SystemDesk

- AUTOSAR authoring tool for ECU software architectures
- Generation of virtual ECUs with and without AUTOSAR
- Comprehensive rule set for automatic validation
SystemDesk

Modeling system architecture and generating virtual ECUs

Highlights

- ECU software architecture and system development
- Efficient modeling, including plausibility checks and correction suggestions
- Flexible generation of virtual ECUs, e.g., with your own basic software, for early validation with VEOS
- Complete and documented API for automation

Application Areas

SystemDesk can be used for two application areas. As a system architecture tool, it provides sophisticated and extensive support for modeling AUTOSAR architectures and systems for application software. Comprehensive graphical support facilitates the first use but also enables efficient and error-reduced working in large-scale projects. The second application area creating virtual ECUs (V-ECUs) for validating ECU software at an early development stage. V-ECUs contain series code for the functionalities to be tested, either at application level only or also including basic software. They can then be used as unit under test with the dSPACE simulation platforms, such as the PC-based simulation platform VEOS.

Key Benefits

- Efficient use due to convenient dialogs, editors, and diagrams
- Easy and fast generation of V-ECUs based on AUTOSAR and non-AUTOSAR software for validation purposes
- V-ECUs including application software and dSPACE basic software for testing application software or your own basic software for integration tests
- Reliable project quality by means of comprehensive rule-based validation of consistency and correctness
- Convenient development process due to the automation of recurring tasks or remote control with third-party tools via the automation API

AUTOSAR and Non-AUTOSAR Support

The generation of V-ECUs is based on AUTOSAR-compliant and non-AUTOSAR-compliant code. SystemDesk offers support for both approaches.

For more information, please see page 9.

1) SystemDesk 3.x can be used for projects based on AUTOSAR R3.
Functionality Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR modeling</td>
<td>■ Modeling of software architectures and systems according to both Classic AUTOSAR and Adaptive AUTOSAR standard</td>
</tr>
<tr>
<td></td>
<td>■ Clear overview due to graphical modeling with diagrams</td>
</tr>
<tr>
<td></td>
<td>■ Sophisticated editors and dialogs for easy and faultless data entry</td>
</tr>
<tr>
<td></td>
<td>■ Data exchange: flexible and easy-to-configure import and export of ARXML files</td>
</tr>
<tr>
<td></td>
<td>■ Complete support of the AUTOSAR meta model¹)</td>
</tr>
<tr>
<td></td>
<td>■ Import of communication matrices (DBC, LDF, FIBEX)</td>
</tr>
<tr>
<td>V-ECU generation</td>
<td>■ V-ECU generation based on AUTOSAR-compliant and non-AUTOSAR-compliant code</td>
</tr>
<tr>
<td></td>
<td>■ Creating V-ECUs for the AUTOSAR Adaptive Platform</td>
</tr>
<tr>
<td></td>
<td>■ For testing application software, V-ECUs can include dSPACE basic software</td>
</tr>
<tr>
<td></td>
<td>■ For testing basic software or functions that require realistic basic software functions, production basic software can be integrated</td>
</tr>
<tr>
<td></td>
<td>■ For testing application software, you can use the mostly automatic configuration and generation of required basic software modules</td>
</tr>
<tr>
<td></td>
<td>■ Integration of your own basic software modules (optional), e.g., for diagnostics tests and test of the complete ECU software</td>
</tr>
<tr>
<td></td>
<td>■ Now supporting basic software based on the ICU, AUTOSAR MCAL modules for VEOS</td>
</tr>
<tr>
<td></td>
<td>■ Generation of A2L files including memory sections</td>
</tr>
<tr>
<td></td>
<td>■ With a wrapper, existing Classic AUTOSAR Software Components can be turned into Adaptive Applications</td>
</tr>
<tr>
<td>Process support</td>
<td>■ Comprehensive rule-based validation of the system architecture and ECU software regarding consistency, correctness, and completeness</td>
</tr>
<tr>
<td></td>
<td>■ Easy data exchange with dSPACE’s code generator, TargetLink</td>
</tr>
<tr>
<td></td>
<td>■ Usable with every AUTOSAR-compliant basic software configuration tool</td>
</tr>
<tr>
<td></td>
<td>■ Support of Classic AUTOSAR Revisions R4.4.0, R4.3.1, R4.3.0, R4.2.2, R4.2.1, R4.1.3, R4.1.2, R4.1.1, R4.0.3, and R4.0.2</td>
</tr>
<tr>
<td></td>
<td>■ Support of Adaptive AUTOSAR Release 18.10</td>
</tr>
<tr>
<td></td>
<td>■ Completely documented automation API for automating tasks such as report generation, software architecture creation, and automatic V-ECU generation</td>
</tr>
<tr>
<td></td>
<td>■ Hyperlink connection between model elements and single requirements in a requirements management tool, such as IBM® Rational® DOORS®</td>
</tr>
<tr>
<td></td>
<td>■ Scripts running on Python 3.6</td>
</tr>
</tbody>
</table>

