily from the advantages of the AUTOSAR standard for automotive development projects. For example, they have a high degree of software interchangeability and reusability.