FlexRay Configuration Tool

Guide

For dSPACE FlexRay Configuration Package 4.7

Release 2021-A – May 2021

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 Guide

Contents

This guide gives you an overview of FlexRay and introduces you to the main features of the FlexRay Configuration Tool, with basic information and instructions on working with the tool. You can view FlexRay network information, create configurations, and generate code and Simulink configuration data for modeling and simulating with the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset.

Required knowledge

This guide gives no information on the FlexRay network. It is assumed that you have a good knowledge of FlexRay network and its basics.

Knowledge in handling the host PC and the Microsoft Windows operating system is assumed.

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.

Symbol	Description
🕮	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.

Basics of the FlexRay Configuration Tool

Where to go from here

Information in this section

Introduction to FlexRay Networks

Where to go from here

Information in this section

FlexRay Basics

Basics of FlexRay

FlexRay is a time-triggered communication system for high-performance in-vehicle control applications. The development of such applications needs to take into account the specifics of a time-triggered, deterministic, and fault-tolerant communication protocol. Tasks are executed and the related messages are sent according to predefined schedules. In FlexRay, activities are aligned to a global time base, whereas in non-deterministic communication protocols such as CAN, they are mainly event-triggered at run time.

Related topics

Basics

Features of FlexRay Networks........ 10

Examples

Example of a FlexRay Communication........ 11

Features of FlexRay Networks

Introduction

Below is a brief overview of the main features of FlexRay networks.

Data transmission

The FlexRay protocol supports synchronous and asynchronous data transmission. Because the FlexRay controller works on a time trigger basis, the time intervals at which data is transmitted are deterministic. In contrast, message handling in CAN networks is event-triggered and therefore unpredictable.

Data rate

FlexRay is initially targeted for a data rate of approximately 10 Mbit/sec. This is higher than the data rate of CAN networks.

Fault tolerance

FlexRay offers a deterministic and fault-tolerant communication layer with a guaranteed latency and jitter for an application.

Global time base

As a FlexRay bus is time-triggered, a global time base is absolutely essential. At a given physical time, the fault-tolerant clock synchronization algorithm ensures that the global times of different FlexRay controllers do not differ significantly, even if there are faults in the network.

Related topics

Basics

FlexRay Basics.. 10

Examples

Example of a FlexRay Communication.. 11

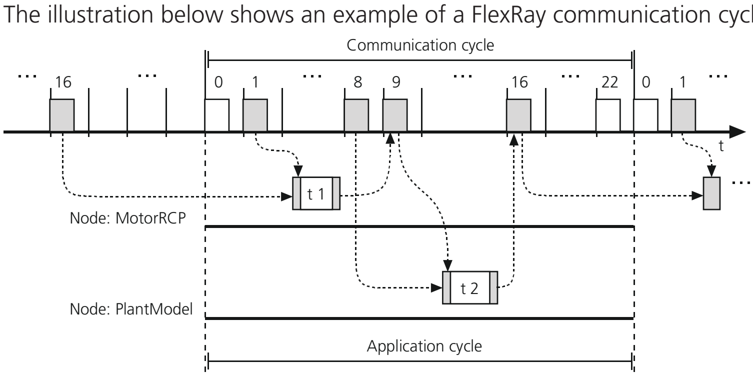
Example of a FlexRay Communication

Introduction

The following example illustrates how data is transmitted over a FlexRay bus.

FlexRay communication cycle

The illustration below shows an example of a FlexRay communication cycle.



In this example the communication cycle is divided into 22 time intervals called slots, in which data is transmitted. Two example nodes are shown; they process data that is transmitted via the FlexRay bus. The times when data is transmitted and when the application running on each node is started must be specified in a design process. The design process must include:

- The points in time when the tasks start
- The worst-case execution times (WCET) of the tasks
- The dependencies between the tasks (which node needs the data of another node)
- The time it takes to send or receive data

In the above example, the MotorRCP node reads the data transmitted in the 1st and 16th time slots (data of the 1st time slot was sent by a node not shown here, data of the 16th time slot was sent by the PlantModel node in the previous cycle).

It writes the data calculated in the 9th time slot to the bus. The application running on the MotorRCP node must therefore be started after the 1st time slot and finished before the 9th time slot.

The PlantModel node reads the data of the 8th and 9th time slots (data of the 8th time slot was sent by a node not shown here, the 9th time slot contains the data previously written by the MotorRCP node). It processes the data and writes the results in the 16th time slot to the bus. The application running on the PlantModel node must therefore be started after the 9th time slot and finished before the 16th time slot.

Related topics

Basics

Features of FlexRay Networks.......... 10
FlexRay Basics.......... 10

Introduction to the FlexRay Configuration Tool

Where to go from here

Information in this section

dSPACE FlexRay Configuration Package

Introduction

You get an overview of the dSPACE FlexRay Configuration Package.

dSPACE FlexRay Configuration Package

The dSPACE FlexRay Configuration Package contains the FlexRay Configuration Tool and the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset, respectively. You can use the package for hardware-in-the-loop simulations and prototyping purposes with FlexRay networks. You can simulate one FlexRay node or simulate a restbus (simulating all the other FlexRay nodes that communicate with one real FlexRay node).

