ControlDesk

Electrical Error Simulation via XIL API EESPort

For ControlDesk 7.4

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

Contents

About This Document

Content

This document introduces you to electrical error simulation via XIL API EESPort with ControlDesk.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Documents folder A standard folder for user-specific documents.

`%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>`

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the icon in dSPACE Help. The PDF opens on the first page.

Tutorial Videos

Tutorial Videos

Introduction

The dSPACE website provides electrical error simulation tutorial videos.

Electrical error simulation

You can use electrical error simulation to test ECU software under error conditions.

The tutorial videos show you:

- Basics on electrical error simulation
- How to create an Electrical Error Simulation Port (EESPort) to interface dSPACE failure simulation hardware.
- How to configure and switch an electrical error.
- How to work with multiple error sets when performing electrical error simulation.
- How to monitor the switching behavior of discrete failure simulation hardware.
- How to monitor the switching behavior of SCALEXIO failure simulation hardware.
- How to use software triggers for electrical error simulation.

Refer to https://www.dspace.com/go/tutorial_cd_el_err_sim.

Public product videos

For public product videos, refer to ControlDesk product videos.

Related topics

Basics

Tutorial Videos for ControlDesk (ControlDesk Introduction and Overview 🕮)

Basics and Instructions

Where to go from here

Information in this section

Introduction to Simulating Electrical Errors

Where to go from here

Information in this section

Basics on Electrical Error Simulation

Introduction

During hardware-in-the-loop (HIL) simulation to test ECU software, not only the normal (undisturbed) behavior of an ECU is to be tested, but also its behavior under error conditions, which are typically caused by problems in the ECU wiring.

Simulating electrical errors in the wiring

To test ECU software under error conditions, *electrical error simulation* is used. Electrical error simulation deals with typical wiring errors like loose contacts,

broken cables, short-circuits to neighboring pins, to ground (chassis) or to battery voltage. Supported by software, the electrical error simulation is performed by the failure simulation hardware of an HIL simulator.

Failure simulation hardware The failure simulation hardware can be a separate component in the HIL simulator or integrated in the overall HIL hardware. For basics on the failure simulation hardware, refer to:

- Electrical Error Simulation with a Discrete FIU on page 23
- Electrical Error Simulation with the Integrated SCALEXIO FIU on page 18

Supporting the ASAM AE XIL API standard

In the context of dSPACE products, electrical error simulation is based on the ASAM AE XIL API 2.1.0 standard.

For complete information on the capabilities of the ASAM AE XIL API, refer to the documentation of the ASAM AE XIL API standard that is delivered with the dSPACE XIL API .NET implementation and the Failure Simulation Package.

In the context of dSPACE products that are not based on the ASAM AE XIL API standard, the electrical error simulation is called *failure simulation*. The errors are therefore *failures* that are executed on failure simulation hardware. Usually the hardware provides a failure insertion unit (FIU). You find both terms in this document.

Electrical error simulation with dSPACE products

ControlDesk for initial operation and manual tests ControlDesk provides a graphical user interface (GUI) for electrical error simulation that is delivered with the Failure Simulation Package. This GUI for electrical error simulation is targeted only at the initial operation of the HIL simulator's failure simulation hardware and to perform first manual tests.

dSPACE XIL API .NET for automated tests For extensive HIL tests, it is recommended to use automated testing based on dSPACE XIL API .NET and its EESPort implementation, which is also delivered with the Failure Simulation Package. For further information on electrical error simulation via automated testing, refer to dSPACE XIL API Implementation Guide 🕮.

Note

- To perform electrical error simulation with ControlDesk, you also have to install the Test Automation APIs product set and have a Failure Simulation Package (FAILURE_SIM) license.
- Only EESPort implementations of XIL API version 2.1.0 are supported.
- If you use an EESPort implementation from dSPACE, the version of ControlDesk and the version of the EESPort implementation must be of the same dSPACE Release. For example, if you use ControlDesk 7.4, which is part of dSPACE Release 2021-A, you must select the XIL API EESPort implementation that is delivered with dSPACE Release 2021-A.

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)
Basics on Electrical Errors....16
Overview of the Graphical User Interface in ControlDesk....25

Basics on Electrical Error Simulation Ports

Introduction

Electrical Error Simulation ports (EESPorts) provide access to the failure simulation hardware for simulating electrical errors in an ECU wiring according to the ASAM AE XIL API standard.

Interface to the failure simulation hardware

EESPorts serve as interfaces to the failure simulation hardware, which is controlled by the test cases during HIL simulation, not by the real-time application. Using EESPorts lets you handle your test cases without knowing the failure simulation hardware in detail.

The configuration of an EESPort is described by a hardware-dependent *port configuration* and one or more *error configurations*.

Port configuration

To interface the failure simulation hardware, an EESPort needs the hardware-dependent *port configuration file* (PORTCONFIG file). The file's contents must fit the connected HIL simulator architecture and its failure simulation hardware.

The port configuration file is based on XML. If you replace the port configuration file, you might not need to modify your EESPort application when you want to run the electrical error simulation on another HIL simulator with a similar architecture that supports the configured errors.

The port configuration can also comprise interface definitions of the failure simulation hardware, information on vendor-specific extensions, and (for dSPACE hardware) information on potential mapping and signal mapping. While signal mapping is optional, potential mapping is required for most of the dSPACE simulation platforms.

There are two ways to create a port configuration file:

- By using the graphical user interface of ControlDesk.
- By using the automation of the dSPACE XIL API .NET Implementation. Refer to Creating dSPACE EESPort Configuration Files (dSPACE XIL API Implementation Guide 🕮).

Potential mapping Potential mapping is usually required if your simulation has errors against ground, battery supply voltage or another potential. Via potential mapping, a potential name and a potential type are mapped to a

unique identifier (e.g., a natural number, starting with 0). Refer to Basics on Potential Mapping on page 31.

Signal mapping (optional) According to the ASAM AE XIL API standard, signals are identified by abstract signal names, i.e., user-defined strings. For dSPACE hardware, optional signal mapping lets you map the ECU pins that you are using in your HIL system to these abstract signal names, e.g., `Signal 3`. Refer to Basics on Signal Mapping on page 35.

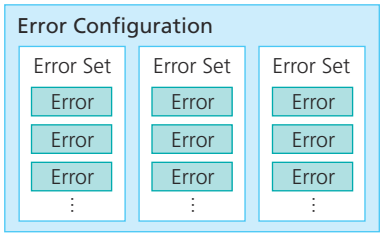
Real-time configuration (optional) You can specify properties for monitoring the switching behavior of the failure simulation hardware in connection with a related real-time application and HIL simulator. Refer to Basics on Monitoring the Switching Behavior of the Failure Simulation Hardware on page 66.

Tip

You find examples for different port configuration (PORTCONFIG) files in the XIL API EESPort Demo project. Refer to XIL API EESPort Demo on page 28.

Error configuration

An error configuration is an XML file that describes a sequence of errors you want to switch during electrical error simulation. Each error configuration comprises error sets with one or more errors.



Error configurations are defined for EESPorts and can be downloaded to the failure simulation hardware to be executed during electrical error simulation.

Tip

- For initial operation of a failure simulation hardware and first manual tests via ControlDesk, you can use single error configuration (XML) files.
- For extensive and automated HIL tests via scripting, it is recommended to implement the error configuration within your source code instead of loading an error configuration (XML) file. This makes testing errors more flexible.

Related topics

Basics

Basics on Electrical Error Simulation....12
Basics on Electrical Errors....16

HowTos

How to Create a New EESPort....38
How to Perform Potential Mapping....41

References

dSPACE EESPort Configuration File (dSPACE XIL API Reference 🕮)

Basics on Electrical Errors

Introduction

An error is defined by the *error category*, the *error type*, and the *load type* of the related signal(s). Which errors you can create for a signal depend on the connected failure simulation hardware.

Error categories

The error category defines how a signal is disturbed.

A signal can be interrupted or connected to other signals or a specific potential (including battery voltage and ground). An error category can affect one or more signals at the same time.

Usually, an error that affects only one pin is called *single pin failure*, except for a short circuit between two pins, which is also called a single pin failure. If an error affects multiple signals, it is called *multiple pin failure* or *multi-pin failure*.

In detail, there are the following error categories.

Error Category	Description
Interrupt	To interrupt a signal line
Pin to Pin	To create a short circuit to other ECU pins
Short to Ground	To create a short circuit to ground
Short to $U_{Battery}$	To create a short circuit to battery voltage
Short to Potential	To create a short circuit to a potential
Interrupt at position	To interrupt a bus line at a specific position

Error Category	Description
Interchange Pin	To interchange two or more signal lines **Note** The Interchange Pin error category is not supported by dSPACE failure simulation hardware.

Potentials To perform a short circuit to a potential, the connected failure simulation hardware must provide additional potentials beside battery voltage ($U_{Battery}$) and ground.

According to the ASAM AE XIL API standard, there are the following *potential types*: Gnd (for ground), Ubat (for $U_{Battery}$), and Potential (for a potential between ground and $U_{Battery}$).

The potentials provided by the failure simulation hardware are specified by unique identifiers (e.g., natural numbers, starting with 0). When you are using failure simulation hardware from dSPACE, you can perform *potential mapping* that lets you map potential names and types to these unique identifiers. Refer to Basics on Potential Mapping on page 31.

Error type

The error type specifies the way an error category – i.e., an interruption or short circuit of signals – is provided. The error type defines the disturbance itself.

There are the following possibilities that differ in the dynamic of the disturbance (static, for a defined duration, controlled by a PWM signal) and the resistance in case of an error.

Error Type (ASAM)	Description
Simple	The error is set statically.
Dynamic	The error is set dynamically. This means that the error is set for a specified duration.
Resistor	The error provides an additional resistor to be switched. The resistor is specified in Ω.
Dynamic Resistor	The error is a combination of a dynamic error with a switchable resistor.
Loose Contact	The error is set dynamically by a PWM signal specified by frequency and duty cycle. This error type is also known as *pulsed switching*.
Loose Contact Resistor	The error is a combination of a loose contact error with a switchable resistor.

Signal

The communication between a hardware-in-the-loop simulator and a connected device under test (DUT), i.e., an ECU, is performed via electrical signals. Each

electrical signal is related to an ECU pin. To disturb a signal via electrical error simulation, the signal must be mapped to an error.

You can map multiple signals (ECU pins) to one error.

Load type

The load type specifies the option to disturb a signal with or without load rejection. This means the load can be connected to (**With load**) or disconnected from (**Without load**) the device under test, i.e., the ECU, during error simulation.

Related topics

Basics

Basics on Electrical Error Simulation....12
Basics on Electrical Error Simulation Ports....14

References

Error....80
Signal....85

Electrical Error Simulation with the Integrated SCALEXIO FIU

Introduction

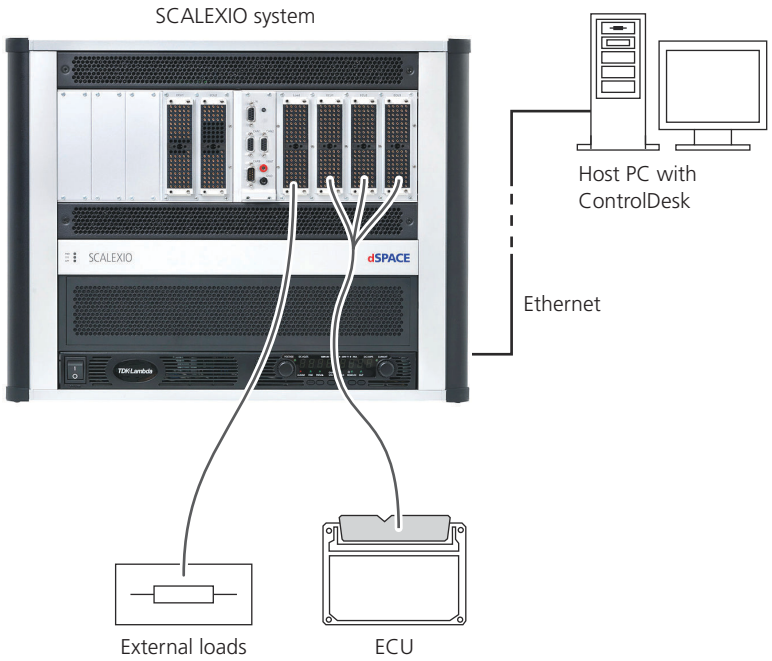
To perform electrical error simulation, a SCALEXIO system can contain an integrated SCALEXIO failure insertion unit (FIU).

Overview

The important components for electrical error simulation are:

- A SCALEXIO failure insertion unit (FIU) that consists of several hardware components. Refer to Components of the integrated SCALEXIO FIU.
- A Host PC with ControlDesk and the Failure Simulation Package.

The following illustration shows an example of a SCALEXIO system that contains a SCALEXIO FIU. It is connected to an ECU, external loads, and the host PC.



Required licenses

To simulate electrical errors, two kinds of licenses are required:

- A license for using the Failure Simulation Package of ControlDesk
- A license for using the components of the SCALEXIO FIU

For more information, refer to License Required for Electrical Error Simulation (SCALEXIO – Hardware and Software Overview 🕮).

Components of the integrated SCALEXIO FIU

To perform electrical error simulation via the integrated SCALEXIO FIU, your SCALEXIO system must provide the required failure simulation hardware, which consists of the following components.

Central failure insertion unit The central failure insertion unit (central FIU), which is located, for example, on an FIU & power switch board or an I/O unit.

Failure routing units Individual failure routing units (FRUs), which are located on the signal and bus lines of the individual channels of SCALEXIO MultiCompact I/O units and boards and HighFlex I/O boards. (Standard SCALEXIO I/O boards do not have any failure routing units.)

Failrails The failrails, which connect the central FIU to the individual failure routing units (FRUs) via the backplanes of the SCALEXIO slot units or I/O units.

Failrail segment switches (optional) One or more (optional) failrail segment switches, which are used for selectively connecting/disconnecting SCALEXIO slot or I/O units to/from the common failrails of the SCALEXIO system.

For more information, refer to Electrical Error Simulation Concept (SCALEXIO Hardware Installation and Configuration 🕮).

Electrical errors

The integrated SCALEXIO failure simulation hardware lets you switch the following electrical errors:

- *Open circuit* to simulate a broken wire or loose contact
- *Short circuit*
  - To ground (KL31)
  - To power switch channels with battery voltage (KL15, KL30)
  - Between channels of the same signal category (e.g., a short circuit between signal measurement channels)
  - Between channels of different signal categories (e.g., a short circuit between signal measurement channels and signal generation channels)
  - Between signal measurement channels or signal generation channels and bus channels
- Pulsed switching for signals of signal measurement channels and signal generation channels (e.g., to simulate loose contacts or relay contact bouncing)

Note

You cannot use pulsed switching for bus signals or when simulating multiple electrical errors.

Except for the *Analog In 2* and *Analog Out 2* channel types, switching electrical errors is performed only for the signal/bus lines, not for reference lines.

Load or signal disconnection

In most cases, you can disconnect the loads or signals during failure simulation (*load rejection*). For example, to protect sensitive loads of signal measurement channels. Limitations apply only when you work with multi-pin failures (refer to Switching short circuits between multiple signals and/or bus channels on page 22).

Loads and signals can be disconnected by switching semiconductor switches or relays. For technical reasons, semiconductor switches cannot be used in all cases. Relays must switch the load or signal disconnection in the following cases:

- Short circuit between two signal generation or signal measurement channels
- Short circuit of a bus channel to any other channel type
- Multiple failures when activation by relay is necessary

Note

The relay might be carrying a current when you are using it to disconnect the loads or signals. Note the related warnings listed in Safety Precautions for Simulating Electrical Errors with a SCALEXIO System (dSPACE XIL API Implementation Guide 🕮).

To switch a relay that might be carrying a current, you must set the corresponding property of the channel in ConfigurationDesk. Otherwise you cannot switch this kind of failure in combination with this channel.

Pulsed switching

The central FIU uses semiconductor switches for switching the failures. It is able to switch very fast (pulsed switching). This makes it possible to simulate loose contacts or defined switch bouncing. However, pulsed switching is not possible in all combinations of channel types.

Pulsed switching is not supported for

- Failures when bus channels are involved
- Multiple failures when activation by relay is necessary

Switching multiple electrical errors

The integrated SCALEXIO failure simulation hardware supports switching multiple electrical errors at the same time or in succession. For example, you can simulate an open circuit for one channel and a short circuit for another channel at the same time, without deactivating the first error.

Note

When multiple electrical errors are simulated, switching can be done by relays that are carrying a current. Note the related warnings listed in Safety Precautions for Simulating Electrical Errors with a SCALEXIO System (dSPACE XIL API Implementation Guide 🕮).

Allowed electrical errors You can simulate the following electrical errors at the same time:

- Any number of open circuits (interrupts)
- Multiple short circuits to one power switch channel (short to battery voltage or another potential)
  - Up to 10 short circuits with a DS2642 FIU & Power Switch Board
  - Up to 6 short circuits with a DS2680 I/O Unit
- Multiple short circuits to ground (short to GND)
  - Up to 10 short circuits with a DS2642 FIU & Power Switch Board
  - Up to 6 short circuits with a DS2680 I/O Unit
- Multiple short circuits between signal measurement, signal generation and bus channels (short to pins, multi-pin errors (refer to Switching short circuits between multiple signals and/or bus channels on page 22))
  - Short circuits between up to 10 channels (one multi-pin error)
  - Two multi-pin errors in parallel (2 × 10 channels)
- Combinations of any number of open circuits (interrupts) with:
  - Multiple short circuits to one power switch channel (short to VBAT)
  - Multiple short circuits to ground (short to GND)
  - Multiple short circuits between channels (short to pins)
  - One error (short circuit to any potential or open circuit) of a signal that uses current enhancement
  - One error (short circuit to any potential or open circuit) with pulsed switching

Limitations

- In general, switching multiple electrical errors is limited by the allowed maximum current of the channels and failrails involved.
- Switching multiple electrical errors with channels that use current enhancement is not supported (except for channel multiplication of the Power Switch 1 channel type of the DS2642 FIU & Power Switch Board).
- Pulsed switching is supported for one signal only.

Switching short circuits between multiple signals and/or bus channels

Multi-pin errors Multi-pin errors let you simulate a short circuit between three or more signal channels and/or bus channels. The channels can be located on the same or different boards or I/O units. You can simulate a short circuit between:

- Channels of the same signal category (e.g., four signal generation channels)
- Channels of different signal categories (e.g., three signal generation channels and two signal measurement channels)
- Signal channels and bus channels (e.g., two signal generation channels, one signal measurement channel, and one bus channel)

Switching multi-pin errors The both failrails of a SCALEXIO system are used automatically by the XIL API EESPort according to the specified error configuration.

Multi-pin errors can be switched only by relays. Therefore, you must use ConfigurationDesk to allow the activation by FRU relays for each involved channel.

Load or signal disconnection You can disconnect loads or signals from channels that are used for multi-pin errors. To disconnect loads or signals, you must switch the channels to failrail 1. You cannot use failrail 2 in this case.

Loads or signals can be disconnected only by relays.

Note

You can switch multi-pin errors and disconnect loads or signals by using relays that might be carrying a current. Note the related warnings listed in Safety Precautions for Simulating Electrical Errors with a SCALEXIO System (dSPACE XIL API Implementation Guide 🕮).

Monitoring the switching behavior

With an experiment software such as ControlDesk, you can monitor and trace the switching behavior of the SCALEXIO failure simulation hardware. For further information, refer to Monitoring the Switching Behavior of the Failure Simulation Hardware on page 66.

Related topics

Basics

Hardware for Electrical Error Simulation on SCALEXIO Systems (SCALEXIO Hardware Installation and Configuration 🕮)
Overview of the Graphical User Interface in ControlDesk 25

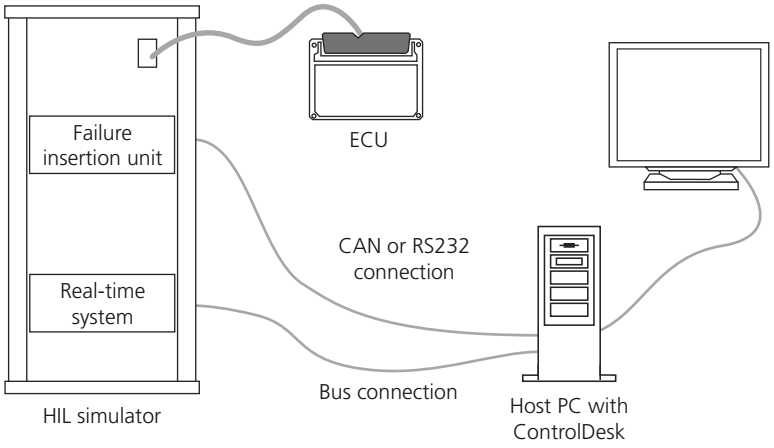
Electrical Error Simulation with a Discrete FIU

Introduction

To perform electrical error simulation, a dSPACE hardware-in-the-loop (HIL) simulator can be extended by a discrete failure insertion unit (FIU).

Overview

The following illustration shows a HIL simulator that is extended by a discrete failure insertion unit (FIU). This schematic is also valid for a SCALEXIO system that does not use the concept of the integrated SCALEXIO failure insertion unit (FIU).



The important components of the failure simulation hardware are the failure insertion unit (FIU) and the host PC with ControlDesk and its Failure Simulation Package.

Failure insertion unit The HIL simulator must be equipped with a discrete failure insertion unit (FIU). The FIU is connected via a CAN or RS232 serial connection, depending on its type. For an overview of the FIUs, refer to Hardware for Failure Simulation (dSPACE XIL API Implementation Guide 🕮).

Host PC with ControlDesk The electrical error simulation is controlled via a host PC with ControlDesk and the Failure Simulation Package. The Failure Simulation Package is an extension for ControlDesk. It lets you configure the errors for the ECU pins graphically and finally switch the failure simulation hardware. For an overview of the related graphical user interface in ControlDesk, refer to Overview of the Graphical User Interface in ControlDesk on page 25.

Supported interfaces

The discrete failure insertion unit can be controlled by CAN interfaces and RS232 interfaces:

CAN interfaces For an overview on supported CAN hardware, refer to supported interface types (refer to Interface type on page 90).

The software and device driver are installed with ControlDesk.

RS232 interfaces COM ports 1 … N of the host PC

Note

Reduced Performance by Using External RS232 Converters

You are strongly recommended to use a physical RS232 port of the host PC to control the failure simulation hardware. If external RS232 ports are missing, try to use an internal RS232 port of the host PC's motherboard.

Software triggers and dynamic errors are not supported if you use an external RS232 converter. Communication via an external RS232 converter is also time-critical and can cause communication errors.

If there is no alternative to using an external RS232 converter:

- Use the IOLAN DS1 from Perle as an Ethernet-to-RS232 converter. For configuring this tool, refer to http://www.dspace.com/go/eth2rs232.
- Otherwise, use an USB-to-RS232 converter with an FTDI chipset and the newest FTDI driver, refer to http://www.ftdichip.com/FTDrivers.htm.

Signal file

For HIL simulators with discrete FIUs, the electrical errors (also named failure classes) allowed for electrical error simulation are specified in the *signal file* that comes with the simulator.

For more information, refer to Defining Failure Classes with Signal Files (dSPACE XIL API Implementation Guide 🕮).

Related topics

Basics

Basics on Failure Classes (dSPACE XIL API Implementation Guide 🕮)
Electrical Error Simulation with the Integrated SCALEXIO FIU 18
Overview of the Graphical User Interface in ControlDesk 25

References

Hardware for Failure Simulation (dSPACE XIL API Implementation Guide 🕮)

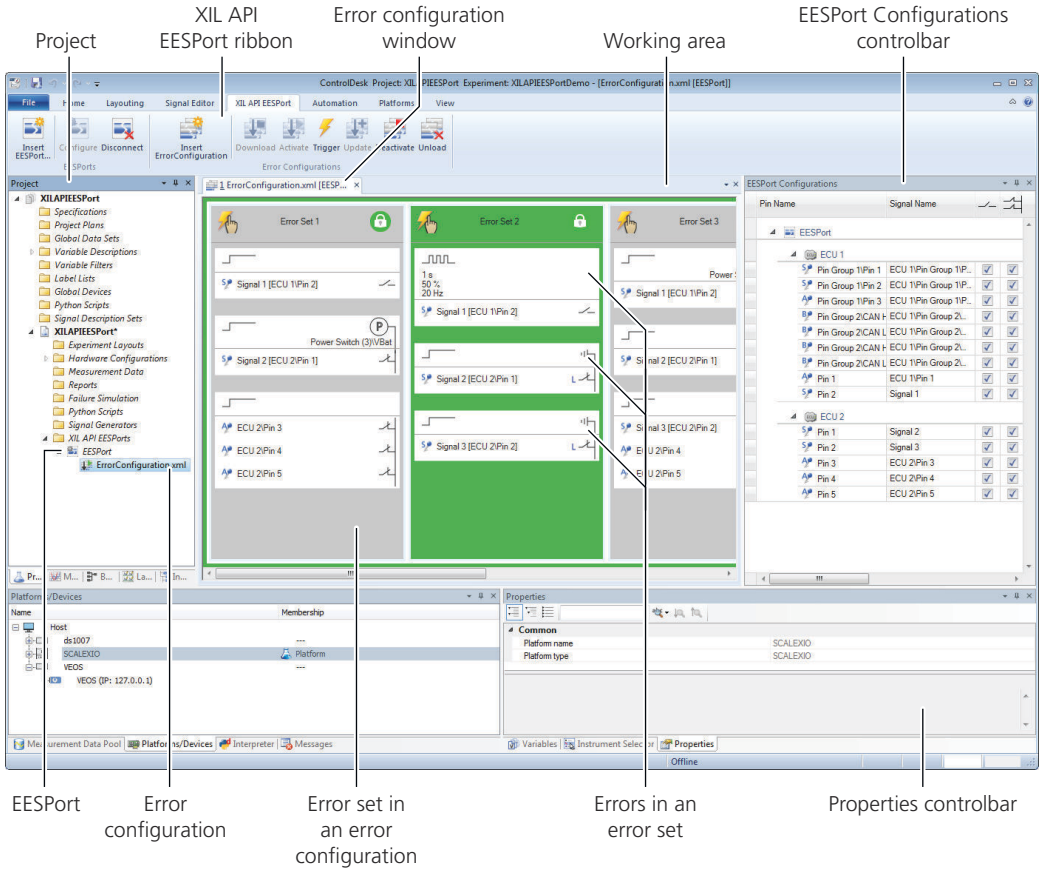
Overview of the Graphical User Interface in ControlDesk

Introduction

The Failure Simulation Package extends ControlDesk for configuring and performing electrical error simulation.

GUI elements for electrical error simulation

The following illustration shows ControlDesk's user interface for electrical error simulation.



The main elements of the graphical user interface are:

- The XIL API EESPort ribbon
- The error configuration window in ControlDesk's working area
- The EESPort Configurations controlbar

XIL API EESPort ribbon

The XIL API EESPort ribbon provides commands for handling EESPorts and performing electrical error simulation with the connected failure simulation hardware.

Error configuration window

The error configuration window in ControlDesk's working area lets you display and specify error configurations, error sets, errors, and the related signals in a graphical environment. All the elements you are focusing in the error configuration window can be configured via the **Properties** controlbar.

Different symbols illustrate the state of the configuration. The trigger type of an error set is indicated by a symbol in its upper left corner, for example, a  symbol for a manually triggered error set or a  symbol for an error set that is triggered by software . Multiple  or  symbols indicate if the single settings are valid or invalid.

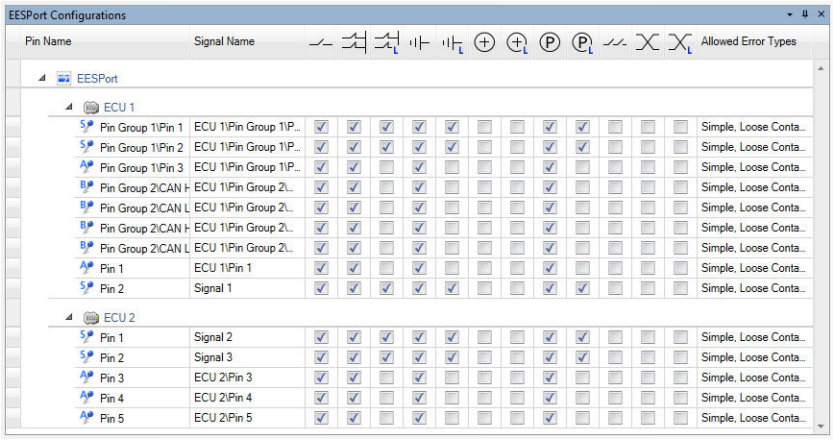
If a signal is to be disturbed without load rejection, this is indicated with an additional L symbol. Further symbols are used to illustrate the configured error categories and error types. For examples on different error configurations, refer to Examples for Configured Errors on page 60.

Different frame colors and background colors display the states of an error configuration and its error sets. For further details, refer to Error Configuration on page 81 and Error Set on page 83.

In the working area, you can move errors and signals between error sets and error configuration windows via drag & drop. You can also copy errors and signals. To do this, press the **Ctrl** key during the drag & drop. If you move or copy errors between different error configuration windows, the related signal mapping might change the involved ECU pins.

EESPort Configurations controlbar

The controlbar displays wiring information on the related hardware-in-the-loop (HIL) simulator and specifications for the electrical error simulation: the ECU signals (i.e., their names), their wiring (i.e., the connected ECU pins) and the allowed error categories, error types, and load types for each ECU pin (see the following example).



If the EESPort is configured, the displayed information is derived from the signal file of a non-SCALEXIO system or the real-time application (RTA) file of a SCALEXIO system.

You can use the controlbar to map ECU pins to errors in the working area via drag & drop.

The controlbar provides a wide range of commands for arranging and filtering the displayed entries, for example, you can group rows or specify comprehensive filter rules.

For information on the controlbar's commands, refer to EESPort Configurations on page 120.

For information on signal files of non-SCALEXIO systems, refer to Defining Failure Classes with Signal Files (dSPACE XIL API Implementation Guide 🕮).

Related topics

Basics

Workflow for Performing Electrical Error Simulation

Workflow steps

Carry out the following steps to simulate electrical errors in the wiring of an ECU:

1. *Specify the allowed errors (failures classes) for each ECU pin*
   - *For non-SCALEXIO systems*, specify the errors (failure classes) allowed for electrical error simulation in the simulator's *signal file*, which is provided with the simulator by dSPACE. For further information, refer to Defining Failure Classes with Signal Files (dSPACE XIL API Implementation Guide 🕮).
   - *For SCALEXIO systems*, specify the errors (failures) allowed for the ECU signals and the load rejection of signal measurement channels in ConfigurationDesk. This information is included in the *real-time application file (RTA file)*. For further information, refer to Specifying Allowed Failure Classes for ECU Pins (SCALEXIO – Hardware and Software Overview 🕮).
2. *Connect the failure simulation hardware to the host PC physically*
   - *For a HIL simulator with a discrete FIU*, connect the host PC to the failure simulation hardware via an RS232 or CAN connection.
   - *For a SCALEXIO system that uses an integrated SCALEXIO FIU*, use the Ethernet connection between the host PC and the SCALEXIO system. (The SCALEXIO FIU is integrated in the SCALEXIO system and requires no additional connection.)
3. *Start ControlDesk*

   Start ControlDesk. Create a new project and experiment or load an existing project and experiment.

   The experiment contains an **XIL API EESPorts** folder for managing Electrical Error Simulation ports (EESPorts) and their error configurations.

For SCALEXIO systems, the experiment must contain a registered SCALEXIO platform with a variable description.

4. *Create an Electrical Error Simulation port (EESPort)*

   Create an EESPort to interface the connected failure simulation hardware. Refer to How to Create a New EESPort on page 38.

5. *Connect the EESPort to the failure simulation hardware*

   Configure the EESPort and connect it with the related failure simulation hardware. Refer to Configure (EESPort) on page 118.

6. *Create and configure an error configuration*

   Create an error configuration and specify the error sets and errors you want to switch. Refer to How to Create and Configure an Electrical Error on page 48.

7. *Perform the electrical error simulation*

   Finally, you can perform the electrical error simulation according to the created error configuration with the following steps:

   - Download an error configuration to the failure simulation hardware.
   - Activate the error configuration on the failure simulation hardware.
   - Trigger the error sets of the error configuration to activate the related errors.
   - Deactivate the error configuration to stop the electrical error simulation.
   - Unload the error configuration from the failure simulation hardware.

   For further details on performing the electrical error simulation, refer to How to Perform Electrical Error Simulation on page 63.

XIL API EESPort Demo

Opening demo projects

For instructions on opening demo projects, refer to Opening a demo project (ControlDesk Introduction and Overview 🕮).

