Virtual Validation with dSPACE
Benefits the whole ECU development process
# Virtual Validation

Virtual Validation __________________________ 3  
Definition of a Virtual ECU ____________________ 4  
Tool Chain for Virtual Validation ______________ 5  
Openness Through Support of Automotive Standards ______ 5  
Standard Interfaces __________________________ 7  

**Use Cases**  
Virtual Test Bench for Function Tests ____________ 8  
Virtual Test Drive ______________________________ 9  
Virtual ECUs Without AUTOSAR _________________ 10  
Function Validation with Virtual ECUs _____________ 11  
Frontloading HIL Tests __________________________ 12  
Preparation of HIL Tests _________________________ 13  
Virtual Bus Simulation __________________________ 14  
Cluster Simulation for Highly Automated Test Drives _______ 15  

**Products for Virtual Validation**  
SystemDesk® V-ECU Generation Module _____________ 16  
VEOS® _________________________________________ 18  
Bus Manager _________________________________ 21  
Automotive Simulation Models (ASM) ______________ 23  
Further Products for Virtual Validation _____________ 25
Virtual Validation
Benefits the whole ECU development process

Highlights

- Develop new functions and validate ECU software much earlier by using virtual ECUs (V-ECUs)
- Use identical tools throughout the entire development process
- Openness through support of automotive standards

The technology of virtual validation means:

- Running PC-based simulations to validate, verify and test ECU software in the form of V-ECUs
- No additional hardware needed
- Preparing and frontloading HIL tests and scenarios on a PC
- Using V-ECUs during function development: verifying new control algorithms in the context of existing ECU software

By using virtual validation, you can perform development, verification and validation tasks much earlier, and also reduce the number of additional test systems and ECU prototypes needed. This answers the need for early simulation that the automotive and aerospace industries are currently experiencing.

dSPACE tools cover all your virtual validation requirements: SystemDesk® for generating virtual ECUs (V-ECUs) from the ECU software architecture, VEOS® for PC-based simulation, as well as software for experiments, visualization, and test automation.

Application Area

There are application areas for virtual validation throughout the whole ECU software development process.

- Function validation with virtual test benches and virtual ECUs, see p. 8 and 11
- Virtual test drives, see p. 9
- Virtual ECUs without AUTOSAR, see p. 10
- Frontloading HIL tests, see p. 12
- Preparing HIL tests, see p. 13
- Virtual bus simulation, see p. 14
- Cluster simulations for highly automated test drives, see p. 15

Key Benefits

- You can develop and test complex new functions in a totally virtual environment instead of on expensive test benches.
- You can simulate a whole ECU on a PC before a prototype is available
- You can prepare simulation models and test libraries on a developer PC, which reduces your preparation time on the HIL simulator.
- You can reuse the experiment software for instrumenting and controlling the HIL simulation when you run simulations on your PC.
Definition of a Virtual ECU

A virtual ECU (V-ECU) is software that emulates a real ECU in a simulation scenario. The V-ECU comprises components from the application and the basic software, and provides functionalities comparable to those of a real ECU. Unlike a soft ECU, which uses only a simplified Simulink®/Stateflow® model, a V-ECU usually has the same software components as the finished ECU. There is no strict dividing line between a soft ECU and a V-ECU, but a V-ECU generally represents the real ECU more realistically.

The abstraction level of a V-ECU depends on its application case:
- V-ECU for developing a single ECU function (contains selected parts of the application software; the RTE and necessary parts of the basic software are provided automatically)
- V-ECU at application level (application software components, RTE, operating system)

Generation of a V-ECU

There are two ways to create a V-ECU, depending on the starting point and project needs, and on whether the development is based on AUTOSAR.

Function and software developers who just have single components can create a V-ECU directly with Simulink® or TargetLink®. The result is a simple V-ECU with only a selected part of the application layer of the ECU software. It enables basic functional tests.

Software integrators who want to test a more complex network of functions can combine software components, functions or just legacy code from different sources in SystemDesk to create the ECU’s software architecture. They then use the SystemDesk V-ECU Generation Module to make a full V-ECU. This contains the run-time environment (RTE) and optional basic software in addition to the application layer. The V-ECUs are used for PC-based simulation with VEOS.

V-ECU including parts of basic software (application software components, RTE, operating system, hardware-independent basic software such as DEM, NVRAM, ECU state manager, COM etc.)
Tool Chain for Virtual Validation

All the products in the dSPACE tool chain for virtual validation interact with each other, covering every aspect from generating a virtual ECU, integrating it with other V-ECUs, building an overall system including environment models and simulating it, to visualizing and automating the simulation. For example, VEOS is an easy-to-use PC-based simulation platform for virtual validation. Models, data, layouts and experiments from ControlDesk® or tests from AutomationDesk can be exchanged between the different simulation platforms without further modifications. Because they support automotive standards such as AUTOSAR, ASAM (XIL API, XCP), or Functional Mock-up Interface (FMI), VEOS and SCALEXIO can easily be integrated into your existing tool chain, so you can continue using all your familiar tools.

Openness Through Automotive Standards

VEOS can easily be integrated in your existing tool chain, as it supports automotive standards. So when you add VEOS to your rapid control prototyping or HIL tool chain to perform PC-based simulation, you can keep your existing tools. By deciding to use dSPACE software and hardware, you gain high flexibility and investment protection for new projects and challenges.