FlexRay Configuration Tool With the FlexRay Configuration Tool, you can configure a dSPACE system as a simulation node in a FlexRay network. You can generate all the code files that are necessary for modeling with the RTI FlexRay Configuration Blockset / FlexRay Configuration Blockset and simulating with ControlDesk. For details, refer to Features of the FlexRay Configuration Tool on page 16.

RTI FlexRay Configuration Blockset The *RTI FlexRay Configuration Blockset* is used to connect the dSPACE real-time system to a FlexRay bus. The current blockset version supports a DS1006, DS1007, and MicroAutoBox II with FlexRay IP modules. Refer to Supported Platforms on page 14.

- For detailed information on modeling with the RTI FlexRay Configuration Blockset and simulating with real-time systems, refer to FlexRay Configuration Features 📖.
- For detailed information on the blockset, refer to RTI FlexRay Configuration Blockset Reference 📖.

FlexRay Configuration Blockset The *FlexRay Configuration Blockset* is used to connect the MicroAutoBox III or SCALEXIO system to a FlexRay bus. The current blockset version supports:

- SCALEXIO systems with a DS2671 Bus Board, DS2672 Bus Module, and/or DS6311 FlexRay Board
- MicroAutoBox III with a DS1514 FPGA Base Board with DS4340 FlexRay Interface Modules and/or DS1521 Bus Board

Refer to Supported Platforms on page 14.

- For detailed information on modeling with the FlexRay Configuration Blockset and simulating with a MicroAutoBox III or a SCALEXIO system, refer to Model Interface Package for Simulink - Modeling Guide 📖.
- For detailed information on the blockset, refer to FlexRay Configuration Blockset Reference 📖.

Related topics

Basics

Supported Platforms

Introduction

Using the FlexRay Configuration Tool and either the RTI FlexRay Configuration Blockset or the FlexRay Configuration Blockset allows you to configure a dSPACE system as a simulation node in a FlexRay network. The required blockset depends on the platform (modular system based on DS1006/DS1007 and MicroAutoBox II, or MicroAutoBox III and SCALEXIO system) you want to connect to the FlexRay system. Various platform types are supported.

Supported real-time systems

The FlexRay Configuration Tool and the RTI FlexRay Configuration Blockset support the following real-time platforms:

- Modular system based on a DS1006 with DS4501 IP Carrier Boards or DS4505 Interface Boards
- Modular system based on a DS1007 with DS4501 IP Carrier Boards or DS4505 Interface Boards

- MicroAutoBox II in the following variants:
  - 1401/1507
  - 1401/1511/1514
  - 1401/1513/1514

The following table shows the supported FlexRay modules which can be used on the real-time platforms.

Platform	Supported Modules
DS1006 with DS4501	▪ DS4340 FlexRay Interface Module ▪ DECOMSYS::FlexIM(MFR4200) (DECOMSYS::FlexIM2+ or DECOMSYS::FlexIM4+, V9.x) ▪ DECOMSYS::FlexIM(V11 IP-Core) (DECOMSYS::FlexIM4+, V11.x) ▪ DECOMSYS::FlexIM(E-Ray) (DECOMSYS::FlexIM2+ or DECOMSYS::FlexIM4+, E-Ray)
DS1006 with DS4505	▪ DS4340 FlexRay Interface Module
DS1007 with DS4501	▪ DS4340 FlexRay Interface Module ▪ DECOMSYS::FlexIM(MFR4200) (DECOMSYS::FlexIM2+ or DECOMSYS::FlexIM4+, V9.x) ▪ DECOMSYS::FlexIM(V11 IP-Core) (DECOMSYS::FlexIM4+, V11.x) ▪ DECOMSYS::FlexIM(E-Ray) (DECOMSYS::FlexIM2+ or DECOMSYS::FlexIM4+, E-Ray)
DS1007 with DS4505	▪ DS4340 FlexRay Interface Module
MicroAutoBox II	▪ DS4340 FlexRay Interface Module ▪ DECOMSYS::FlexIM(MFR4200) (DECOMSYS::FlexIM2+ or DECOMSYS::FlexIM4+, V9.x) ▪ DECOMSYS::FlexIM(V11 IP-Core) (DECOMSYS::FlexIM4+, V11.x) ▪ DECOMSYS::FlexIM(E-Ray) (DECOMSYS::FlexIM2+ or DECOMSYS::FlexIM4+, E-Ray)

Supported channel types

The FlexRay Configuration Tool and the FlexRay Configuration Blockset support several channel types of SCALEXIO systems and the MicroAutoBox III that allow communication via a FlexRay bus.

SCALEXIO The following SCALEXIO channel types are supported:

- Bus 1: Channel type on a DS2671 Bus Board.
- FlexRay 1: Channel type on a DS2672 Bus Module.
- FlexRay 2: Channel type on a DS6311 FlexRay Board.

The following table shows the supported FlexRay modules which can be used on the SCALEXIO systems:

Platform	Supported Modules
SCALEXIO	FlexRay IP module *Bosch ERay Core*

MicroAutoBox III The following MicroAutoBox III channel types are supported:

- FlexRay 3: Channel type on a DS4340 FlexRay Interface Module.
- FlexRay 4: Channel type on a DS1521 Bus Board.