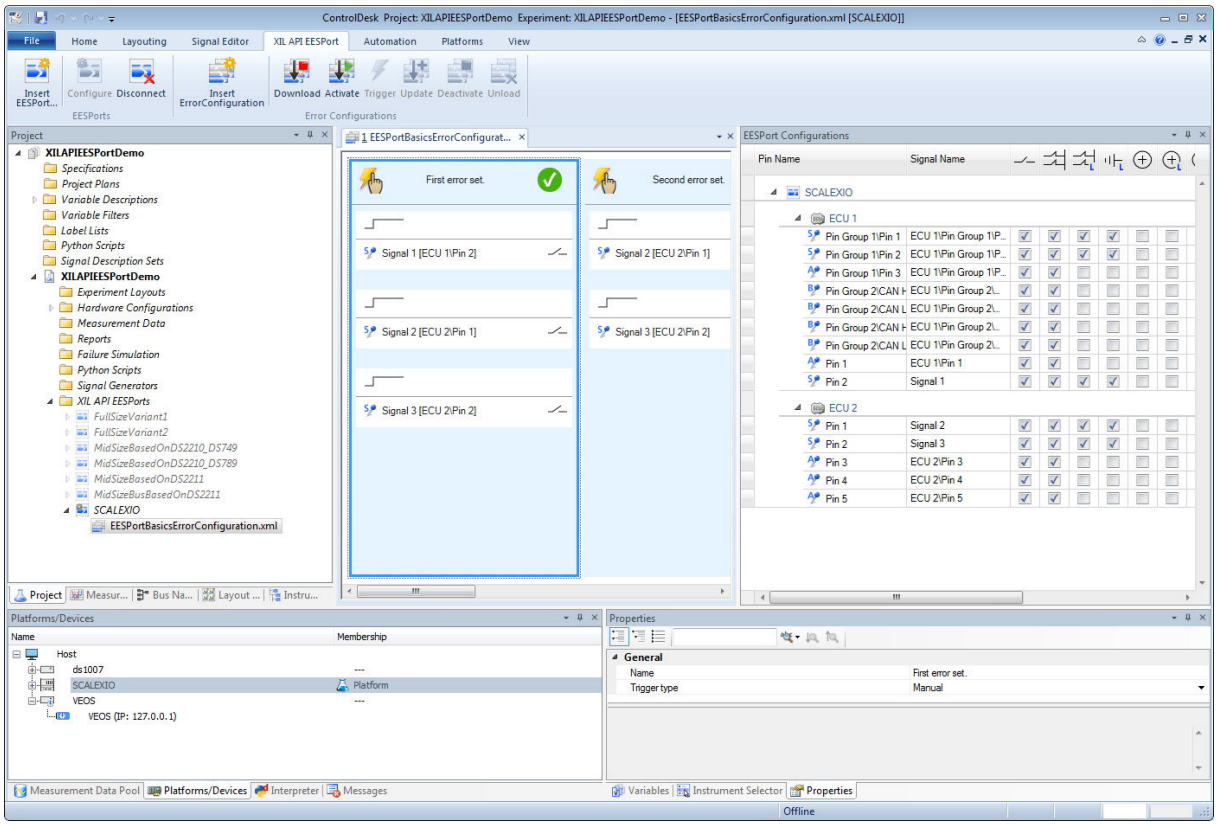
Description of the demo project

The XILAPIEESPortDemo project shows you ControlDesk and its electrical error simulation features in connection with dSPACE failure insertion hardware. As your simulator probably does not match the configurations used for this demo, you cannot use it to perform electrical error simulation, but you can gain an impression of how electrical error simulation works.

Required products and modules Working with this demo requires:

- *ControlDesk*
- *Failure Simulation Package*
- (If you work with SCALEXIO failure simulation units) *ConfigurationDesk - Implementation Version: SCALEXIO Failure Simulation*

Demo overview The illustration below shows the error configuration for the SCALEXIO XIL API EESPort as an example:



Contained XIL API EESPorts The following table shows which simulator hardware you can use for the XIL API EESPorts in the demo project:

XIL API EESPort	For Simulator Hardware
FullSizeVariant1	▪ dSPACE Simulator Full-Size with DS291 FIU Modules ▪ SCALEXIO system with DS291 FIU Modules
FullSizeVariant2	dSPACE Simulator Full-Size with DS293 FIU Modules
MidSizeBasedOnDS2210_DS749	dSPACE Simulator Mid-Size based on DS2210 with DS749 FIU Modules
MidSizeBasedOnDS2210_DS789	dSPACE Simulator Mid-Size based on DS2211 with DS789 Sensor FIU Modules
MidSizeBasedOnDS2211	dSPACE Simulator Mid-Size based on DS2211 with DS791 and DS793 FIU Modules
MidSizeBusBasedOnDS2211	dSPACE Simulator Mid-Size based on DS2211 with a DS1450 Bus FIU Board
SCALEXIO	SCALEXIO system with its integrated FIU

Related topics

Basics

Basics on Electrical Error Simulation..... 12

Configuring Electrical Error Simulation

Where to go from here

Information in this section

Basics on Potential Mapping

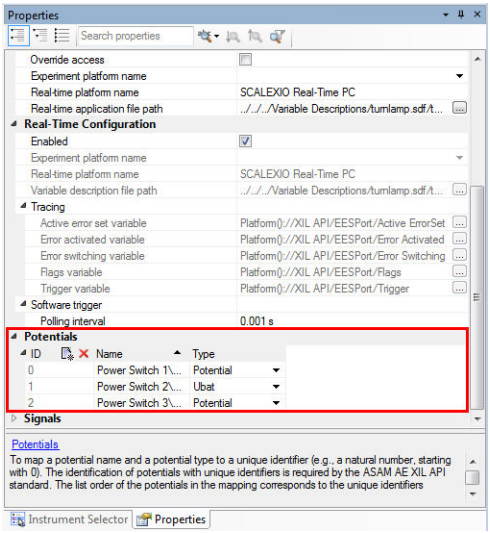
Introduction

Potential mapping is usually required if your electrical error simulation works with errors against ground, battery supply voltage or another potential.

Potential mapping for dSPACE hardware

The ASAM AE XIL API standard requires the identification of potentials with unique identifiers (IDs), starting with 0. For a more practical approach, electrical error simulation with ControlDesk and the dSPACE XIL API implementation lets you map these identifiers to the potential names that you use with your real-time application and your HIL simulator.

Via potential mapping, potential names and potential types are mapped to unique identifiers. Refer to the following example:



According to the ASAM AE XIL API standard, there are the following *potential types*: **Gnd** (for ground), **Ubat** (for $U_{Battery}$), and **Potential** (for a potential between ground and $U_{Battery}$). The list order of the potentials in the mapping corresponds to the unique identifiers assigned for the single potentials.

In ControlDesk's user interface for electrical error simulation, ControlDesk will display the potential names instead of the inexpressive potential IDs. See the following example.

To show the potential names in the errors in ControlDesk's working area, the related EESPort must be configured. Only if there is no valid potential mapping, ControlDesk will display a potential ID in the GUI.

Basics on potential names

The potential names are specified by the failure simulation hardware of your dSPACE simulator.

Potential names of discrete FIUs For discrete FIUs, the potential names (for potentials 0 … n) are defined in the format area of the simulator's signal file. See the following example.

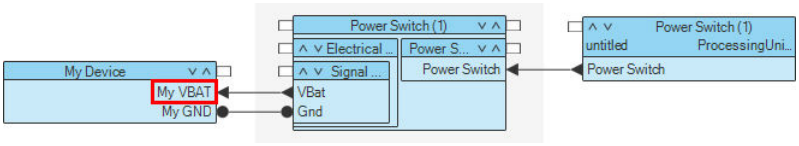
	A	B	C
21	/Format		
22	Version	1.3	
23	BoardType	DS293	
24	Potential0	Ubat	
25	Potential1	Gnd	
26	Potential2	Pot2	
27	Potential3	Pot3	
28	Potential4	Pot4	
29	OffsetCANAddress	none	
30	OffsetLoadAddress	1	
31	OffsetPowerSwitchAddress	none	
32	FIULoadAddress	1_02_4	
33	RSIMLoadAddress	1_02_5	
34	FIUCANAddress	none	
35	RSIMCANAddress(1)	none	
36	RSIMCANAddress(2)	none	
37	AdditionalFiles	none	
38	FaultClassFile	none	
39	/FormatEnd		

signallist full-size ds293

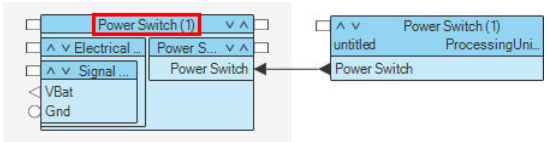
For further details, refer to Defining Failure Classes with Signal Files (dSPACE XIL API Implementation Guide 🕮).

Potential names of the integrated SCALEXIO FIU For the integrated SCALEXIO FIU, you define potentials in ConfigurationDesk when you configure Power Switch function blocks.

- If you connect a Power Switch function block to an external device, the name of the device pin that is connected to the VBat pin of the Power Switch function block is used as potential name in ControlDesk. In the following example, the potential name in ControlDesk is My VBAT.



- If you do not connect a Power Switch function block to an external device, the name of Power Switch function block is used as potential name in ControlDesk. In the following example, the potential name in ControlDesk is Power Switch (1).



For information on Power Switch function blocks, refer to Power Switch (ConfigurationDesk I/O Function Implementation Guide 🕮).

Specific hardware dependencies

You are recommended to use potential mapping for dSPACE failure simulation hardware. Potential mapping is not required for all the hard-wired potentials of

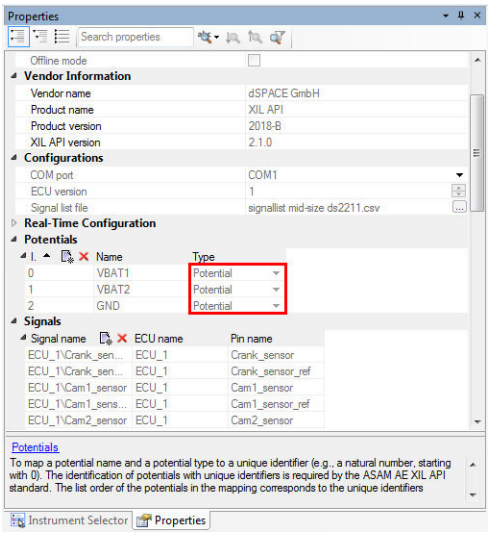
your failure simulation hardware, because the dSPACE XIL API implementation knows these potentials.

Observe the following hardware dependencies:

- Potential mapping is not required for DS291, DS749, DS5355/DS5390, and DS1450.
- You can specify one **Ubat** potential type for the following failure simulation hardware:
  - DS293
  - DS789
  - DS791
  - DS793
  - SCALEXIO
- You can specify one **GND** potential type for the following failure simulation hardware:
  - DS293
  - DS791
  - DS793

Using a preconfigured potential mapping

You can create a preconfigured potential mapping when creating a *new* port configuration (PORTCONFIG) file with ControlDesk. ControlDesk performs a preconfigured potential mapping based on the potential names that are provided for your failure simulation hardware. By default, the **Type** of all the potentials is set to **Potential**. Refer to the following illustration:



You can change this preconfigured potential mapping in the **Properties** controlbar.

For more information, refer to How to Create a New Port Configuration File on page 44.

Related topics

Basics

Basics on Electrical Error Simulation Ports....14
Basics on Signal Mapping....35

HowTos

How to Perform Potential Mapping....41

References

EESPort – Potentials Properties....93
Insert EESPort....131

Basics on Signal Mapping

Introduction

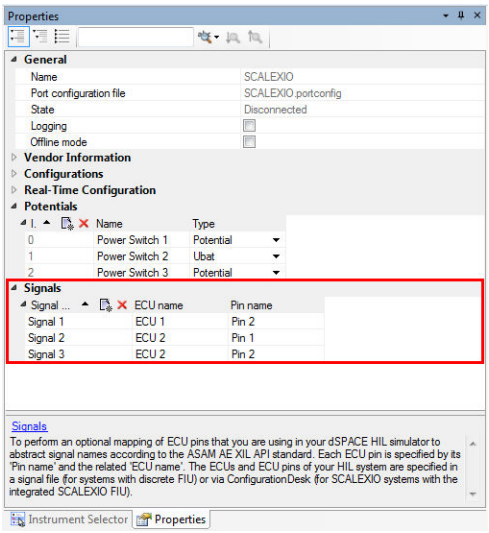
Signal mapping specifies an optional mapping between the signal names and abstract names used in the XIL API application.

Signal mapping for dSPACE hardware

According to the ASAM AE XIL API standard, the signals to be disturbed via electrical error simulation are identified by abstract signal names, i.e., user-defined strings. For dSPACE hardware, optional signal mapping lets you map the ECU pins that you are using in your HIL system to these abstract signal names.

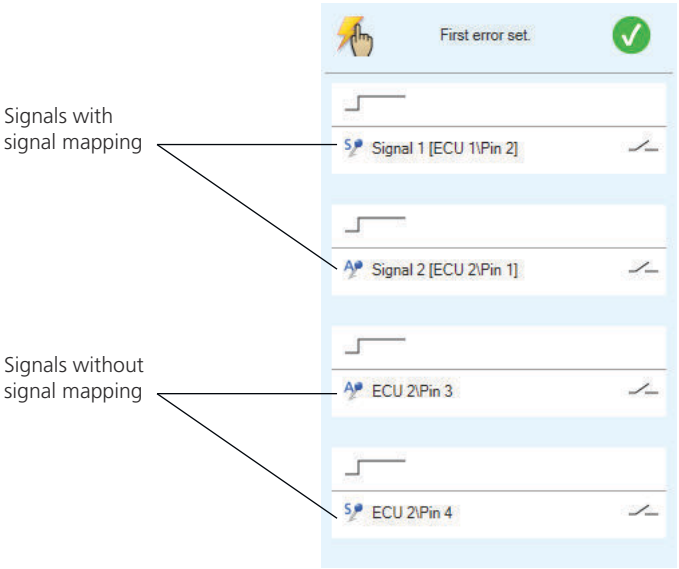
Signal mapping lets you replace the port configuration file without modifying your EESPort application when you want to run the electrical error simulation on another HIL simulator with a similar architecture.

You can specify signal mapping when creating a new EESPort or later in the **Properties** controlbar. The signal mapping is stored in the port configuration (PROTCONFIG) file of the EESPort. The following illustration shows how signal mapping for an EESPort is displayed in the **Properties** controlbar.



In this example, the ECU pin named **Pin 2** of the ECU named **ECU 1** is mapped to a signal named **Signal 1**.

If you do not perform signal mapping, ControlDesk generates an abstract signal name automatically when you map an ECU pin to an error. The resulting signal name is composed of the original ECU name and the pin name, e.g., **ECU 2\Pin 4**. The following illustration shows how signals with and without signal mapping are displayed in an error set.



Generating a default signal mapping

You can generate a default signal mapping when creating a new EESPort. To generate the default signal mapping, enable **Configure after creation** and **Generate default signal mapping** when creating the new EESPort. Refer to How to Create a New Port Configuration File on page 44.

ControlDesk generates the default signal mapping based on the ECU names and pin names that are specified for your dSPACE simulator. The resulting signal names are composed of the original ECU names and the pin names, e.g., **ECU 2\Pin 4**. You can change this default signal mapping in the **Properties** controlbar.

If you create the new EESPort by referencing an existing port configuration (PORTCONFIG) file that already contains a signal mapping, the original signal mapping is kept and the new mapping is added. Unused mappings are also kept.

Example Existing signal mapping in the referenced PORTCONFIG file:

- Signal 1 – ECU 1\Pin 1
- Signal 2 – ECU 1\Pin 2
- Signal 3 – ECU 42\Pin 23

Signals specified for the dSPACE simulator:

- ECU 1\Pin 1
- ECU 1\Pin 2
- ECU 1\Pin 3
- ECU 2\Pin 1

New signal mapping for the new EESPort:

- Signal 1 – ECU 1\Pin 1 *(signal mapping is kept)*
- Signal 2 – ECU 1\Pin 2 *(signal mapping is kept)*
- Signal 3 – ECU 42\Pin 23 *(signal mapping is kept, although unused)*
- ECU 1\Pin 3 – ECU 1\Pin 3 *(signal mapping is added by using default names)*
- ECU 2\Pin 1 – ECU 2\Pin 1 *(signal mapping is added by using default names)*

ECU and pin names for dSPACE hardware

The ECUs and ECU pins of your HIL system are specified in a signal file (for systems with discrete FIU) or via ConfigurationDesk (for SCALEXIO systems with the integrated SCALEXIO FIU).

Related topics

Basics

Basics on Electrical Error Simulation Ports....14
Basics on Potential Mapping....31

HowTos

How to Create a New Port Configuration File....44

References

EESPort – Signals Properties....97
Insert EESPort....131

How to Create a New EESPort

Objective

To interface the failure simulation hardware, you have to create an Electrical Error Simulation port (EESPort).

Preconditions

To create an EESPort for the failure simulation hardware of a SCALEXIO system:

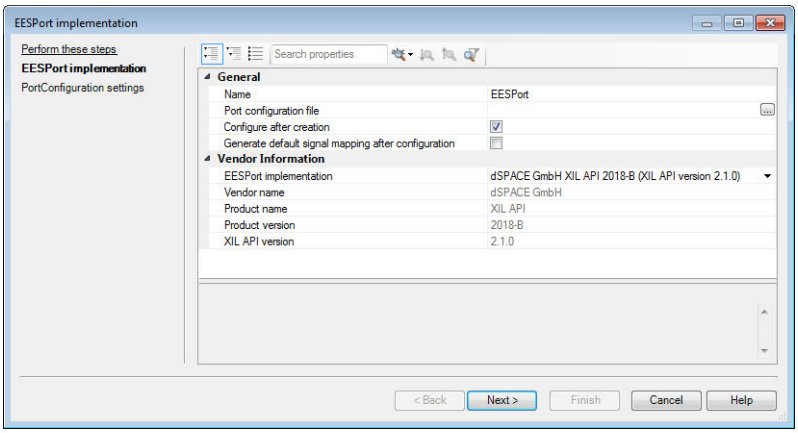
- The related SCALEXIO platform must be registered in ControlDesk. For further details, refer to How to Register a Platform (ControlDesk Platform Management 🕮).
- The registered SCALEXIO platform is added to the ControlDesk experiment you are working with. For further details, refer to How to Add a Platform/Device to an Experiment (ControlDesk Platform Management 🕮).
- You added a variable description to the SCALEXIO platform. For further details, refer to How to Add a Variable Description to a Platform/Device (ControlDesk Variable Management 🕮)
- The experiment has been saved at least once.

Method

To create a new EESPort

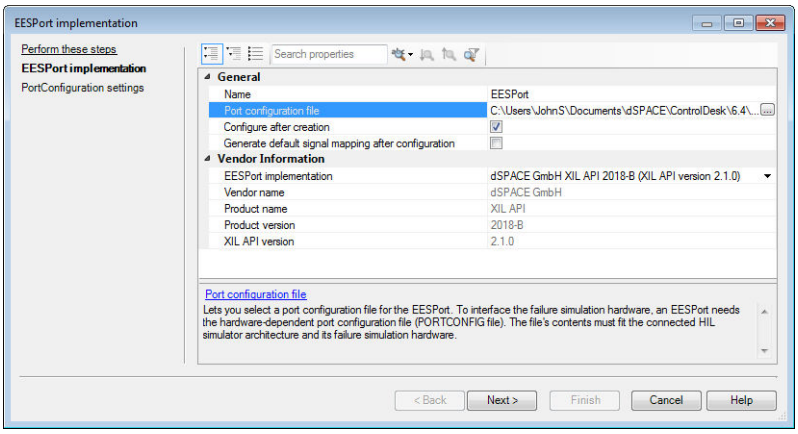
1 From the context menu of the XIL API EESPort folder in the Project controlbar, select Insert EESPort.

ControlDesk opens the EESPort Implementation dialog.



By default, the Configure after creation option is selected to configure the XIL API EESPort automatically after its creation to connect it with the related failure simulation hardware. (The list on the EESPort Configurations controlbar is empty as long as no dSPACE XIL API EESPort is configured.)

2 Click the **Port configuration file** Browse button to select a port configuration (PORTCONFIG) file for the EESPort.

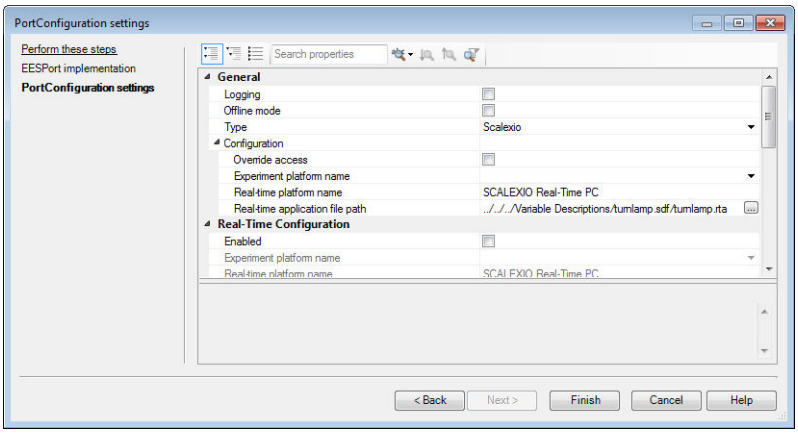


Tip

For dSPACE failure simulation hardware, you can also use the dialog to create a new port configuration (PORTCONFIG) file if you do not select an existing PORTCONFIG file. Refer to How to Create a New Port Configuration File on page 44.

3 If you use an XIL API EESPort implementation from dSPACE, you must click the **Next** button to change to the **PortConfiguration** dialog.

The dialog displays the contents of the port configuration file (see the following example).



4 You can change the configuration of the EESPort according to your needs. For example, you can add, edit, or remove the potential mapping and/or signal mapping. Your settings must fit the related failure simulation hardware.

For details on the available properties, refer to Insert EESPort on page 131.

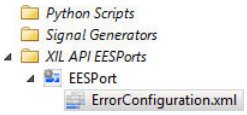
Tip

You can also change specific configuration properties of the EESPort later via the **Properties** controlbar.

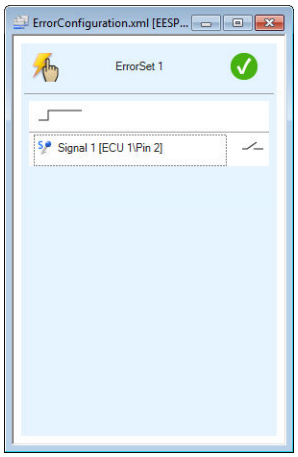
5 Click **Finish** to create the EESPort.

Result

In the **Project** controlbar, a new EESPort with an error configuration appears in the **XIL API EESPorts** folder.



The new error configuration is automatically opened in ControlDesk's working area. The error configuration contains an error set with an error specifying an interrupt of a signal with the default name **Signal 1** (see the following example).



If the optional signal mapping of the port configuration is missing, symbols will indicate that the error must be configured properly.

In the Windows file system, a related EESPort folder is created. The folder contains:

- A local working copy of the selected port configuration (PORTCONFIG) file or a new PORTCONFIG file.
- A local working copy of the signal list (CSV) file (for a HIL simulator that uses a discrete FIU).
- A first error configuration (XML) file

Next step

You can now configure the error configuration, refer to How to Create and Configure an Electrical Error on page 48.

Related topics

Basics

Basics on Electrical Error Simulation Ports....14

HowTos

How to Perform Potential Mapping....41

References

EESPort....78
Insert EESPort....131

How to Perform Potential Mapping

Objective

For dSPACE hardware, you have to perform a potential mapping.

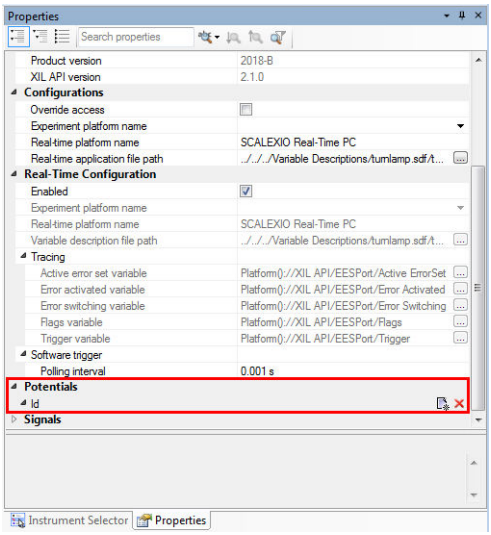
Preconditions

- You created an EESPort. To create an EESPort, refer to How to Create a New EESPort on page 38.
- The EESPort is disconnected from the failure simulation hardware. To disconnect an EESPort, refer to Disconnect (EESPort) on page 119.

Method

To perform potential mapping

1 In the Project controlbar, select the EESPort and open the Properties controlbar for it.



Tip

You can also perform potential mapping when creating an EESPort.

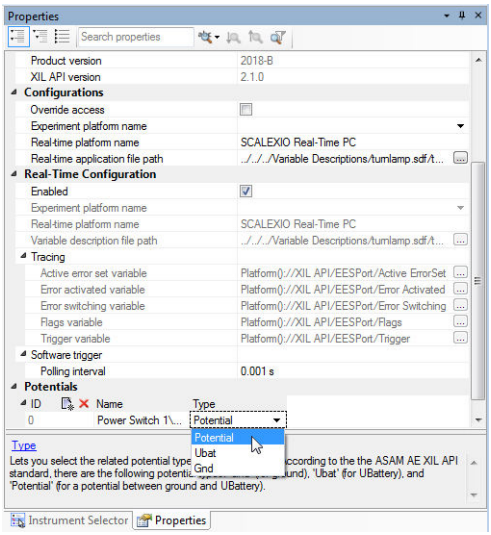
2 In the Potentials – Id row, click .
A new row with edit fields appears in the Potentials list.

Tip

You can click ✕ to remove a row from the Potentials list.

3 In the Name edit field, type the potential name as used for your HIL simulator: e.g., `Power Switch 1\VBat`.

4 Change to the Type field and select the related potential type: e.g., Potential.

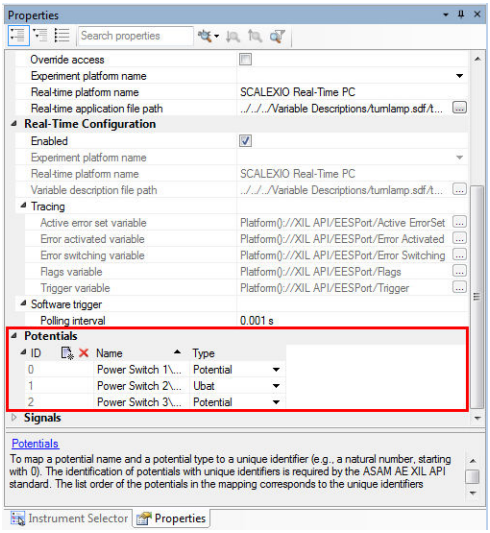


The potential ID is assigned automatically (in sequential numbering): e.g., `0`.

5 Repeat steps 3 … 4 for all the potentials of your HIL system.

Result

You performed potential mapping. See the following example.



For the Short to Potential error category, the GUI for electrical error simulation will use the specified potential names instead of the related potential IDs.

Your potential mapping is stored in the hardware-dependent port configuration file (PORTCONFIG file). See the following example.

```
<?xml version="1.0" encoding="utf-8"?>
<PortConfigurations>
  <EESPortConfig Logging="false" OfflineMode="false" Version="2018-B">
    <HardwareConfigurations>
      <Drivers>
         <Driver ID="0" DriverType="SCLX">
           <PlatformName>SCALEXIO Real-Time PC</PlatformName>
           <OverrideAccess>false</OverrideAccess>
         </Driver>
      </Drivers>
      <HardwareConfiguration ID="1" SignalListPath="../../../Variable Descriptions/turnlamp.sdf/turnlamp.rta"
DriverId="0" EcuVersion="1" />
      <PotentialMapping>
        <Potential ID="0" Name="Power Switch 1\VBat" Type="Potential" HardwareConfigurationId="1" />
        <Potential ID="1" Name="Power Switch 2\VBat" Type="Ubat" HardwareConfigurationId="1" />
        <Potential ID="2" Name="Power Switch 3\VBat" Type="Potential" HardwareConfigurationId="1" />
      </PotentialMapping>
      <SignalMapping>
        <Signal MappingName="Signal 1" ECUName="ECU 1" PinName="Pin 2" />
        <Signal MappingName="Signal 2" ECUName="ECU 2" PinName="Pin 1" />
        <Signal MappingName="Signal 3" ECUName="ECU 2" PinName="Pin 2" />
      </SignalMapping>
    </HardwareConfigurations>
    <RealTimeConfiguration PlatformName="SCALEXIO Real-Time PC" SystemDescriptionFilePath="../../../Variable
Descriptions/turnlamp.sdf/turnlamp.sdf">
      <Tracing Enabled="false">
        <Variable Value="Platform():/XIL API/EESPort/Error Activated" Type="ErrorActivated" />
        <Variable Value="Platform():/XIL API/EESPort/Active ErrorSet" Type="ActiveErrorSet" />
        <Variable Value="Platform():/XIL API/EESPort/Error Switching" Type="ErrorSwitching" />
        <Variable Value="Platform():/XIL API/EESPort/Flags" Type="Flags" />
        <Variable Value="Platform():/XIL API/EESPort/Trigger" Type="Trigger" />
      </Tracing>
      <SoftwareTrigger PollingInterval="0.001" />
    </RealTimeConfiguration>
  </EESPortConfig>
</PortConfigurations>
```

Related topics

HowTos

How to Create a New Port Configuration File.....44

References

Configuration Files for Electrical Error Simulation (dSPACE XIL API Reference 🕮)

How to Create a New Port Configuration File

Objective

For dSPACE failure simulation hardware, you can create a new port configuration (PORTCONFIG) file when creating a new XIL API EESPort.

Method

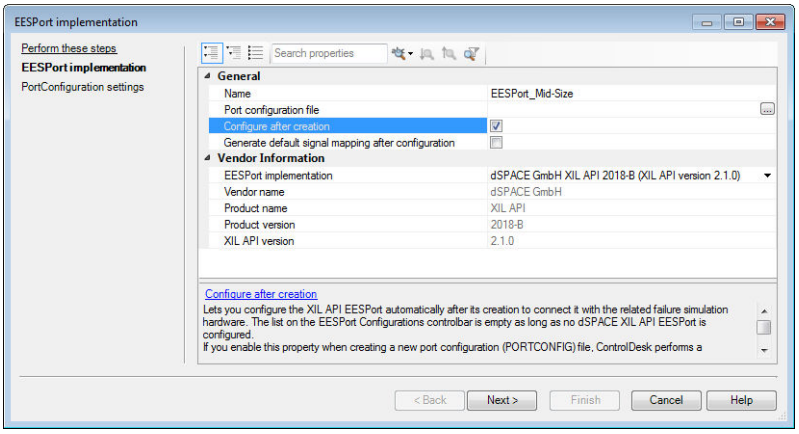
To create a new port configuration file

1. To create a new port configuration file, start by creating a new XIL API EESPort. Select **Insert EESPort** from the context menu of the XIL API EESPort folder.

   ControlDesk opens the **EESPort Implementation** dialog.

2. In the dialog, specify the name of the new EESPort, for example, EESPort_Mid-Size. This name is also used for the new PORTCONFIG file.

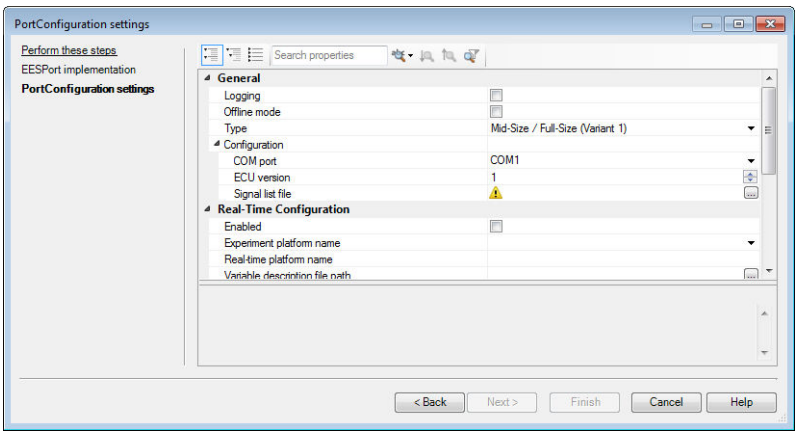
3. Select **Configure after creation** to create a preconfigured potential mapping for the new EESPort. (This option also configures the EESPort automatically after its creation.)



4. Select **Generate default signal mapping** to create a default signal mapping for the new EESPort.

5. To create a new port configuration file with the new EESPort, do not edit the **Port configuration file** edit field. Click the **Next** button to go to the **PortConfiguration** dialog.

   ControlDesk opens the **PortConfiguration** dialog.



   The **PortConfiguration** dialog is not available if you use a third-party ASAM XIL API EESPort server.

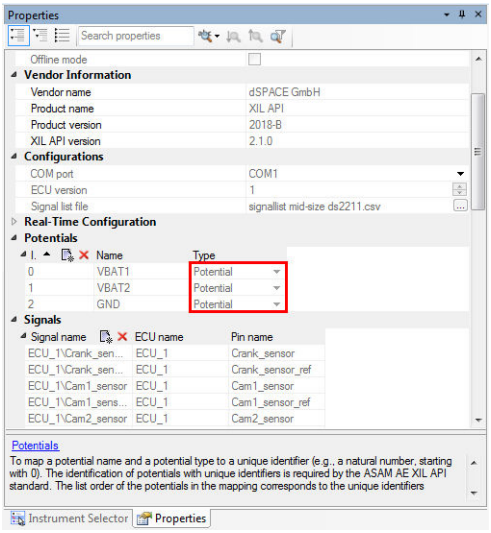
6 Select the **Type** of the dSPACE hardware that you are using for the electrical error simulation.

dSPACE Hardware	Description
Mid-Size / Full-Size (Variant 1)	A HIL system with a DS291 FIU Module, DS749 FIU Module, DS789 Sensor FIU Module, DS791 Actuator FIU Module, DS793 Sensor FIU Module, or a DS1450 Bus FIU Board. (Also valid for SCALEXIO systems that do not use the integrated SCALEXIO FIU.) Refer to Hardware for Failure Simulation (dSPACE XIL API Implementation Guide 🕮).
Full-Size (Variant 2)	A HIL system with a DS293 FIU Module. Refer to Hardware for Failure Simulation (dSPACE XIL API Implementation Guide 🕮).
SCALEXIO	A SCALEXIO system with the integrated SCALEXIO FIU. Refer to Hardware for Electrical Error Simulation on SCALEXIO Systems (SCALEXIO Hardware Installation and Configuration 🕮).