ASAM
In July 2009, ASAM (Association for Standardisation of Automation and Measuring Systems) released the new XIL API standard, defining an interface to connect test automation tools like AutomationDesk with any simulation platform, such as VEOS or SCALEXIO. The standard enables truly platform-independent test development.

The AUTOSAR Standard
AUTOSAR (AUTomotive Open System ARchitecture) is a de-facto open industry standard for automotive electric/electronic (E/E) architectures.
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.

Functional Mock-up Interface (FMI)
The Functional Mock-up Interface (FMI) is an open standard for the exchange and integration of plant models provided by different tool vendors. dSPACE has signed the Codex of PLM Openness and works actively in the ProSTEP Smart Systems Engineering project, the Modelica Association FMI project to further develop the FMI standard, and the Modelica Association project for System Structure and Parameterization of Components for Virtual System Design (SSP). Through these activities dSPACE gathers the necessary knowledge and insights to support our customers in projects using FMI.
Support of Functional Mock-up Interface (FMI)

Efficient Integration of Different Modeling Approaches
Compliance with FMI ensures that models created in different modeling tools can be exported as functional mock-up units (FMUs) based on the FMI standard. Afterwards these FMUs can readily be integrated in simulation environments with FMI support. This simplifies the use of best-in-class tools for specific modeling tasks and the consistent reuse of models in different development phases (e.g., virtual validation and HIL simulations) and different company departments. VEOS supports FMUs based on FMI for Co-Simulation. FMUs can be integrated in a comprehensive virtual validation project together with other FMUs, V-ECUs and Simulink® models. The user workflow in VEOS for importing and connecting these FMUs to V-ECUs and other model interfaces is identical to the user-friendly workflow for V-ECUs and Simulink models. New modeling approaches can therefore be integrated into new or existing projects fast and efficiently. The reliable dSPACE tool chain ensures consistent simulation and parameter access in different use cases. dSPACE ensures smooth interfacing between all the tools in the dSPACE tool chain for virtual validation and HIL projects. This means you can reuse not only the real-time capable FMUs, but also corresponding tests and experiments based on tools such as AutomationDesk and ControlDesk. This completes the FMI reusability approach for virtual validation and HIL use cases.

Services around FMI
dSPACE also provides additional services around FMI to customize the FMI-based workflow for your requirements. For example, you can make existing plant model legacy code available as FMUs to benefit from the advantages of FMI with your existing code base.

Example Workflow
The figure below shows an example workflow in comprehensive virtual validation projects that use FMI. An automatic gearbox is modeled with detailed elastic and frictional behavior via a Modelica-based physical modeling approach and then exported as an FMU based on FMI for Co-Simulation. This FMU is integrated in an overall system model that includes the V-ECU representation of the automatic gearbox controller described in AUTOSAR and the Simulink-based ASM model of the vehicle dynamics. ControlDesk is used to access and monitor all the parameters and variables of the integrated system model simulated by VEOS on a standard PC.
VEOS provides various interfaces for integrating it in an existing tool chain. The software can be coupled with any tool that supports the provided standard interfaces, for example XCP or XIL API.

For measuring, stimulating, and calibrating, VEOS supports description files commonly used in the automotive industry:

- For accessing plant models TRC files are used, like during HIL simulation
- For accessing V-ECUs VEOS uses A2L files; the same format is used for real ECUs

This enables a continuous tool chain using the same configuration and layouts for your calibration and automation software, no matter if for virtual validation or hardware-in-the-loop projects.
Virtual Test Bench for Function Tests: Constantly available on your PC

The advantages:

**VEOS as a virtual test bench**

- PC as virtual test bench available at all times
- Integration of specialized models from the computation department using FMI
- Realistic simulation using virtual ECUs and sophisticated simulation models on the PC
- Use of an integrated tool chain

---

**The challenge: Complex tests on a test bench**

Today’s function development for electronic control units (ECUs) is a complex task in which new functions often have to be tested in real units under test or in the vehicle. Real test benches do allow detailed tests and calculations, but they are cost-intensive and cannot always be adjusted to meet changing demands. In addition, the test benches are not always available to all the developers who need to test new controllers.

---

**The idea: Virtual test benches**

Since all function developers have a PC, using it for function tests is a rather pragmatic approach. The simulation platform dSPACE VEOS® turns the PC into a personal virtual test bench, including simulation models and virtual ECUs. The PC-based simulation of engine effects is accurate enough to test ECU functions in a realistic context. If more complex models have to be used for function validation, the Functional Mock-up Interface (FMI) standard can be used to integrate computational engineering tools into VEOS.

---

**One example: Developing ECU functions for fully-variable valve trains**

To meet the stricter requirements for fuel consumption, new control functions have to be used, such as a fully-variable valve train. However, the new freedom to control valve opening and closing times and the valve lift involves extra work when developing and calibrating engine control functions. With VEOS, functions can first be tested on a PC by using virtual motor models. The design engineer can use special models that HIL test engineers already validated in numerous test runs on a HIL simulator. In addition, the dSPACE experiment software ControlDesk® offers photo-realistic layouts that can be used interactively during running simulations on a PC. The variables and the calibration parameters of the controller and of the controlled system can therefore be accessed during simulation, so they can be integrated effortlessly into an automation or optimization process. The first test drives with new functions can be performed purely virtually before going onto the actual test bench or into the vehicle.
Virtual Test Drive: Test driver assistance functions on your own PC

The advantages: VEOS for virtual test drives
- Use your own PC for closed-loop tests of ADAS functions
- Reuse plant and environment models
- Use models and automated tests seamlessly throughout all development phases
- Support of Simulink®, TargetLink®, legacy C code, AUTOSAR, and FMI

The challenge: Test scenarios not available
Validating ECU functions in realistic test scenarios during the early stages of development is becoming a more common practice. Simple unit tests do not provide enough coverage anymore, especially for safety-critical driver assistance functions which, by nature, are networked with many other systems. In cases like this, function developers must test the functions’ interaction with other control algorithms closed-loop with complex environment models.