The following table shows the supported FlexRay modules which can be used on the MicroAutoBox III systems:

Platform	Supported Modules
MicroAutoBox III with DS1514	DS4340 FlexRay Interface Module
MicroAutoBox III with DS1521	FlexRay IP module *Bosch ERay Core*

Related topics

Basics

Features of the FlexRay Configuration Tool

Introduction

The FlexRay Configuration Tool allows you to import and view FlexRay network information, create configurations, and generate code for modeling and simulation. The following introduction gives you an overview of the features.

Configuration projects

Configuration projects are the containers for your configuration. The FlexRay Configuration Tool manages your project and handles the FlexRay network elements. You can configure a dSPACE system as a simulation node in a FlexRay network. Refer to Handling Configuration Projects on page 21.

Visualization of FlexRay configuration

The FlexRay Configuration Tool provides different views to display the FlexRay network information and handle your configuration project. Refer to Graphical User Interface of the FlexRay Configuration Tool on page 18.

Creating configurations

The FlexRay Configuration Tool lets you create your network configuration.

Simulation configuration The FlexRay Configuration Tool has a Configuration view for configuring the simulation. Refer to Basics of Simulation on page 55.

Monitoring The FlexRay Configuration Tool allows you to monitor the FlexRay bus communication. Refer to Basics of Monitoring on page 113.

Specifying XCP master for usage with the RTI Bypass Blockset To perform function bypassing via XCP on FlexRay using the RTI Bypass Blockset, you must set up an XCP master service. The FlexRay Configuration Tool allows you to configure the communication layer of the selected XCP master. For information on the configuration steps in the FlexRay Configuration Tool necessary to perform function bypassing via XCP on FlexRay, refer to the RTI Bypass Blockset Application Note 📖.

Configuring hardware

The FlexRay Configuration Tool lets you select the hardware your real-time system is based on. For more details, refer to How to Configure Hardware on page 117.

Creating tasks

The FlexRay Configuration Tool has a Task view for managing communication and application tasks. Default task generation schemes for hardware-in-the-loop and restbus simulation, and prototyping purposes are available. You can also create tasks manually. For more details, refer to Creating Tasks on page 121.

Configuring controllers

The FlexRay Configuration Tool lets you view the settings of the FlexRay network controllers. For more details, refer to How to View the Controller Configuration on page 128.

Generating code

With the FlexRay Configuration Tool´s Code Generator, you can generate communication layer (Com) code, controller host interface (CHI) code, Simulink configuration data, a TRC file for ControlDesk, a configuration file for ControlDesk's Bus Navigator, and optionally a report file. For more details, refer to Generating Code on page 131.

Workflow

For detailed information on the configuration workflow, refer to FlexRay Configuration with the FlexRay Configuration Tool on page 22.

Related topics

Basics

How to Start the FlexRay Configuration Tool

Objective

You can start the FlexRay Configuration Tool via the Microsoft Windows Start menu or a shortcut on the desktop.

Method

To start the FlexRay Configuration Tool

1. Click the Windows Start button and select dSPACE RCP & HIL <x.y> – dSPACE FlexRay Configuration Tool <x.y> – dSPACE FlexRay Configuration Tool <x.y>.

   The FlexRay Configuration Tool opens.

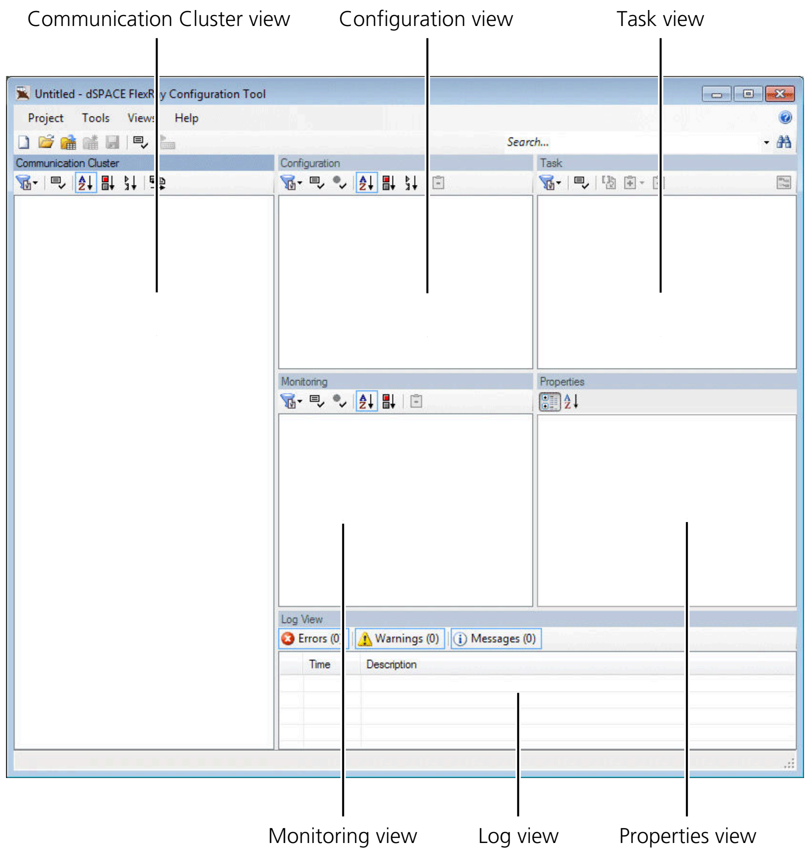
Graphical User Interface of the FlexRay Configuration Tool

Introduction

The graphical user interface offers the basic environment for all other elements and tool windows provided by the FlexRay Configuration Tool.