7 Specify the **Configuration** properties according to the dSPACE hardware that you are using for the electrical error simulation. Refer to EESPort - Configuration Properties on page 88.

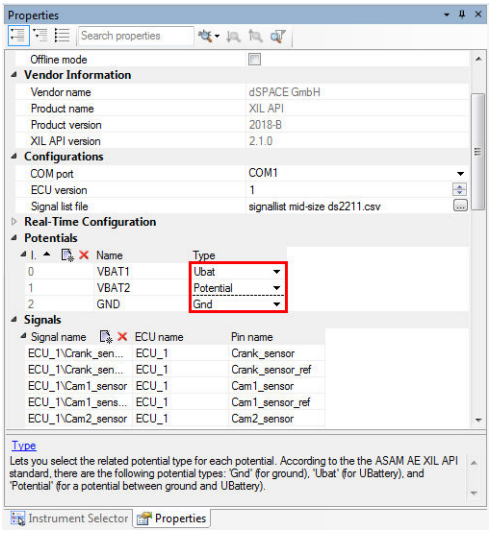
8 Click **Finish** to create the EESPort (and a new port configuration file).

ControlDesk creates and configures a new EESPort. The EESPort has a *preconfigured* potential mapping based on the potential names that are provided for your selected failure simulation hardware. By default, the **Type** of all the potentials is set to **Potential**. Refer to the following illustration:



9 On the XIL API EESPort ribbon, click **Disconnect** to disconnect the EESPort and make its properties editable.

10 In the **Properties** controlbar, edit the potential mapping according to the dSPACE hardware that you are using for the electrical error simulation. Refer to the following example:



For more information, refer to Basics on Potential Mapping on page 31 and How to Perform Potential Mapping on page 41.

11 Save the ControlDesk project.

ControlDesk saves your changes.

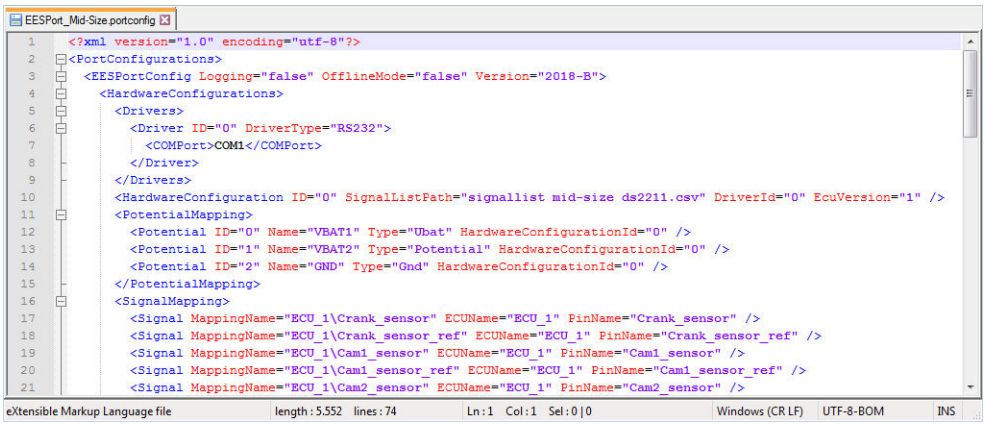
12 In the **Project** controlbar, select **Explore Folder** from the context menu of the EES port.

ControlDesk opens the folder where you can find the related port configuration file.

Result

You created a new EESPort and the related new port configuration file, completed potential mapping and saved your changes.

You can open the port configuration file in an XML editor to view its content. Refer to the following illustration:



Because you selected **Generate default signal mapping**, the PORTCONFIG file also contains a default signal mapping which is based on the ECU names and pin names that are specified for your dSPACE simulator.

Related topics

HowTos

How to Perform Potential Mapping 41

References

Configuration Files for Electrical Error Simulation (dSPACE XIL API Reference 🕮)

How to Create and Configure an Electrical Error

Objective

To create and configure an electrical error in ControlDesk's working area.

Configuring electrical errors

ControlDesk lets you configure multiple error categories in combination with multiple error and load types. Due to the abundance of combinations, this topic can show you only an example of how to create and configure an electrical error.

If an XIL API EESPort is configured, the error set's validity is checked automatically during its configuration. The configuration state of an error set, an error or a signal can be become temporarily invalid, which is indicated by a ❗ symbol.

Tip

You can display the related error message by moving and holding the mouse over the ❗ icon.

However, at the end of the configuration, the state of an error set, an error or a signal must be valid, which is indicated by a ✅ symbol. In the end, which errors you can simulate depends on the connected failure simulation hardware and the configuration of your HIL simulator for the connected ECU(s).

Precondition

You created an Electrical Error Simulation port (EESPort), refer to How to Create a New EESPort on page 38.

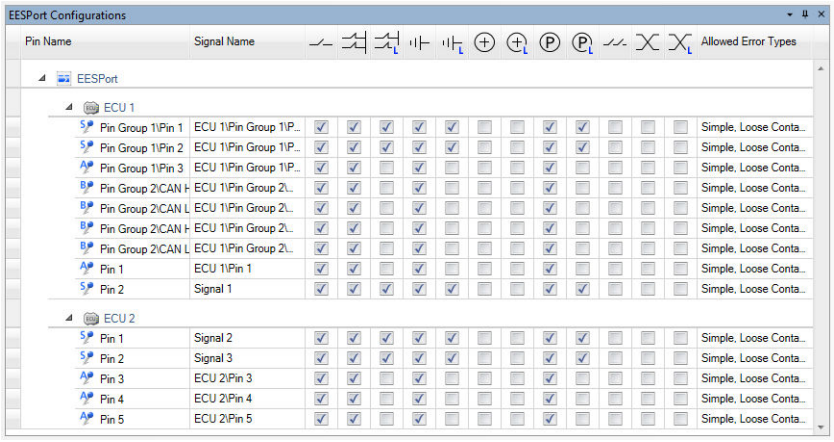
Method

To create and configure electrical errors

1 From the context menu of the EESPort in the **Project** controlbar, select **Configure** to configure the XIL API EESPort and connect it with the related failure simulation hardware.

2 On the View ribbon, click Controlbar – Switch Controlbars - EESPort Configurations to open the EESPort Configurations controlbar.

ControlDesk opens the EESPort Configurations controlbar (see the following example).



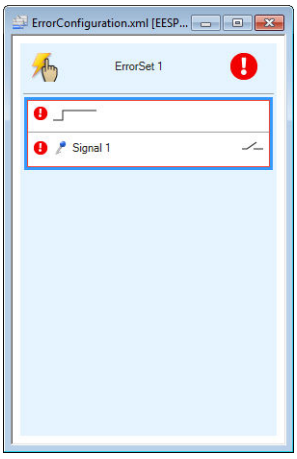
Because the XIL API EESPort is configured, the list on the EESPort Configurations controlbar now shows the related ECU pins and their allowed states for electrical error simulation.

3 Open an existing error configuration or create a new one:

- To open an error configuration, select Open from the context menu of an EESPort in the XIL API EESPort folder in the Project controlbar.
- To create a new error configuration, select New ErrorConfiguration from the context menu of an EESPort in the XIL API EESPort folder in the Project controlbar.

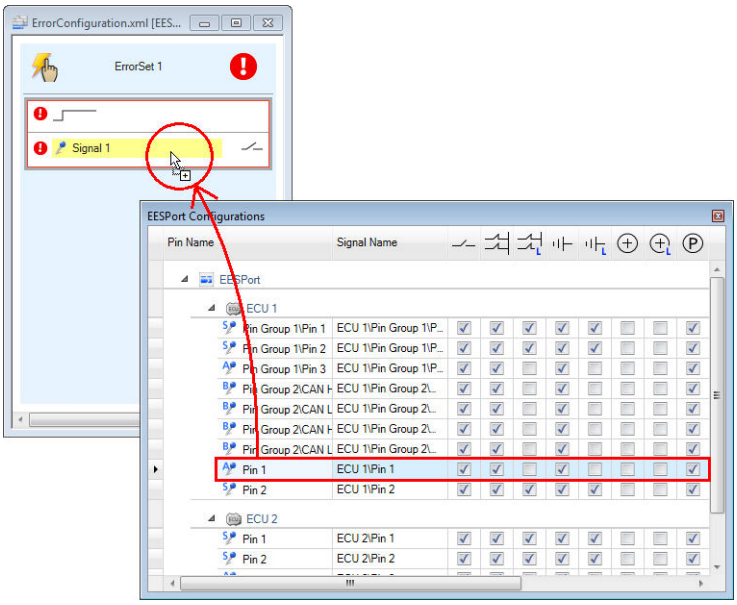
The error configuration is opened in ControlDesk's working area.

The following illustration shows an example of an error configuration containing one error set and one interrupt error where the optional signal mapping of the port configuration is missing. This is indicated by a ❗ symbol in the signal row.



You now have to map an ECU pin (i.e., a signal) to the error.

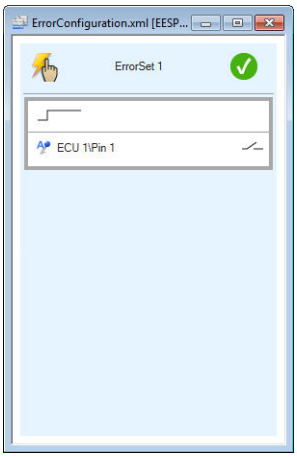
4 From the **EESPort Configurations** controlbar, select an ECU pin and drag it to the signal row of the error (see the following example).



Tip

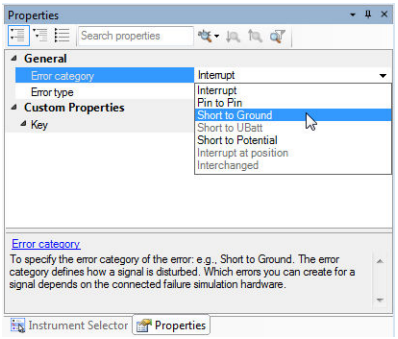
You can use the **Show Search Panel** command or press **Ctrl** + **F** to filter the list for the pin you want to use. Refer to Show Search Panel on page 146.

The ECU pin is mapped to the error.

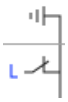


A ✓ symbol indicates if the configuration is valid.

5 Select the error in the working area. In the Properties controlbar, change the Error category property: e.g., Short to Ground.

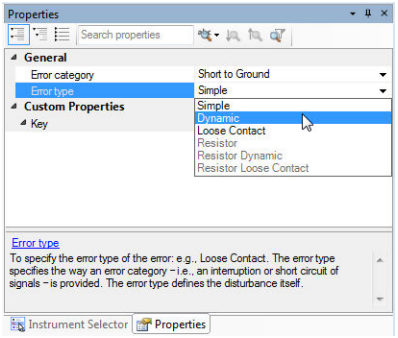


A short circuit to battery voltage is indicated by the following symbols:



The L symbol in the signal row indicates that the signal is to be disturbed without load rejection.

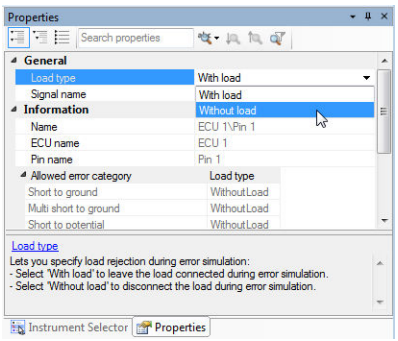
6 In the Properties controlbar, change the Error type property: e.g., Dynamic.



The Dynamic error category is indicated by the following symbol:

7 Change the Duration property: e.g., `0.1 s`.

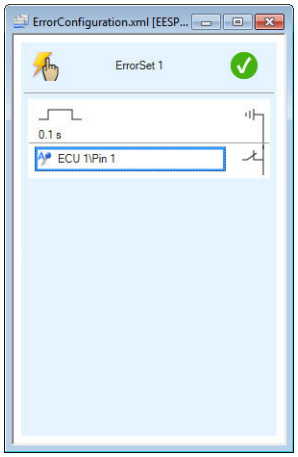
8 Select the signal in the working area. In the Properties controlbar, change the signal's Load type property, for example, to Without Load to configure a load rejection for this signal when the error is to be simulated.



The L symbol disappears from the signal row to indicate the load rejection.

Result

You created and configured an electrical error.



You can now create further errors in the error set or create further error sets.

Related topics

Basics

Basics on Electrical Errors....16
Tips and Tricks for Configuring Electrical Errors....58

References

Configure (EESPort)....118
EESPort Configurations....120
Error Configuration....81

Basics on Software Triggers

Introduction

You can activate error sets manually or via software triggers. Software triggers let you activate error sets according to defined trigger conditions or durations.

Triggers for error sets

You can specify the trigger type for each error set individually which is indicated by the following symbols.

Symbol	Description
	The error set is to be triggered manually.
	The error set is to be triggered by software.

Note

To allow the use of software triggers, you must enable this for each EESPort separately via the **Enabled** property. Refer to How to Configure a Software Trigger on page 54.

Types of software triggers

There are two types of software triggers: Software triggers defined by duration or by a condition.

Duration trigger type The error set is activated automatically after a specified duration, for example, 5 seconds. (This implies that the previous error set is deactivated automatically at the same time.)

Condition trigger type The error set is activated automatically if a specified trigger condition is fulfilled before a timeout. (This implies that the previous error set is deactivated automatically at the same time.)

You specify the trigger condition according to the ASAM General Expression Syntax (GES) by means of the **Expression Editor**. Refer to Expression Editor on page 106.

If you use a trigger condition you must specify a timeout in seconds to deactivate the error set and the error configuration automatically if the specified trigger condition cannot be fulfilled. This timeout stops the execution of the error configuration.

If you select **Offline mode** in the **General** properties of the EESPort, checking trigger conditions and monitoring the switching behavior and transition states of the failure simulation hardware is not possible. Error sets that are using software triggers with a condition are activated immediately without checking the specified trigger condition.

Using aliases/symbol names

When specifying a trigger condition by means of the **Expression Editor**, you can also work with aliases/symbol names that stand for specific measurement variables or parameters of your simulation application.

The symbol names are displayed in the **Properties** controlbar:

- If you select an error set, a list displays all aliases that you are using to specify trigger conditions for the error set.

- If you select an error configuration, a list displays all the aliases that you are using to specify trigger conditions for the single error sets of the error configuration.

You can map a measurement variable or parameter to a **Symbol name** by editing the variable path in the related edit field or via the Browse button and the **Select Variable Dialog**.

Tip

To have a clear overview, it is recommended to use only unique alias names in an error configuration.

Related topics

HowTos

How to Configure a Software Trigger 54

References

EESPort – Real-Time Configuration Properties 94
Error Set - General Properties 104
Error Set - Software Trigger Properties 105
Error Set - Variable Mapping Properties 105
Expression Editor 106
Select Variable Dialog (ControlDesk Variable Management 🕮)

How to Configure a Software Trigger

Objective

To enable and configure a software trigger.

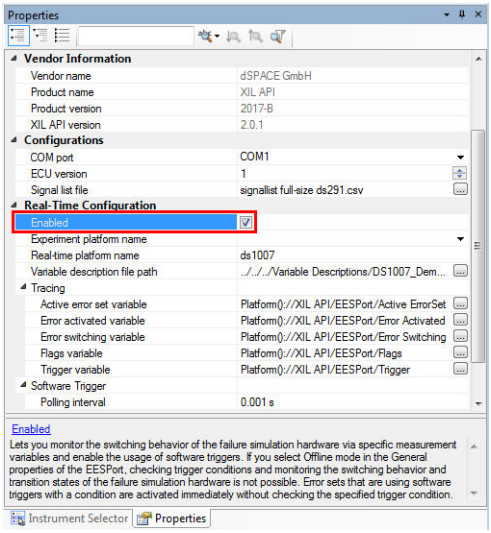
Method

To configure a software trigger

1 In the **Project** controlbar, select the EESPort for which you want to allow the use of software triggers.

The **Properties** controlbar displays the properties of the selected EESPort.

2 Select Real-Time Configuration – Enabled to allow the use of software triggers.



Note

- This property also enables you to monitor the switching behavior of the failure simulation hardware via specific measurement variables when you perform electrical error simulation.
- If you select Offline mode in the General properties of the EESPort, checking trigger conditions and monitoring the switching behavior and transition states of the failure simulation hardware is not possible. Error sets that are using software triggers with a condition are activated immediately without checking the specified trigger condition.

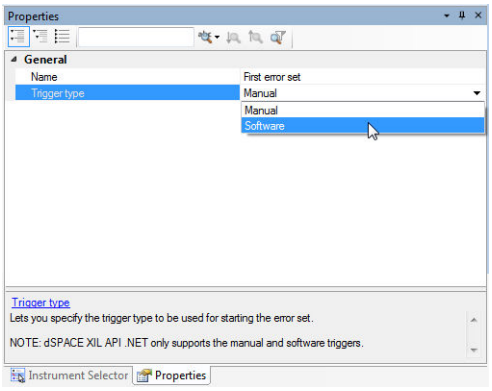
3 Open one of the EESPort's error configurations.

The error sets of the error configuration are displayed in the working area.

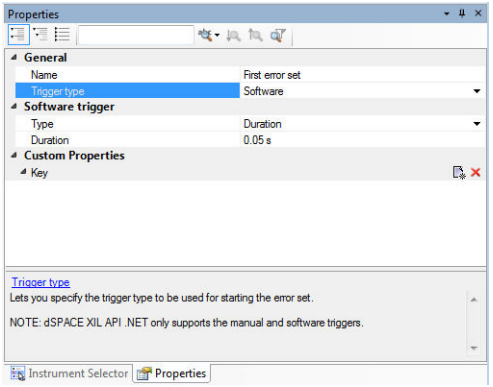
4 In the working area, select the error set you want to configure.

The Properties controlbar displays the properties of the selected error set.

5 In the Properties controlbar, specify the Trigger type as Software.



The **Software trigger** properties appear in the **Properties** controlbar.



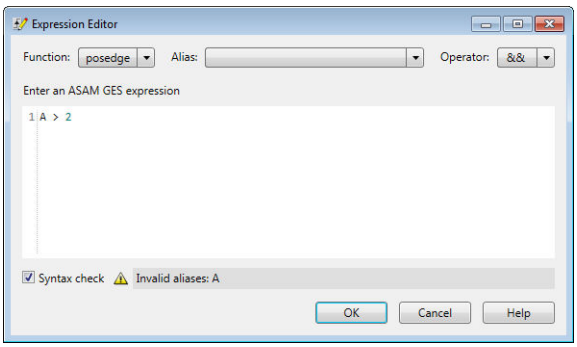
By default, the **Type** of the software trigger is preselected as **Duration**.

6 Change the **Type** of the software trigger to **Condition** to specify a trigger condition instead of the preselected duration trigger.

Further properties are displayed.

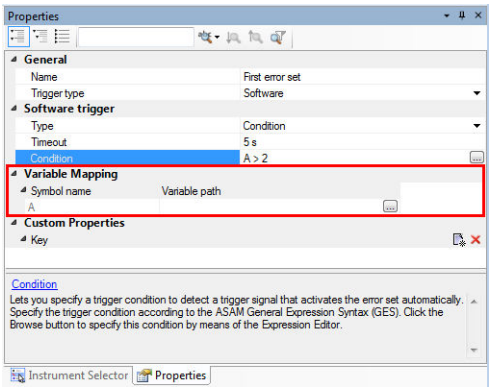
7 Click the Browse button in the row of the **Condition** property.

ControlDesk opens the **Expression Editor** to specify a trigger condition in the ASAM General Expression Syntax (GES) as a string. Refer to the following example.



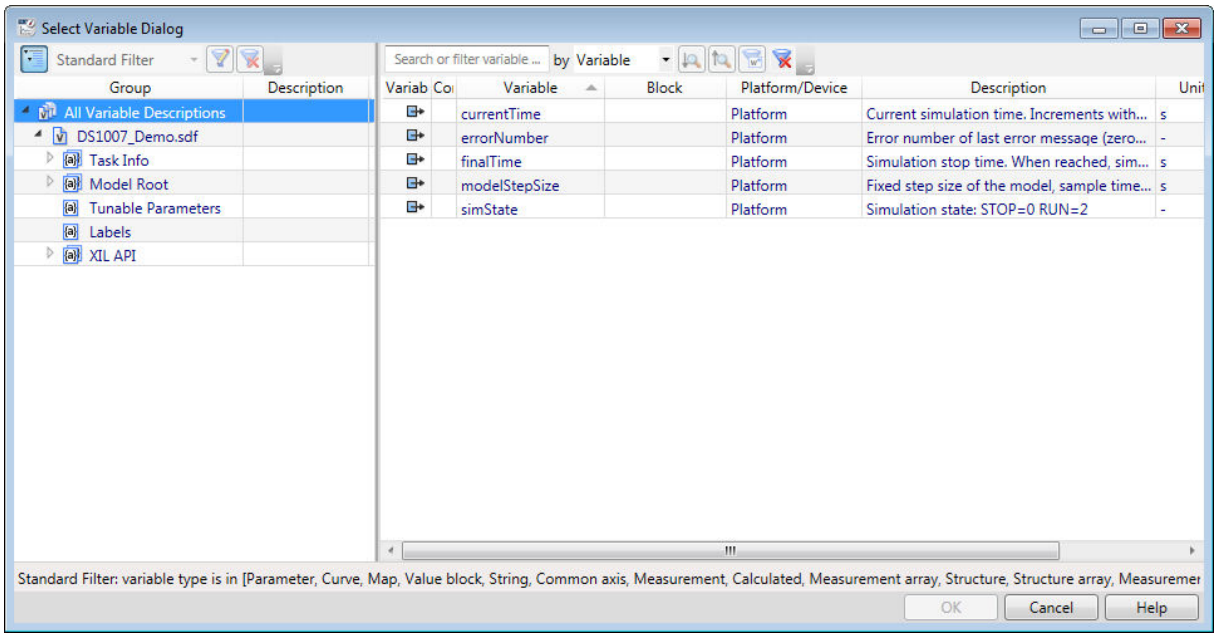
For more information, refer to Expression Editor on page 106.

8 If you enter an expression that contains a symbol name, the symbol name appears in the **Variable Mapping** properties section of the error set when you close the **Expression Editor**. Refer to the following example.



9 To map a measurement variable or parameter to this alias, click the Browse button in the related **Variable path** field.

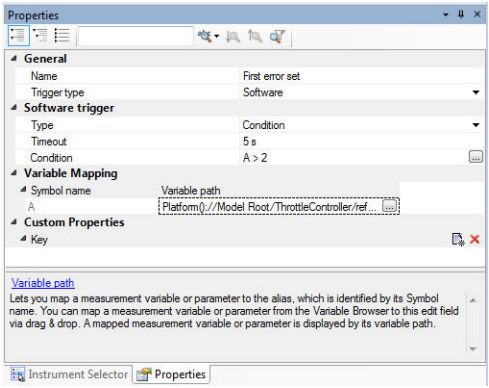
The **Select Variable Dialog** opens.



10 Select a measurement variable or parameter in the **Select Variable Dialog**. Click **OK** to confirm your selection and close the dialog.

Result

You enabled software triggers and configured a condition for a software trigger. The related variable mapping is displayed in the **Properties** controlbar. Refer to the following example.



In the working area, the software trigger of the error set is indicated by a  symbol.

Related topics

Basics

Basics on Software Triggers..... 52

References

EESPort – Real-Time Configuration Properties..... 94
Error Set - General Properties..... 104
Error Set - Software Trigger Properties..... 105
Error Set - Variable Mapping Properties..... 105
Expression Editor..... 106
Select Variable Dialog (ControlDesk Variable Management 🕮)

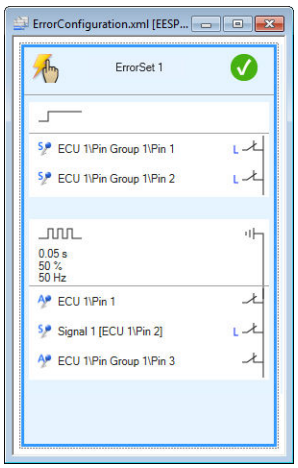
Tips and Tricks for Configuring Electrical Errors

Introduction

When configuring errors, you can use standard commands or drag & drop to simplify your work.

Creating pin-to-pin and multi-pin errors

The following illustration shows an example of an error set with a pin-to-pin error and a multi-pin error.



There are two ways to create pin-to-pin and multi-pin errors:

- You can drag further pins from the **EESPort Configurations** controlbar to an existing error in the working area.
- You can select multiple pins in the **EESPort Configurations** controlbar and drag them to an error set in the working area.

Changing the selected pins

The following methods are available for changing the pins you mapped to one or more electrical errors:

- You can change a single pin of an error by dragging another pin from the EESPort Configurations controlbar to the related signal row.
- You can replace a pin in all the error sets of an error configuration in one step if you drag another pin to a related signal row by holding down the *right* mouse key. This will open a context menu which lets you change the pin globally.
- You can change a single pin by editing the Signal name property in the Properties controlbar.

Moving errors

You can move errors in an error set or from one error set to another via drag & drop.

Copying errors

The following methods are available for copying errors from one error set to another or to the same error set:

- You can the use standard Copy and Paste commands to copy errors.
- You can copy errors between error sets via drag & drop while holding the **Ctrl** key.

Moving error sets

You can move error sets in an error configuration via drag & drop.

Copying error sets

The following methods are available for copying error sets:

- You can use the standard Copy and Paste commands to copy errors sets between error configurations or in an error configuration.
- You can also copy error sets in an error configuration via drag & drop while holding the **Ctrl** key.

Related topics

Basics

Overview of the Graphical User Interface in ControlDesk 25

HowTos

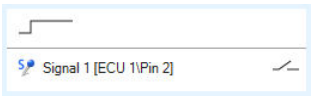
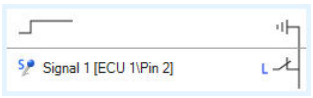
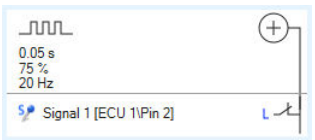
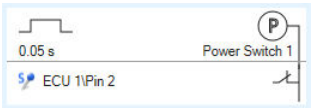
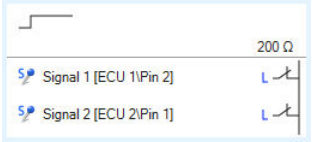
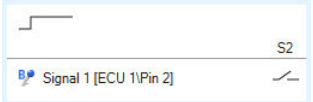
How to Create and Configure an Electrical Error 48

Examples for Configured Errors

Introduction

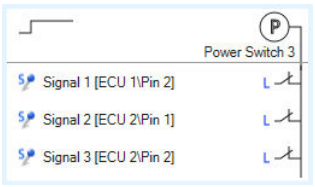
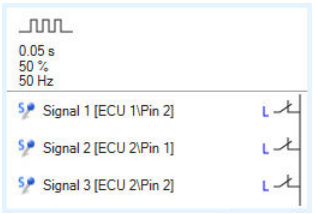
The following tables show how typical errors are displayed in ControlDesk. Which errors you can create for a signal depends on the connected failure simulation hardware.

Single pin failures

Error	Error Properties			Signal Properties	
	Error Category	Error Type	Parameters	Signal Name	Load Type
Cable break (open circuit) 	Interrupt	Simple	None	Signal 1	–
Short circuit to ground (GND) 	Short to ground	Simple	None	Signal 1	With load
Pulsed short circuit to $U_{Battery}$ 	Short to UBat	Loose contact	Duration: 0.05 s Duty cycle: 75 % Frequency: 20 Hz	Signal 1	With load
Dynamic short circuit to a potential 	Short to potential	Dynamic	Duration: 0.05 s Potential: e.g., 0[1)] Potential name: e.g., Power Switch 1[1)]	Signal 1	Without load
Short circuit between two pins 	Pin to pin	Resistor	Resistor: 200 Ω	Signal 1 Signal 2	With load With load
Cable break (open circuit) of a bus channel 	Interrupt at position	Simple	Interrupt position: S2	Signal 1	–

[1)] This value depends on the individual potential mapping.

Multiple pin failures

Error	Error Properties			Signal Properties	
	Error Category	Error Type	Parameters	Signal name	Load Type
Short circuits between multiple pins and a potential 	Short to potential	Simple	Potential: e.g., 2[1)] Potential name: e.g., Power Switch 3[1)]	Signal 1 Signal 2 Signal 3	With load With load With load
Pulsed short circuits between multiple pins 	Pin to pin	Resistor loose contact	Duration: 0.05 s Duty cycle: 50 % Frequency: 50 Hz	Signal 1 Signal 2 Signal 3	With load With load With Load

[1)] This value depends on the individual potential mapping.

Related topics

References

Error 80

Performing Electrical Error Simulation

Where to go from here

Information in this section

Basics on Performing Electrical Error Simulation

Introduction

ControlDesk lets you download and control error configurations on a connected failure simulation hardware to perform electrical error simulation with it.

Basics on downloaded error configurations

Executing error configurations To execute an error configuration, it has to be downloaded to the failure simulation hardware and started (activated).

The error sets of an error configuration are executed in consecutive order on the failure simulation hardware. The first error set is activated when a first trigger is set. An error set remains active until the next trigger is set.

The errors of the last error set remain active as long as the error configuration is active.

Using empty error sets To get a defined end of the electrical error simulation without any signal disturbance, an empty error set can be used as the last error set in the error configuration.

Empty error sets can also be used to create error-free phases.

Expanding error configurations An error configuration can be dynamically expanded with further error sets after the download.

If you added further error sets, use the **Update** command to update the error configuration on the failure simulation hardware.

Related topics

HowTos

How to Perform Electrical Error Simulation

Objective

You can use ControlDesk with the Failure Simulation Package to test and initiate your failure simulation hardware.

Precondition

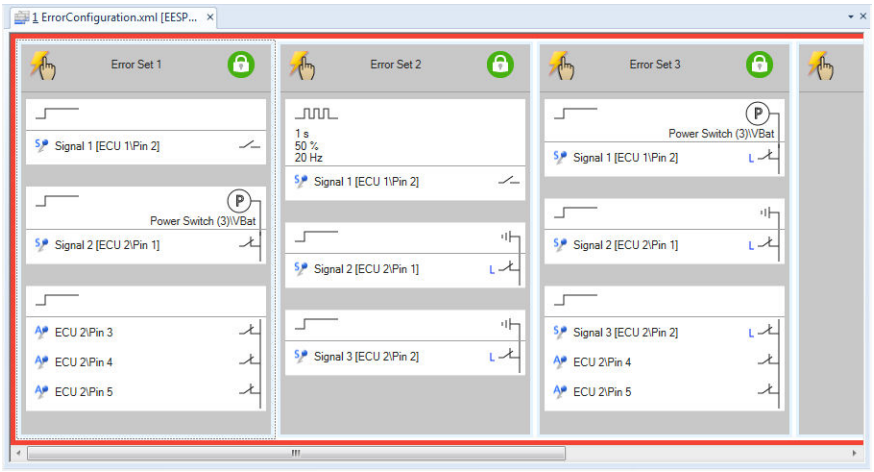
You opened and configured an error configuration in ControlDesk's working area.

Method

To perform electrical error simulation

1 On the XIL API EESPort ribbon, click ErrorConfigurations – Download or use the Download context menu command of the error configuration, to download the error configuration to the failure simulation hardware that is connected to the host PC.

ControlDesk checks the error configuration for inconsistencies and downloads it to the failure simulation hardware if the configuration is valid. In the working area, the error configuration is displayed with a red frame when it is downloaded and the background color of the error sets changes to gray to indicate that they are in read-only mode (see the following example).



The symbols indicate that the error sets can no longer be edited or removed.

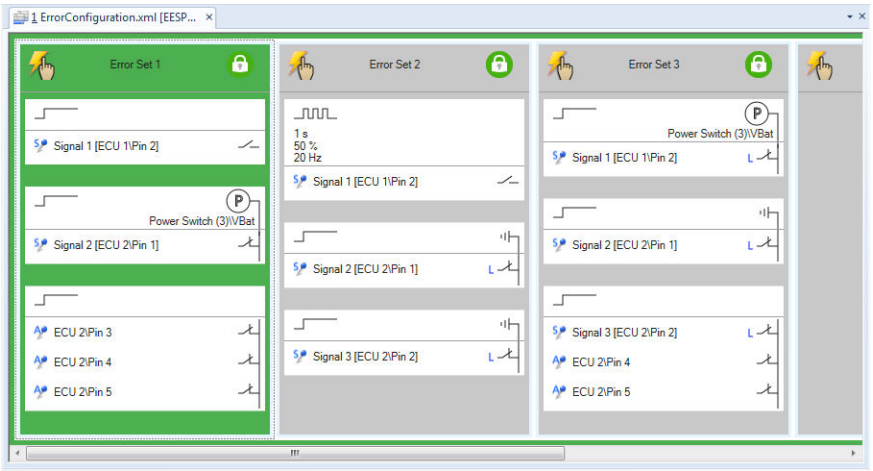
2 Click XIL API EESPort – ErrorConfigurations – Activate to activate it on the failure simulation hardware.