¹) SystemDesk 5.x only.

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemDesk Modeling Module (p. 4)</td>
<td>Modeling AUTOSAR compositions and systems containing one or more networked AUTOSAR ECUs</td>
<td>SYD_MOD</td>
</tr>
<tr>
<td>SystemDesk V-ECU Generation Module</td>
<td>Generating V-ECUs for application software tests or with your own basic software and support of AUTOSAR- and non-AUTOSAR-based ECU code</td>
<td>SYD_GEN</td>
</tr>
</tbody>
</table>

Relevant Software

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>TargetLink</td>
</tr>
<tr>
<td></td>
<td>■ Production code generation</td>
</tr>
<tr>
<td></td>
<td>■ Integration of models, functions, and production code in software components and architectures designed in SystemDesk</td>
</tr>
<tr>
<td>Optional</td>
<td>VEOS</td>
</tr>
<tr>
<td></td>
<td>■ Offline simulation of virtual ECUs</td>
</tr>
<tr>
<td></td>
<td>■ Offline simulation of Simulink® plant models together with V-ECUs generated in SystemDesk</td>
</tr>
</tbody>
</table>
Working with SystemDesk

SystemDesk supports the development process with numerous features for intuitive working:

- **AUTOSAR-Compliant Modeling**
  Use SystemDesk’s authoring capabilities to create and edit software architecture and system descriptions according to the AUTOSAR standard.

- **Generation of Virtual ECUs for Simulation**
  Use SystemDesk to generate virtual ECUs (V-ECUs) that include dSPACE’s or your own basic software modules to get a realistic representation of the real ECU. The V-ECUs are based on AUTOSAR- or non-AUTOSAR-compliant code. Simulate the V-ECU with VEOS to validate your software in a PC-based simulation.

- **Dialogs and Diagrams**
  For an easy and efficient capture and configuration of common AUTOSAR data, SystemDesk offers convenient, preconfigured dialogs. Additionally, special diagrams of the software architecture and system elements help keep an overview of your model.

- **Data Exchange with TargetLink**
  You can exchange data seamlessly between SystemDesk and dSPACE’s production code generator, TargetLink. This gives you a secure process for including the actual functionality in the software architecture in a model-based development environment.

- **Project Validation**
  Use SystemDesk’s validation functionality to ensure that your AUTOSAR architecture is complete and consistent before giving it to a project partner for basic software (BSW) configuration and before generating a V-ECU – and save time by finding problems as early as possible. You can define your own validation rules to also check compliance with company-specific project requirements.

- **Complete and Documented Tool API**
  Use the open tool API to automate recurring design tasks and extend the features of SystemDesk according to project-specific requirements or even use it to create a completely automated tool chain for V-ECU generation.

- **Application Software and BSW Integration**
  For testing the ECU software, you can either combine your application software with dSPACE basic software (BSW) modules or import your own BSW. Importing your own production BSW lets you test your basic software or functions that require realistic BSW input, such as diagnostics functions. If your test focus is the application software, using the dSPACE BSW is more convenient, as it is automatically generated and configured according to the systems requirements.

SystemDesk Modeling Module

**Application Areas**

The SystemDesk Modeling Module enables the modeling of Classic AUTOSAR and Adaptive AUTOSAR software architectures and systems containing one or more networked AUTOSAR ECUs. It supports the software architecture development process from modeling software components (SWCs) to specifying system aspects including the hardware topology and network communication.