FlexRay Configuration Tool´s graphical user interface

The following illustration shows the user interface after you start the FlexRay Configuration Tool for the first time.



User interface elements

The user interface provides the following views:

Communication Cluster view The Communication Cluster view displays the content (for example, frames and signals) of the currently imported FIBEX and/or AUTOSAR system description files hierarchically and provides functions to handle the elements.

Configuration view The Configuration view lets you add elements from the Communication Cluster view for creating your configuration for simulation and manage your configuration.

Task view The Task view allows you to create and manage tasks.

Monitoring view The Monitoring view allows you to add elements from the Communication Cluster view for monitoring the bus communication.

Properties view The Properties view shows the attributes of an element or task selected in one of the other views.

Log view The Log view displays the FlexRay Configuration Tool´s log messages. The messages displayed in the view are also written to the `FlexRayConfigurationTool.log` file.

Handling Configuration Projects

Where to go from here

Information in this section

FlexRay Configuration with the FlexRay Configuration Tool

Introduction

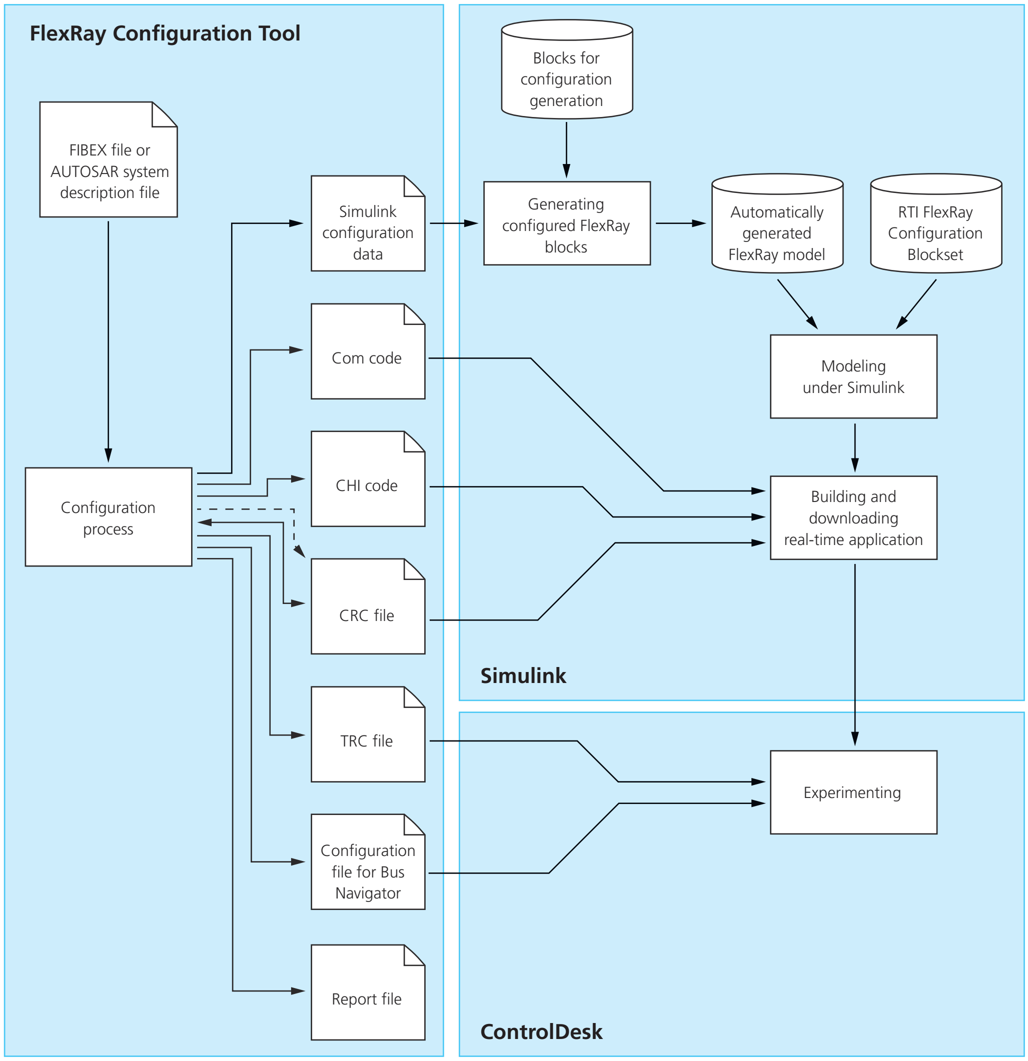
The FlexRay Configuration Tool supports you in viewing FlexRay network information, creating configurations, and generating code for modeling and execution of simulation.