The frame color of the error configuration changes to green to indicate that the error configuration is activated, which means that it is running on the failure simulation hardware. The failure simulation hardware waits for the first trigger to execute the first error set.

3 If you specified a manual trigger, click XIL API EESPort – ErrorConfigurations – Trigger to activate the first error set.

If you specified a software trigger, the first error set is activated automatically according to the related specification, for example, after the duration of 5 seconds.

The background color of the first error set changes to green to indicate that its errors are active on the failure simulation hardware (see the following example). The name of the active error set is also displayed in the **Properties** controlbar. Refer to the following example.



The errors are active until the next error set is triggered.

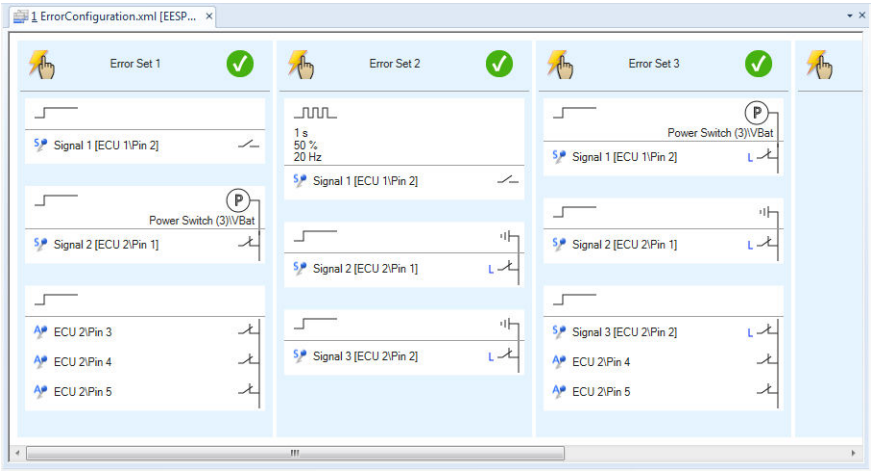
4 Repeat step 3 for all the error sets of the error configuration.

5 Click **XIL API EESPort – ErrorConfigurations – Deactivate** to deactivate and stop the execution of the error configuration on the failure simulation hardware.

The frame color of the error configuration changes to red.

6 Click **XIL API EESPort – ErrorConfigurations – Unload** to unload the error configuration from the failure simulation hardware.

In the working area, the colored frame of the error configuration is removed and the background color of the error sets changes to light blue.



Result	You performed electrical error simulation on the connected failure simulation hardware.
Next steps	For extensive and automated HIL tests, you must use dSPACE XIL API .NET and its EESPort implementation, which is also delivered with the Failure Simulation Package. For further information, refer to Implementing an EESPort Client Application (dSPACE XIL API Implementation Guide 🕮) and Electrical Error Simulation Port Implementation (dSPACE XIL API Reference 🕮).

Monitoring the Switching Behavior of the Failure Simulation Hardware

Where to go from here

Information in this section

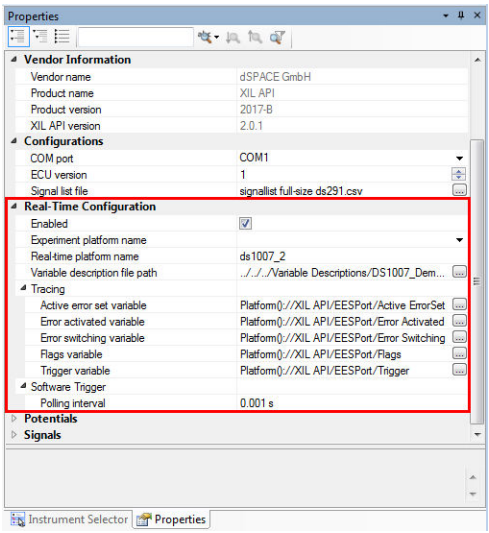
Basics on Monitoring the Switching Behavior of the Failure Simulation Hardware

Introduction

You can monitor the estimated switching behavior and transition states of the failure simulation hardware via specific measurement variables.

Basics

Each EESPort provides **Real-Time Configuration** properties for monitoring the switching behavior of the failure simulation hardware in connection with a related real-time application and HIL simulator. Refer to the following example:



The **Tracing** variables for monitoring the switching behavior are generated into the resulting variable description file when you build the simulation application of a real-time model with software of dSPACE-Release 2016-A and later. These variables are independent from other model variables. They do not measure the real behavior of the failure simulation hardware, they only estimate it according to empirical values. The variables are written and read only by the XIL API EESPort.

You can enable the monitoring of the switching behavior via the **Enable** property of the EESPort. Refer to EESPort – Real-Time Configuration Properties on page 94.

Variables for monitoring the switching behavior

Using the default tracing variables When you build the simulation application, the following tracing variables are generated into the related variable description file.

Variable Name	Purpose	Values
`ActiveErrorSet`	To display the number of the error set that is currently activated.	▪ 0 = No error set is activated ▪ 1 … 500 = Number of the error set that is currently activated
`ErrorActivated`	To indicate whether one or more errors are activated.	▪ 0 = No error is activated ▪ 1 = One or more errors are activated
`ErrorSwitching`	To indicate the undefined transition state when switching the failure simulation hardware.	▪ 0 = No switching is in process. The failure simulation hardware is in the steady state. ▪ 1 = Switching is in process. The failure simulation hardware is in the transition state. The error is switched physically during this transition state.

Variable Name	Purpose	Values
`Flags`	This variable is used internally. Do not use this variable for monitoring.	–
`Trigger`	This variable is used internally to detect software triggers.	–

These tracing variables for monitoring are located in the `XIL API/EESPort` subgroup. For multiprocessor models, the subgroup is generated into each submodel separately.

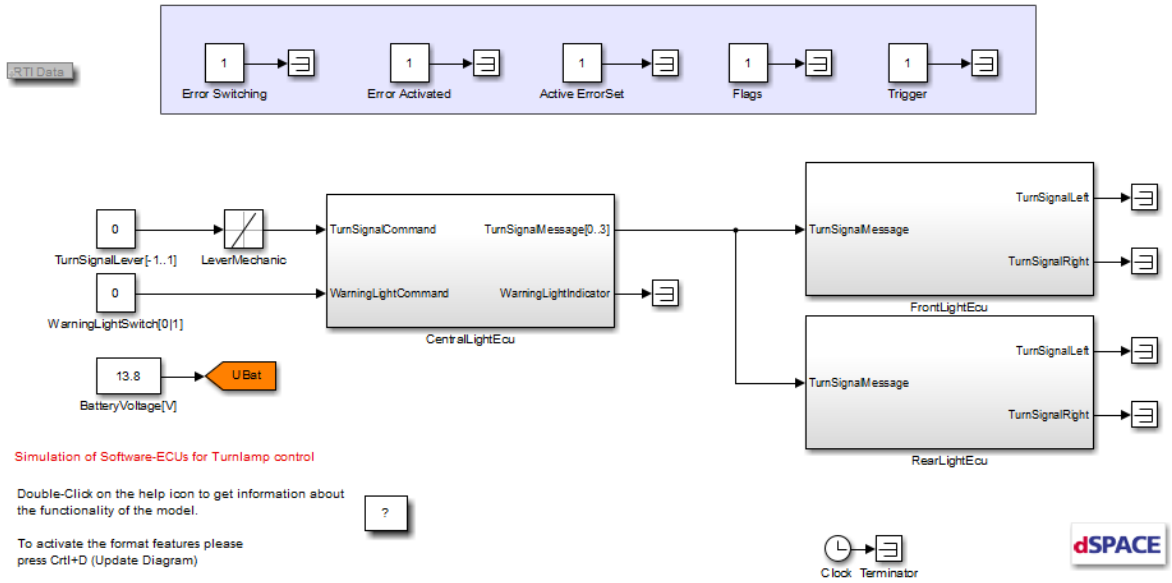
Note

- The tracing variables are not available for DS1104 platforms.
- If you execute multiple EESPorts in parallel, you must ensure that the tracing variables of the EESPorts have different name paths or variable names. Refer to Using alternative variables.
- If you use the integrated SCALEXIO FIU in a SCALEXIO multi-processor system, you must use the tracing variables of the SCALEXIO processing unit that is specified to control the electrical error simulation. (Although the tracing variables are generated into each application process, they are only written on the processing unit that controls the electrical error simulation.)

Tip

In addition to the measurement variables described in this topic, variable description (TRC) files of SCALEXIO systems that use the integrated SCALEXIO FIU also contain the `Diagnostics/Failure Simulation` subgroup. This subgroup provides further variables for monitoring the switching behavior of the SCALEXIO failure simulation hardware. Refer to Monitoring Additional Behavior of an Integrated SCALEXIO FIU on page 73.

Using alternative variables If you want to use alternative variables to monitor the switching behavior of the failure simulation hardware, you have to prepare your Simulink model manually. It is recommended to add Simulink Constant blocks to the model to provide representations for the EESPort tracing variables in the `Model Root` subgroup of the variable description file. These Constant blocks must be terminated and must not have connections to other blocks that might modify their values. Refer to the following illustration.



When you specify the port configuration (PORTCONFIG) file for the EESPort, you have to replace the default tracing variables with your alternative variables.

Refer to the following example.

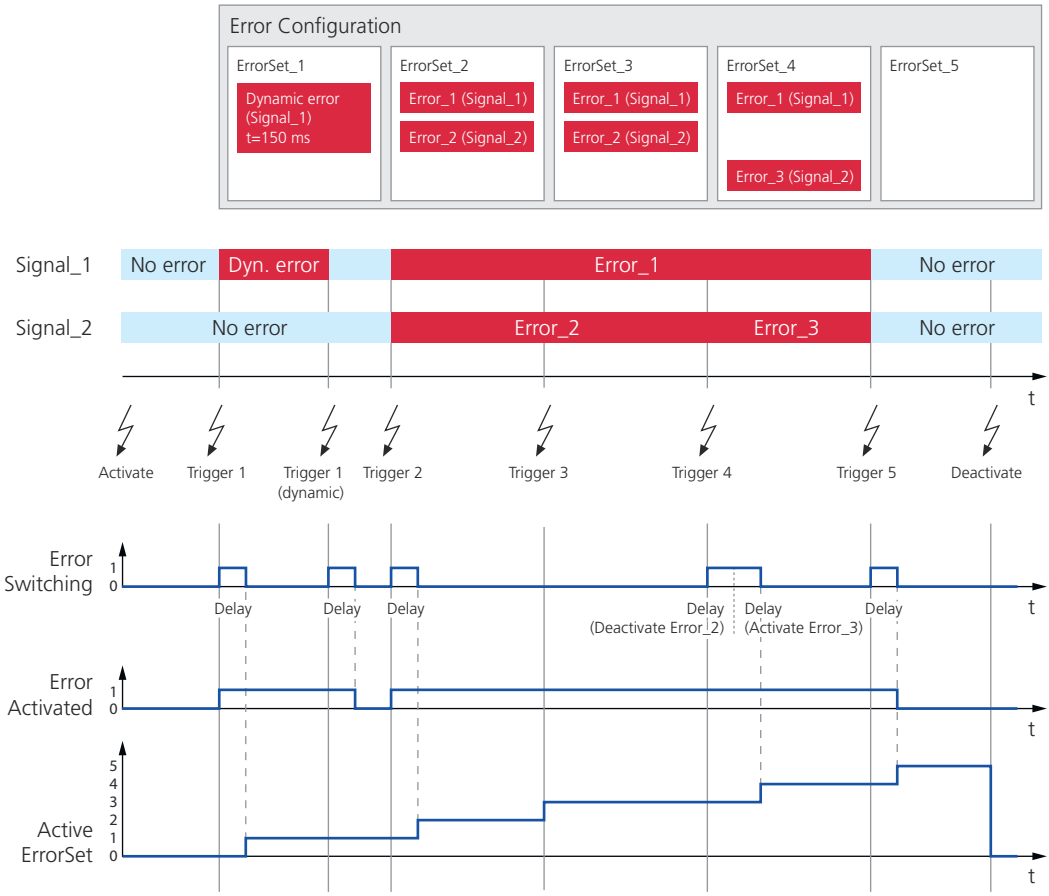
```
<RealTimeConfiguration PlatformName="SCALEXIO Real-Time PC" SystemDescriptionFilePath="../../../Variable Descriptions/
turnlamp.sdf/turnlamp.sdf">
    <Tracing Enabled="true">
    <Variable Value="Platform():/XIL API/EESPort/Error Activated" Type="ErrorActivated" />
    <Variable Value="Platform():/XIL API/EESPort/Active ErrorSet" Type="ActiveErrorSet" />
    <Variable Value="Platform():/XIL API/EESPort/Error Switching" Type="ErrorSwitching" />
    <Variable Value="Platform():/XIL API/EESPort/Flags" Type="Flags" />
    <Variable Value="Platform():/XIL API/EESPort/Trigger" Type="Trigger" />
    </Tracing>
</RealTimeConfiguration>
```

Note

If you perform electrical error simulation with an integrated SCALEXIO FIU, you can not specify alternative variables. You must use the default tracing variables.

Example

Switching the failure simulation hardware consumes time for communication and relay switching. The following example shows how the various measurement variables will display the estimated switching behavior of the failure simulation hardware in response to different commands (e.g., triggers) from the host PC where the XIL API EESPort is executed.



The example shows an error configuration with five error sets that disturb two signals with errors. The displayed switching behavior of the failure simulation hardware mainly depends on hardware-dependent delay times (e.g., for relay switching or communication) which can differ from each other.

Related topics

Basics

Latencies when Performing Electrical Error Simulation (dSPACE XIL API Implementation Guide 🕮)
Monitoring the Switching Behavior of Electrical Error Simulation Hardware (dSPACE XIL API Implementation Guide 🕮)

HowTos

How to Configure to Monitor the Switching Behavior of the Failure Simulation Hardware....71

References

EESPort – Real-Time Configuration Properties....94

How to Configure to Monitor the Switching Behavior of the Failure Simulation Hardware

Objective

You can configure to monitor the switching behavior of the failure simulation hardware when creating an EESPort or later via the EESPort's properties.

This topic shows how to configure the EESPort's properties via the Properties controlbar.

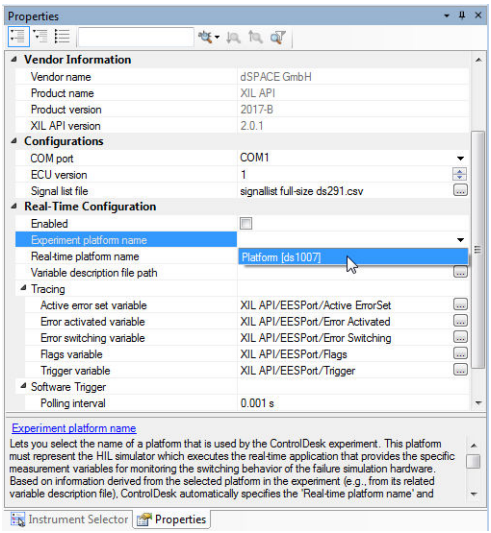
Method

To configure to monitor the switching behavior of the failure simulation hardware

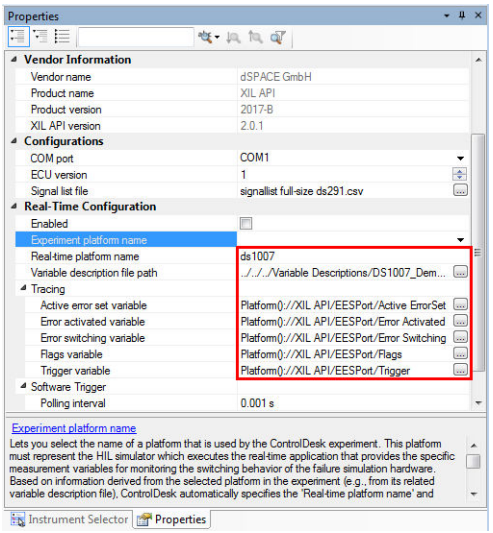
1 Open the Properties controlbar for the EESPort you want to configure.

2 If not configured, use the Real-Time Configuration – Experiment platform name property to select the name of a platform that is used by the ControlDesk experiment. This platform must represent the HIL simulator which executes the real-time application that provides the specific measurement variables for monitoring the switching behavior of the failure simulation hardware.

Refer to the following example:



Based on information derived from the selected platform in the experiment (e.g., from its related variable description file), ControlDesk automatically specifies the **Real-time platform name** and **Variable description file path** properties and the names of the **Tracing** variables. Refer to the following example:



The properties and variable names are stored in the port configuration (PORTCONFIG) file of the EESPort.

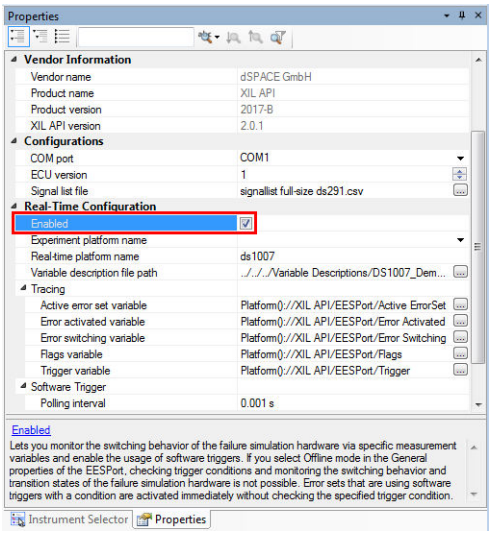
Note

If you use a SCALEXIO system with the integrated SCALEXIO FIU, the **Configurations – Real-time platform name** and the **Real-Time Configuration – Real-time platform name** must be the same.

Note

The **Experiment platform name** is not stored in the port configuration (PORTCONFIG) file of the EESPort. It is only a convenience function in ControlDesk for specifying the **Real-time platform name** and **Variable description file path** properties and variable names in one step. You can also specify these properties and variable names manually without selecting an **Experiment platform name**.

3 Select **Enabled** to monitor the switching behavior of the failure simulation hardware via specific measurement variables when you perform electrical error simulation. (This property also enables the use of software triggers.)



Note

If you select **Offline mode** in the **General** properties of the EESPort, checking trigger conditions and monitoring the switching behavior and transition states of the failure simulation hardware is not possible. Error sets that are using software triggers with a condition are activated immediately without checking the specified trigger condition.

Result

You configured the **Real-Time Configuration** properties of the EESPort for monitoring the switching behavior of the failure simulation hardware.

When you perform electrical error simulation, you can now capture the specified **Tracing** variables with ControlDesk to monitor the switching behavior of the failure simulation hardware. For more information on the **Tracing** variables, refer to Basics on Monitoring the Switching Behavior of the Failure Simulation Hardware on page 66.

Related topics

References

EESPort – Real-Time Configuration Properties.....94

Monitoring Additional Behavior of an Integrated SCALEXIO FIU

Introduction

When performing electrical error simulation, you can monitor additional switching behavior of the integrated SCALEXIO FIU. Different SCALEXIO-specific

measurement variables let you trace the failure simulation hardware's transition states in addition to the tracing variables of an EESPort.

Variables for monitoring the switching behavior

The `Diagnostics/Failure Simulation` subgroup of the TRC file contains the following variables for monitoring the switching behavior of the integrated SCALEXIO FIU.

Variable Name	Purpose	Values
`Client connected`	To indicate whether a connection to the failure simulation hardware is established.	0 = No connection to the failure simulation hardware is established. 1 = A connection to the failure simulation hardware is established.
`Failure activated`	To indicate whether one or more failures are activated.	0 = No failure is activated. 1 = One or more failures are activated.
`Failure configured`	To indicate whether one or more failures are configured by the failure simulation hardware.	0 = No failure is configured. 1 = One or more failures are configured.
`Failure sequence count`	To display the number of state changes of the failure simulation hardware.[1)]	0 = No failure is configured. 1 … 65535 = Number of the failure sequence that is currently activated.
`Relay switching`	To display the response times of FRU relays. The response times are worst-case assumptions for switching relays[2)]. In reality, relay switching can be faster. During the response time, the failure simulation hardware can be in an undefined transition state.	0 = No relay switching in process. Failure simulation system is in steady state. 1 = Relay switching in process. Failure simulation system is in transition state.

1) The state changes of the failure simulation hardware might not equal the number of errors you specified.

2) For details on the response time of FRU relays, refer to FRU Relays Data Sheet (SCALEXIO Hardware Installation and Configuration 🕮).

For technical details on the SCALEXIO failure simulation hardware, refer to Electrical Error Simulation Concept (SCALEXIO Hardware Installation and Configuration 🕮).

Related topics

Basics

Basics on Monitoring the Switching Behavior of the Failure Simulation Hardware..... 66

Examples

Example of Monitoring the States of the Integrated SCALEXIO FIU..... 75

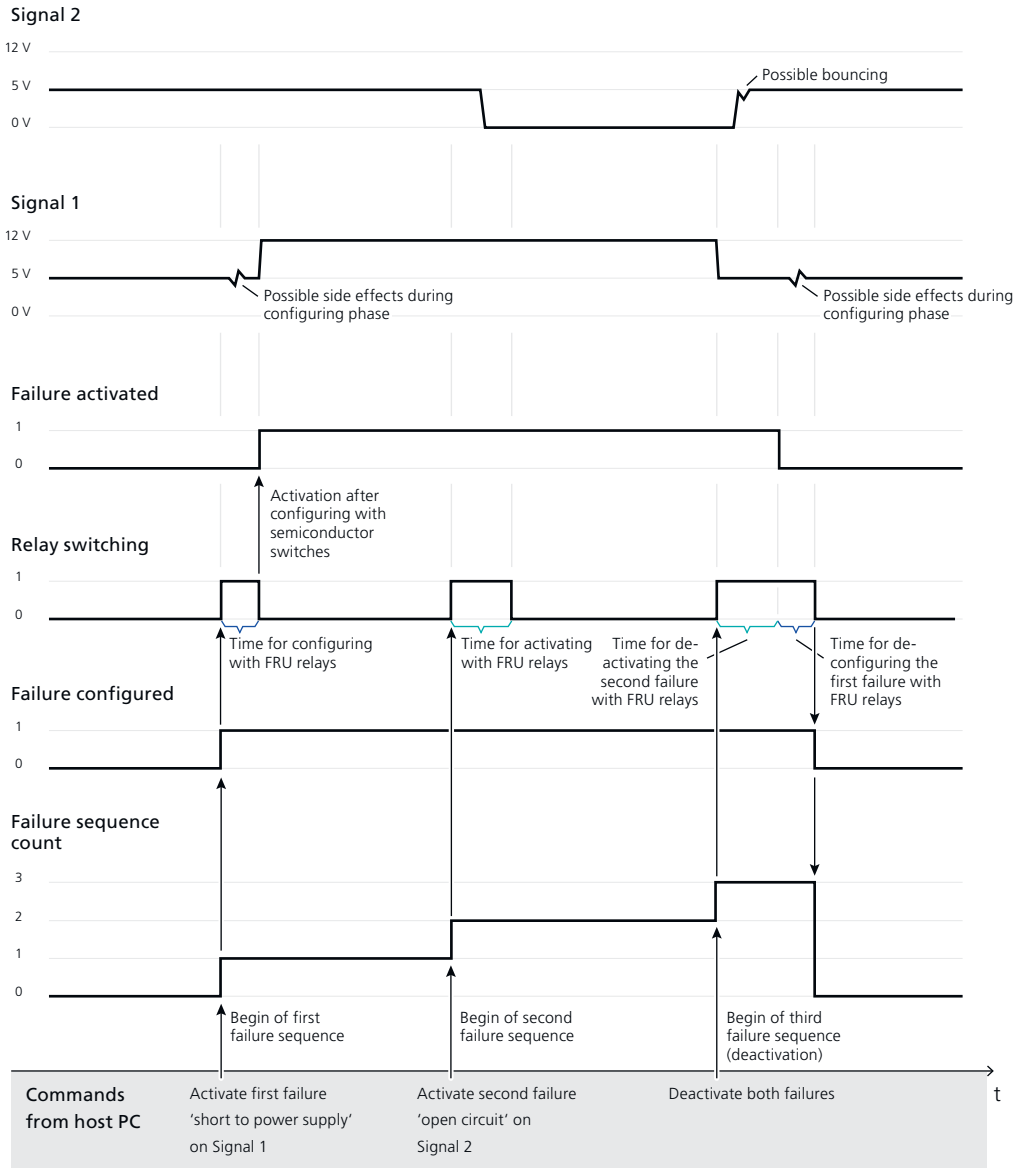
Example of Monitoring the States of the Integrated SCALEXIO FIU

Introduction

The following example shows the switching behavior of the integrated SCALEXIO FIU in response to different commands.

Example description

In the example, two failures are switched in succession and then cleared at the same time.



Switching the first failure *Signal 1* is to be shorted to V_{BAT} (12 V). The failure is first (pre)configured by FRU relays in the signal's failure routing unit. During this configuration phase, which equals an assumed relay response time,

side effects can occur. After the configuration phase, the failure is activated by the semiconductor switches of the central FIU.

Switching the second failure An open circuit is to be simulated for *Signal 2*. (Pre)configuration is impossible because the semiconductor switches of the central FIU are still in use. That is why this second failure is activated by FRU relays in the signal's failure routing unit. The state change of *Signal 2* is performed during the assumed response time of the FRU relays. In the example, the open circuit results in a potential drop of *Signal 2* from 5 V to 0 V.

Clearing the failures The deactivation of both failures is started at the same time. While the first failure is deactivated immediately via semiconductor switches, the second failure must be deactivated by FRU relays. This can be accompanied by contact bouncing. After an assumed relay response time for deactivating the second failure, the first failure is deconfigured by FRU relays, which can be accompanied by side effects.

When all the failures are cleared, the failure sequence count is zero.

Related topics

Basics

Monitoring Additional Behavior of an Integrated SCALEXIO FIU.....73

Reference Information

Where to go from here

Information in this section

Electrical Error Simulation Descriptions

Where to go from here

Information in this section

EESPort

Introduction

Electrical Error Simulation port (EESPort) An *Electrical Error Simulation port* (EESPort) provides access to a failure simulation hardware for simulating electrical errors in an ECU wiring according to the ASAM AE XIL API standard. The configuration of the EESPort is described by a hardware-dependent *port configuration* and one or more *error configurations*.

If you are using different types of failure insertion units (FIUs) in your HIL system, you must specify an EESPort for each FIU type. These EESPorts will operate independently.

Description

EESPorts are located in the **XIL API EESPorts** folder of the **Project** controlbar.

In the Windows file system, each EESPort has a related folder that can contain:

- A local working copy of the port configuration (PORTCONFIG) file
- Error configuration (XML) files
- A local working copy of the signal list (CSV) file (for a HIL simulator that uses a discrete FIU).

EESPort symbols

An EESPort has the following symbols in the Project controlbar:

-  (closed)
-  (opened, but not connected to the failure simulation hardware)
-  (opened and connected to the failure simulation hardware)

EESPort properties

An EESPort provides the following properties and settings:

Purpose	Refer to
To specify properties that are related to the hardware you are using for electrical error simulation.	EESPort - Configuration Properties on page 88
To display the general properties of the selected XIL API EESPort.	EESPort - General Properties on page 91
To specify a mapping of the potentials provided by the simulator to unique identifiers you need according to the ASAM AE XIL API standard.	EESPort – Potentials Properties on page 93
To specify properties for monitoring the switching behavior of the failure simulation hardware in connection with a related real-time application and HIL simulator.	EESPort – Real-Time Configuration Properties on page 94
To perform an optional mapping of ECU pins that you are using in your dSPACE HIL simulator to abstract signal names according to the ASAM AE XIL API standard.	EESPort – Signals Properties on page 97
To display information on the electrical error simulation port (XIL API EESPort) implementation that is installed on the host PC.	EESPort – Vendor Information Properties on page 97

Related commands

An EESPort provides the following commands:

Purpose	Refer to
To close an XIL API EESPort.	Close (EESPort) on page 117
To configure the XIL API EESPort with vendor-specific information, i.e., the properties you specified via the port configuration file.	Configure (EESPort) on page 118
To disconnect the XIL API EESPort from the related failure simulation hardware.	Disconnect (EESPort) on page 119
To export a port configuration as a PORTCONFIG file.	Export PortConfiguration on page 124
To import an error configuration as an XML file from the Windows file system.	Import ErrorConfiguration on page 130
To create a new error configuration.	New ErrorConfiguration on page 137
To open an XIL API EESPort.	Open (EESPort) on page 139
To open the Properties controlbar.	Properties on page 140
To reload the original port configuration for an EESPort.	Reload PortConfiguration on page 141
To replace the port configuration of an EESPort with a new one.	Replace PortConfiguration on page 142

Related topics

HowTos

How to Create a New EESPort 38

References

Insert EESPort 131

Error

Introduction

Error An electrical error that is specified by:

- An error category
- An error type
- A load type

Description

In the working area, an error can display the following symbols:

Symbol	Description
	There is a configuration error. You can display the related error message by moving and holding the mouse over this icon.

Error properties

The item provides the following properties and settings:

Purpose	Refer to
To specify custom properties for an error.	Error - Custom Properties on page 98
To configure the general properties of an electrical error, such as the error category and the error type.	Error - General Properties on page 99
To specify specific error parameters.	Error - Parameters Properties on page 100

Related commands

The item provides the following commands:

Purpose	Refer to
To specify the error category of an error.	Error Category on page 123
To lock the view of the error sets displayed in the working area when triggering errors.	Lock Scrolling on page 136
To create a new error in an error set in the working area.	New Error on page 136
To create a new error set in an error configuration in the working area.	New ErrorSet on page 138
To create a new signal in an error in the working area.	New Signal on page 138
To open the **Properties** controlbar.	Properties on page 140

Related topics

Basics

Basics on Electrical Errors....16

Examples

Examples for Configured Errors....60

References

Error Configuration....81
Error Set....83
Signal....85

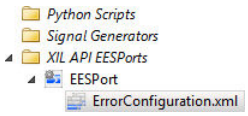
Error Configuration

Introduction

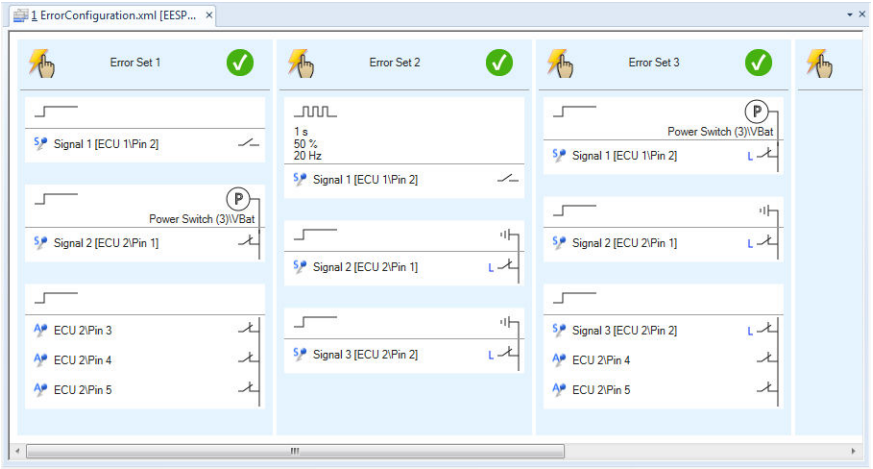
Error configuration An XML file that describes a sequence of errors you want to switch during electrical error simulation. Each error configuration comprises error sets with one or more errors.

Description

Error configuration in the Project controlbar Error configurations are displayed as children of EESPorts in the Project controlbar.



Error configuration in the working area An error configuration can be opened and configured in ControlDesk's working area. The following illustration shows an example of an error configuration that comprises three error sets.



Opened in the working area, an error configuration can have the following frame colors:

Frame Color	Description
None	The error configuration is not downloaded to the failure simulation hardware. It can be configured. You can add and remove error sets.
Red	The error configuration is downloaded to the failure simulation hardware, but not activated. You can only add new error sets.
Green	The error configuration is downloaded to the failure simulation hardware and activated. You can only add new error sets.

Error configuration in the file system An error configuration is stored as an XML file in the Windows file system. You can find the XML file in the folder of the related EESPort.