**Key Benefits**

- Easy workflow and intuitive working environment
- Easy start in working with AUTOSAR
- Ideally suited for large production projects
- Reliable data exchange between different project members
Example Workflow: 
Modeling an ECU Software Architecture According to AUTOSAR

- **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 with a graphical representation of the SWCs and their interconnections. New SWCs as well as their ports and connections are modeled in this diagram, providing developers with an intuitive working method and offering a well-defined overview of even the most complex system.

- **Designing Software Components**
  The next step is to define and configure the internal behavior of the software components. Among others, this comprises 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 offer guidance so you can enter data easily and avoid potential errors. Additionally, several integrated validation procedures make sure the modeled elements and references are AUTOSAR-compliant and consistent.

- **Data Exchange with TargetLink**
  After the software components are designed, their functions can be implemented from any C code generation tool. SystemDesk is closely coupled with dSPACE’s production code generator TargetLink. Both tools support an SWC container concept for transferring all required data. All the files in this container that belong to a modeled software component, such as ARXML files for the software component, data types and interfaces, are exported from SystemDesk. TargetLink users import the SWC container and use the defined elements to automatically create an AUTOSAR frame model for the components. Then they develop the actual functionality and generate AUTOSAR-compliant code. This code is transferred back to SystemDesk together with any generated ASAP2 files, plus the ARXML files with additional implementation information. The SWC containers are handled and merged in the SWC Container Manager, a specialized tool with a graphical user interface which can be opened from SystemDesk and TargetLink.

Once the options for the export are defined, such as which files and data to include, the following import/export just requires a few clicks.
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 designing the software components.
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.

Data Transfer to Basic Software Configuration Tools
To develop the basic software components for the final ECU, the system description modeled in SystemDesk has to be transferred to a basic software configuration tool. SystemDesk can be connected to any AUTOSAR-compliant basic software configuration tool for generating and configuring basic software modules such as OS and COM. You can flexibly define which data to include in the export and save this configuration for the next export. Additionally, before you export your system description, an integrated validation in SystemDesk enables you to check the model for correctness and integrity against project specifications.
Cooperation with EB Automotive

dSPACE and EB Automotive cooperate to offer an optimal tool chain for developing AUTOSAR-based ECU software: dSPACE TargetLink, dSPACE SystemDesk, and EB tresos® Studio.

EB tresos® Studio allows complete ECU basic software configuration, integration and validation in one single tool environment. Based on the system description files exported from SystemDesk, the parameters for the basic software can be configured in EB tresos® Studio. The tool automatically generates the implementation code for the operating system and other AUTOSAR-compliant basic software modules. Together with the C code for the SWCs, the implementation code from EB tresos® Studio makes up the final code for the ECU.

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). To create a V-ECU, software components, functions or C code from different sources can be combined in SystemDesk. To test the system’s overall behavior, V-ECUs can be used for PC-based simulation with VEOS as soon as the C code implementation is available.

Depending on your test focus, you can do the following:

- Create V-ECUs from AUTOSAR application software by additionally integrating dSPACE basic software into the V-ECU
- Create V-ECUs from complete hardware-independent AUTOSAR software, including application and basic software
- Create V-ECUs from non-AUTOSAR code
- Create V-ECUs from Adaptive applications including Middleware
- Create V-ECUs for Linux applications

Key Benefits

- Support of Classic AUTOSAR, Adaptive AUTOSAR, and non-AUTOSAR-based approaches
- Guided creation of V-ECUs on the basis of the AUTOSAR software architecture
- Automatic configuration of dSPACE basic software for the simple and fast preparation of V-ECUs on-application level
- Automatic configuration of production BSW that can be generated with third-party BSW generator tools and used to generate V-ECUs (available for some modules)
- Integration of production BSW above the Microcontroller Abstraction Layer (MCAL)
- Automatic processes for V-ECU generation possible due to a complete automation API
- Comprehensive validation of the software architecture model for direct feedback in case of problems
Definition of a Virtual ECU
A virtual ECU (V-ECU) is software that represents a real ECU in a simulation. Unlike a soft ECU, which uses only a simplified Simulink®/Stateflow® model, a V-ECU consists of real production code. It consists of code only and does not require any special hardware. In SIL scenarios, V-ECUs are used instead of real ECUs or controller models.