The idea: Virtual test drives
The pragmatic approach takes the environment simulation models that already exist for ECU testing and reuses them on the developer PC. dSPACE VEOS® is the bridge between these two worlds. It lets function developers perform virtual tests of functions whenever they want to so that they can easily test many different environment scenarios. This holds the number of real test drives at an affordable level. It also makes reproducing virtual test drives much easier, which is useful for checking a corrected function. VEOS also includes established error analysis methods such as debugging and code coverage, which are not possible in real test drives. If function developers do not have access to the environment models, they can use the dSPACE models that cover many ADAS areas.

One example: Developing complex intersection assistant functions
Performing automated tests of intersection assistance functions on a hardware-in-the-loop simulator involves constructing many test scenarios that describe the exact traffic situations (the course of the road, how many traffic participants there are there, what the roadside looks like, etc.) to test all the participating ECUs in detail. With VEOS, function developers can take the relevant function algorithms that were developed in Simulink or AUTOSAR and simulate them in interaction with these environment models. VEOS provides them the same features for visualization, simulation control, and automated tests as a HIL simulator.
Virtual ECUs Without AUTOSAR: Test software components early on your PC

The advantages:
Virtual ECU tests with VEOS
- Integration tests on the developer’s PC
- Use of established tools for debugging, code coverage, and parameter analysis
- Early integration tests without a hardware prototype
- Virtual ECU tests even without AUTOSAR

The challenge: Integration tests during the development process
The necessity for integrating new functions and validating their interaction with environment models at an early development stage is growing every day. The C code modules of functions that were developed in the C language had to be tested either individually or at a relatively late stage, when they were integrated on real ECU prototypes and tested with a HIL simulator. Potential errors were difficult to detect and correct. Until now.

The idea: Virtual ECUs
A more practical way of testing the interaction of the various functions as early as possible is using virtual ECUs for validation on a PC. Function developers can use dSPACE’s PC-based simulation platform, VEOS®, to test the interaction of their new functions on their PC. In addition, developers can use established methods such as debugging or code coverage to analyze errors and implement the required changes even during the development phase.

One example: Integration tests of TargetLink® module
dSPACE’s production code generator, TargetLink, generates C code modules for a variety of functions. These modules can now be used in SystemDesk for the automated generation of virtual ECUs, including the appropriate task scheduling. In VEOS, developers can connect the virtual ECUs to a plant model to perform the first function tests on the PC.
Function Validation with Virtual ECUs: Testing Function Interactions on your PC

The advantages:
Virtual ECU tests with VEOS
- Continuous integration of ECU software using the PC
- Benefit from familiar debugging, code coverage and parameter analysis settings
- Support of AUTOSAR-based and non-AUTOSAR-based workflows

The challenge: Iterative software updates
Today, simple unit or module tests are no longer sufficient for function validation, because new ECU functions are becoming more and more complex. Some control functions have to be integrated together with the ECU basic software to validate the overall behavior of the ECU software and test it together with other ECUs. At the same time, corrected functions must also be easy to integrate into the overall system and easy to test. This is where real ECU prototypes reach their limits because the flash process required to update them is very time-consuming.

The idea: Software integration tests on the PC
Virtual electronic control units (V-ECUs) provide a more flexible approach. They are generated directly on the developer PC and, to a large extent, contain the same software components and basic software as the final ECU prototype. Software changes and updates can therefore be integrated quickly at any time. With the PC-based simulation platform dSPACE VEOS®, you can validate the overall behavior of the software by using virtual ECUs. Realistic plant models are easy to integrate for software-in-the-loop simulation.

One example: Integrating and testing an ACC ECU
For the development of an ACC (adaptive cruise control) ECU for automatic distance and speed control, three components are integrated into one virtual electronic control unit: distance control, preceding vehicle detection, and user interface control. When testing the interaction of these three components, an error in the distance control is detected and corrected. Since the build process for the generation of V-ECUs is executed entirely on the developer PC, updating the overall system with new functions is easy. At the same time, the simulation itself is performed in a very realistic test environment, because VEOS uses the same vehicle dynamics models for the closed-loop simulation as are used in HIL simulation. The test scenarios can also be easily reproduced after each correction until the desired behavior of the ECU is achieved.
Virtual Validation / Use Case

Frontloading HIL Tests:
Testing virtual ECUs fully automated on a PC

The advantages:
Functional tests on a PC with VEOS
- No ECU prototype necessary
- Quick updates if changes occur
- Easy to duplicate V-ECUs for simultaneous use in different tests
- Increased software quality due to early automated tests

The challenge: Integration tests too late
Until now, automated tests had to wait until a HIL simulator and an electronic control unit (ECU) prototype were available. However, because the functionality of an ECU control is often provided by different development teams, the overall behavior of the ECU must be validated at an early stage. Due to the increasing complexity of these functions and the diverse range of their applications, only automated tests can handle this test scope.