Error configuration symbols

An error configuration has the following symbol in the Project controlbar:

- (closed)
- (opened, but not downloaded to the failure simulation hardware)
- (downloaded to the failure simulation hardware, but deactivated)
- (downloaded to the failure simulation hardware and activated)

Error configuration properties

The item provides the following properties and settings:

Purpose	Refer to
To configure the general properties of the error configuration.	Error Configuration - General Properties on page 102
To display and specify aliases for the condition triggers of the error sets of the error configuration.	Error Configuration - Variable Mapping Properties on page 103
To specify custom properties for an error configuration.	Error Configuration - Custom Properties on page 102

Related commands The item provides the following commands:

Purpose	Refer to
To activate an error configuration.	Activate (Error Configuration) on page 114
To close an error configuration that is opened in ControlDesk's working area.	Close (Error Configuration) on page 117
To deactivate and stop the execution of an error configuration on the failure simulation hardware.	Deactivate (Error Configuration) on page 118
To download an error configuration to the failure simulation hardware that is connected to the host PC.	Download (Error Configuration) on page 119
To export an error configuration as an XML file.	Export (Error Configuration) on page 124
To open an error configuration in ControlDesk's working area.	Open (Error Configuration) on page 139
To open the **Properties** controlbar.	Properties on page 140
To reload the last saved error configuration to the experiment.	Reload (Error Configuration) on page 140
To save the error configuration (XML) file.	Save (Error Configuration) on page 143
To activate the first or next error set of an error configuration.	Trigger (Error Configuration) on page 148
To update an error configuration on the failure simulation hardware if you configured one or more new error sets after activating an error configuration.	Update (Error Configuration) on page 150
To unload an error configuration from the failure simulation hardware.	Unload (Error Configuration) on page 149

Related topics

Basics

Basics on Electrical Error Simulation Ports 14

HowTos

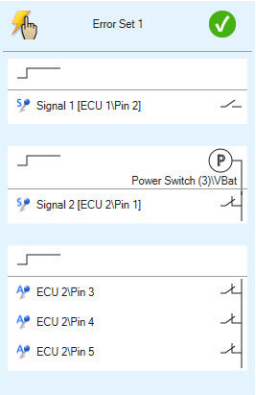
How to Create and Configure an Electrical Error 48

Error Set

Introduction **Error set** An error set is used to group errors (pin failures).

Description The errors of an error set are activated when the error set is triggered. The order of errors in the error set does not play any role in their activation.

Error sets in the working area An error set can be configured in ControlDesk's working area. The following illustration shows an example of an error set that comprises three errors.



Background Color	State Description
Light blue	The error set is not downloaded to the failure simulation hardware. The error set can be configured.
Gray	The error set is downloaded to the failure simulation hardware, but not activated. It cannot be configured.
Green	The error set is downloaded to the failure simulation hardware and activated. It cannot be configured.

In the working area, an error set can display the following symbols:

Symbol	Description
	The error set is configured correctly.
	There is a configuration error. You can display the related error message by moving and holding the mouse over this icon.
	The error set is configured correctly but read-only, for example, because it is downloaded to the failure simulation hardware.
	The error set is to be triggered manually.
	The error set is to be triggered by hardware.
	The error set is to be triggered by software.

Note

dSPACE XIL API .NET EESPort implmentation supports only manual and software triggers.

Item properties

The item provides the following properties and settings:

Purpose	Refer to
To specify custom properties for an error set.	Error Set - Custom Properties on page 104
To specify the general properties of an error set, such as the name and trigger type.	Error Set - General Properties on page 104
To specify software trigger properties that activate the error set automatically.	Error Set - Software Trigger Properties on page 105
To specify aliases to be used for the trigger condition of the error set.	Error Set - Variable Mapping Properties on page 105

Related commands

The item provides the following commands:

Purpose	Refer to
To lock the view of the error sets displayed in the working area when triggering errors.	Lock Scrolling on page 136
To create a new error in an error set in the working area.	New Error on page 136
To create a new error set in an error configuration in the working area.	New ErrorSet on page 138
To open the Properties controlbar.	Properties on page 140

Related topics

References

Signal

Introduction

Signal The communication between a hardware-in-the-loop simulator and a connected device under test (DUT), i.e., an ECU, is performed via electrical signals. Each electrical signal is related to an ECU pin. To disturb a signal via electrical error simulation, the signal must be mapped to an error.

Description

In the working area, a signal can display the following symbols:

Symbol	Description
	There is a configuration error. You can display the related error message by moving and holding the mouse over this icon.

Item properties

The item provides the following properties and settings:

Purpose	Refer to
To specify the general properties of a signal, such as name and load type.	Signal - General Properties on page 110
To display information on the signal if the EESPort is configured.	Signal - Information Properties on page 110

Related commands

The item provides the following commands:

Purpose	Refer to
To display the ECU pin in the EESPort Configurations controlbar.	Highlight Pin on page 129
To lock the view of the error sets displayed in the working area when triggering errors.	Lock Scrolling on page 136
To create a new error in an error set in the working area.	New Error on page 136
To create a new error set in an error configuration in the working area.	New ErrorSet on page 138
To create a new signal in an error in the working area.	New Signal on page 138
To open the Properties controlbar.	Properties on page 140

Related topics

References

Electrical Error Simulation Properties

Where to go from here

Information in this section

EESPort - Configuration Properties

Purpose

To specify properties that are related to the hardware you are using for electrical error simulation.

Properties for serial communication

These properties are available only if you use an XIL API EESPort implementation from dSPACE.

COM port Lets you specify the COM port when using an RS232 serial interface to control the failure simulation hardware: e.g., COM1.

The COM port must be physically connected to the failure simulation hardware.The COM port must not be used by another client.

Note

Reduced Performance by Using External RS232 Converters

You are strongly recommended to use a physical RS232 port of the host PC to control the failure simulation hardware. If external RS232 ports are missing, try to use an internal RS232 port of the host PC's motherboard.

Software triggers and dynamic errors are not supported if you use an external RS232 converter. Communication via an external RS232 converter is also time-critical and can cause communication errors.

If there is no alternative to using an external RS232 converter:

- Use the IOLAN DS1 from Perle as an Ethernet-to-RS232 converter. For configuring this tool, refer to http://www.dspace.com/go/eth2rs232.
- Otherwise, use an USB-to-RS232 converter with an FTDI chipset and the newest FTDI driver, refer to http://www.ftdichip.com/FTDrivers.htm.

Properties for CAN communication

These properties are available only if you use an XIL API EESPort implementation from dSPACE.

Card index (available for dSPACE CAN API 1.0) Lets you specify the index of the CAN device: e.g., 1. For further information, refer to the documentation of the used CAN interface.

Channel identifier (available for dSPACE CAN API 2.0) Lets you specify the identifier of the channel.

Controller index (available for dSPACE CAN API 1.0) Lets you specify the CAN controller index used for the CAN interface: e.g., 1.

The host PC must be connected to the failure insertion unit via the CAN interface, which must not be used by another client.

dSPACE CAN API Version Lets you select the dSPACE CAN API version. It is recommended to select dSPACE CAN API 2.0.

dSPACE CAN API 2.0 was introduced with dSPACE Release 2016-B. It is the successor of dSPACE CAN API 1.0, includes all previous features, and additionally supports CAN FD.

As of dSPACE Release 2020-A, dSPACE CAN API 1.0 is no longer supported.

Interface name (available for dSPACE CAN API 2.0) Lets you specify the CAN interface that you are using to control the connected failure simulation hardware.

Value	Description
Unknown interface	Unknown interface
DCI-CAN2	dSPACE DCI-CAN2
DCI-CAN/LIN1	dSPACE DCI-CAN/LIN1
Leaf	Kvaser Leaf
Memorator Professional	Kvaser Memorator Professional
USBcan II	Kvaser USBcan II
USBcan Professional	Kvaser USBcan Professional
CANcaseXL	Vector CANcaseXL
VN1610	Vector VN1610
VN1611	Vector VN1611
VN1630	Vector VN1630
VN1640	Vector VN1640
VN5610	Vector VN5610
VN5610A	Vector VN5610A
VN7600	Vector VN7600

Value	Description
VN8900	Vector VN8900
Virtual	Virtual interface

Interface type (available for dSPACE CAN API 1.0) Lets you specify the host PC's CAN interface that you are using to control the connected failure simulation hardware: e.g., dSPACE CAN Interface.

The following CAN interfaces are supported:

- dSPACE CAN Interface
- Vector CANcaseXL
- Vector VN1610
- Vector VN1611
- Vector VN1630
- Vector VN1640
- Vector VN5610
- Vector VN7600
- Vector VN8900
- Unknown Vector interface
- Kvaser USBcan II
- Kvaser Leaf Family
- Kvaser USBcan Professional
- Kvaser Memorator Professional

Serial number Lets you specify the serial number of the CAN interface as integer. It can be used instead of the card index.

Vendor name (available for dSPACE CAN API 2.0) Lets you specify the vendor name of the used CAN interface.

Value	Description
dSPACE	dSPACE GmbH
Eberspaecher	Eberspächer GmbH
Kvaser	Kvaser
Vector Informatik	Vector Informatik GmbH
Unknown vendor	Unknown vendor

Properties for a discrete FIU

The following properties are available only if you use an XIL API EESPort implementation from dSPACE and a discrete failure insertion unit (FIU) in your simulator.

ECU version Lets you specify the version of an ECU, if multiple ECU versions are specified in the simulator's signal list file: e.g., 1.

Signal list file Lets you specify the related signal list (CSV) file via an absolute path or a relative path to the related port configuration file. (In the port

configuration file, this property is represented by the `SignalListPath` attribute of the `HardwareConfiguration` element.)

For details on signal list files, refer to Defining Failure Classes with Signal Files (dSPACE XIL API Implementation Guide 🕮).

Properties for an integrated SCALEXIO FIU

The following properties are available only if you use an integrated SCALEXIO failure insertion unit (FIU).

Experiment platform name Lets you select the name of a SCALEXIO platform that is used by the ControlDesk experiment.

Override access Lets you automatically disconnect another client that might be connected to the SCALEXIO failure simulation hardware when you configure the XIL API EESPort with ControlDesk. The disconnected client will fall into an error status. It is recommended to disable this property in normal operation.

Real-time application file path Lets you specify the path to the real-time application (RTA) file which is related to the variable description (SDF) file. (In the port configuration file, this property is represented by the `SignalListPath` attribute of the `HardwareConfiguration` element.)

This property is set automatically if the platform that is selected via the Experiment platform name property provides the related information.

Real-time platform name Lets you specify the name of the SCALEXIO processing unit that controls the electrical error simulation. (In the port configuration file, this property is represented by the value of the `PlatformName` element.)

This property is set automatically if the platform that is selected via the Experiment platform name property provides the related information.

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)
Creating dSPACE EESPort Configuration Files (dSPACE XIL API Implementation Guide 🕮)
Using the EESPortConfiguration API (dSPACE XIL API Implementation Guide 🕮)

References

dSPACE EESPort Configuration File (dSPACE XIL API Reference 🕮)
Insert EESPort..... 131

EESPort - General Properties

Purpose

To display the general properties of the selected XIL API EESPort.

Properties

Logging (available if the EESPort is not configured yet) For support only, lets you enable logging. Logging essentially increases the execution time. Do not enable logging in normal operating situations. Enable logging only if you want to contact dSPACE Support.

Name Displays the name of the EESPort.

Offline Mode (available if the EESPort is not configured yet) For tests on the host PC without a physical connection to a failure simulation hardware, lets you simulate the activation and triggering of errors without connecting the EESPort to the failure simulation hardware. If you select this property, the use of the tracing variables to monitor the switching behavior and transition states of the failure simulation hardware is disabled automatically. If you are working with software triggers, trigger conditions cannot be checked in offline mode. Error sets that use trigger conditions are therefore activated immediately without checking the specified condition.

Port configuration file Displays the name of the port configuration file. To interface the failure simulation hardware, an EESPort needs the hardware-dependent *port configuration file* (PORTCONFIG file). The file's contents must fit the connected HIL simulator architecture and its failure simulation hardware.

State Displays the state of the EESPort: e.g., Connected.

There are the following states:

State	Description
Disconnected	In its initial state, the EESPort is disconnected from the failure simulation hardware. You can configure the EESPort. Electrical error simulation is not possible.
Connected	The EESPort is connected with the configured failure simulation hardware. You can add, remove, and configure error configurations.
Downloaded	An error configuration is downloaded. You can start the error configuration or add error sets to it. You cannot edit or remove the downloaded error sets.
Activated	The downloaded error configuration is started. The failure simulation hardware waits for the first trigger event to execute the first error set of the error configuration. You can trigger the error sets of the error configuration in the specified order. You can also add new error sets to the error configuration.

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)

References

Insert EESPort 131

EESPort – Potentials Properties

Purpose

To specify a mapping of the potentials provided by the simulator to unique identifiers you need according to the ASAM AE XIL API standard. These properties are available only if you use an XIL API EESPort implementation from dSPACE.

Properties

ID Displays the ID of the potential. The potential ID is assigned automatically (in sequential numbering): e.g., `0`.

Name Lets you specify the name for a potential provided by your dSPACE simulator.

Potentials To map a potential name and a potential type to a unique identifier (e.g., a natural number, starting with 0). The identification of potentials with unique identifiers is required by the ASAM AE XIL API standard. The list order of the potentials in the mapping corresponds to the unique identifiers assigned for the single potentials. For systems with discrete FIU, you find the potential names and potential types in the simulator's signal file. For SCALEXIO systems with the integrated SCALEXIO FIU, you define potentials with ConfigurationDesk as power switches.

Type Lets you select the related potential type for each potential. According to the ASAM AE XIL API standard, there are the following *potential types*: **Gnd** (for ground), **Ubat** (for $U_{Battery}$), and **Potential** (for a potential between ground and $U_{Battery}$).

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)

Basics on Potential Mapping 31

Creating dSPACE EESPort Configuration Files (dSPACE XIL API Implementation Guide 🕮)

HowTos

How to Perform Potential Mapping 41

References

dSPACE EESPort Configuration File (dSPACE XIL API Reference 🕮)

Insert EESPort 131

EESPort – Real-Time Configuration Properties

Purpose

To specify properties for monitoring the switching behavior of the failure simulation hardware in connection with a related real-time application and HIL simulator.

Properties

These properties are available only if you use an XIL API EESPort implementation from dSPACE.

Enabled Lets you:

- Monitor the switching behavior of the failure simulation hardware via specific measurement variables. Refer to Tracing properties on page 95.
- Enable the use of software triggers. Refer to Software trigger properties on page 96.

If you select **Offline mode** in the **General** properties of the EESPort, checking trigger conditions and monitoring the switching behavior and transition states of the failure simulation hardware is not possible. Error sets that are using software triggers with a condition are activated immediately without checking the specified trigger condition.

Experiment platform name Lets you select the name of a platform that is used by the ControlDesk experiment. This platform must represent the HIL simulator which executes the real-time application that provides the specific measurement variables for monitoring the switching behavior of the failure simulation hardware.

Based on information derived from the selected platform in the experiment (e.g., from its related variable description file), ControlDesk automatically specifies the **Real-time platform name** and **Variable description file path** properties and the names of the **Tracing** variables

Note

The Experiment platform name is not stored in the port configuration (PORTCONFIG) file of the EESPort. It is only a convenience function in ControlDesk for specifying the Real-time platform name and Variable description file path properties and variable names in one step. You can also specify these properties and variable names manually without selecting an Experiment platform name.

Real-time platform name Lets you specify the platform name of the real-time platform which executes the real-time application that provides the specific measurement variables for monitoring the switching behavior. This platform name must be the same as the platform name in the ControlDesk Platforms/Devices controlbar. (In the port configuration file, this property is represented by the `PlatformName` attribute of the `RealTimeConfiguration` element.)

This property is set automatically if the platform that is selected via the Experiment platform name property provides the related information.

Note

- For SCALEXIO single-processor systems, the Real-time platform name is the name of the SCALEXIO processing unit.
- For SCALEXIO multi-processor systems, the Real-time platform name is the name of the superordinate SCALEXIO cluster.

Variable description file path Lets you specify the path of the variable description (SDF) file that contains the specific measurement variables for monitoring the switching behavior of the failure simulation hardware. (In the related port configuration file, this property is represented by the `SystemDescriptionFilePath` attribute of the `RealTimeConfiguration` element.)

This property is set automatically if the platform that is selected via the Experiment platform name property provides the related information.

Tracing properties

To monitor the switching behavior and transition states of the failure simulation hardware via specific measurement variables. These variables are independent from other model variables. They do not measure the real behavior of the failure simulation hardware, they only estimate it according to empirical values. The variables are written and read only by the XIL API EESPort.

These properties are available only if you use an XIL API EESPort implementation from dSPACE.

Active error set variable This variable displays the number of the error set that is currently activated. If no error set is activated, the value is '0'. You can enter the name of another measurement variable to replace the default variable. Select only an independent variable that cannot be overwritten by the real-time

model you are simulating. Otherwise, this variable might not monitor the switching behavior of the failure simulation hardware correctly.

Error activated variable This variable indicates whether one or more errors are activated (= '1') or not (= '0'). You can enter the name of another measurement variable to replace the default variable. Select only an independent variable that cannot be overwritten by the real-time model you are simulating. Otherwise, this variable might not monitor the switching behavior of the failure simulation hardware correctly.

Error switching variable This variable indicates the transition state when you switch the failure simulation hardware. '0' = No switching is in process. (The failure simulation hardware is in the steady state.) '1' = Switching is in process. (The failure simulation hardware is in the transition state.) The error is switched physically during this transition state.You can enter the name of another measurement variable to replace the default variable. Select only an independent variable that cannot be overwritten by the real-time model you are simulating. Otherwise, this variable might not monitor the switching behavior of the failure simulation hardware correctly.

Flags variable This variable is used internally. Do not use this variable to monitor the switching behavior of the failure simulation hardware.

Trigger variable This variable is used internally. Do not use this variable to monitor the switching behavior of the failure simulation hardware.

Software trigger properties

Software triggers let you activate error sets in response to a defined trigger condition or duration. You can specify the trigger type for each error set individually.

These properties are available only if you use an XIL API EESPort implementation from dSPACE.

Polling interval Lets you specify the sampling rate for polling the trigger variable of a trigger condition.

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)
Basics on Monitoring the Switching Behavior of the Failure Simulation Hardware..... 66
Using the EESPortConfiguration API (dSPACE XIL API Implementation Guide 🕮)

HowTos

How to Configure to Monitor the Switching Behavior of the Failure Simulation Hardware..... 71

References

Insert EESPort..... 131

EESPort – Signals Properties

Purpose

To perform an optional mapping of ECU pins that you are using in your dSPACE HIL simulator to abstract signal names according to the ASAM AE XIL API standard.

Properties

These properties are available only if you use an XIL API EESPort implementation from dSPACE.

ECU name Lets you specify the related ECU name for the pin.

Pin name Lets you specify the pin name.

Signal name Lets you specify an abstract name (user-defined string) for the signal according to the ASAM AE XIL API standard.

Signals To perform an optional mapping of ECU pins that you are using in your dSPACE HIL simulator to abstract signal names according to the ASAM AE XIL API standard. Each ECU pin is specified by its **Pin name** and the related **ECU name**.

The ECUs and ECU pins of your HIL system are specified in a signal file (for systems with discrete FIU) or via ConfigurationDesk (for SCALEXIO systems with the integrated SCALEXIO FIU).

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)

References

dSPACE EESPort Configuration File (dSPACE XIL API Reference 🕮)
Insert EESPort 131

EESPort – Vendor Information Properties

Purpose

To display information on the electrical error simulation port (XIL API EESPort) implementation that is installed on the host PC.

Properties

Product name Displays the product name of the XIL API EESPort implementation.

Product version Displays the product version of the XIL API EESPort implementation.

Vendor name Displays the vendor of the XIL API EESPort implementation.

XIL API version Displays the XIL API version of the EESPort implementation.

Related topics

Basics

Basic Information on the EESPort Implementation (dSPACE XIL API Implementation Guide 🕮)

References

Insert EESPort 131

Error - Custom Properties

Purpose

To specify custom properties for an error.

Properies

Custom properties Custom properties are user-defined (and vendor-specific) properties that consist of a key word and a related value. You can add one or more custom properties to an error configuration, an error set, or an error. All these custom properties are stored in the XML file of the related error configuration.

Click  to add a new custom property.

Click ❌ to remove a custom property.

Key Lets you specify a key word for a custom property.

Value Lets you specify a value for the related key word.

Related topics

References

Error - General Properties 99
Error - Parameters Properties 100
Error Configuration - Custom Properties 102
Error Set - Custom Properties 104

Error - General Properties

Purpose

To configure the general properties of an electrical error, such as the error category and the error type.

Properties

Error category To specify the error category of the error: e.g., Short to Ground. The error category defines how a signal is disturbed. Which errors you can create for a signal depends on the connected failure simulation hardware.

The following table gives you a short overview on the basic error categories mentioned in the ASAM standard and the related terminology used by dSPACE for the failure classes. For more detailed information, refer to Failure Classes (dSPACE XIL API Reference 🕮).

Error Category (ASAM)	Failure Class (dSPACE)
ErrorPin2Pin	Short circuit to another ECU pin
InterruptError	Cable break[1)]
ErrorToGround	Short circuit to GND
ErrorToUbatt	Short circuit to Ubat
ErrorToPotential	Short circuit to Potential
InterruptAtPosition	Cable break[2)]
InterchangedPins	Not supported

1) Open circuit
2) Open circuit on the high or the low bus line. Only supported by the DS1450 Bus FIU Board.

Error type To specify the error type of the error: e.g., Loose Contact. The error type specifies the way an error category – i.e., an interruption or short circuit of signals – is provided. The error type defines the disturbance itself.

Error Type (ASAM)	Description
Simple	The error is set statically.
Dynamic	The error is set dynamically. This means that the error is set for a specified duration.
Resistor	The error provides an additional resistor to be switched. The resistor is specified in Ω.
Dynamic Resistor	The error is a combination of a dynamic error with a switchable resistor.
Loose Contact	The error is set dynamically by a PWM signal specified by frequency and duty cycle. This error type is also known as *pulsed switching*.
Loose Contact Resistor	The error is a combination of a loose contact error with a switchable resistor.

Related topics

Basics

Basic Information on Configuring Errors (dSPACE XIL API Implementation Guide 🕮)

Examples

Examples for Configured Errors..... 60

References

EESPort Configurations..... 120
Error..... 80
Error - Custom Properties..... 98
Error - Parameters Properties..... 100
New Error..... 136

Error - Parameters Properties

Purpose

To specify specific error parameters.

Properties

Duration Lets you specify a duration of the disturbance (error signal) in s.

Duty cycle Lets you specify the duty cycle of the disturbance (error signal) in %. Observe the technical limitations for DS793 modules and the integrated SCALEXIO FIU.

Limitations for DS793 modules:

- The duty cycle must be in the range 0.025 … 99.975 %.
- The frequency must be in the range 6.105006 … 25,000.0 Hz.

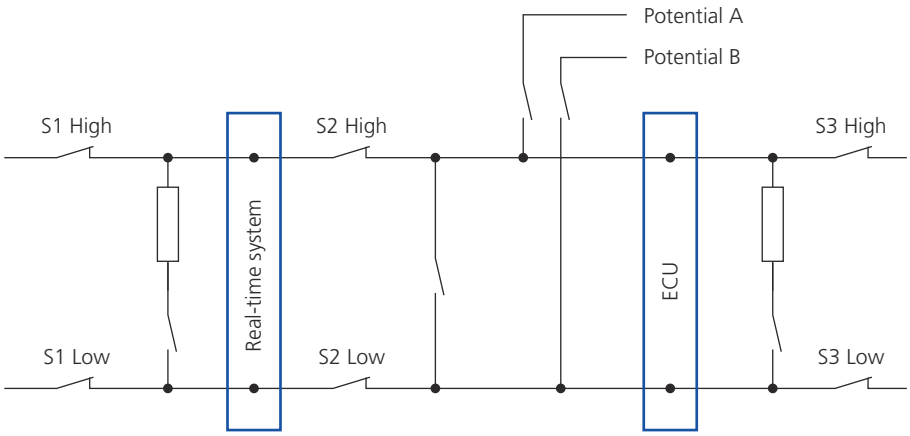
Limitations for the integrated SCALEXIO FIU:

- The duty cycle must be in the range 0.0000344827 … 99.9999655173 %.
- The frequency must be in the range 0.001724 … 5000.0 Hz.

Frequency Lets you specify the frequency of the disturbance (error signal) in Hz.

Interrupt position Lets you specify the position for the interrupt of a bus line if you are using a DS1450 Bus FIU Board. You can enter S1, S2, or S3.

The following illustration shows the interrupt positions **S1**, **S2**, and **S3** in the bus lines.



ControlDesk automatically detects if a CAN High or CAN Low signal line is used to switch the related interrupt relay (Sx High or Sx Low).

Potential ID Lets you select a potential to simulate a short circuit to. According to the XIL API definition, each potential is specified by an index number. The index starts with 0.

Potential name Displays the name for the selected potential as specified in the EESPort's **Potentials** properties. Potential names are displayed only if the EESPort is in the **Connected** state.

For further details, refer to EESPort – Potentials Properties on page 93.

Resistor Lets you specify the resistance in Ω if you are using a DS293 FIU Module or DS1450 Bus FIU Board.

For technical details, refer to DS293 FIU Module, DS282 Load Module and DS289MK RSim Module (dSPACE XIL API Implementation Guide 🕮) and DS1450 Data Sheet (PHS Bus System Hardware Reference 🕮).

Related topics

Examples

Examples for Configured Errors....60

References

EESPort – Potentials Properties....93
Error - Custom Properties....98
Error - General Properties....99
Insert EESPort....131

Error Configuration - Custom Properties

Purpose

To specify custom properties for an error configuration.

Properies

Custom properties Custom properties are user-defined (and vendor-specific) properties that consist of a key word and a related value. You can add one or more custom properties to an error configuration, an error set, or an error. All these custom properties are stored in the XML file of the related error configuration.

Click  to add a new custom property.

Click  to remove a custom property.

Key Lets you specify a key word for a custom property.

Value Lets you specify a value for the related key word.

Related topics

References

Error Configuration - General Properties

Purpose

To configure the general properties of the error configuration.

Properties

Active error set Displays the name of the error set that is currently active. This property entry is empty if there is no active error set.

File name Displays the name of the error configuration's XML file in the Windows file system.

File path Displays the file path of the error configuration's XML file in the Windows file system.

Name Lets you specify the name of the error configuration.

As specified by the ASAM AE XIL API standard, the name is only stored in the error configuration's XML file. The name is not used in the GUI except for this property entry.

The name of the error configuration can be unequal with the name of the error configuration's XML file.

Related topics

References

Error Configuration - Variable Mapping Properties

Purpose

To display and specify aliases for the condition triggers of the error sets of the error configuration.

Properties

Defines This list displays all the aliases for model parameters or variables of the simulation application used to specify condition triggers for the single error sets of the error configuration. You can use the list to change the mapping of the model parameters or variables to the aliases. The list does not display the related error set in which an alias is used. To have a clear overview, it is recommended to use only unique symbol names in an error configuration.

Symbol name Displays the alias that specifies a condition trigger for one or more error sets. Although symbol names can be edited and used independently for each error set, it is recommended to use only unique symbol names in an error configuration to have a clear overview.

Variable path Lets you map a measurement variable or parameter to the alias, which is identified by its **Symbol name**. A mapped measurement variable or parameter is displayed by its variable path. You can map a measurement variable or parameter to a **Symbol name** by editing the variable path in the related edit field or via the Browse button and the **Select Variable Dialog**. This edit field is empty if no mapping is performed or if the **Symbol name** is used in different error sets with different mappings.

Related topics

References

Error Set - Custom Properties

Purpose

To specify custom properties for an error set.

Properies

Custom properties Custom properties are user-defined (and vendor-specific) properties that consist of a key word and a related value. You can add one or more custom properties to an error configuration, an error set, or an error. All these custom properties are stored in the XML file of the related error configuration.

Click  to add a new custom property.

Click  to remove a custom property.

Key Lets you specify a key word for a custom property.

Value Lets you specify a value for the related key word.

Related topics

References

Error Set - General Properties

Purpose

To specify the general properties of an error set, such as the name and trigger type.

Properties

Name Lets you specify the name of the error set. The name must be unique in the related error configuration.

Trigger Type Lets you specify the trigger type to be used for starting the error set.

Note

dSPACE XIL API .NET EESPort implmentation supports only manual and software triggers.

Related topics

References

Error Set - Software Trigger Properties

Purpose

To specify software trigger properties that activate the error set automatically.

Properties

Duration Lets you specify a duration after which the error set is activated automatically.

Condition Lets you specify a trigger condition to activate the error set automatically if the trigger condition is fulfilled. Specify the trigger condition according to the ASAM General Expression Syntax (GES). Click the Browse button to specify this condition by means of the **Expression Editor**. Refer to Expression Editor on page 106.

Timeout Lets you specify a timeout in seconds to deactivate the error configuration automatically if the specified trigger condition cannot be detected.

Type Lets you specify the type of the software trigger: **Duration** or **Condition**.

Related topics

HowTos

References

Error Set - Variable Mapping Properties

Purpose

To specify aliases to be used for the trigger condition of the error set.

Properties

Defines This list lets you specify aliases for specific model parameters or variables of the simulation application. You can use the aliases to specify a condition trigger for the error sets.

Symbol name Displays the alias that specifies the trigger condition for the error set. You must map a measurement variable or parameter to this alias. Although not required, it is recommended to use only unique symbol names in an error configuration to have a clear overview.

Variable path Lets you map a measurement variable or parameter to the alias, which is identified by its **Symbol name**. A mapped measurement variable or parameter is displayed by its variable path. You can map a measurement variable or parameter to a **Symbol name** by editing the variable path in the related edit field or via the Browse button and the **Select Variable Dialog**.

Related topics

HowTos

How to Configure a Software Trigger 54

References

Error Configuration - Variable Mapping Properties 103
Error Set - General Properties 104
Error Set - Software Trigger Properties 105
Select Variable Dialog (ControlDesk Variable Management 🕮)

Expression Editor

Access

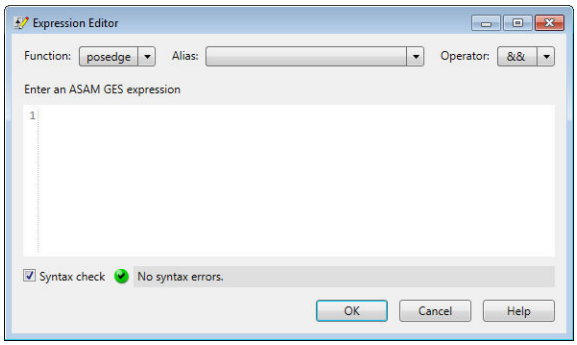
You can access this editor via the Browse button of the **Software trigger – Condition** property of an error set.

Purpose

To specify a trigger condition in the ASAM General Expression Syntax (GES) as a string.

Description

You can use the **Expression Editor** to specify a trigger condition that complies with the ASAM General Expression Syntax (GES).



Expressions that comply with this GES syntax let you specify conditions that relate not only to one value at a certain time, but also to the change of values over time (e.g., in signals).

In such an expression, you can use common logical operations (e.g., **and**) and mathematical functions (e.g., **sin**). Furthermore, the GES provides functions, such as **posedge**, that consider values and their predecessors for validation. You can nest ASAM GES expressions to specify more complex conditions.

To edit the condition string, you can use the **Expression Editor** in two different ways:

- You can specify the expression by using the editor buttons **Function**, **Parameter**, and **Operator** to select its functions, parameters, and operators.
- You can manually enter the condition as an ASAM GES expression.

By default, the syntax of the condition is continuously checked, but you can disable syntax checking to use aliases that you want to define later.

Tip

For more information on the *General Expression Syntax*, refer to *Appendix A* of the ASAM documentation ASAM_AE_XIL_Generic-Simulator-Interface_BS-1-4-Programmers-Guide_V2-1-0.pdf.

Note

If you check two floating-point values for equality, using the `==` operator might lead to unexpected results. For example, `(3 * 0.1 == 0.3)` returns *False*.

To take into account precision effects of the binary representation of float values, it is useful to allow tolerances. For example, to check the equality of a and b, use:

```
abs(b-a) < 1e-14
```

Settings of the editor

The Expression Editor lets you specify the following settings:

Alias Lets you select an alias from the mapping table or a definition from the Defines data object that is available for the condition. The selected item is inserted into the condition at the text cursor position.

ASAM GES expression Lets you edit a condition manually. This means you can manually add parentheses and aliases that you want to define later.

Function Lets you select an ASAM XIL API function that is inserted into the condition at the text cursor position.

Operator	Syntax	Meaning
posedge	`posedge(Expr1, Expr2)`[1)]	Returns true when the value of `Expr1` changes from a value lower than `Expr2` to a value greater than or equal to `Expr2` (positive edge detection).
negedge	`negedge(Expr1, Expr2)`[1)]	Returns true when the value of `Expr1` changes from a value greater than `Expr2` to a value lower than or equal to `Expr2` (negative edge detection).
changed	`changed(Expr1, Expr2)`[1)]	Returns true when the difference between the current value of `Expr1` and its direct successor is greater or equal to `Expr2` (value change detection).
changedpos	`changedpos(Expr1, Expr2)`[1)]	Returns true when the current value of `Expr1` is greater than its predecessor at least by `Expr2` (positive value change detection).
changedneg	`changedneg(Expr1, Expr2)`[1)]	Returns true when the current value of `Expr1` is smaller than its predecessor at least by `Expr2` (negative value change detection).
sin	`sin(Expr)`[2)]	Returns the sine of `Expr`.
cos	`cos(Expr)`[2)]	Returns the cosine of `Expr`.
abs	`abs(Expr)`	Returns the absolute value of `Expr`.
pow	`pow(Expr1, Expr2)`	Returns the value of `Expr1` to the power of `Expr2`.
min	`min(Expr1, Expr2)`	Returns the value of the smaller expression.
max	`max(Expr1, Expr2)`	Returns the value of the greater expression.