V-ECUs can have different levels of abstraction, depending on what they are used for:
- V-ECUs at application level include selected parts of the application software and an operating system.
- For AUTOSAR use cases, the RTE and required parts of the basic software are added, which are typically provided by dSPACE.
- V-ECUs can also include the application software and parts of the production basic software, such as Dem, NvM, and COM.
- V-ECUs can include the complete application software and hardware-independent basic software, except modules for the Microcontroller Abstraction Layer (MCAL).
- For Adaptive AUTOSAR use cases, V-ECUs contain adaptive applications and a middleware.

Feature Overview
V-ECUs created with SystemDesk can include components of the application and the basic software as well as functionalities comparable to those of real ECUs. The SystemDesk V-ECU Generation Module lets you generate V-ECUs for different use cases. For Classic AUTOSAR use cases, you can configure and generate BSW modules, e.g., the RTE, OS, the ECU State Manager or the NVRAM Manager. In order to generate V-ECUs as fast and conveniently as possible, 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. You can integrate production BSW into V-ECUs if you want to test the complete hardware-independent ECU software. You can integrate just some BSW functionalities or the complete ECU code on the basis on the dSPACE MCAL modules for VEOS. For Adaptive AUTOSAR use cases, dSPACE provides mechanisms to generate the required artifacts for the applications, on the basis of service interfaces. Middleware can be integrated in SystemDesk, which makes V-ECU generation convenient. Preparing the V-ECU for connection to plant models is easy with SystemDesk, both on the RTE and the MCAL level. You can even include intervention points to insert errors for stimulating the RTE, which is much easier than in a real ECU. This makes it easy to test the application software in various error scenarios.

SystemDesk not only lets you configure and generate V-ECUs. It also supports you during the whole development process. A powerful validation functionality lets you check 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.

Integration into the Development Tool Chain
V-ECUs can be simulated with the PC-based simulation platform dSPACE VEOS for an early verification of the ECU software. VEOS offline simulation can be used for simulating single V-ECUs and ECU networks, optionally including bus simulation. The V-ECUs can also be executed on real-time platforms: MicroAutoBox II for rapid control prototyping applications and SCALEXIO for hardware-in-the-loop tests. Please contact dSPACE for further information. Regardless of the simulation platform you work with, you can use ControlDesk for experimenting and visualizing the simulation, AutomationDesk for test automation, and ModelDesk for model and environment parameterization. Since VEOS supports various interfaces, it can easily be integrated into an existing tool chain. This means you can create and test layouts and automated tests for the HIL simulation on VEOS and reuse them on your HIL system.
**V-ECUs Based on Non-AUTOSAR Code**

If your ECU software is based on non-AUTOSAR code, SystemDesk also lets you generate V-ECUs. For this, you need the code of the functionalities to be included in the V-ECU and a list of tasks as well as functions to be called in these tasks. If variables have to be measured or calibrated, the corresponding A2L files have to be provided as well. SystemDesk generates a V-ECU with realistic behavior, e.g., realistic scheduling.

Please contact dSPACE for further information.

---

**Use Cases**

- **Simulation at the Application Software Level**
  With SystemDesk and VEOS, you can verify and test application software in an early phase, even without considering the underlying hardware topology (virtual functional bus (VFB) mode). Thus, 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 support for conveniently configuring BSW modules. Multiple V-ECUs can be connected at the signal level and simulated together.

- **Tests of Diagnostics Functions**
  You can frontload tests of diagnostics functions from the HIL simulator to VEOS. For this, the V-ECU has to include the application software as well as specific BSW modules, such as the COM stack (CAN), NvRAM, or diagnosis modules (Dem). As in HIL tests, you can use dSPACE ControlDesk to access the diagnostics functions while simulating them with VEOS.

- **Bus Simulation**
  You can also import predefined communication matrices and generate V-ECUs including a COM stack. The V-ECUs can then be connected with each other or with a restbus model 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.

- **ECU Integration Tests**
  The focus of ECU integration tests is to simulate and test the complete ECU code, including the hardware-independent BSW modules. By using V-ECUs, you can perform these tests on your PC and immediately debug them, if necessary. You also benefit from a continuous integration, meaning that new functions are integrated and can be tested as soon as they are available.