The idea: Virtual ECUs
One solution involves integrating the existing software components to create a virtual ECU (V-ECU). With the PC-based simulation platform VEOS®, this virtual ECU can then be tested comprehensively on a PC in real HIL test scenarios, including automated test sequences. If necessary, two or more virtual ECUs can be combined to test how they communicate in a network. Because it is easy to duplicate virtual ECUs, they can be used simultaneously for different scenarios. That is how functional tests can be frontloaded from the HIL simulator onto a PC. Errors are found earlier and the ECU software quality is already high when the subsequent HIL tests are started.

One example: Integrating an ESP control unit
When an ESP control unit is integrated, the entire function model must be integrated to create an overall function. This involves using dSPACE SystemDesk® to generate a virtual ECU from the AUTOSAR software components (SWCs). First, AUTOSAR compliance and the SWC interfaces are automatically validated with SystemDesk. Then VEOS is used to help validate the overall functionality, including the task scheduling. If the development process is not AUTOSAR-based, the dSPACE tool chain also lets you use an approach based on S-functions or functional mock-up units. In all of these cases, you can reuse the existing HIL test scenarios, sequences, configurations and layouts to validate the V-ECUs.
Preparation of HIL Tests:
Developing and validating tests on the PC

The advantages:
Preparing Hardware-in-the-Loop Tests with VEOS
- Preparing HIL test artifacts such as automated tests, layouts, and model configurations on the PC
- Identifying errors in the HIL tests even before execution
- Maximizing HIL simulator utilization
- Support of the XIL API standard

The challenge: Downtimes of the HIL simulator
HIL simulators should run around the clock for optimal use. But today they are used to prepare, design, and validate tests, which leads to undesirable downtimes. Add to this overnight tests that are terminated prematurely due to faulty test sequences – wasting valuable time and delaying projects.

The idea: Preparing tests on the PC
The PC-based simulation platform dSPACE VEOS® has the same interfaces as a HIL simulator, so you can design and validate test scenarios and layouts for the HIL simulation in advance. In addition, you can develop, parameterize, and test plant models without actually needing a HIL system. VEOS can run automated HIL test simulations to later on avoid termination of the actual HIL tests due to faulty test sequences. Layouts that were previously developed for the HIL system can be reused and adapted to new requirements. Layouts and tests are created once and can be reused across the different development stages to save both time and effort. Frontloading HIL tests yields higher test quality and cuts the total cost of ownership (TCO) by using the HIL simulator more efficiently.

One example: Autonomous parking in a parking garage
To test control functions for autonomous parking in a realistic parking garage environment, the entire parking garage including other vehicles has to be modeled and designed. Different parking scenarios with different available parking spaces and borders require individual HIL test scenarios. With VEOS, you can design models, layouts, and tests on the PC in advance and identify the right HIL test for each project. The simulation models can be parameterized early on while the HIL simulator can continue testing. Testing and validating tests before the actual HIL simulation avoids corrections and terminated overnight tests.
Virtual Bus Simulation: Testing ECUs and their communication on a PC

The Advantages: ECU Tests with Virtual Bus Simulation
- No ECU prototypes or additional hardware necessary due to virtual ECUs
- Independent of real-time limitations
- Use of existing communication matrices and restbus configurations
- Reuse of test and experiment environments

The Challenge: Developing and Validating Distributed Functions in an ECU Network
Some functionalities involve multiple software components (SWCs) that are distributed over various ECUs. These ECUs communicate via bus networks. In order to test these functionalities thoroughly, the complete system, including the communication between the ECUs, has to be simulated. Ideally, these tests should take place as early as possible during the development.

The Idea: Bus Simulation in Early Development Stages
Early tests of the bus communication are possible by using the PC-based simulation platform dSPACE VEOS®. Since virtual ECUs (V-ECUs) are used, the only hardware needed is a standard PC. During ECU software development, the necessary V-ECUs are generated on the basis of an existing communication matrix that defines inter-ECU communication. For each new software iteration, the V-ECUs can be updated easily, ensuring that the communication tests are always up to date. Because V-ECUs and VEOS are purely software-based, they make the simulation independent of real-time limitations. The restbus simulation needed for testing the whole system communication is defined on the basis of the same communication matrix as the V-ECU. A detailed bus monitoring down to the signal level is available for the virtual bus simulation, just as for real-time simulation. This virtual bus communication simulation lets function developers find and fix errors long before the hardware-in-the-loop (HIL) simulation takes place, thus saving valuable time at the HIL simulator. Additionally, all artifacts created for communication tests with VEOS, such as test scripts, bus and other configurations, can be reused for the HIL tests, saving further configuration efforts.