Basics of FlexRay configuration

FlexRay configuration means configuring a dSPACE system as a simulation node in a FlexRay network. The FlexRay Configuration Tool supports you in creating a configuration for simulation based on FlexRay network information and generating Simulink configuration data and code. The following illustrations will help you to see the FlexRay configuration in context.

Workflow for setting up a FlexRay network for real-time simulation with a modular system based on DS1006 or DS1007, or a MicroAutoBox II The configuration process is performed in the FlexRay Configuration Tool. It creates Simulink configuration data, communication code, and CHI code for simulation purposes with the RTI FlexRay Configuration Blockset.

In MATLAB, you can generate a Simulink model containing FlexRay blocks which are configured according to generated Simulink configuration data from the FlexRay Configuration Tool. You can use the generated model together with the RTI FlexRay Configuration Blockset library to implement the functionality and I/O capabilities of FlexRay boards in Simulink models. You can then use the model to simulate time-triggered task execution in FlexRay communication systems.

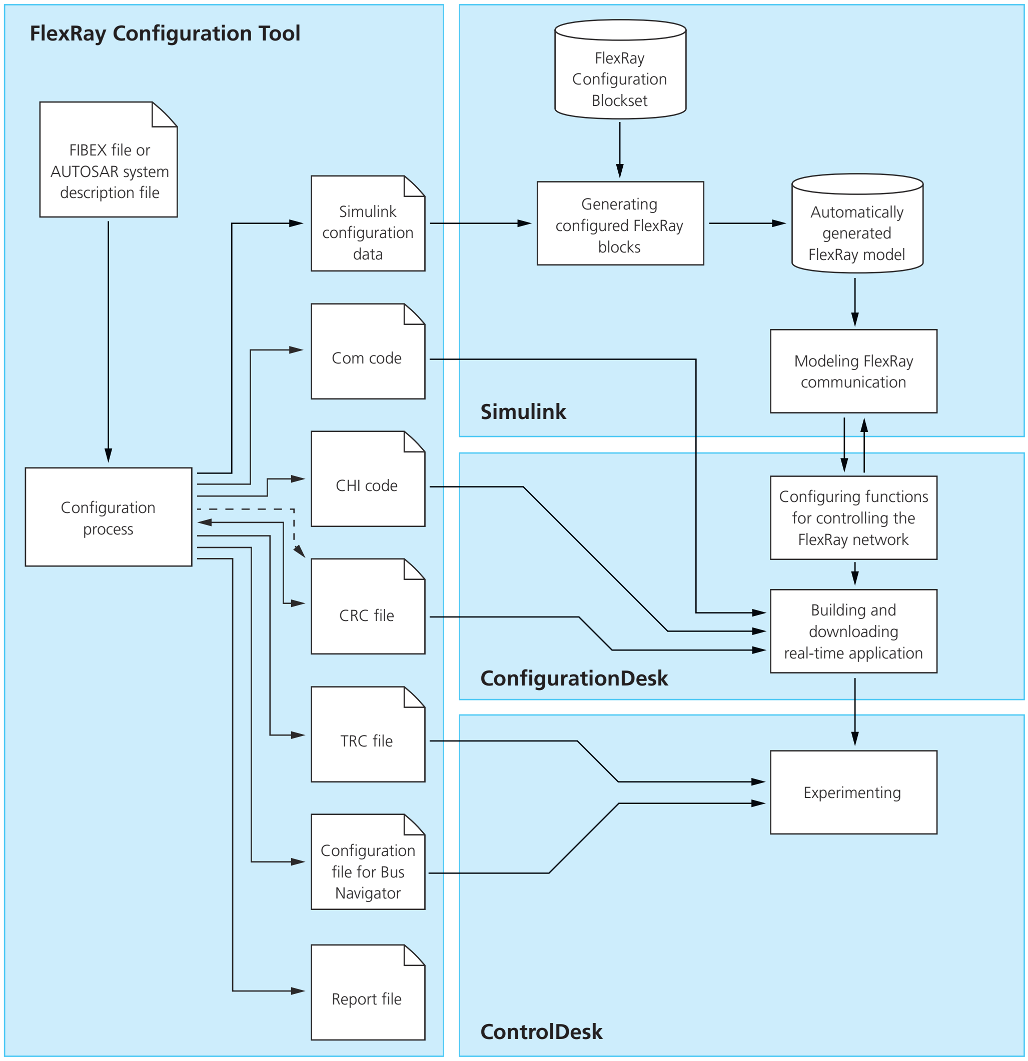


For further information, refer to Overview of the Workflow (FlexRay Configuration Features 📖).

Workflow for setting up a FlexRay network for real-time simulation with a MicroAutoBox III or a SCALEXIO system The configuration process is performed in the FlexRay Configuration Tool. It creates Simulink configuration data, communication code, and CHI code for simulation purposes with the FlexRay Configuration Blockset. The Simulink configuration data holds all the parameters for building the Simulink blocks needed to model the FlexRay communication.