1) Where `Expr1` defines a signal and `Expr2` defines a number.
2) Where Expr is specified in radian.

Operator Lets you select an operator to insert into the condition at the text cursor position.

Operator	Meaning
&&	Logical *and* of the left and right operands.
||	Logical *or* of the left and right operands.
^^	Logical *exclusive or* of the left and right operands.
!	Logical *not* of the right operand.
&>	Logical *and then* of the left and right operands.
<	The left operand is smaller than the right operand.
<=	The left operand is smaller than or equal to the right operand.

Operator	Meaning
>	The left operand is greater than the right operand.
>=	The left operand is greater than or equal to the right operand.
==	The left operand is equal to the right operand.
!=	The left operand is not equal to the right operand.
+	The sign of the right operand or the sum of the right and left operands.
-	The sign of the right operand or the difference of the right and left operands.
*	The product of the right and left operands.
/	The ratio of the right and left operands.
**	The left operand to the power of the right operand.

Syntax check Lets you enable or disable continuous syntax checking of the ASAM GES expression. By default, syntax checking is selected.

If syntax checking is selected, its result is displayed in a status icon and a status message. The following status icons are provided:

Status Icon	Meaning
	The condition is syntactically correct.
	The condition contains syntax errors. In the ASAM GES expression, the error is highlighted. The OK button is disabled.
	The condition contains unknown aliases. The names of the invalid aliases are displayed in the status message.

If enabled, the syntax is checked continuously. Therefore, the check status frequently indicates an error while you enter the expression.

Features of the editor

The Expression Editor provides the following features to make reading and writing conditions more convenient:

Syntax highlighting Keywords, identifiers, literals, and comments are displayed in different colors to increase the readability of the code.

Bracket matching When you position the text cursor after a bracket, the text cursor and the matching bracket are highlighted.

Name completion When you enter the first characters of a name or command, you can press Ctrl+Spacebar to automatically complete the entry. If the entered name segment is not unique, all the possible matches are displayed in a list. Only elements that were already executed are available.

Related topics

HowTos

How to Configure a Software Trigger 54

References

Error Set - Software Trigger Properties 105

Signal - General Properties

Purpose

To specify the general properties of a signal, such as name and load type.

Properties

Load type Lets you specify load rejection during error simulation:

- Select **With load** to leave the load connected during error simulation.
- Select **Without load** to disconnect the load during error simulation.

Signal name Lets you specify the signal name.

For dSPACE hardware, the signal name is either a user-defined string you used in the optional signal mapping definition of the port configuration or the composition of the related ECU and pin names, e.g., `ECU 2\Pin 5`.

Related topics

References

EESPort Configurations 120
Signal 85
Signal - Information Properties 110

Signal - Information Properties

Purpose

To display information on the signal if the EESPort is configured.

Properties

If the EESPort is configured, the displayed information is derived from the signal file of a non-SCALEXIO system or the real-time application (RTA) file of a SCALEXIO system.

Allowed error categories Displays a list of the allowed error categories and the related load types of the ECU pin.

Allowed error category Displays an allowed error category of the ECU pin.

Allowed error types Displays a list of the allowed error types of the ECU pin.

Allowed error type Displays an allowed error type of the ECU pin.

ECU name Displays the ECU name.

Load type Displays one possible load type for the related error category.

Name Displays the signal name.

Pin name Displays the pin name.

Related topics

References

Signal....85
Signal - General Properties....110

Electrical Error Simulation Commands

Where to go from here

Information in this section

Activate (Error Configuration)

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	▪ Error configuration in the **Project** controlbar ▪ Error configuration page in the working area
Shortcut key	None
Icon	

Purpose

To activate an error configuration.

Related topics

References

Deactivate (Error Configuration)..... 118
Trigger (Error Configuration)..... 148

Best Fit

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

To optimize the width of the selected column.

Related topics

References

Best Fit (All Columns)..... 115
EESPort Configurations..... 120

Best Fit (All Columns)

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	None

Purpose

To optimize the widths of all the displayed columns.

Related topics

References

Best Fit 115
EESPort Configurations 120

Clear Filter

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the EESPort Configurations controlbar
Shortcut key	None
Icon	

Purpose

To clear a filter that is set for a column.

Related topics

References

EESPort Configurations 120

Clear Sorting

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the EESPort Configurations controlbar
Shortcut key	None
Icon	

Purpose

To deactivate the sort attribute applied to the selected column.

Related topics

References

EESPort Configurations ... 120
Sort Ascending ... 147
Sort Descending ... 147

Close (EESPort)

Access

You can access this command via:

Ribbon	None
Context menu of	EESPort in the Project controlbar
Shortcut key	None
Icon	None

Purpose

To close an XIL API EESPort.

Related topics

References

EESPort ... 78
Open (EESPort) ... 139

Close (Error Configuration)

Access

You can access this command via:

Ribbon	None
Context menu of	Error configuration in the Project controlbar
Shortcut key	None
Icon	None

Purpose

To close an error configuration that is opened in ControlDesk's working area.

Related topics

References

Error Configuration....81
Open (Error Configuration)....139

Configure (EESPort)

Access

You can access this command via:

Ribbon	XIL API EESPort – EESPorts
Context menu of	EESPort in the Project controlbar
Shortcut key	None
Icon	

Purpose

To configure the XIL API EESPort with vendor-specific information, i.e., the properties you specified via the port configuration file.

Related topics

References

Disconnect (EESPort)....119
EESPort....78

Deactivate (Error Configuration)

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	▪ Error configuration in the Project controlbar ▪ Error configuration page in the working area
Shortcut key	None
Icon	

Purpose

To deactivate and stop the execution of an error configuration on the failure simulation hardware.

Related topics

References

Activate (Error Configuration)..... 114
Trigger (Error Configuration)..... 148

Disconnect (EESPort)

Access

You can access this command via:

Ribbon	XIL API EESPort – EESPorts
Context menu of	EESPort in the Project controlbar
Shortcut key	None
Icon	

Purpose

To disconnect the XIL API EESPort from the related failure simulation hardware.

Description

The interface to the failure simulation hardware can now be used by another XIL API EESPort client.

Related topics

References

Configure (EESPort)..... 118
EESPort..... 78

Download (Error Configuration)

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	▪ Error configuration in the Project controlbar ▪ Error configuration page in the working area
Shortcut key	None
Icon	

Purpose

To download an error configuration to the failure simulation hardware that is connected to the host PC.

Related topics

References

Unload (Error Configuration)..... 149

EESPort Configurations

Access

You can access this command via:

Ribbon	View – Controlbar – Switch Controlbars
Context menu of	None
Shortcut key	None
Icon	

Purpose

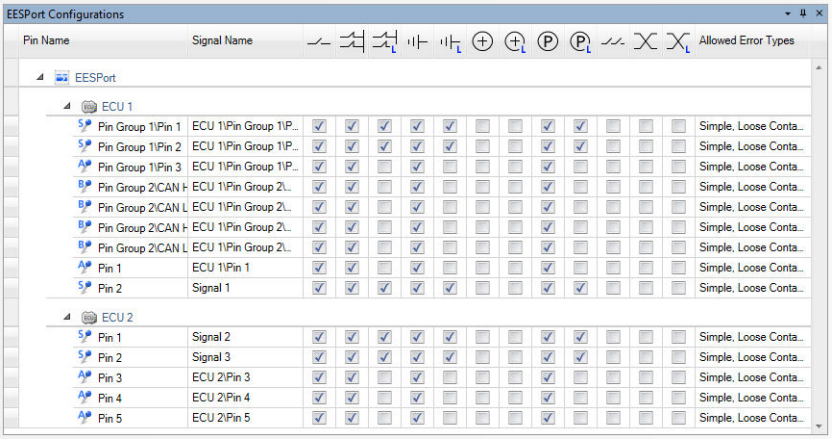
To show the EESPort Configurations controlbar.

Result

The EESPort Configurations controlbar opens.

EESPort Configurations controlbar

The controlbar displays wiring information on the related hardware-in-the-loop (HIL) simulator and specifications for the electrical error simulation: the ECU signals (i.e., their names), their wiring (i.e., the connected ECU pins) and the allowed error categories, error types, and load types for each ECU pin (see the following example).



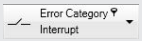
If the EESPort is configured, the displayed information is derived from the signal file of a non-SCALEXIO system or the real-time application (RTA) file of a SCALEXIO system.

You can use the controlbar to map ECU pins to errors in the working area via drag & drop.

The controlbar provides a wide range of commands for arranging and filtering the displayed entries, for example, you can group rows or specify comprehensive filter rules.

Tip

- If you enlarge the column of an error category, the error category is also displayed in text. Refer to the following example.

- The Reset Settings command lets you return to the original configuration of the controlbar.

Icons for connected hardware

The EESPort Configurations controlbar has the following icons for connected hardware:

Icon	Description
	EESPort
	ECU
	ECU pin (without specific information about the connected channel)
	ECU pin of a connected signal measurement channel (for actuator simulation)
	ECU pin of a connected bus channel (for bus simulation)
	ECU pin connected to a high current failure simulation hardware
	ECU pin of a connected signal generation channel (for sensor simulation)

Icons for error categories and load types

The EESPort Configurations controlbar has the following icons for error categories and load types:

Icon	Error Category	Load Type
	Interrupt	–
	Pin to pin	Without load
	Pin to pin	With load

Icon	Error Category	Load Type
	Short to ground	Without load
	Short to ground	With load
	Short to UBat	Without load
	Short to UBat	With load
	Short to potential	Without load
	Short to potential	With load
	Interrupt at position	–
	Interchanged pins[1)]	Without load
	Interchanged pins[1)]	With load

[1)] Not supported by dSPACE failure simulation hardware.

Related commands

The EESPort Configurations controlbar provides the following commands:

Purpose	Refer to
To optimize the width of the selected column.	Best Fit on page 115
To optimize the widths of all the displayed columns.	Best Fit (All Columns) on page 115
To clear a filter that is set for a column.	Clear Filter on page 116
To deactivate the sort attribute applied to the selected column.	Clear Sorting on page 116
To open the Filter Editor to edit a filter rule.	Filter Editor on page 125
To group the signals according to the entries of the column.	Group by This Column on page 127
To open the Group Summaries editor.	Group Summary Editor on page 127
To hide the group panel.	Hide Group Panel on page 128
To close the search panel.	Hide Search Panel on page 129
To reset the current configuration of the controlbar.	Reset Settings on page 143
To open a dialog for specifying which columns to display.	Show Column Chooser on page 144
To show the group panel.	Show Group Panel on page 145
To show the search panel.	Show Search Panel on page 146
To sort the rows alphabetically in ascending order by the selected column.	Sort Ascending on page 147
To sort the rows alphabetically in descending order by the selected column.	Sort Descending on page 147
To remove a column header from the group panel to revoke the related grouping of rows.	Ungroup on page 149

Related topics

References

Error - General Properties....99
Signal - General Properties....110

Error Category

Access

You can access this command via:

Ribbon	None
Context menu of	Error in an error set
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To specify the error category of an error.

Description

The error category defines how a signal is disturbed. Which errors you can create for a signal depends on the connected failure simulation hardware.

The following table gives you a short overview on the basic error categories mentioned in the ASAM standard and the related terminology used by dSPACE for the failure classes. For more detailed information, refer to Failure Classes (dSPACE XIL API Reference 🕮).

Error Category (ASAM)	Failure Class (dSPACE)
ErrorPin2Pin	Short circuit to another ECU pin
InterruptError	Cable break[1)]
ErrorToGround	Short circuit to GND
ErrorToUbatt	Short circuit to Ubat
ErrorToPotential	Short circuit to Potential
InterruptAtPosition	Cable break[2)]
InterchangedPins	Not supported

1) Open circuit
2) Open circuit on the high or the low bus line. Only supported by the DS1450 Bus FIU Board.

Tip

You can also specify the error category in the **Properties** controlbar. Refer to Error - General Properties on page 99.

Related topics

References

Error 80

Export (Error Configuration)

Access

You can access this command via:

Ribbon	None
Context menu of	Error configuration in the **Project** controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To export an error configuration as an XML file.

Result

Opens a standard **Save As** dialog to export the error configuration as an XML file to the Windows file system.

Export PortConfiguration

Access

You can access this command via:

Ribbon	None
Context menu of	EESPort in the **Project** controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To export a port configuration as a PORTCONFIG file.

Related topics

References

EESPort 78

Filter Editor

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

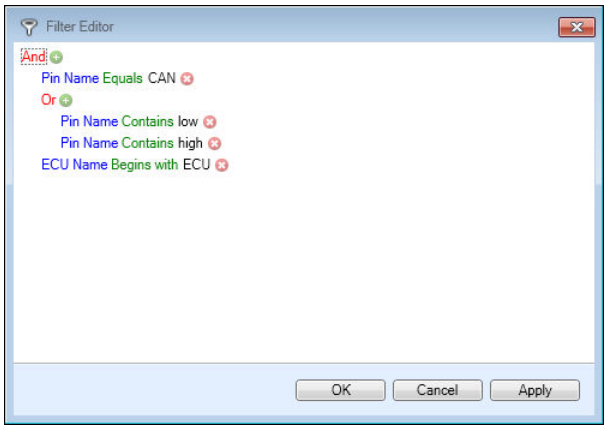
To open the Filter Editor to edit a filter rule.

Result

The **Filter Editor** is opened to create and apply a filter rule.

Filter Editor

The Filter Editor provides a tree view for specifying a hierarchical filter rule, for example, by using Boolean operators.



A filter can consist of several conditions that can be combined by a logical operation. In a condition, a column header is compared with a specified value. To specify a filter, the colored words can be modified.

Red word Click the red word to select a logical operator or a command for combining conditions. The following logical operators are available:

Logical Operation	Description
And	All combined conditions must be true.
Or	At least one of the combined condition must be true.
NotAnd	Not all of the combined conditions are true.
NotOr	Neither of the combined conditions is true.

The following commands are available:

Command	Description
Add Condition, , +, or `Insert`	Lets you add a condition.
Add Group	Lets you add a condition group.
Clear All	Lets you clear all conditions.
	Lets you clear a condition.
Remove Group	Lets you remove a group.

Blue word Click the blue word to select a column of the table whose value is then compared with a specified value.

Green word Click the green word to select a relational operator.

You can apply a filter rule by pressing the OK or Apply button. Pressing the OK button closes the editor.

Related topics

References

EESPort Configurations 120
Reset Settings 143

Group by This Column

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

To group the signals according to the entries of the column.

Description

You can create nested groups. If you selected a column for grouping, you can unselect it via **Ungroup** in the context menu of its column header.

Related topics

References

EESPort Configurations 120
Group Summary Editor 127
Hide Group Panel 128
Show Group Panel 145
Ungroup 149

Group Summary Editor

Access

You can access this command only via the context menu of column headers that are moved to the group panel of the **EESPort Configurations** controlbar.

Note

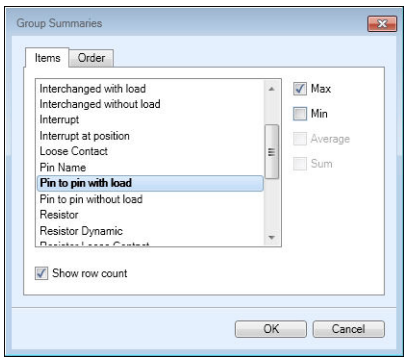
This command is not available in operator mode.

Purpose

To open the Group Summaries editor.

Group summary editor

Lets you display summary information for selected items in each group row.



Max To display the maximum value. (Only useful for alphanumeric information.)

Min To display the minimum value. (Only useful for alphanumeric information.)

Show row count To display the number of rows per group.

Related topics

References

EESPort Configurations..... 120
Group by This Column..... 127
Hide Group Panel..... 128
Show Group Panel..... 145
Ungroup..... 149

Hide Group Panel

Access

This command is available only if the group panel is displayed. You can access this command via:

Ribbon	None
Context menu of	Column header in the EESPort Configurations controlbar
Shortcut key	None
Icon	None

Purpose

To hide the group panel.

Related topics

References

EESPort Configurations 120
Group by This Column 127
Show Group Panel 145
Ungroup 149

Hide Search Panel

Access

This command is available only if the search panel is shown. You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

To close the search panel.

Related topics

References

EESPort Configurations 120
Show Search Panel 146

Highlight Pin

Highlight Pin

You can access this command via:

Ribbon	None
Context menu of	Signal in an error
Shortcut key	None
Icon	None

Purpose To display the ECU pin in the **EESPort Configurations** controlbar.

Result The ECU pin is displayed in the **EESPort Configurations**.

Related topics

References

EESPort Configurations 120

Import ErrorConfiguration

Access You can access this command via:

Ribbon	None
Context menu of	EESPort in the **Project** controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose To import an error configuration as an XML file from the Windows file system.

Result ControlDesk opens a standard **Open** dialog for you to select an error configuration (XML) file to be imported to the EESPort.

In the Windows file system, the imported error configuration (XML) file is stored as a local working copy in the folder of the EESPort.

Related topics

References

EESPort 78

Insert EESPort

Access

You can access this command via:

Ribbon	XIL API EESPort – EESPorts
Context menu of	XIL API EESPorts folder in the Project controlbar
Shortcut key	None
Icon	

Note

This command is not available in operator mode.

Purpose

To create a new XIL API EESPort in the Project controlbar.

Result

The EESPort Implementation dialog opens for you to specify and create a new XIL API EESPort.

EESPort Implementation dialog

The dialog lets you select or create a port configuration (PORTCONFIG) file for the new EESPort.

Cancel Lets you cancel the creation of the new XIL API EESPort.

Configure after creation Lets you configure the XIL API EESPort automatically after its creation to connect it with the related failure simulation hardware. (The list on the EESPort Configurations controlbar is empty as long as no dSPACE XIL API EESPort is configured.)

If you enable this property when creating a *new* port configuration (PORTCONFIG) file, ControlDesk performs a preconfigured potential mapping based on the potential names that are provided for your failure simulation hardware. Refer to How to Create a New Port Configuration File on page 44.

(You can change this preconfigured potential mapping afterwards in the Properties controlbar.)

EESPort implementation Lets you specify the XIL API EESPort implementation.

Note

- Only EESPort implementations of XIL API version 2.1.0 are supported.
- If you use an EESPort implementation from dSPACE, the version of ControlDesk and the version of the EESPort implementation must be of the same dSPACE Release. For example, if you use ControlDesk 7.4, which is part of dSPACE Release 2021-A, you must select the XIL API EESPort implementation that is delivered with dSPACE Release 2021-A.

Finish Lets you create the specified XIL API EESPort in the Project controlbar.

For dSPACE failure simulation hardware:

- If you selected an existing port configuration file, a local working copy of the port configuration file is copied to the related EESPort subfolder of the ControlDesk experiment in the Windows file system.
- If you did not select an existing port configuration file, a new port configuration file is created in the related EESPort subfolder of the ControlDesk experiment in the Windows file system.

Generate default signal mapping after configuration (available if Configure after creation is enabled) Lets you generate a default signal mapping which is based on the ECU names and pin names that are specified for your dSPACE simulator.

Name Lets you specify the name of the XIL API EESPort to be created. The name must be unique in the ControlDesk experiment.

Next (for an XIL API EESPort implementation from dSPACE only) Lets you change to the PortConfiguration Settings dialog to specify further hardware-specific properties for the EESPort.

Port configuration file Lets you select a port configuration file for the EESPort. To interface the failure simulation hardware, an EESPort needs the hardware-dependent *port configuration file* (PORTCONFIG file). The file's contents must fit the connected HIL simulator architecture and its failure simulation hardware.

Product name Displays the product name of the XIL API EESPort implementation.

Product version Displays the product version of the XIL API EESPort implementation.

Vendor name Displays the vendor of the XIL API EESPort implementation.

XIL API version Displays the XIL API version of the EESPort implementation.

PortConfiguration Settings dialog

If you use an XIL API EESPort implementation from dSPACE, this dialog lets you configure further hardware-specific properties of the EESPort.

Note

The properties that are displayed in this dialog depend on the selected type of dSPACE hardware that you are using for the electrical error simulation. Refer to the **Type** property.

Back Lets you change to the **EESPort Implementation** dialog.

Cancel Lets you cancel the creation of the new XIL API EESPort.

Configuration Lets you specify properties that are related to the dSPACE hardware that you are using for the electrical error simulation.

For further information, refer to EESPort - Configuration Properties on page 88.

Enabled Lets you:

- Monitor the switching behavior of the failure simulation hardware via specific measurement variables. Refer to Tracing on page 134.
- Enable the usage of software triggers. Refer to Software trigger on page 134.

If you select **Offline mode** in the **General** properties of the EESPort, checking trigger conditions and monitoring the switching behavior and transition states of the failure simulation hardware is not possible. Error sets that are using software triggers with a condition are activated immediately without checking the specified trigger condition.

Experiment platform name Lets you select the name of a platform that is used by the ControlDesk experiment. This platform must represent the HIL simulator which executes the real-time application that provides the specific measurement variables for monitoring the switching behavior of the failure simulation hardware.

Note

The **Experiment platform name** is not stored in the port configuration (PORTCONFIG) file of the EESPort. It is only a convenience function in ControlDesk for specifying the **Real-time platform name** and **Variable description file path** properties and variable names in one step. You can also specify these properties and variable names manually without selecting an **Experiment platform name**.

Finish Lets you create the specified XIL API EESPort in the **Project** controlbar.

For dSPACE failure simulation hardware:

- If you selected an existing port configuration file, a local working copy of the port configuration file is copied to the related EESPort subfolder of the ControlDesk experiment in the Windows file system.
- If you did not select an existing port configuration file, a new port configuration file is created in the related EESPort subfolder of the ControlDesk experiment in the Windows file system.

Logging For support only, lets you enable logging. Logging essentially increases the execution time. Do not enable logging in normal operating situations. Enable logging only if you want to contact dSPACE Support.

Offline mode For tests on the host PC without a physical connection to a failure simulation hardware, lets you simulate the activation and triggering of errors without connecting the EESPort to the failure simulation hardware. If you select this property, the use of the tracing variables to monitor the switching behavior and transition states of the failure simulation hardware is disabled automatically. If you are working with software triggers, trigger conditions cannot be checked in offline mode. Error sets that use trigger conditions are therefore activated immediately without checking the specified condition.

Potentials To map a potential name and a potential type to a unique identifier (e.g., a natural number, starting with 0). The identification of potentials with unique identifiers is required by the ASAM AE XIL API standard. The list order of the potentials in the mapping corresponds to the unique identifiers assigned for the single potentials. For systems with discrete FIU, you find the potential names and potential types in the simulator's signal file. For SCALEXIO systems with the integrated SCALEXIO FIU, you define potentials with ConfigurationDesk as power switches.

Refer to EESPort – Potentials Properties on page 93.

Real-time platform name Lets you specify the platform name of the real-time platform which executes the real-time application that provides the specific measurement variables for monitoring the switching behavior. This platform name must be the same as the platform name in the ControlDesk Platforms/Devices controlbar. (In the port configuration file, this property is represented by the `PlatformName` attribute of the `RealTimeConfiguration` element.)

Signals To perform an optional mapping of ECU pins that you are using in your dSPACE HIL simulator to abstract signal names according to the ASAM AE XIL API standard. Each ECU pin is specified by its Pin name and the related ECU name.

The ECUs and ECU pins of your HIL system are specified in a signal file (for systems with discrete FIU) or via ConfigurationDesk (for SCALEXIO systems with the integrated SCALEXIO FIU).

Refer to EESPort – Signals Properties on page 97.

Software trigger Software triggers let you activate error sets in response to a defined trigger condition or duration. You can specify the trigger type for each error set individually.

Tracing To monitor the switching behavior and transition states of the failure simulation hardware via specific measurement variables. These variables are independent from other model variables. They do not measure the real behavior of the failure simulation hardware, they only estimate it according to empirical values. The variables are written and read only by the XIL API EESPort.

Refer to EESPort – Real-Time Configuration Properties on page 94.

Type Lets you select the dSPACE hardware that you are using for the electrical error simulation.

dSPACE Hardware	Description
Mid-Size / Full-Size (Variant 1)	A HIL system with a DS291 FIU Module, DS749 FIU Module, DS789 Sensor FIU Module, DS791 Actuator FIU Module, DS793 Sensor FIU Module, or a DS1450 Bus FIU Board. (Also valid for SCALEXIO systems that do not use the integrated SCALEXIO FIU.) Refer to Hardware for Failure Simulation (dSPACE XIL API Implementation Guide 🕮).
Full-Size (Variant 2)	A HIL system with a DS293 FIU Module. Refer to Hardware for Failure Simulation (dSPACE XIL API Implementation Guide 🕮).
SCALEXIO	A SCALEXIO system with the integrated SCALEXIO FIU. Refer to Hardware for Electrical Error Simulation on SCALEXIO Systems (SCALEXIO Hardware Installation and Configuration 🕮).

Variable description file path Lets you specify the path of the variable description (SDF) file that contains the specific measurement variables for monitoring the switching behavior of the failure simulation hardware. (In the related port configuration file, this property is represented by the `SystemDescriptionFilePath` attribute of the `RealTimeConfiguration` element.)

Related topics

Basics

Basics on Electrical Error Simulation Ports....14
Basics on Potential Mapping....31
Basics on Signal Mapping....35

HowTos

How to Create a New EESPort....38
How to Perform Potential Mapping....41

References

EESPort....78
EESPort - Configuration Properties....88

Lock Scrolling

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Error configuration in the working area ▪ Error set in the working area ▪ Error in an error set ▪ Signal in an error set
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To lock the view of the error sets displayed in the working area when triggering errors.

Description

The working area is usually too small to display all the error sets of an error configuration. That is why, ControlDesk scrolls the view to the currently activated error set by default when you are triggering errors. If you do not want ControlDesk to change the view automatically, you can use the **Lock Scrolling** command to lock the error sets that are displayed. ControlDesk proceeds with the triggering, but the currently activated error set is then outside the working area until you revoke the command.

Related topics

References

Trigger (Error Configuration)..... 148

New Error

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Error set in the working area ▪ Error in an error set ▪ Signal in an error set

Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To create a new error in an error set in the working area.

Related topics

References

Error 80

New ErrorConfiguration

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	EESPort in the Project controlbar
Shortcut key	None
Icon	

Note

This command is not available in operator mode.

Purpose

To create a new error configuration.

Related topics

References

EESPort 78
Error Configuration - General Properties 102
Open (Error Configuration) 139

New ErrorSet

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Error configuration in the working area ▪ Error set in the working area ▪ Error in an error set ▪ Signal in an error set
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To create a new error set in an error configuration in the working area.

Related topics

References

Error Configuration 81
Error Set 83

New Signal

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Error in an error set ▪ Signal in an error set
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To create a new signal in an error in the working area.

Related topics

References

Signal 85

Open (EESPort)

Access

You can access this command via:

Ribbon	None
Context menu of	EESPort in the Project controlbar
Shortcut key	None
Icon	None

Purpose

To open an XIL API EESPort.

Related topics

References

Close (EESPort) 117
EESPort 78

Open (Error Configuration)

Access

You can access this command via:

Ribbon	None
Context menu of	Error configuration in the Project controlbar
Shortcut key	None
Icon	None

Purpose

To open an error configuration in ControlDesk's working area.

Related topics

References

Close (Error Configuration)..... 117
Error Configuration..... 81

Properties

Access

You can access this command via:

Ribbon	None
Context menu of	▪ EESPort in the Project controlbar ▪ Error configuration in the Project controlbar ▪ Error configuration in the working area ▪ Error configuration page in the working area ▪ Error set in the working area ▪ Error in an error set ▪ Signal in an error set
Shortcut key	None
Icon	

Purpose

To open the Properties controlbar.

Related topics

References

Properties (Controlbar) (ControlDesk User Interface Handling)

Reload (Error Configuration)

Access

You can access this command via:

Ribbon	None
Context menu of	Error configuration in the Project controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To reload the last saved error configuration to the experiment.

Description

All the unsaved changes are lost.

Reload PortConfiguration

Access

You can access this command via:

Ribbon	None
Context menu of	EESPort in the Project controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To reload the original port configuration for an EESPort.

Result

ControlDesk reloads the original port configuration (PORTCONFIG) file that you selected when you created the EESPort. Changes to the port configuration you made since the EESPort was created are lost.

If you use dSPACE failure simulation hardware, in the related EESPort subfolder of the ControlDesk experiment in the Windows file system, the local working copy of the port configuration file is replaced with a new copy of original port configuration file.

Related topics

References

EESPort....78
Insert EESPort....131
Replace PortConfiguration....142

Replace PortConfiguration

Access

You can access this command via:

Ribbon	None
Context menu of	EESPort in the **Project** controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To replace the port configuration of an EESPort with a new one.

Result

ControlDesk opens a standard **Open** dialog for you to select a new port configuration (PORTCONFIG) file for the EESPort from the Windows file system. Changes to the port configuration you made since the EESPort was created are lost.

If you use dSPACE failure simulation hardware, in the related EESPort subfolder of the ControlDesk experiment in the Windows file system, the local working copy of the port configuration file is replaced with a copy of the new port configuration file.

Description

If you replace the port configuration file, you might not need to modify your EESPort application when you want to run the electrical error simulation on another simulator with a similar architecture that supports the configured error sets and errors.

Related topics

References

EESPort....78
Insert EESPort....131
Reload PortConfiguration....141

Reset Settings

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	None

Purpose

To reset the current configuration of the controlbar.

Related topics

References

EESPort Configurations 120
Filter Editor 125
Show Group Panel 145

Save (Error Configuration)

Access

You can access this command via:

Ribbon	None
Context menu of	Error configuration in the Project controlbar
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

Purpose

To save the error configuration (XML) file.

Show Column Chooser

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the EESPort Configurations controlbar
Shortcut key	None
Icon	

Purpose

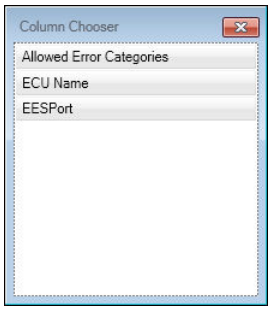
To open a dialog for specifying which columns to display.

Result

The Column Chooser is opened.

Column Chooser

With the Column Chooser, you can specify which columns to display in the EESPort Configurations controlbar. The column headers displayed in the Column Chooser represent the columns that are not displayed (see the following example).



To remove a column from the controlbar, drag it to the Column Chooser. To add a column to the controlbar, drag it from the dialog back to the controlbar.

Related topics

References

EESPort Configurations..... 120

Show Group Panel

Access

This command is available only if the group panel is not shown. You can access this command via:

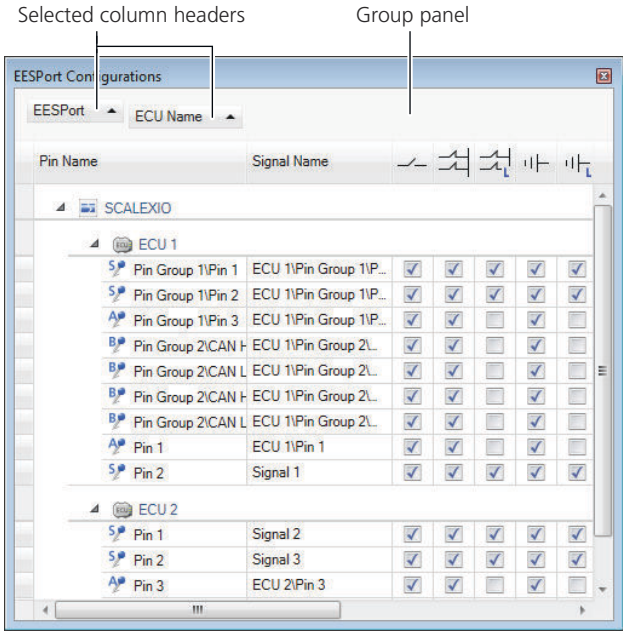
Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

To show the group panel.

Group panel

The group panel lets you group rows according to selected column headers (see the following example).



You can select or deselect column headers by dragging them to or from the group panel. If you drag multiple column headers to the group panel, the resulting row grouping is arranged hierarchically.

If the group panel is not shown, you can display it with the **Show Group Panel** command.

If you want to reset the current configuration of the group panel, use the **Reset Settings** command.

The group panel has the following context menu commands:

Clear Grouping (available from the context menu of the group panel) Lets you clear the currently selected grouping.

Full Collapse (available from the context menu of the group panel) Lets you collapse all the rows of the grid.

Full Expand (available from the context menu of the group panel) Lets you fully expand all the rows of the grid.

Related topics

References

EESPort Configurations 120
Group by This Column 127
Group Summary Editor 127
Hide Group Panel 128
Reset Settings 143
Ungroup 149

Show Search Panel

Access

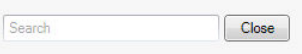
This command is available only if the search panel is not shown. You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	**Ctrl+F**
Icon	None

Purpose

To show the search panel.

Search panel



In the edit field of the search panel, you can specify a search substring. Immediately, only rows containing this substring are displayed. Each occurrence of the substring is highlighted in yellow. Numbers and letters are not distinguished.

You can click ⊗, to clear a search substring.

You can click **Close** to hide the search panel.

Related topics

References

EESPort Configurations 120
Hide Search Panel 129

Sort Ascending

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

To sort the rows alphabetically in ascending order by the selected column.

Related topics

References

Clear Sorting 116
EESPort Configurations 120
Sort Descending 147

Sort Descending

Access

You can access this command via:

Ribbon	None
Context menu of	Column header in the **EESPort Configurations** controlbar
Shortcut key	None
Icon	

Purpose

To sort the rows alphabetically in descending order by the selected column.