One Example: Early Validation of a Comfort ECU
An existing comfort functionality is extended by a new component that runs on a comfort ECU in the rear of the car. This ECU communicates with the main comfort ECU at the front of the vehicle via CAN. The main ECU has already been tested in the form of a V-ECU. The restbus simulation configuration was generated with the Bus Manager. As soon as the new component is available, e.g., as a Simulink® model, it is used to generate a V-ECU. The bus configuration of the main ECU can be reused, which makes setting up the virtual bus simulation effortless. Because the same communication matrix is used for defining the communication between the V-ECUs and configuring the restbus simulation, the two parts interact seamlessly and do not have to be modified at all. Later in the test process, the already tested restbus simulation can be reused for HIL tests, including the configurations, test layouts and monitorings created for the virtual bus simulation.
Cluster Simulation for Highly Automated Test Drives:
Driving millions of kilometers on your PC

The Advantages:
Test Drives with VEOS
- Comprehensive testing at an early development stage
- Highly scalable due to virtual ECUs
- Deterministic and reproducible test execution
- High test throughput through fast test execution

The Challenge: Millions of Test Kilometers as Fast as Possible
For testing highly automated driving functions or self-driving cars, millions of kilometers of test drives are necessary. These test drives must cover a broad range of roads, weather conditions, and traffic scenarios. Using only hardware-in-the-loop (HIL) simulators for these tests would require a vast amount of HIL systems and real ECU prototypes, which would increase investment costs beyond reason.

The Idea: Numerous Virtual Test Drives with a PC
As a PC-based simulation platform, VEOS offers the ideal solution for a high number of simulation requests. Due to its open automation interface, test scenarios with VEOS can be executed completely automated. You can set up as many VEOS instances in one cluster as needed for the parallel execution of virtual test drives. The PC cluster is controlled by one central unit that schedules the execution of the simulation jobs and test cases. If you use virtual ECUs (V-ECUs), which can be copied easily and distributed throughout the VEOS cluster, VEOS and V-ECUs together make for a highly scalable test system. Additionally, VEOS supports the XIL API for accessing and recording the simulation data. The data can be transferred to a central storage device and can be analyzed later, for example. The PC-based simulation lets you execute the tests faster than in real time. Depending on the number of VEOS instances used, you can complete millions of test drive kilometers each day. Any test drive that fails can be reproduced and debugged in detail, also by using VEOS.

One Example: Testing Development Iterations Continuously
Due to the complexity of a highly automated driving function, the software must be tested extensively at several stages of the development. Every other week, the current software version is integrated into one or more V-ECUs and simulated in the VEOS cluster. The resulting data is loaded to a big data analysis tool for analyzing and comparing it to the previous results. This way, improvements or setbacks are easy to identify and correct, if needed. Performing this many test drive kilometers during the first development phase results in a very high software quality even before starting HIL tests or real test drives.
SystemDesk V-ECU Generation Module
Generation of virtual ECUs

Highlights

- Generating simulation platform-independent V-ECUs for validation and verification
- Creating V-ECUs from external code
- Several editors and process support for V-ECU convenient configuration and generation

Application Areas

For virtual validation applications, the SystemDesk V-ECU Generation Module lets you configure and generate virtual ECUs (V-ECUs). Software integrators who want to test complex networks of functions can combine software components, functions or just legacy code from different sources in SystemDesk to create a full V-ECUs. To test the system’s overall behavior V-ECUs can be used for PC-based simulation with VEOS as soon as the implementation with C code is available.

Key Benefits

- Guided creation of virtual ECUs out of the software architecture
- Support of AUTOSAR-based and non-AUTOSAR-based approaches
- Automatic configuration of the basic software for simple and fast preparation of V-ECUs
- Automatic processes for V-ECU generation possible due to complete automation API
- Comprehensive validation of software architecture model for direct feedback in case of problems

Feature Overview

V-ECUs created with SystemDesk comprise components from the application and the basic software, and provide functionalities comparable to those of real ECUs. The SystemDesk V-ECU Generation Module makes it possible to configure and generate BSW modules, e.g., the RTE, OS, the ECU State Manager or the NVRAM Manager. For configuring the basic software 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. Additionally, it is possible to integrate series BSW into a V-ECU.

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 much easier than in a real ECU. This makes it easy to 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 file, enabling you to fix these problems instead of running into them later during RTE generation. And, thanks to the complete automation API, SystemDesk can also be used in automated processes for generating the V-ECUs.
Use Cases

- Simulation at Software Architecture Level
  The modeled software architectures can be verified and tested, without considering the underlying hardware topology (virtual functional bus (VFB) mode). Hereby functional errors or error behavior in the communication between software components can be found and fixed as soon as possible. SystemDesk provides the basic software modules needed for simulation, such as RTE and OS, and offers options for modeling, configuring and integrating basic software components. Together with the PC-based simulation platform VEOS, software architectures can be simulated early in the development phase.

- Bus Simulation
  You can also distribute software components in SystemDesk to different ECUs, import predefined communication matrices and generate V-ECUs including a COM stack. The V-ECUs can then be connected via a virtual bus. For each ECU, SystemDesk provides the required basic software components. You can use VEOS to simulate the bus communication between the ECUs. Arbitration effects, transfer times and transmission delays of bus messages or ECU and bus breakdowns can be simulated for CAN and LIN buses and verified in an early phase of development. Networks including FlexRay buses can be modeled and simulated as idealized buses.