Using the FlexRay Configuration Blockset in MATLAB, you generate a Simulink model based on the Simulink configuration data from the FlexRay Configuration Tool. You can use the generated model to model the FlexRay communication in Simulink models. In ConfigurationDesk, you use FlexRay function blocks to implement the I/O functionality to control the FlexRay network and assign hardware resources of the MicroAutoBox III or the SCALEXIO system to FlexRay controllers. You can synchronize the Simulink model with the data in

ConfigurationDesk. When the design of your FlexRay network is finished, you can use the model to simulate time-triggered task execution in FlexRay communication systems.



For further information, refer to Overview of the Workflow (Model Interface Package for Simulink - Modeling Guide 📖).

Configuration projects

Configuration projects are the containers for your FlexRay configuration. The FlexRay Configuration Tool provides a means of handling configuration projects.

Communication cluster files In the FlexRay Configuration Tool, you make your configuration based on FIBEX (Field Bus Exchange Format) or AUTOSAR System Templates. For further information, refer to Communication Cluster Files Usable for Configuration on page 26.

Handling communication cluster file elements The FlexRay Configuration Tool allows you to handle the communication cluster elements, display their parameters in the Properties view, and search for elements. Refer to Handling the Elements of a Communication Cluster File on page 48.

Managing projects You can save and open configuration projects. Refer to Managing Configuration Projects on page 47.

Workflow for FlexRay configuration within the FlexRay Configuration Tool

The workflow with the FlexRay Configuration Tool is as follows:

1. Import the FIBEX or AUTOSAR system description file including the FlexRay network description you want to make your configuration for. Refer to How to Import a Communication Cluster File on page 40.

   If another project is currently open, create a new project first. Refer to How to Create a New Project on page 40.
2. Select and configure signals for simulation and/or monitoring. Refer to:
   - Basics on Sending and Receiving Signals on page 71
   - How to Select and Configure Signals for Monitoring on page 114
3. Select the hardware your real-time system is based on. Refer to How to Configure Hardware on page 117.
4. Create tasks. You can create communication and application tasks manually, or let the FlexRay Configuration Tool create communication tasks automatically. Refer to How to Create Tasks on page 126.
5. Check the controller configuration. Refer to How to View the Controller Configuration on page 128.
6. Generate Simulink configuration data, Com code, CHI code, a TRC file, a configuration file for the Bus Navigator, and optionally a report file to use outside the FlexRay Configuration Tool. Refer to How to Generate Code on page 137.

Next steps

You can use the generated code files for modeling and simulation purposes with the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset. Refer to Using the Generated Code on page 139.

Related topics

Basics

Communication Cluster Files Usable for Configuration

Introduction

Most of the data describing the entire car communication is generated by software tools and contained in a database. Data is exported from the database to a communication cluster file (FIBEX file or AUTOSAR system description file).

The FlexRay Configuration Tool supports the FIBEX and AUTOSAR standards for describing FlexRay networks.

Communication cluster file

A communication cluster file describes the entire FlexRay bus system, for example in a car, and contains the following data:

- Network topology description including all the nodes, controllers, and channels
- Communication-protocol-specific configuration data
- Communication schedule
- Frames and signals

The FlexRay Configuration Tool supports the FIBEX and AUTOSAR standards for describing FlexRay networks. You can base your configuration on an imported FIBEX file or AUTOSAR system description file.

FIBEX format

The Field Bus Exchange (FIBEX) is a format for data exchange between tools, such as the FlexRay Configuration Tool, that work with message-oriented bus communication. The description of a FlexRay communication system, its topology, and configuration data are usually stored in databases. Extracts from these databases are then exported to an XML exchange format file. One of the commonly used file formats is FIBEX, developed by ASAM e.V.

Supported FIBEX versions Several FIBEX versions have been released by ASAM e.V. The FlexRay Configuration Tool supports FIBEX baseline version 1.1.5a, 1.2.0a, 2.0.0d, 2.0.1, 3.0.0, 3.1.0, 3.1.1, 4.1.0, 4.1.1, 4.1.2, and FIBEX+.

The FIBEX versions use different terms for the messages: frames and PDUs.

- PDUs are defined in FIBEX+, FIBEX 3.x, and FIBEX 4.x versions.
- Frames are defined in FIBEX 2.0 and lower versions.

You can get further information on FIBEX at http://www.asam.net.

AUTOSAR System Templates

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership that aims to develop and establish an open standard for automotive electric/electronic (E/E) architectures.

AUTOSAR system description files are XML files that describe a system according to AUTOSAR. A system is a combination of a hardware topology, a software architecture, a network communication, and information on the mappings between these elements. AUTOSAR system description files are instances of the AUTOSAR System Template. The AUTOSAR System Template contains a description of the network communication and hardware topology according to the FIBEX standard defined by ASAM e.V.

AUTOSAR system description files are files of AUTOSAR XML file type that can be used to export or exchange information on system descriptions.

Supported versions The FlexRay Configuration Tool supports the AUTOSAR System Template of AUTOSAR Release 3.1.4, 3.2.1, 3.2.2, 4.0.3, 4.1.1, 4.1.2, 4.2.1, 4.2.2, 4.3.0, 4.3.1, 4.4.0, and AUTOSAR Classic Platform Release R19-11 and R20-11.