Related topics

References

Clear Sorting 116
EESPort Configurations 120
Sort Ascending 147

Trigger (Error Configuration)

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	▪ Error configuration in the Project controlbar ▪ Error configuration page in the working area
Shortcut key	None
Icon	

Purpose

To activate the first or next error set of an error configuration.

Description

You can activate error sets by using different triggers:

- *Manual trigger*

  The trigger is explicitly specified in the XIL API application.
- *Hardware trigger*

  The FIU hardware reacts to a trigger input line.
- *Software trigger*

  The FIU hardware reacts to a trigger condition or duration defined in software, for example, defined in the model or in other software connected to the FIU hardware.

Note

dSPACE XIL API .NET EESPort implmentation supports only manual and software triggers.

Related topics

References

Error Set....83
Lock Scrolling....136

Ungroup

Access

You can access this command only via the context menu of column headers that are moved to the group panel of the EESPort Configurations controlbar.

Purpose

To remove a column header from the group panel to revoke the related grouping of rows.

Related topics

References

EESPort Configurations....120
Group by This Column....127
Group Summary Editor....127
Hide Group Panel....128
Show Group Panel....145

Unload (Error Configuration)

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	▪ Error configuration in the Project controlbar ▪ Error configuration page in the working area
Shortcut key	None
Icon	

Purpose

To unload an error configuration from the failure simulation hardware.

Description

The failure simulation hardware is disconnected from the original host PC. The interface of the failure simulation hardware can now be used by another host PC.

Related topics

References

Download (Error Configuration)..... 119

Update (Error Configuration)

Access

You can access this command via:

Ribbon	XIL API EESPort – ErrorConfigurations
Context menu of	▪ Error configuration in the Project controlbar ▪ Error configuration page in the working area
Shortcut key	None
Icon	

Note

This command is not available in operator mode.

Purpose

To update an error configuration on the failure simulation hardware if you configured one or more new error sets after activating an error configuration.

Description

You can extend an error configuration that is downloaded to a failure simulation hardware with new error sets. New error sets are always added behind the last error set of an error configuration. You can use the **Update** command to download the added error sets to the failure simulation hardware.

Related topics

References

Error Configuration..... 81
Error Set..... 83

Automation

Where to go from here

Information in this section

Programming ControlDesk Automation

Where to go from here

Information in this section

Automating Electrical Error Simulation via XIL API EESPort..... 152
ControlDesk's XIL API EESPort automation interface extends the dSPACE XIL API by ControlDesk-specific automation interfaces.

Information in other sections

Tool Automation Demos (ControlDesk Automation 🕮)
Demonstrate how to automate ControlDesk and use ControlDesk events.

Automating Electrical Error Simulation via XIL API EESPort

Introduction

ControlDesk's XIL API EESPort automation interface extends the dSPACE XIL API by ControlDesk-specific automation interfaces.

Note

If you only use ControlDesk's XIL API EESPort automation interface you cannot fully automate electrical error simulation.

For example, you cannot:

- Create and configure new errors
- Download error configurations and trigger error sets

To prepare an electrical error simulation via automation, use the dSPACE XIL API .NET implementation, which supports the Electrical Error Simulation Port (EESPort).

Automation interfaces

The following list shows the main interfaces for automating electrical error simulation via XIL API EESPort.

Interface	Description
IXaXILAPIEESPort (refer to XILAPIEESPort / IXaXILAPIEESPort <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing the XIL API EESPort.
IXaAvailableEESPortImplementations (refer to AvailableEESPortImplementations / IXaAvailableEESPortImplementations <<Collection>> (ControlDesk Automation 🕮))	Interface for accessing the list of ErrorConfigurations.

Interface	Description
IXaEESPorts (refer to EESPorts / IXaEESPorts <<Collection>> (ControlDesk Automation 🕮))	Interface for accessing the list of XIL API EESPorts.
IXaEESPort (refer to EESPort / IXaEESPort <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing a single EESPort.
IXaEESPortImplementation (refer to EESPortImplementation / IXaEESPortImplementation <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing information about a single XIL API EESPort implementation.
IXaErrorConfiguration (refer to ErrorConfiguration / IXaErrorConfiguration <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing a single ErrorConfiguration.
IXaErrorConfigurations (refer to ErrorConfigurations / IXaErrorConfigurations <<Collection>> (ControlDesk Automation 🕮))	Interface for accessing the list of ErrorConfigurations.

Related topics

Basics

Basics on Electrical Error Simulation..... 12
Implementing an EESPort Client Application (dSPACE XIL API Implementation Guide 🕮)

XIL API EESPort

XIL API EESPort-Related Interfaces

Introduction

ControlDesk's XIL API EESPort lets you add and edit XIL API EESPort configurations.

Description

The object that implements the *IXaXILAPIEESPort* interface is used to configure the XIL API EESPort.

The collection that implements the *IXaEESPorts* interface is used to add and edit XIL API EESPort configurations.

Related interfaces

Interface	Description
IXaXILAPIEESPort (refer to XILAPIEESPort / IXaXILAPIEESPort <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing the XIL API EESPort.
IXaAvailableEESPortImplementations (refer to AvailableEESPortImplementations / IXaAvailableEESPortImplementations <<Collection>> (ControlDesk Automation 🕮))	Interface for accessing the list of ErrorConfigurations.
IXaEESPorts (refer to EESPorts / IXaEESPorts <<Collection>> (ControlDesk Automation 🕮))	Interface for accessing the list of XIL API EESPorts.
IXaEESPort (refer to EESPort / IXaEESPort <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing a single EESPort.
IXaEESPortImplementation (refer to EESPortImplementation / IXaEESPortImplementation <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing information about a single XIL API EESPort implementation.
IXaErrorConfiguration (refer to ErrorConfiguration / IXaErrorConfiguration <<Interface>> (ControlDesk Automation 🕮))	Interface for accessing a single ErrorConfiguration.
IXaErrorConfigurations (refer to ErrorConfigurations / IXaErrorConfigurations <<Collection>> (ControlDesk Automation 🕮))	Interface for accessing the list of ErrorConfigurations.

Related documentation

Topic	Description
Automating Electrical Error Simulation via XIL API EESPort on page 152	ControlDesk's XIL API EESPort automation interface extends the dSPACE XIL API by ControlDesk-specific automation interfaces.

Troubleshooting

Problem with Failure Simulation on SCALEXIO Systems

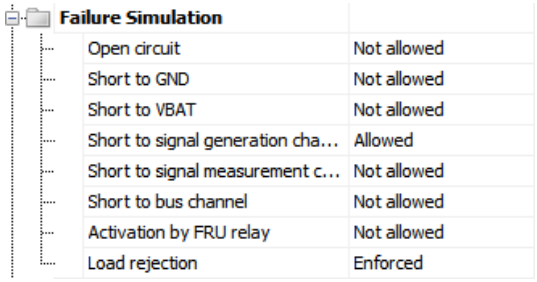
Problem

You cannot activate short circuits between pins of different channel categories in ControlDesk, if the allowed failure classes are specified incompletely in ConfigurationDesk.

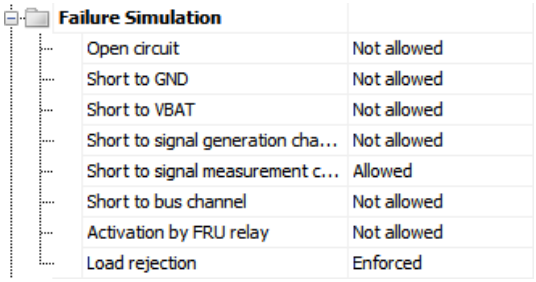
Solution

To activate short circuits between pins of different channel categories (i.e., signal measurement channels, signal generation channels, and bus channels), the desired signals have to allow each other their corresponding failure class in ConfigurationDesk.

The following example shows the required setting in ConfigurationDesk for a pin of a signal measurement channel that is allowed for a short circuit to a signal generation channel.



For the corresponding signal generation channel, you have to allow a short circuit to a signal measurement channel.



Failure Simulation	
Open circuit	Not allowed
Short to GND	Not allowed
Short to VBAT	Not allowed
Short to signal generation cha...	Not allowed
Short to signal measurement c...	Allowed
Short to bus channel	Not allowed
Activation by FRU relay	Not allowed
Load rejection	Enforced

Limitations

Limitations for Electrical Error Simulation via XIL API EESPort

Limitations

For information on the limitations of the dSPACE XIL API implementation when using EESPort, refer to Limitations (dSPACE XIL API Implementation Guide 🕮).

Related topics

Basics

Limitations (dSPACE XIL API Implementation Guide 🕮)

Glossary

Introduction

Briefly explains the most important expressions and naming conventions used in the ControlDesk documentation.

Where to go from here

Information in this section

Numerics

3-D Viewer An instrument for displaying items in a 3-D environment.

A

A2L file A file that contains all the relevant information on measurement and calibration variables in an ECU application and the ECU's communication interface(s). This includes information on the variables' memory addresses and conversion methods, the memory layout and data structures in the ECU as well as interface description data (IF_DATA).

Acquisition An object in the Measurement Configuration controlbar that specifies the variables to be measured and their measurement configuration.

Active variable description The variable description that is currently active for a platform/device. Multiple variable descriptions can be assigned to one platform/device, but only one of them can be active at a time.

Additional write variable A scalar parameter or writable measurement variable that can be connected to an instrument in addition to the main variable. When the value of the main variable changes, the changed value is also applied to all the additional write variables connected to the instrument.

Airspeed Indicator An instrument for displaying the airspeed of a simulated aircraft.

Altimeter An instrument for displaying the altitude of a simulated aircraft.

Animated Needle An instrument for displaying the value of a connected variable by a needle deflection.

Application image An image file that contains all the files that are created when the user builds a real-time application. It particularly includes the variable

description (SDF) file. To extend a real-time application, ControlDesk lets the user create an updated application image from a data set. The updated application image then contains a real-time application with an additional set of parameter values.

Artificial Horizon An instrument displaying the rotation on both the lateral and the longitudinal axis to indicate the angle of pitch and roll of a simulated aircraft. The Artificial Horizon has a pitch scale and a roll scale.

Automatic Reconnect Feature for automatically reconnecting to platform/device hardware, for example, when the ignition is turned off and on, or when the physical connection between the ControlDesk PC and the ECU is temporarily interrupted.

If the feature is enabled for a platform/device and if the platform/device is in the 'unplugged' state, ControlDesk tries to re-establish the logical connection to the platform/device hardware. After the logical connection is re-established, the platform/device has the same state as before the unplugged state was detected. A measurement started before the unplugged state was detected is resumed.

Automation A communication mechanism that can be used by various programming languages. A client can use it to control a server by calling methods and properties of the server's automation interface.

Automation script A script that uses automation to control an automation server.

Axis point object Common axis

B

Bar An instrument (or a value cell type of the Variable Array) for displaying a numerical value as a bar deflection on a horizontal or vertical scale.

Bitfield A value cell type of the Variable Array for displaying and editing the source value of a parameter as a bit string.

Bookmark A marker for a certain event during a measurement or recording.

Browser An instrument for displaying HTML and TXT files. It also supports Microsoft Internet Explorer© plug-ins that are installed on your system.

Bus communication replay A feature of the Bus Navigator that lets you replay logged bus communication data from a log file. You can add replay nodes

to the Bus Navigator tree for this purpose. You can specify filters to replay selected parts of the logged bus communication.

Bus configuration A configuration of all the controllers, communication matrices, and messages/frames/PDUs of a specific communication bus such as CAN. ControlDesk lets you display and experiment with bus configurations in the Bus Navigator.

Bus connection A mode for connecting dSPACE real-time hardware to the host PC via bus. The list below shows the possible bus connections:

- dSPACE real-time hardware installed directly in the host PC
- dSPACE real-time hardware installed in an expansion box connected to the host PC via dSPACE link board

Bus Instrument An instrument available for the Bus Navigator. It can be configured for different purposes, for example, to display information on received messages (RX messages) or to manipulate and transmit messages (TX messages). The instrument is tailor-made and displays only the message- and signal-specific settings which are enabled for display and/or manipulation by ControlDesk during run time.

Bus logging A feature of the Bus Navigator that lets you log raw bus communication data. You can add logger nodes on different hierarchy levels of the Bus Navigator tree for this purpose. You can specify filters to log filtered bus communication. The logged bus communication can be replayed.

Bus monitoring A feature of the Bus Navigator that lets you observe bus communication. You can open monitoring lists and add monitor nodes on different hierarchy levels of the Bus Navigator tree for this purpose. You can specify filters to monitor filtered bus communication.

Bus Navigator A controlbar for handling bus messages, such as CAN messages, LIN frames, and Ethernet packets.

Bus statistics A feature of the Bus Navigator that lets you display and log statistical information on the bus load during bus monitoring.

Bypassing A method for replacing an existing ECU function by running a new function.

C

Calculated variable A scalar variable that can be measured and recorded, and that is derived from one or more *input variables*.

The following input variable types are supported:

- Measurement variables
- Single elements of measurement arrays or value blocks
- Scalar parameters, or existing calculated variables

The value of a calculated variable is calculated via a user-defined *computation formula* that uses one or more input variables.

Calculated variables are represented by the  symbol.

CalDemo ECU A demo program that runs on the same PC as ControlDesk. It simulates an ECU on which the Universal Measurement and Calibration (XCP) protocol and the Unified Diagnostic Services (UDS) protocol are implemented.

The CalDemo ECU allows you to perform parameter calibration, variable measurement, and ECU diagnostics with ControlDesk under realistic conditions, but without having to have a real ECU connected to the PC. Communication between the CalDemo ECU and ControlDesk can be established via XCP on CAN or XCP on Ethernet, and UDS on CAN.

Tip

If communication is established via XCP on Ethernet, the CalDemo ECU can also run on a PC different from the PC on which ControlDesk is running.

The memory of the CalDemo ECU consists of two areas called memory page. Each page contains a complete set of parameters, but only one page is accessible by the CalDemo ECU at a time. You can easily switch the memory pages of the CalDemo ECU to change from one parameter to another in a single step.

Two ECU tasks run on the CalDemo ECU:

- ECU task #1 runs at a fixed sample time of 5 ms. In ControlDesk's **Measurement Configuration**, ECU task #1 is related to the time-based **5 ms**, **10 ms**, **50 ms** and **100 ms** measurement rasters of the CalDemo ECU.
- ECU task #2 has a variable sample time. Whenever the CalDemo ECU program is started, the initial sample time is 5 ms. This can then be increased or decreased by using the **dSPACE CalDemo** dialog.

  ECU task #2 is related to the **extEvent** measurement raster of the CalDemo ECU.

The CalDemo ECU can also be used to execute diagnostic services and jobs, handle DTCs and perform measurement and calibration via ECU diagnostics.

The CalDemo ECU program is run by invoking `CalDemo.exe`. The file is located in the `.\Demos\CalDemo` folder of the ControlDesk installation.

Calibration Changing the parameter values of real-time applications or ECU applications.

Calibration memory segment Part of the memory of an ECU containing the calibratable parameters. Memory segments can be defined as `MEMORY_SEGMENT` in the A2L file. ControlDesk can use the segments to evaluate the memory pages of the ECU.

ControlDesk lets you perform the calibration of:

- Parameters inside memory segments
- Parameters outside memory segments
- Parameters even if no memory segments are defined in the A2L file.

CAN Bus Monitoring device A device that monitors the data stream on a CAN bus connected to the ControlDesk PC.

The CAN Bus Monitoring device works, for example, with PC-based CAN interfaces such as the DCI-CAN2 or the DCI-CAN/LIN1.

The device supports the following variable description file types:

- DBC
- FIBEX
- AUTOSAR system description (ARXML)

CANGenerator A demo program that simulates a CAN system, that is, it generates signals that can be measured and recorded with ControlDesk. The program runs on the same PC as ControlDesk.

The CANGenerator allows you to use the CAN Bus Monitoring device under realistic conditions, but without having to have any device hardware connected to the PC.

The CAN (Controller Area Network) protocol is used for communication between the CANGenerator and ControlDesk. However, since the CANGenerator runs on the same PC as ControlDesk, ControlDesk does not communicate with the device via a real CAN channel, but via a *virtual CAN channel* implemented on the host PC.

You can start the CAN generator program by running `CANGenerator.exe`. The file is located in the `.\Demos\CANGenerator` folder of the ControlDesk installation.

Capture A data packet of all the measurement variables assigned to a measurement raster. The packet comprises the data that results from a single triggering of the raster.

CCP Abbreviation of CAN Calibration Protocol. This protocol can be implemented on electronic control units (ECUs) and allows users to access ECUs with measurement and calibration systems (MCS) such as ControlDesk.

The basic features of CCP are:

- Read and write access to the ECU memory, i.e., providing access for calibration
- Synchronous data acquisition
- Flash programming for ECU development purposes

The CCP protocol was developed by ASAM e.V. (Association for Standardization of Automation and Measuring Systems e.V.). For the protocol specification, refer to http://www.asam.net.

The following device supports ECUs with an integrated CCP service:

- CCP device

CCP device A device that provides access to an ECU with CCP connected to the ControlDesk PC via CAN, for example, for measurement and calibration purposes via CCP (CAN Calibration Protocol).

Check Button An instrument (or a cell type of the Variable Array) for displaying whether the value of a connected variable matches predefined values or for writing a predefined value to a connected variable.

cmdloader A command line tool for handling applications without using the user interface of an experiment software.

Common axis A parameter that consists of a 1-dimensional array containing axis points. A common axis can be referenced by one or more curves and/or maps. Calibrating the data points of a common axis affects all the curves and/or maps referencing the axis.

Common axes are represented by the symbol.

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Computation method A formula or a table that defines the transformation of a source value into a converted value (and vice versa). In addition to the computation methods defined in the variable description file, ControlDesk provides the *__Identity* computation method which means the converted and the source value are equal.

Connected A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- A platform/device must be in the 'connected' state before it can change to the 'measuring/recording' or 'online calibration started' state.
- Online calibration is impossible. ControlDesk did not yet adjust the memory segments containing calibration data in the platform/device and on the corresponding hardware. Offline calibration is possible.
- Platform/device configuration is not possible. However, you can invoke platform/device configuration for a platform/device that is in the connected state. ControlDesk temporarily sets the platform/device to the disconnected state.

The 'connected' platform/device state is indicated by the icon.

Connection mode dSPACE real-time systems can be installed within the host PC or connected to the host via a bus interface and/or via Ethernet. When the Ethernet is being used, different network clients might exist. The connection type being used and, in the case of Ethernet, the network client being used, determine the dSPACE systems that can be accessed.

Control primitive A special diagnostic communication object for changing communication states or protocol parameters, or for identifying (ECU) variants.

Controlbar A window or pane outside the working area. Can be docked to an edge of the main window or float in front of it. A controlbar can contain a

document, such as a layout, or a tool, such as the **Bus Navigator**. It can be grouped with other controlbars in a window with tabbed pages.

ControlDesk The main version of ControlDesk for creating and running experiments, and for accessing dSPACE real-time hardware and VEOS. The functionality can be extended by optional software modules.

ControlDesk - Operator Version A version of ControlDesk that provides only a subset of functionality for running existing experiments. The functionality can be extended by optional software modules.

ControlDesk Bus Navigator Module An optional software module for ControlDesk for handling bus messages, such as CAN, LIN, and FlexRay messages, frames, and PDUs and Ethernet packets.

ControlDesk ECU Diagnostics Module An optional software module for ControlDesk that facilitates the calibration and validation of ECU diagnostic functions.

ControlDesk ECU Interface Module An optional software module for ControlDesk for calibration and measurement access to electronic control units (ECUs). The module is also required for calibration and measurement access to virtual ECUs (V-ECUs) used in SIL testing scenarios.

ControlDesk Signal Editor Module An optional software module for ControlDesk for the graphical definition and execution of signal generators for stimulating model variables of real-time/offline simulation applications.

Controller board Single-board hardware computing the real-time application. Contains a real-time processor for fast calculation of the model and I/O interfaces for carrying out the control developments.

Conversion table A table that specifies the value conversion of a source value into a converted value. In the case of verbal conversion, the converted value is a string that represents one numerical value or a range of numerical values.

Conversion type The type of a computation method, for example a linear function or a verbal computation method.

Curve A parameter that consists of

- A 1-dimensional array containing the axis points for the x-axis. This array can also be specified by a reference to a common axis.
- Another 1-dimensional array containing data points. The curve assigns one data point to each axis point.

Curves are represented by the symbol.

D

DAQ module A hardware module for the acquisition of physical quantities

Data Cursor One or two cursors that are used to display the values of selected chart positions in a Time Plotter or an Index Plotter.

Data logger An object in the Measurement Configuration controlbar that lets you configure a data logging.

Data logger signal list A list that contains the variables to be included in subsequent data loggings on real-time hardware.

Data logging The recording of data on dSPACE real-time hardware that does not require a physical connection between the host PC and the real-time hardware. In contrast to flight recording, data logging is configured in ControlDesk.

Data set A set of the parameters and their values of a platform/device derived from the variable description of the platform/device. There are different types of data sets:

- Reference data set
- Sub data set
- Unassigned data set
- Working data set

DCI-CAN/LIN1 A dSPACE-specific interface between the host PC and the CAN/CAN FD bus and/or LIN bus. The DCI-CAN/LIN1 transfers messages between the CAN-/LIN-based devices and the host PC via the universal serial bus (USB).

DCI-CAN2 A dSPACE-specific interface between the host PC and the CAN bus. The DCI-CAN2 transfers CAN and CAN FD messages between the CAN-based devices and the host PC via the universal serial bus (USB).

DCI-GSI2 Abbreviation of *dSPACE Communication Interface - Generic Serial Interface 2*. A dSPACE-specific interface for ECU calibration, measurement and ECU interfacing.

DCI-GSI2 device A device that provides access to an ECU with DCI-GSI2 connected to the ControlDesk PC for measurement, calibration, and bypassing purposes via the ECU's debug interface.

DCI-KLine1 Abbreviation of *dSPACE Communication Interface - K-Line Interface*. A dSPACE-specific interface between the host PC and the diagnostics bus via K-Line.

Debug interface An ECU interface for diagnostics tasks and flashing.

Default raster A platform-/device-specific measurement raster that is used when a variable of the platform/device is connected to a plotter or a recorder, for example.

Deposition definition A definition specifying the sequence in which the axis point values of a curve or map are deposited in memory.

Device A software component for carrying out calibration and/or measurement, bypassing, ECU flash programming, or ECU diagnostics tasks.

ControlDesk provides the following devices:

- Bus devices:
  - CAN Bus Monitoring device
  - Ethernet Bus Monitoring device
  - LIN Bus Monitoring device
- ECU Diagnostics device
- GNSS device
- Measurement and calibration devices:
  - CCP device
  - DCI-GSI2 device
  - XCP on CAN device
  - XCP on Ethernet device

Each device usually has a variable description that specifies the device's variables to be calibrated and measured.

Diagnostic interface Interface for accessing the fault memory of an ECU.

Diagnostic job (often called Java job) Programmed sequence that is usually built from a sequence of the diagnostic service. A diagnostic job is either a single-ECU job or a multiple-ECU job, depending on whether it communicates with one ECU or multiple ECUs.

Diagnostic protocol A protocol that defines how an ECU communicates with a connected diagnostic tester. The protocol must be implemented on the ECU and on the tester. The diagnostics database specifies the diagnostic protocol(s) supported by a specific ECU.

ControlDesk's ECU Diagnostics device supports CAN and K-Line as the physical layers for communication with an ECU connected to the ControlDesk PC. For information on the supported diagnostic protocols with CAN and K-Line, refer to Basics of ECU Diagnostics with ControlDesk (ControlDesk ECU Diagnostics 🕮).

Diagnostic service A service implemented on the ECU as a basic diagnostic communication element. Communication is performed by selecting a service, configuring its parameters, executing it, and receiving the ECU results. When a service is executed, a defined request is sent to the ECU and the ECU answers with a specific response.

Diagnostic trouble code (DTC) A hexadecimal index for the identification of vehicle malfunctions. DTCs are stored in the fault memory of ECUs and can be read by diagnostic testers.

Diagnostics database A database that completely describes one or more ECUs with respect to diagnostics communication. ControlDesk supports the ASAM MCD-2 D ODX database format, which was standardized by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.). For the format specification, refer to http://www.asam.net.

Proprietary diagnostics database formats are not supported by ControlDesk.

Diagnostics Instrument An instrument for communicating with an ECU via the diagnostic protocol using diagnostic services, diagnostic jobs, and control primitives.

Disabled A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware.
- When a platform/device is disabled, ControlDesk does not try to establish the logical connection for that platform/device. Any communication between the platform/device hardware and ControlDesk is rejected.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disabled' platform/device state is indicated by the  icon.

Disconnected A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware.
- When a platform/device is in the disconnected state, ControlDesk does not try to re-establish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disconnected' platform/device state is indicated by the  icon.

Display An instrument (or a value cell type of the Variable Array) for displaying the value of a scalar variable or the text content of an ASCII variable.

Documents folder A standard folder for user-specific documents.

```
%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>
```

DS1006 Processor Board platform A platform that provides access to a DS1006 Processor Board connected to the host PC for HIL simulation and function prototyping purposes.

DS1007 PPC Processor Board platform A platform that provides access to a single multicore DS1007 PPC Processor Board or a DS1007 multiprocessor system consisting of two or more DS1007 PPC Processor Boards, connected to the host PC for HIL simulation and function prototyping purposes.

DS1104 R&D Controller Board platform A platform that provides access to a DS1104 R&D Controller Board installed in the host PC for function prototyping purposes.

DS1202 MicroLabBox platform A platform that provides access to a MicroLabBox connected to the host PC for function prototyping purposes.

DsDAQ service A service in a real-time application or offline simulation application (OSA) that provides measurement data from the application to the

host PC. Unlike the host service, the DsDAQ service lets you perform, for example, triggered measurements with complex trigger conditions.

The following platforms support applications that contain the DsDAQ service:

- DS1007 PPC Processor Board platform
- DS1202 MicroLabBox platform
- MicroAutoBox III platform
- SCALEXIO platform
- VEOS platform
- XIL API MAPort platform

dSPACE Calibration and Bypassing Service An ECU service for measurement, calibration, bypassing, and ECU flash programming. The dSPACE Calibration and Bypassing Service can be integrated on the ECU. It provides access to the ECU application and the ECU resources and is used to control communication between an ECU and a calibration and/or bypassing tool.

With the dSPACE Calibration and Bypassing Service, users can run measurement, calibration, bypassing, and flash programming tasks on an ECU via the DCI-GSI2. The service is also designed for bypassing ECU functions using dSPACE prototyping hardware by means of the RTI Bypass Blockset in connection with DPMEM PODs. The dSPACE Calibration and Bypassing Service allows measurement, calibration, and bypassing tasks to be performed in parallel.

dSPACE Internal Bypassing Service An ECU service for on-target prototyping. The dSPACE Internal Bypassing Service can be integrated in the ECU application. It lets you add additional functions to be executed in the context of the ECU application without the need for recompiling the ECU application.

dSPACE Log A collection of errors, warnings, information, questions, and advice issued by all dSPACE products and connected systems over more than one session.

dSPACE system A hardware system such as a MicroAutoBox III or SCALEXIO system on which the real-time application runs.

Duration trigger A trigger that defines a duration. Using a duration trigger, you can, for example, specify the duration of data acquisition for a measurement raster. A duration trigger can be used as a stop trigger.

E

ECU Abbreviation of *electronic control unit*.

ECU application A sequence of operations executed by an ECU. An ECU application is mostly represented by a group of files such as ECU Image files, MAP files, A2L files and/or software module description files.

ECU calibration interface Interface for accessing an ECU by either emulating the ECU's memory or using a communication protocol (for example, XCP on CAN).

ECU diagnostics Functions such as:

- Handling the ECU fault memory: Entries in the ECU's fault memory can be read, cleared, and saved.
- Executing diagnostic services and jobs: Users can communicate with an ECU via a diagnostic protocol using diagnostic services, diagnostic jobs, and control primitives.

ControlDesk provides the ECU Diagnostics device device to access ECUs for diagnostic tasks. Communication is via diagnostic protocols implemented on the ECUs.

ECU diagnostics with ControlDesk are completely based on Open Diagnostic Data Exchange (ODX), the ASAM MCD-2 D diagnostics standard.

ControlDesk provides the Fault Memory Instrument and the Diagnostics Instrument for ECU diagnostics tasks.

ECU Diagnostics device A device that provides access to ECUs connected to the ControlDesk PC via CAN or K-Line for diagnostics or flash programming purposes.

ControlDesk provides the *ECU Diagnostics v2.0.2* device, which supports the ASAM MCD-3 D V2.0.2 standard.

ControlDesk supports the following ODX database standards:

- ASAM MCD-2 D V2.0.1
- ASAM MCD-2 D V2.2.0 (ISO 22901-1)

ECU flash programming A method by which new code or data is stored in ECU flash memory.

ECU Image file A binary file that is part of the ECU application. It usually contains the code of an ECU application and the data of the parameters within the application. It can be stored as an Intel Hex (HEX) or Motorola S-Record (MOT or S19) file.

EESPort Configurations controlbar A controlbar for configuring error configurations.

Electrical error simulation Simulating electrical errors such as loose contacts, broken cables, and short-circuits, in the wiring of an ECU. Electrical error simulation is performed by the failure simulation hardware of an HIL simulator.

Electrical Error Simulation port (EESPort) An *Electrical Error Simulation port* (EESPort) provides access to a failure simulation hardware for simulating electrical errors in an ECU wiring according to the ASAM AE XIL API standard.

The configuration of the EESPort is described by a hardware-dependent *port configuration* and one or more *error configurations*.

Environment model A model that represents a part or all of the ECU's environment in a simulation scenario.

The environment model is a part of the simulation system.

Environment VPU The executable of an environment model built for the VEOS platform. An environment VPU is part of an offline simulation application (OSA).

Error An electrical error that is specified by:

- An error category
- An error type
- A load type

Error category The error category defines how a signal is disturbed. Which errors you can create for a signal depends on the connected failure simulation hardware.

Error configuration An XML file that describes a sequence of errors you want to switch during electrical error simulation. Each error configuration comprises error sets with one or more errors.

Error set An error set is used to group errors (pin failures).

Error type The error type specifies the way an error category – i.e., an interruption or short circuit of signals – is provided. The error type defines the disturbance itself.

Ethernet Bus Monitoring device A device that monitors the data stream on an Ethernet network connected to the ControlDesk PC.

The device supports the following variable description file type:

- AUTOSAR system description (ARXML)

Ethernet connection A mode for connecting dSPACE real-time hardware to the host PC via Ethernet. The list below shows the possible Ethernet connections:

- dSPACE real-time hardware installed in an expansion box connected to the host PC via Ethernet.
- MicroAutoBox II/III and MicroLabBox connected via Ethernet.

Ethernet decoding A feature of the Bus Navigator that lets you view protocol data and raw data of an Ethernet frame.

Event An event that is triggered by an action performed in ControlDesk.

Event context The scope of validity of event sources and events. There is one event handler code area for each event context.

Event handler Code that is executed when the related event occurs.

Event management Functionality for executing custom code according to actions triggered by ControlDesk.

Event source An object providing and triggering events. *LayoutManagement* is an example of an event source.

Event state State of an event. ControlDesk provides the following event states:

- No event handler is defined
- Event handler is defined and enabled
- Event handler is defined and disabled
- Event handler is defined, but no Python code is available
- Event handler is deactivated because a run-time error occurred during the execution of the Python code

Expansion box A box that hosts dSPACE boards. It can be connected to the host PC via bus connection or via network.

Experiment A container for collecting and managing information and files required for a parameter calibration and/or measurement task. A number of experiments can be collected in a project but only one of them can be active.

Extension script A Python script (PY or PYC file) that is executed each time ControlDesk starts up. An extension script can be executed for all users or user-specifically.

F

Failure insertion unit Hardware unit used with dSPACE simulators to simulate failures in the wiring of an ECU, such as broken wire and short circuit to ground.

Fault memory Part of the ECU memory that stores diagnostic trouble code (DTC) entries with status and environment information.

Fault Memory Instrument An instrument for reading, clearing, and saving the content of the ECU's fault memory.

Firmware update An update for the firmware installed in the board's flash memory. Firmware should be updated if it is older than required by the real-time application to be downloaded.

Fixed axis An axis with data points that are not deposited in the ECU memory. Unlike a common axis, a fixed axis is specified within a curve or map. The parameters of a fixed axis cannot be calibrated.

Fixed parameter A parameter that has a fixed value during a running simulation. Changing the value of a fixed parameter does not immediately affect the simulation results. The affect occurs only after you stop the simulation and

start it again. A fixed parameter is represented by an added pin in its symbol, for example: .

Flash job A specific diagnostic job for flashing the ECU memory. A flash job implements the process control for flashing the ECU memory, such as initialization, security access, writing data blocks, etc.

Flight recording The recording of data on dSPACE real-time hardware that does not require a physical connection between the host PC and the real-time hardware. In contrast to data logging, flight recording is not configured in ControlDesk but via RTI and RTLib.

Frame An instrument for adding a background frame to a layout, for example, to visualize an instrument group.

G

Gauge An instrument for displaying the value of the connected variable by a needle deflection on a circular scale.