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>SystemDesk 4.x</th>
<th>SystemDesk 3.x</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemDesk Modeling Module</td>
<td>Modeling AUTOSAR compositions and systems containing one or more networked AUTOSAR ECUs</td>
<td>✔️</td>
<td>✔️</td>
<td>SYD_MOD</td>
</tr>
<tr>
<td>SystemDesk V-ECU Generation Module</td>
<td>Generating virtual ECUs (V-ECUs) including nonoptimized RTE and building simulation systems for offline simulation with VEOS</td>
<td>✔️</td>
<td>Add-on for SystemDesk Modeling Module</td>
<td>SYD_GEN</td>
</tr>
<tr>
<td>SystemDesk RTE Generation Module</td>
<td>Generating optimized RTE production code</td>
<td>–</td>
<td>Add-on for SystemDesk Modeling Module</td>
<td>SYD_RTE</td>
</tr>
</tbody>
</table>
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) 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.
- Software-software integration can be tested from the component to the system level.
- Sophisticated environment models can be integrated with virtual ECUs to validate complex controller strategies or simulate and test entire virtual vehicles.
- 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
- Automotive Simulation Models for sophisticated environment models
- ModelDesk for graphically configuring and parameterizing environment models
- ControlDesk® for experimenting and visualizing simulations that monitor bus communication with ControlDesk’s Bus Navigator
- MotionDesk for visualizing simulation scenarios
- AutomationDesk for creating tests and automating simulation runs
Functionality Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC-based simulation</td>
<td>Simulation of heterogeneous models, from function models to virtual ECUs, bus systems, and vehicle models</td>
</tr>
<tr>
<td></td>
<td>No additional hardware needed for simulation</td>
</tr>
<tr>
<td>Simulink support</td>
<td>Simulation of Simulink function models and Simulink environment models (such as dSPACE ASMs)</td>
</tr>
<tr>
<td></td>
<td>Support of S-functions, model referencing, multitasking, triggered or enabled subsystems and tunable parameters</td>
</tr>
<tr>
<td></td>
<td>Simulation of Simulink implementation containers (SICs) generated from different projects</td>
</tr>
<tr>
<td>FMI support</td>
<td>Simulation of Functional Mock-up Units (FMUs) based on the Functional Mock-up Interface (FMI) for Co-Simulation</td>
</tr>
<tr>
<td></td>
<td>Support of FMI 2.0 functionalities and access/monitor support for all variables and parameters defined by an FMU</td>
</tr>
<tr>
<td>TargetLink support</td>
<td>Simulation of TargetLink-generated code as virtual ECUs (V-ECUs) or FMUs</td>
</tr>
<tr>
<td></td>
<td>Support for AUTOSAR as well as non-AUTOSAR TargetLink code</td>
</tr>
<tr>
<td>AUTOSAR support</td>
<td>Simulation of virtual ECUs generated by SystemDesk</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR-compliant operating system</td>
</tr>
<tr>
<td></td>
<td>Support of AUTOSAR basic software modules</td>
</tr>
<tr>
<td></td>
<td>Support of AUTOSAR R4</td>
</tr>
<tr>
<td>Bus simulation</td>
<td>Simulation of ECU network communication on CAN and LIN buses, including messages, scheduling and arbitration</td>
</tr>
<tr>
<td></td>
<td>Idealized bus simulation for FlexRay</td>
</tr>
<tr>
<td>XIL API support</td>
<td>Support of XIL API Model Access Port</td>
</tr>
<tr>
<td>XCP support</td>
<td>Access to Simulink and TargetLink models as well as V-ECUs via XCP on Ethernet</td>
</tr>
<tr>
<td>Debugging</td>
<td>C code debugging during a running simulation</td>
</tr>
<tr>
<td>Code coverage</td>
<td>Analyzing the extent to which code has been tested with CTC++ from Testwell</td>
</tr>
<tr>
<td>Tool chain integration</td>
<td>Off-the-shelf integration into the dSPACE rapid control prototyping (RCP) and hardware-in-the-loop (HIL) tool chain</td>
</tr>
</tbody>
</table>

Order Information

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline simulation of one or more Simulink models or FMUs</td>
<td>Basic variant</td>
<td>VEOS_BASE</td>
</tr>
<tr>
<td></td>
<td>Simulation of several Simulink models or FMUs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The dSPACE Run-Time Target is needed for generating Simulink implementation containers (SICs) from Simulink models (Simulink Coder required)</td>
<td></td>
</tr>
<tr>
<td>Offline simulation of one or more virtual ECUs</td>
<td>Add-on to VEOS_BASE</td>
<td>VEOS_ECU</td>
</tr>
<tr>
<td></td>
<td>Simulation of Simulink models or FMUs together with V-ECUs</td>
<td></td>
</tr>
<tr>
<td>Offline simulation of virtual ECUs with virtual CAN or LIN bus simulation</td>
<td>Add-on to VEOS_ECU</td>
<td>VEOS_CAN</td>
</tr>
<tr>
<td></td>
<td>Connecting SystemDesk V-ECUs via CAN or LIN bus</td>
<td>VEOS_LIN</td>
</tr>
</tbody>
</table>
Features and Benefits

PC-Based Simulation of Heterogeneous Models
With VEOS, you can simulate Simulink and TargetLink models, FMUs, AUTOSAR software components, virtual ECUs, and ECU networks in one single environment right on your PC. This enables a fast integration and validation process for your ECU software very early in the development process. You will find errors long before the first hardware prototype exists. Another advantage of a PC-based simulation platform is that the parameters, models and results can easily be exchanged between different kinds of user groups in the development process. An error which has been found by a software architect, integrator or tester is much easier for function developers to understand, investigate and fix if they can use the same simulation and testing environment.