In AUTOSAR system description files, frame-specific data is always interpreted as PDUs.

For more information on AUTOSAR System Templates, refer to http://www.autosar.org.

Limitations applying to communication cluster files

Some rules and limitations apply to the FIBEX or AUTOSAR system description file you want to import. Refer to Limitations Applying to the Communication Cluster File on page 27.

Related topics

Basics

HowTos

Limitations Applying to the Communication Cluster File

Limitations

The following rules and limitations apply to the FIBEX or AUTOSAR system description file you want to import.

- A FIBEX or AUTOSAR system description file must contain at least the following elements:
  - One cluster
  - One channel
  - Two ECUs connected to a FlexRay bus (only for FIBEX 1.1 and 1.2)
- Not supported frame types:
  - DIAG-REQUEST
  - DIAG-RESPONSE
  - DIAG-STATE

- Not supported data types (FIBEX only):
  - A_UNICODE2STRING
  - A_BYTEFIELD
  - A_BITFIELD
- The following data types are only supported with a bit length ≤ 64:
  - A_ASCIISTRING
  - OTHER
- Not supported computation method categories:
  - SCALE-LINEAR
  - SCALE-RAT-FUNC
  - RAT-FUNC
  - TAB-NOINTP
  - FORMULA
- Frames/PDUs:
  - Only one multiplexer can be defined per frame/PDU.
  - Subframes can be defined only as part of event frames for FIBEX 1.1 and 1.2 versions.
  - Multiplexed PDUs with multiple segments are supported for AUTOSAR system description files only.
- Signals:
  - The coded bit length of a signal must be specified.
  - The signal update bit is not supported for FIBEX+.
  - Signal groups are not supported for FIBEX.
- Scale constraints:
  - For FIBEX version ≤ 2.0 and FIBEX+, only the scale constraints 'VALID' and 'NOT VALID' are supported.
  - One valid signal value range at most can be imported per signal (i.e., a maximum of one signal value range for the validity status 'VALID' per signal).
  - For every signal, a scale constraint != VALID is supported only if its value range contains exactly one single signal value.
  - For each signal, the value ranges of the different scale constraints must not overlap.
  - Scale constraints that are defined by physical data that does not match the physical data type are not supported.
  - Importing more than one scale constraint for a specific validity status per signal is not possible.
- Computation scales:
  - The FlexRay Configuration Tool supports only SCALE-LINEAR TEXTTABLE computation methods for signal encoding/decoding that consist of at most one linear (or identical) scaling and arbitrary many text table scalings.
  - Computation scales that are combined in a single SCALE-LINEAR TEXTTABLE computation method must have non-overlapping value ranges.
  - The TEXTTABLE computation method is supported for scalar coded values only.

- The value ranges of computation scales must not overlap in the scope of the containing computation method.
- The BITFIELD-TEXTTABLE computation method is interpreted as IDENTICAL during import. The BITFIELD-TEXTTABLE computation method results in a concatenated value set.
- Naming convention:
  - Signal, ECU and PDU short names must be unique.
- XCP functionality:
  - XCP functionality is not supported for AUTOSAR 4.x and AUTOSAR Classic Platform Release R19-11 and R20-11 system description files.
- AUTOSAR end-to-end communication protection:
  - The FlexRay Configuration Tool does not support the AUTOSAR E2E Transformer. E2E protection profile descriptions that are based on the E2E Transformer are not imported.

Related topics

Basics

Communication Cluster Files Usable for Configuration..26

Aspects of Miscellaneous Supported AUTOSAR Features

Multiplexed IPDUs

The FlexRay Configuration Tool supports multiplexed IPDUs. A multiplexed IPDU consists of one dynamic part, a selector field, and one optional static part. Multiplexing is used to transport varying ISignal IPDUs via the same bytes of a multiplexed IPDU.

- The dynamic part is one ISignal IPDU that is selected for transmission at run time. Several ISignal IPDUs can be specified as dynamic part alternatives. One of these alternatives is selected for transmission.
- The selector field value indicates which ISignal IPDU is transmitted in the dynamic part during run time. For each selector field value, there is one corresponding ISignal IPDU of the dynamic part alternatives. The selector field value is evaluated by the receiver of the multiplexed IPDU.
- The static part is one ISignal IPDU that is always transmitted.

'Multiplexed IPDU' is a term according to AUTOSAR.

Container IPDUs and contained IPDUs

The FlexRay Configuration Package supports *container IPDUs*. A container IPDU is an IPDU that contains one or several smaller IPDUs (i.e., *contained IPDUs*). The contained IPDUs can be ISignal IPDUs or multiplexed IPDUs, for example.

Contained IPDUs are not sent directly on the bus. When a trigger condition for a contained IPDU is fulfilled, the contained IPDU is written to its container IPDU.

The container IPDU itself is sent on the bus when one of its trigger conditions is met. Refer to Triggering of container IPDUs. When a container IPDU is mapped to a frame, all its contained IPDUs are included in that frame as well. Thus, container IPDUs separate in time the packing of an IPDU from the actual transmission of the IPDU. According to AUTOSAR, a contained IPDU must be included in exactly one container IPDU.