Gigalink module A dSPACE board for connecting several processor boards in a multiprocessor system. The board allows high-speed serial data transmission via fiber-optic cable.

GNSS data Positioning and timing data that is transmitted by a Global Navigation Satellite System (GNSS), such as GPS, GLONASS, or Galileo. GNSS receivers use this data to determine their location.

GNSS device A device that provides positioning data from a GNSS receiver (e.g., a serial GPS mouse) in ControlDesk.

ControlDesk provides the *GNSS (GPS, GLONASS, Galileo, ...)* device that supports various global navigation satellite systems.

GPX file An XML file that contains geodata, such as waypoints, routes, or tracks. In ControlDesk, you can import GPX files to visualize GNSS positioning data in a Map instrument.

Group A collection of variables that are grouped according to a certain criterion.

H

Heading Indicator An instrument displaying the heading direction of a simulated aircraft on a circular scale.

Host service A service in a real-time application that provides measurement data from the application to the host PC.

The following platforms support applications that contain the host service:

- DS1006 Processor Board platform
- DS1104 R&D Controller Board platform
- MicroAutoBox platform
- Multiprocessor System platform

Index Plotter A plotter instrument for displaying signals that are measured in an event-based raster (index plots).

Input quantity A measurement variable that is referenced by a common axis and that provides the input value of that axis.

Instrument An on-screen representation that is designed to monitor and/or control simulator variables interactively and to display data captures. Instruments can be arranged freely on layouts.

The following instruments can be used in ControlDesk:

- 3-D Viewer
- Airspeed Indicator
- Altimeter
- Animated Needle
- Artificial Horizon
- Bar
- Browser
- Bus Instrument
- Check Button
- Diagnostics Instrument
- Display
- Fault Memory Instrument
- Frame
- Gauge
- Heading Indicator
- Index Plotter
- Invisible Switch
- Knob
- Multistate Display
- Multiswitch
- Numeric Input
- On/Off Button

- Push Button
- Radio Button
- Selection Box
- Slider
- Sound Controller
- Static Text
- Steering Controller
- Table Editor
- Time Plotter
- Variable Array
- XY Plotter

Instrument Navigator A controlbar that displays a tree with all the instruments of the active layout and all the variables that are connected to them. The Instrument Navigator's main function is easy selection of instruments in complex layouts.

Instrument script A Python script used to extend the functionality of an instrument.

Instrument Selector A controlbar that provides access to ControlDesk's instruments. The instruments can be placed on a layout via double-click or drag & drop.

Interface description data (IF_DATA) An information structure, mostly provided by an A2L file, describing the type, features and configuration of an implemented ECU interface.

Internal Interpreter ControlDesk's built-in programming interface for editing, running and importing Python scripts. It contains an Interpreter controlbar where the user can enter Python commands interactively and which displays output and error messages of Python commands.

Interpreter controlbar A controlbar that can be used to execute line-based commands. It is used by the Internal Interpreter to print out Python standard error messages and standard output during the execution or import of Python scripts.

Invisible Switch An instrument for defining an area that is sensitive to mouse operations.

IOCNET IOCNET (I/O carrier network) is a dSPACE-specific high-speed serial communication bus that connects all the real-time hardware in a SCALEXIO system. IOCNET can also be used to build a multiprocessor system that consists of multiple SCALEXIO processor hardware components.

K

Knob An instrument for displaying and setting the value of the connected variable by means of a knob on a circular scale.

L

Label list A list of user-defined variables that can be used for saving connected variables, etc.

Layout A window with instrument s connected to variables of one or more simulation models.

Layout Navigator A controlbar that displays all opened layout s. It can be used for switching between layouts.

Layout script A Python script used to extend the functionality of a layout .

Leading raster The measurement raster that specifies the trigger settings for the Time Plotter display. The leading raster determines the time range that is visible in the plotter if a start and stop trigger is used for displaying the signals.

LIN Bus Monitoring device A device that monitors the data stream on a LIN bus connected to the ControlDesk PC.

The LIN Bus Monitoring device works, for example, with PC-based LIN interfaces.

The device supports the following variable description file types:

- LDF
- FIBEX
- AUTOSAR system description (ARXML)

Load type The load type specifies the option to disturb a signal with or without load rejection.

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Logical link A representation of an ECU specified in the diagnostics database. A logical link contains information on the ECU itself, and all the information required for accessing it, such as the diagnostic protocol used for

communication between the ECU and ControlDesk. Each logical link is represented by a unique short name in the ODX database.

Look-up table A look-up table maps one or more input values to one output value. You have to differentiate between the following look-up table types:

- A 1-D look-up table maps one input value to one output value.
- A 2-D look-up table maps two input values to one output value.
- An n-D look-up table maps multidimensional table data with 3 or more input values to one output value.

Look-up table is a generic term for curves and maps.

M

Main variable A scalar variable that is visualized in an instrument that can be used to change parameter values. In addition to the main variable, additional write variables can also be connected to (but not visualized in) the same instrument. When you change the value of the main variable in an instrument, the changed value is also applied to all the additional write variables connected to that instrument.

Map A parameter that consists of

- A 1-dimensional array containing the axis points for the x-axis. This array can also be specified by a reference to a common axis.
- A 1-dimensional array containing the axis points for the y-axis. This array can also be specified by a reference to a common axis.
- A 2–dimensional array containing data points. The map assigns one data point of the array to each pair of x-axis and y-axis points.

Maps are represented by the  symbol.

Map file A file that contains symbols (symbolic names) and their physical addresses. It is generated during a build process of an ECU application.

Map instrument A customized Browser instrument. It uses an instrument script to open a web map and connect positioning data to the map. The Map instrument offers prepared connection nodes to connect variables with GNSS data.

Measurement Viewing and analyzing the time traces of variables, for example, to observe the effects of ECU parameter changes.

ControlDesk provides various instruments for measuring variables.

Measurement (variable type) A scalar variable that can be measured, including individual elements of a measurement array.

Measurement variables are represented by the  symbol.

Measurement array A 1-, 2-, or 3-dimensional array of measurement variables. In variable lists, ControlDesk displays entries for the measurement array itself and for each array element.

Measurement arrays are represented by the  symbol.

Measurement buffer A ring buffer that buffers measurement data at the start of a measurement. The measurement buffer size determines the amount of data that can be buffered. Earlier values are overwritten by later values when the buffer capacity is exceeded (buffer overflow).

Measurement Configuration A controlbar that allows you to configure measurement, recording and data logging.

Measurement Data API Application programming interface for accessing measurement data. The API lets the user access measurement data without having to use ControlDesk.

Measurement Data Pool A controlbar that provides access to measurement data recorded in measurement data files.

Measurement raster Specification of how often a value of a variable is updated during a measurement. A measurement raster can be derived from a measurement service.

Measurement service The generic term for the following services:

- CCP service
- DsDAQ service
- Host service
- XCP service

Measurement signal list A list containing the variables to be included in subsequent measurements and recording. The list is global for all platforms/devices of the current experiment. The measurement signal list is available in the configuration area of the **Measurement Configuration** controlbar.

Measurement variable Any variable type that can be measured but not calibrated.

Measuring/recording A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- Online calibration is possible. Parameter values can be changed directly on the platform/device hardware.
- A measurement (or recording) is running.
- Platform/device configuration is not possible.

The 'measuring' / 'recording' platform/device state is indicated by the icon.

Memory page An area of a calibration memory. Each page contains a complete set of parameters of the platform/device hardware, but only one of the pages is "visible" to the microcontroller of the ECU or the real-time processor (RTP) of the platform hardware at a time.

ControlDesk supports platform/device hardware with up to two memory pages. These are usually the working page and the reference page. The parameter values on the two memory pages usually are different. ControlDesk lets you switch from one page to the other, so that when parameters are changed on one page, the changes can be made available to the ECU or prototyping hardware via a single page switch.

Messages controlbar A controlbar displaying a history of all error and warning messages that occur during work with ControlDesk.

MicroAutoBox III platform A platform that provides access to a MicroAutoBox III connected to the host PC for function prototyping purposes such as Bypassing.

MicroAutoBox platform A platform that provides access to a MicroAutoBox II connected to the host PC for function prototyping purposes such as bypassing.

Mirrored memory A memory area created by ControlDesk on the host PC that mirrors the contents of the available memory pages of calibration and prototyping hardware. For hardware with two memory pages, the mirrored memory is divided into a reference and a working page, each of them containing a complete set of parameters. When a calibration or prototyping platform/device is added to an experiment, ControlDesk initially fills the available memory pages of the mirrored memory with the contents of the ECU Image file (initial filling for calibration devices) or with the contents of the SDF file (initial filling for platforms).

- Mirrored memory for offline calibration

  Parameter values can even be changed offline. Changes to parameter values that are made offline affect only the mirrored memory.
- Offline-to-online transition for online calibration

  For online calibration, an offline-to-online transition must be performed. During the transition, ControlDesk compares the memory pages of the hardware of each platform/device with the corresponding pages of the mirrored memory. If the pages differ, the user has to equalize them by uploading them from the hardware to the host PC, or downloading them from the host PC to the hardware.
- Mirrored memory for online calibration

  When ControlDesk is in the online mode, parameter value changes become effective synchronously on the memory pages of the hardware and in the mirrored memory. In other words, parameter values on the hardware and on the host PC are always the same while you are performing online calibration.

Modular system A dSPACE processor board and one or more I/O boards connected to it.

Multi-capture history The storage of all the captures acquired during a triggered measurement. The amount of stored data depends on the measurement buffer.

Multi-pin error A feature of the SCALEXIO concept for electrical error simulation that lets you simulate a short circuit between three or more signal

channels and/or bus channels. The channels can be located on the same or different boards or I/O units. You can simulate a short circuit between:

- Channels of the same signal category (e.g., four signal generation channels)
- Channels of different signal categories (e.g., three signal generation channels and two signal measurement channels)
- Signal channels and bus channels (e.g., two signal generation channels, one signal measurement channel, and one bus channel)

Multiple electrical errors A feature of the SCALEXIO concept for electrical error simulation that lets you switch electrical errors at the same time or in succession. For example, you can simulate an open circuit for one channel and a short circuit for another channel at the same time, without deactivating the first error.

Multiprocessor System platform A platform that provides access to:

- A multicore application running on a multicore DS1006 board
- A multiprocessor application on a multiprocessor system consisting of two or more DS1006 processor boards interconnected via Gigalink.

ControlDesk handles a multiprocessor/multicore system as a unit and uses one system description file (SDF file) to load the applications to all the processor boards/cores in the system.

Multistate Display An instrument for displaying the value of a variable as an LED state and/or as a message text.

Multistate LED A value cell type of the Variable Array for displaying the value of a variable as an LED state.

Multiswitch An instrument for changing variable values by clicking sensitive areas in the instrument and for visualizing different states depending on the current value of the connected variable.

N

Numeric Input An instrument (or a value cell type of the Variable Array) for displaying and setting the value of the connected variable numerically.

O

Observing variables Reading variable values cyclically from the dSPACE real-time hardware and displaying their current values in ControlDesk, even if no measurement is running. Variable observation is performed without using a measurement buffer, and no value history is kept.

For platforms that support variable observation, variable observation is available for parameters and measurement variables that are visualized in single-shot instruments (all instruments except for a plotter). If you visualize a variable in a single-shot instrument, the variable is not added to the measurement signal list. Visualizing a parameter or measurement variable in a plotter automatically adds the variable to the measurement signal list.

ControlDesk starts observing variables if one of the following conditions is true:

- Online Calibration is started for the platform.

  All the parameters and measurement variables that are visualized in single-shot instruments are observed.
- Measurement is started for the platform.

  All the visualized parameters and measurement variables that are not activated for measurement in the measurement signal list are observed. Data of the activated parameters and measurement variables is acquired using measurement rasters.

ODX database Abbreviation of Open Diagnostic Data Exchange, a diagnostics database that is the central ECU description for working with an ECU Diagnostics device in ControlDesk. The ODX database contains all the information required to perform diagnostic communication between ControlDesk and a specific ECU or set of ECUs in a vehicle network. ControlDesk expects the database to be compliant with ASAM MCD-2 D (ODX).

Offline State in which the parameter values of platform/device hardware in the current experiment cannot be changed. This applies regardless of whether or not the host PC is physically connected to the hardware.

The mirrored memory allows parameter values to be changed even offline.

Offline simulation A PC-based simulation in which the simulator is not connected to a physical system and is thus independent of the real time.

Offline simulation application (OSA) An offline simulation application (OSA) file is an executable file for VEOS. After the build process with a tool such as the VEOS Player, the OSA file can be downloaded to VEOS.

An OSA contains one or more VPUs, such as V-ECUs and/or environment VPUs.

On/Off Button An instrument (or a value cell type of the Variable Array) for setting the value of the connected parameter to a predefined value when the button is pressed (On value) and released (Off value).

Online calibration started A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- Online calibration is possible. Parameter values can be changed directly on the platform/device hardware.
- Platform/device configuration is not possible.

Before starting online calibration, ControlDesk lets you compare the memory pages on the platform/device hardware with the corresponding pages of the mirrored memory. If the parameter values on the pages differ, they must be

equalized by uploading the values from the hardware to ControlDesk, or downloading the values from ControlDesk to the hardware. However, a page cannot be downloaded if it is read-only.

The 'online calibration started' platform/device state is indicated by the symbol.

Operation signal A signal which represents the result of an arithmetical operation (such as addition or multiplication) between two other signals.

Operator mode A working mode of ControlDesk in which only a subset of the ControlDesk functionality is provided. You can work with existing experiments but not modify them, which protects them from unintentional changes.

Output parameter A parameter or writable measurement whose memory address is used to write the computed value of a calculated variable to.

P

Parameter Any variable type that can be calibrated.

Parameter (variable type) A scalar parameter, as well as the individual elements of a value block.

Scalar parameters are represented by the P symbol.

Parameter limits Limits within which parameters can be changed. Parameters have hard and weak limits.

- Hard limits

  Hard limits designate the value range of a parameter that you *cannot* cross during calibration.

  The hard limits of a parameter originate from the corresponding variable description and cannot be edited in ControlDesk.

- Weak limits

  Weak limits designate the value range of a parameter that you *should not* cross during calibration. When you cross the value range defined by the weak limits, ControlDesk warns you.

  In ControlDesk, you can edit the weak limits of a parameter within the value range given by the parameter's hard limits.

PHS (Peripheral High Speed) bus A dSPACE-specific bus for communication between a processor board and the I/O boards in a modular system. It allows direct I/O operations between the processor board (bus master) and I/O boards (bus slaves).

PHS-bus-based system A modular dSPACE system consisting of a processor board such as the DS1006 Processor Board and I/O boards. They communicate with each other via the PHS (Peripheral High Speed) bus.

Pitch variable A variable connected to the pitch scale of an Artificial Horizon.

Platform A software component representing a simulator where a simulation application is computed in real-time (on dSPACE real-time hardware) or in non-real-time (on VEOS).

ControlDesk provides the following platforms:

- DS1006 Processor Board platform
- DS1007 PPC Processor Board platform
- DS1104 R&D Controller Board platform
- DS1202 MicroLabBox platform
- MicroAutoBox platform
- MicroAutoBox III platform
- Multiprocessor System platform
- SCALEXIO platform
- VEOS platform
- XIL API MAPort platform

Each platform usually has a variable description that specifies its variables.

Platform trigger A trigger that is available for a platform and that is evaluated on the related dSPACE real-time hardware or VEOS.

Platforms/Devices controlbar A controlbar that provides functions to handle devices, platforms, and the applications assigned to the platforms.

Plotter instrument ControlDesk offers three plotter instruments with different main purposes:

- The Index Plotter displays signals in relation to events.
- The Time Plotter displays signals in relation to measurement time.
- The XY Plotter displays signals in relation to other signals.

Port configuration To interface the failure simulation hardware, an EESPort needs the hardware-dependent *port configuration file* (PORTCONFIG file). The file's contents must fit the connected HIL simulator architecture and its failure simulation hardware.

Postprocessing The handling of measured and recorded data by the following actions:

- Displaying measured or recorded data
- Zooming into measured or recorded signals with a plotter
- Displaying the values of measurement variables and parameters as they were at any specific point in time

Processor board A board that computes real-time applications. It has an operating system that controls all calculations and communication to other boards.

Project A container for collecting and managing the information and files required for experiment/calibration/modification tasks in a number of experiments. A project collects the experiments and manages their common data.

Project controlbar A controlbar that provides access to projects and experiments and all the files they contain.

Project root directory The directory on your file system to which ControlDesk saves all the experiments and documents of a project. Every project is associated with a project root directory, and several projects can use the same project root directory. The user can group projects by specifying several project root directories.

ControlDesk uses the Documents folder as the default project root directory unless a different one is specified.

Properties controlbar A controlbar providing access to the properties of, for example, platforms/devices, layouts/instruments, and measurement/recording configurations.

Proposed calibration A calibration mode in which the parameter value changes that the user makes do not become effective on the hardware until they are applied. This allows several parameter changes to be written to the hardware together. Being in proposed calibration mode is like being in the offline calibration mode temporarily.

Push Button An instrument (or a value cell type of the Variable Array) for setting the value of the connected parameter by push buttons.

Python Editor An editor for opening and editing PY files.

Q

Quick start measurement A type of measurement in which all the ECU variables configured for measurement are measured and recorded, starting with the first execution of an ECU task. ControlDesk supports quick start measurements on ECUs with DCI-GSI2, CCP, and XCP (except for XCP on Ethernet with the TCP transmission protocol).

Quick start measurement can be used to perform cold start measurements. Cold start means that the vehicle and/or the engine are cooled down to the temperature of the environment and then started. One reason for performing cold start measurements is to observe the behavior of an engine during the warm-up phase.

R

Radio Button An instrument for displaying and setting the value of the connected parameter by radio buttons.

Real-time application An application that can be executed in real time on dSPACE real-time hardware. A real-time application can be built from a Simulink model containing RTI blocks, for example.

Record layout A record layout is used to specify a data type and define the order of the data in the memory of the target system (ECU, for example). For scalar data types, a record layout allows you to add an address mode (direct or indirect). For structured (aggregated) data types, the record layout specifies all the structure elements and the order they appear in.

The `RECORD_LAYOUT` keyword in an A2L file is used to specify the various record layouts of the data types in the memory. The structural setup of the various data types must be described in such a way that a standard application system will be able to process all data types (reading, writing, operating point display etc.).

Record layout component A component of a record layout. A structured record layout consists of several components according to the ASAP2 specification. For example, the AXIS_PTS_X component specifies the x-axis points, and the FNC_VALUES component describes the function values of a map or a curve.

Recorder An object in the Measurement Configuration controlbar that specifies and executes the recording of variables according to a specific measurement configuration.

Recorder signal list A list that contains the variables to be included in subsequent recordings.

Recording Saving the time traces of variables to a file. Both measurement variables and parameters can be recorded. Recorded data can be postprocessed directly in ControlDesk.

A recording can be started and stopped immediately or via a trigger:

- Immediate recording

  The recording is started and stopped without delay, without having to meet a trigger condition.
- Triggered recording

  The recording is not started or stopped until certain trigger conditions are met. These conditions can be defined and edited in ControlDesk.

Reduction data Additional content in an MF4 file that allows for visualizing the MF4 file data depending on the visualization resolution. Reduction data therefore improves the performance of the visualization and postprocessing of measurement data.

Reference data set A read-only data set assigned to the reference page of a device that has two memory pages. There can be only one reference data set for each device. The reference data set is read-only.

Reference page Memory area containing the parameters of an ECU. The reference page contains the read-only reference data set.

Note

Some platforms/devices provide only a working page. You cannot switch to a reference page in this case.

Resynchronization Mechanism to periodically synchronize the drifting timers of the platform/device hardware ControlDesk is connected to. Resynchronization means adjustment to a common time base.

Roll variable A variable connected to the roll scale of an Artificial Horizon.

S

Sample count trigger A trigger that specifies the number of samples in a data capture.

A sample count trigger can be used as a stop trigger.

SCALEXIO platform A platform that provides access to a single-core, multicore or multiprocessor SCALEXIO system connected to the host PC for HIL simulation and function prototyping purposes.

SCALEXIO system A dSPACE hardware-in-the-loop (HIL) system consisting of at least one processing hardware component, I/O boards, and I/O units. They communicate with each other via the IOCNET. In a SCALEXIO system, two types of processing hardware can be used, a DS6001 Processor Board or a real-time industry PC as the SCALEXIO Processing Unit. The SCALEXIO system simulates the environment to test an ECU. It provides the sensor signals for the ECU, measures the signals of the ECU, and provides the power (battery voltage) for the ECU and a bus interface for restbus simulation.

SDF file The system description file that describes the files to be loaded to the individual processing units of a simulation platform. It also contains the variable description of the relevant simulation application.

The SDF file is generated automatically when the TRC file is built.

Segment The minimum part a segment signal can consist of.

There are different kinds of segments to be used in segment signals:

- Segments to form synthetic signal shapes (sine, sawtooth, ramp, etc.)
- Segments to perform arithmetical operations (addition, multiplication) with other segments
- Segments to represent numerical signal data (measured data)

Segment signal A signal consisting of one or more segments.

Selection Box An instrument for selecting a text-value entry and setting the respective numerical value for the connected variable.

Signal

- Representation of a variable measured in a specific measurement raster.
- Generic term for segment signals and operation signals.

  A signal is part of a signal description set which can be displayed and edited in the working area.

Signal description set A group of one or more signals.

A signal description set and its signals can be edited in the working area by means of the Signal Editor. Each signal description set is stored as an STZ file either in the **Signal Description Sets** folder or in the **Signal Generators** folder.

Signal Editor A software component to create, configure, display, and manage signals in signal description sets.

Signal file A file that contains the wiring information of a simulator and that is part of the standard dSPACE documentation of dSPACE Simulator Full-Size. Normally, dSPACE generates this file when designing the simulator. Before using a failure simulation system, users can adapt the signal file to their needs.

Signal generator An STZ file containing a signal description set and optional information about the signal mapping, the description of variables, and the real-time platform.

The file is located in the **Signal Generators** folder and used to generate, download, and control Real-Time Testing sequences, which are executed on the real-time platform to stimulate model variables in real time.

Signal Mapping A controlbar of the Signal Editor to map model variables to signals and variable aliases of a signal generator.

Signal Selector A controlbar of the Signal Editor. The Signal Selector provides signals and segments for arranging and configuring signal description sets in the working area.

SIL testing Abbreviation of *software-in-the-loop testing*.

Simulation and testing of individual software functions, complete virtual ECUs (V-ECUs), or even V-ECU networks on a local PC or highly parallel in the cloud independently of real-time constraints and real hardware.

Simulation application The generic term for offline simulation application (OSA) and real-time application.

Simulation system A description of the composition of V-ECU models, environment models, real ECUs, and their interconnections required for simulating the behavior of a system. A simulation system is the basis for the generation of a simulation application for a given simulator platform.

Simulation time group Group of platforms/devices in an experiment whose simulation times are synchronized with each other. If resynchronization is enabled, ControlDesk synchronizes a simulation time group as a whole, not the single members of the group individually.

Simulator A system that imitates the characteristics or behaviors of a selected physical or abstract system.

Single-processor system A system that is based on one dSPACE processor or controller board.

Single-shot instrument An instrument that displays an instantaneous value of a connected variable without keeping a value history. In ControlDesk, all instruments except for a plotter are single-shot instruments. For platforms that support the variable observer functionality, you can use single-shot instruments to observe variables.

Slave application An application assigned to the slave DSP of a controller or I/O board. It is usually loaded and started together with the real-time application running on the corresponding main board.

Slave DSP A DSP subsystem installed on a controller or I/O board. Its slave application can be loaded together with the real-time application or separately.

Slider An instrument (or a value cell type of the Variable Array) for displaying and setting the value of the connected variable by means of a slide.

Sound Controller An instrument for generating sounds to be played.

Standard axis An axis with data points that are deposited in the ECU memory. Unlike a common axis, a standard axis is specified within a curve or map. The parameters of a standard axis can be calibrated, which affects only the related curve or map.

Start trigger A trigger that is used, for example, to start a measurement raster. A platform trigger can be used as a start trigger.

Static Text An instrument for displaying explanations or inscriptions on the layout.

Steering Controller An instrument for changing variable values using a game controller device such as a joystick or a steering wheel.

Stimulation Writing signals to variables in real-time models during a simulation run.

Stop trigger A trigger that is used, for example, to stop a measurement raster.

String A text variable in ASCII format.

Strings are represented by the  symbol.

Struct A variable with the struct data type. A struct contains a structured list of variables that can have various data types. In ControlDesk, a struct variable can contain either parameters and value blocks or measurement variables and measurement arrays. ControlDesk supports nested structs, i.e., structs that contain further structs and struct arrays as elements.

Structs are represented by the  symbol.

Struct array An array of homogeneous struct variables.

Struct arrays are represented by the  symbol.

STZ file A ZIP file containing signal descriptions in the STI format. The STZ file can also contain additional MAT files to describe numerical signal data.

Sub data set A data set that does not contain the complete set of the parameters of a platform/device.

Symbol A symbolic name of a physical address in a MAP file. A symbol can be associated to a variable in the Variable Editor, for example, to support an address updates.

System variable A type of variable that represents internal variables of the device or platform hardware and that can be used as measurement signals in ControlDesk to give feedback on the status of the related device or platform hardware. For example, an ECU's power supply status or the simulation state of a dSPACE board can be visualized via system variables.

T

Table Editor An instrument for displaying and setting values of a connected curve, map, value block, or axis in a 2-D, 3-D, and grid view. The Table Editor can also display the values of a measurement array.

The Table Editor can be used for the following variable types:

- Common axis ()
- Curve ()
- Map ()
- Measurement array ()
- Value block ()

Time cursor A cursor which is visible at the same time position in the following instruments:

- In all Time Plotters
- In all XY Plotters
- In all bus monitoring lists

You can use the time cursor to view signal values at a specific point in time. If you move the time cursor, all measured signals and the respective parameters are

updated. Instruments and bus monitoring lists display the values that are available at the selected time position.

Time Plotter A plotter instrument for displaying signals that are measured in a time-based raster (time plots).

Topology A description of the processor boards belonging to a multiprocessor system and their interconnections via Gigalinks. The topology also contains information on which Gigalink port of each processor board is connected to the Gigalink ports of other processor boards in the multiprocessor system.

Topology information is contained in the real-time application (PPC/x86/RTA) files of the multiprocessor system's processor boards.

TRC file A variable description file with information on the variables available in an environment model running on a dSPACE platform.

Trigger A condition for executing an action such as starting and stopping a measurement raster or a recorder.

The generic term for the following trigger types:

- Duration trigger
- Platform trigger
- Sample count trigger

Trigger condition A formula that specifies the condition of a trigger mathematically.

Triggered measurement The measurement of a measurement raster started by a platform trigger. The data flow between the dSPACE real-time hardware or VEOS and the host PC is not continuous.

U

Unassigned data set A data set that is assigned neither to the working page nor to the reference page of a platform/device. An unassigned data set can be defined as the new working or reference data set. It then replaces the "old" working or reference data set and is written to the corresponding memory page, if one is available on the platform/device.

Unplugged A platform/device state defined by the following characteristics:

- The logical connection between ControlDesk and the hardware was interrupted, for example, because the ignition was turned off or the ControlDesk PC and the hardware were disconnected.
- Before the state of a platform/device changes to 'unplugged', the platform/device was in one of the following states:
  - 'Connected'
  - 'Online calibration started'
  - 'Measuring' / 'Recording'

Tip

A device for which the connection between ControlDesk and the device hardware currently is interrupted is also set to the 'unplugged' state when you start online calibration if both the following conditions are fulfilled:

- The device's **Start unplugged** property is enabled.
- The **Start online calibration behavior** property is set to 'Ignore differences'.

This is possible for CCP and XCP devices. For details on the two properties listed above, refer to General Settings Properties (ControlDesk Platform Management 🕮).

- If the Automatic Reconnect feature is enabled for a platform/device and if the platform/device is in the 'unplugged' state, ControlDesk periodically tries to re-establish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'unplugged' platform/device state is indicated by the ⚠ icon.

Untriggered measurement The measurement of a measurement raster not started by a platform trigger. The data flow between the dSPACE real-time hardware or VEOS and the host PC is continuous.

User function An external function or program that is added to the ControlDesk user interface for quick and easy access during work with ControlDesk.

User Functions Output A controlbar that provides access to the output of external tools added to the **Automation** ribbon.

V

Value block A parameter that consists of a 1- or 2-dimensional array of scalar parameters.

In variable lists, ControlDesk displays entries for the value block itself and for each array element.

Value blocks are represented by the 🕮 symbol.

Value conversion The conversion of the original *source values* of variables of an application running on an ECU or dSPACE real-time hardware into the corresponding scaled *converted values*.

Variable Any parameter or measurement variable defined in a variable description. ControlDesk provides various instruments to visualize variables.

Variable alias An alias name that lets the user control the property of a segment by a model parameter of a real-time application.

Variable Array An instrument for calibrating parameters and displaying measurement variable values.

The Variable Array can be used for the following variable types:

- Measurement ( )
- Measurement array ( )
- String ( )
- Struct ( )
- Struct array ( )
- Value ( )
- Value block ( )

Variable connection The connection of a variable to an instrument. Via the variable connection, data is exchanged between a variable and the instrument used to measure or calibrate the variable. In other words, variable connections are required to visualize variables in instrument.

Variable description A file describing the variables in a simulation application, which are available for measurement, calibration, and stimulation.

Variable Editor A tool for viewing, editing, and creating variable descriptions in the ASAM MCD-2MC (A2L) file format. The Variable Editor allows you to create A2L files from scratch, or to import existing A2L files for modification.

Variable Filter A variable filter contains the filter configuration of a combined filter, which is used to filter the variable list in the **Variables** controlbar using a combination of filter conditions.

Variables controlbar A controlbar that provides access to the variables of the currently open experiment.

V-ECU Abbreviation of *virtual ECU*.

ECU software that can be executed in a software-in-the-loop (SIL) testing environment such as a local PC or highly parallel in the cloud independently of real-time constraints and real ECU hardware.

Vehicle information The ODX database can contain information for one or more vehicles. Vehicle information data is used for vehicle identification purposes and for access to vehicles. It references the access paths (logical links) to the ECUs.

VEOS A simulator which is part of the PC and allows the user to run an offline simulation application (OSA) without relation to real time.

VEOS Player is the graphical user interface for VEOS.

VEOS platform A platform that configures and controls the offline simulation application (OSA) running in VEOS and that also provides access to the application's environment VPU.

VEOS Player An application running on the host PC for editing, configuring and controlling an offline simulation application (OSA) running in VEOS.

Verbal conversion A conversion in which a conversion table is used to specify the computation of numerical values into strings. The verbal conversion table is used when you switch the value representation from source to converted mode and vice versa.

Verbal conversion range A conversion in which a conversion table is used to specify the computation of a range of numerical values into strings. The verbal conversion range table is used when you switch the value representation from source to converted mode and vice versa.

View set A named configuration of the controlbars of ControlDesk. A view set has a default state and a current state that can differ from the default state. The configuration includes the geometry, visibility, and docking or floating state of controlbars.

Visualization The representation of variables in instruments:

- Measurement variables are visualized in instruments to view and analyze their time traces.
- Calibration parameters are visualized in instruments to change their values.

VPU Abbreviation of *virtual processing unit*. A VPU is part of an offline simulation application in VEOS. Each VPU runs in a separate process of the PC.

VPU is also the generic term for:

- V-ECUs
- Environment VPUs
- Controller VPUs
- Bus VPUs

W

Working data set The data set currently residing in the memory of a platform/device hardware. There can be only one working data set for each calibration platform/device. The working data set is read/write.

Working page Memory area containing the parameters of an ECU or prototyping hardware (memory page). The working page contains the read/write working data set.

If the platform/device also provides a reference page, ControlDesk lets you switch between both pages.

Writable measurement A scalar variable that can be measured and calibrated.

X

XCP Abbreviation of *Universal Measurement and Calibration Protocol*. A protocol that is implemented on electronic control units (ECUs) and provides access to ECUs with measurement and calibration systems (MCS) such as ControlDesk.

XCP is based on the *master-slave principle*:

- The ECU is the slave.
- The measurement and calibration system is the master.

The "X" stands for the physical layers for communication between the ECU and the MCS, such as CAN (Controller Area Network) and Ethernet.

The basic features of XCP are:

- ECU parameter calibration (CAL)
- Synchronous data acquisition (DAQ)
- Synchronous data stimulation (STIM), i.e., for bypassing
- ECU flash programming (PGM)

The XCP protocol was developed by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.). For the protocol specification, refer to http://www.asam.net.

The following ControlDesk devices support ECUs with an integrated XCP service:

- XCP on CAN device
- XCP on Ethernet device

XCP on CAN device A device that provides access to an ECU with XCP connected to the ControlDesk PC via CAN. Using the XCP on CAN device, you can access the ECU for measurement and calibration purposes via XCP (*Universal Measurement and Calibration Protocol*).

XCP on Ethernet device A device that provides access to an ECU or V-ECU with XCP connected to the ControlDesk PC via Ethernet. The XCP on Ethernet device provides access to the ECU/V-ECU via XCP (*Universal Measurement and Calibration Protocol*) for measurement and calibration purposes.

XIL API EESPort Electrical Error Simulation port (EESPort)

XIL API MAPort platform A platform that provides access to a simulation platform via the ASAM XIL API implementation that is installed on your host PC.

XY Plotter A plotter instrument for displaying signals as functions of other signals.