Convenient Model Exchange
To make exchanging simulation models easy, dSPACE offers a Model Interface Package for Simulink® (MIPS) for generating Simulink implementation container (SIC) files. With the free-of-charge MIPS, modeling experts can generate the (C code) SIC file with Simulink Coder, without needing a VEOS or ConfigurationDesk license. Out of their Simulink models and together with dSPACE Run-Time Target, they can generate code and create ZIP files that contain all the necessary code and artifacts for executing the models on different simulation platforms, such as VEOS and SCALEXIO®. Model integrators using SIC files do not have to generate code again for building the simulation. Using SICs therefore significantly reduces the amount of time needed for reusing the SICs in different projects.

Comprehensive Bus Simulation
Using VEOS, you can also simulate a network of AUTOSAR virtual ECUs. These include a realistic operating system and can be extended with basic software such as DEM, NVRAM or the ECU state manager as the simulation scenario requires. A predefined COM stack can be included or configured for a virtual ECU in SystemDesk. CAN and LIN buses and their bus-specific effects can be simulated with VEOS on a PC without additional hardware. This gives you precise simulation of distributed functions, including ECU network communication, very early in the development process.

Openness Through Automotive Standards
VEOS can easily be integrated into your existing tool chain, as it supports automotive standards such as
- ASAM
- AUTOSAR
- Functional Mock-up Interface (FMI)

So when you add VEOS to your rapid control prototyping or HIL tool chain to perform PC-based simulation, you can keep your existing tools. By deciding to use VEOS, you gain high flexibility and investment protection for new projects and challenges.
Bus Manager
Configuration for LIN, CAN, and CAN FD bus simulation

Highlights
- One configuration tool for different bus systems
- Work with several communication matrices for one configuration
- Customizable restbus configuration with tool automation interface

Application Areas
The dSPACE Bus Manager is the central tool for configuring bus communication for simulation purposes e.g., restbus simulation, and for implementing the bus communication in real-time applications for dSPACE SCALEXIO® systems. It supports different bus systems, such as LIN, CAN, and CAN FD. The Bus Manager is available as an add-on for dSPACE ConfigurationDesk® to configure hardware-in-the-loop real-time (HIL) applications for dSPACE SCALEXIO hardware. You can also use a stand-alone version to configure PC-based restbus simulations with VEOS®.

Key Benefits
- One tool for homogeneously configuring several bus systems at the same time
- Easy bus configuration via drag & drop
- Work with several communication matrices for one configuration
- Modeling-tool-independent model interface
- Tool automation interface
- Consistent bus simulation for CAN and LIN with VEOS®

Workflow
The Bus Manager offers a convenient and straight-forward workflow for implementing bus simulations. In general, the same workflow applies to virtual validation scenarios with VEOS and for HIL tests using SCALEXIO:
- Import one or more bus communication matrices. All relevant information is extracted automatically for the subsequent bus configuration.
- Create a bus configuration. For the configuration, different views for the dedicated tasks are available.
- Assign the communication matrices completely or in part to the bus configurations.
- Specify the real-time hardware access (this step only applies to HIL tests).
- If required, you can configure different parameters and properties of the simulated elements. For example, you can enable the access to signal values during run time via experiment software such as ControlDesk®. If the simulation requires signals whose values must change dynamically during run time, you can use behavior models, e.g., MATLAB/Simulink behavior models or Functional Mock-up Units (FMUs) to use behavior models designed in another modeling tool.
- Finally, start the build process, and download (applies only to HIL tests) and execute the real-time application.
Functionality Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td>- Import of several communication matrices, such as AUTOSAR system description files, FIBEX, DBC, and LDF files</td>
</tr>
<tr>
<td></td>
<td>- Versatile communication matrix visualization with different views, folding, filtering, searching</td>
</tr>
<tr>
<td></td>
<td>- Easy restbus configuration across communication matrices via drag &amp; drop</td>
</tr>
<tr>
<td></td>
<td>- Selectable and customizable restbus configuration view</td>
</tr>
<tr>
<td></td>
<td>- Tree view and property grid for bus-element-based configuration</td>
</tr>
<tr>
<td></td>
<td>- Table view for multiple-element configuration</td>
</tr>
<tr>
<td></td>
<td>- Configuration of the bus hardware</td>
</tr>
<tr>
<td></td>
<td>- Modeling-tool-independent model interface (model port blocks)</td>
</tr>
<tr>
<td></td>
<td>- Support of Functional Mock-up Units (FMUs)</td>
</tr>
<tr>
<td></td>
<td>- Convenient update of existing configurations with new communication matrices</td>
</tr>
<tr>
<td><strong>LIN</strong></td>
<td>- Send and receive of unconditional, event-triggered and sporadic frames/PDUs</td>
</tr>
<tr>
<td></td>
<td>- Generation of LIN schedules</td>
</tr>
<tr>
<td><strong>CAN</strong></td>
<td>- Triggered/cyclic send and receive of CAN frames/PDUs</td>
</tr>
<tr>
<td></td>
<td>- Support of multiplexed PDUs</td>
</tr>
<tr>
<td><strong>CAN FD</strong></td>
<td>- Triggered/cyclic send and receive of CAN FD frames/PDUs</td>
</tr>
<tr>
<td></td>
<td>- Support of multiplexed PDUs</td>
</tr>
<tr>
<td><strong>Bus simulation</strong></td>
<td>- Export of bus configurations as bus configuration containers for import into VEOS</td>
</tr>
</tbody>
</table>

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Manager (add-on for ConfigurationDesk)</td>
<td>BUS_MANAGER_BASE</td>
</tr>
<tr>
<td>Bus Manager (stand-alone version for use with VEOS)</td>
<td></td>
</tr>
</tbody>
</table>

Relevant Software and Hardware

<table>
<thead>
<tr>
<th>Software</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required</strong></td>
<td></td>
</tr>
<tr>
<td>ConfigurationDesk (for SCALEXIO)</td>
<td>Please inquire</td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td>ConfigurationDesk CAN Module</td>
<td>CFD_I_CAN</td>
</tr>
<tr>
<td>ConfigurationDesk LIN Module</td>
<td>CFD_I_LIN</td>
</tr>
<tr>
<td>VEOS with CAN support</td>
<td>VEOS_CAN</td>
</tr>
<tr>
<td>VEOS with LIN support</td>
<td>VEOS_LIN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td>DS2671 Bus Board</td>
<td>DS2671</td>
</tr>
<tr>
<td>DS2680 I/O Unit (with bus support)</td>
<td>DS2680_2672</td>
</tr>
<tr>
<td>DS6301 CAN/LIN Board</td>
<td>DS6301</td>
</tr>
</tbody>
</table>
Automotive Simulation Models (ASM)
Tool suite for simulating the engine, vehicle dynamics, electrical system, and traffic

Highlights
- Open MATLAB®/Simulink® models
- For ECU testing and function development
- Intuitive graphical parameterization, and road, maneuver, and traffic creation in ModelDesk

Application Areas
The Automotive Simulation Models (ASM) are a tool suite of open Simulink models for the real-time simulation of passenger cars and trucks as well as their components. They are used as plant models for the development and testing of engine controls, vehicle dynamics controls, on-board power electronics and driver assistance systems. The ASMs typically run on a dSPACE Simulator/SCALEXIO® for hardware-in-the-loop (HIL) testing of electronic control units (ECUs) or during the design phase of controller algorithms for early validation by offline simulation.

Key Benefits
All the Simulink blocks in the models are visible, so it is easy to add or replace components with custom models to adapt the properties of modeled components perfectly to individual requirements. The ASMs’ standardized interfaces make it easy to expand a single model such as an engine or body, or even create a whole virtual vehicle. Roads and driving maneuvers can be easily and intuitively created using graphical tools with preview and clear visualization.

Seamless throughout the entire test process
The Automotive Simulation Models are used during the design phase of controller algorithms for early validation with VEOS. Later in the development, the same simulation models are used for hardware-in-the-loop simulation. The reuse of simulation models throughout the entire development and test process reduces the necessary configuration effort. It also results in consistent simulation scenarios, which makes it easier to compare test results. The various and extensive application scenarios for ASM are represented in the use cases for VEOS.

Please see pages 8 to 15 for more detailed information or visit www.dspace.com/go/viva_usecases
ASM Tool Suite

Combining Simulation Models
The dSPACE Automotive Simulation Models consist of various packages and libraries for specific application areas. They can be combined as needed to create the simulation model for a specific project.

Parameterization and Visualization
For parametrizing the simulation models, dSPACE ModelDesk is used. The graphical user interface also provides project handling and allows parameter sets to be downloaded for offline and online simulations. During the simulation run itself, dSPACE MotionDesk visualizes the simulation in a virtual world that exactly represents the simulation scenario.
Further Products for Virtual Validation

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
</table>
| TargetLink       | Production code generation software  
                        AUTOSAR software components developed with TargetLink can be exported to SystemDesk and integrated into virtual ECUs there.  
                        Offers a direct link to VEOS for simulating the generated production code as a virtual ECU in offline simulation scenarios. |
| ConfigurationDesk| Configuration and implementation software for dSPACE SCALEXIO hardware  
                        Configuring real-time applications graphically  
                        Managing signal paths between external devices (like ECUs or loads) and behavior model interfaces  
                        Implementing behavior model code and I/O function code on dSPACE hardware  
                        Managing multicore applications and importing virtual ECUs |
| ModelDesk        | Software for parameterizing the Automotive Simulation Models graphically  
                        Parameterization during online (simulator) and offline (Simulink) simulations  
                        Managing parameter sets and entire projects |
| ControlDesk®     | Universal, modular experiment and instrumentation software for accessing simulation platforms, such as VEOS and SCALEXIO  
                        Experimenting and visualizing the simulations  
                        Access to all variables of the virtual ECUs and environment models  
                        Powerful layouting, instrumentation, measurement and postprocessing  
                        Failure simulation model for controlling failure insertion units in real time  
                        Monitoring bus communication  
                        Integrated ECU calibration, measurement and diagnostics access (CCP, XCP, ODX) |
| AutomationDesk   | Environment for powerful and convenient test automation  
                        Automation of virtual ECUs tests  
                        Connects to VEOS and SCALEXIO simulation platforms via XIL/HIL API |
| Real-Time Testing| Real-time tests are executed synchronously with the simulation model  
                        Model variables can be observed and changed in every simulation step  
                        Test programming via standard Python scripting language |
| MotionDesk       | Visualizing driving maneuvers for vehicle dynamics simulation  
                        Perfectly suited for visualizing ADAS scenarios, with a new rendering engine and features like rainfall and snow |
| SYNECT®          | Systematically planning and controlling test execution  
                        Managing test data and scenarios |