Dynamic and static container layout The container layout of container IPDUs can be organized dynamically or statically:

Container Layout	Description
Dynamic container layout	The contained IPDUs do not have a predefined position in the container IPDU. The receiving ECU identifies the contained IPDUs via their header IDs. For this purpose, two header types are available (`SHORT_HEADER`, `LONG_HEADER`). A contained IPDU can provide a separate header ID for each of these header types. The container IPDU defines the required header type: The ID of the required header type must be specified for each contained IPDU, and the receiving ECU evaluates only these IDs.
Static container layout	The contained IPDUs have a fixed position within the container IPDU. The receiving ECU identifies the contained IPDUs unambiguously by their position within the container IPDU. For this purpose, the header type of the container IPDU must be set to `NO_HEADER` and an offset value must be specified for each contained IPDU. The offset value determines the position of the contained IPDU within the container IPDU.

Triggering of container IPDUs The triggering of container IPDUs depends on various timing and triggering conditions that can be specified for container and/or contained IPDUs in the communication matrix. For example, a container IPDU can be transmitted after the following events:

- A specified timeout value elapses.
- Immediately after the first contained IPDU is added.
- A specific contained IPDU triggers its transmission.

The FlexRay Configuration Tool supports various AUTOSAR-compliant timing and triggering conditions.

Queuing of contained IPDUs For each contained IPDU that is transmitted via the bus, the communication matrix must specify the collection semantics:

- With a queued semantics, multiple instances of the contained IPDU can be added to one container IPDU.

  A queued semantics is supported only for contained IPDUs that are included in container IPDUs with a dynamic container layout.
- With a last-is-best semantics, only one instance of the contained IPDU can be added to one container IPDU. In this case, the data of the contained IPDU is buffered and only the latest data is added to the container IPDU just before it is transmitted.

  For contained IPDUs that are included in container IPDUs with a static container layout, only the last-is-best semantics is supported.

The FlexRay Configuration Tool supports both semantics. However, when a container IPDU is received that contains several instances of a contained IPDU, only the last received instance can be displayed for the contained IPDU.

Identifying updated contained IPDUs in container IPDUs with static container layout When a container IPDU with a static container layout is transmitted, some of its contained IPDUs might not have been updated since the previous transmission and therefore provide old data. There are two ways to identify the update state:

Setting	Description
Update bit of contained IPDU	Contained IPDUs can provide an update bit. The bit is set if a contained IPDU has been updated between two transmissions of the related container IPDU.
Unused bit pattern of container IPDU	Container IPDUs can provide an unused bit pattern. If a contained IPDU of the container IPDU is not updated, the content of the contained IPDU is cleared and replaced with the unused bit pattern.

Accepting contained IPDUs received with container IPDUs with dynamic container layout For container IPDUs with a dynamic container layout, the setting of the `RX-ACCEPT-CONTAINED-IPDU` attribute determines which contained IPDUs are accepted by the receiving ECU:

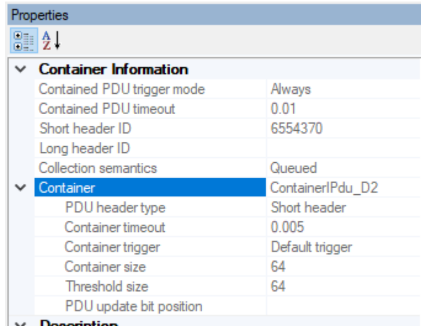
Setting	Description
`ACCEPT-ALL`	In general, the receiving ECU accepts any contained IPDU, regardless of whether it is included in the container IPDU according to the specifications in the communication matrix. However, to accept a contained IPDU that is not included in the container IPDU according to the specifications in the communication matrix, the following conditions must be met: ▪ The communication matrix includes the IPDU in another container IPDU whose `RX-ACCEPT-CONTAINED-IPDU` attribute is set to `ACCEPT-ALL`. ▪ The header ID of the header type that is required by the container IPDU is specified for the contained IPDU.
`ACCEPT-CONFIGURED`	The receiving ECU accepts only contained IPDUs that are included in the container IPDU according to the specifications in the communication matrix.

Contained IPDUs that are not accepted are ignored by the receiving ECU, i.e., they are not extracted from the container IPDU and their data is not processed.

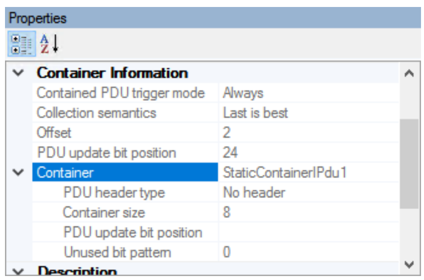
'Container IPDU' and 'contained IPDU' are terms according to AUTOSAR.

Working with container and contained IPDUs in the FlexRay Configuration Tool After you import a communication cluster file in the FlexRay Configuration Tool, the contained IPDUs are displayed in the Communication Cluster view. Container IPDUs are not displayed. However, the FlexRay Configuration Tool displays the connection of a contained IPDU and its container IPDU in the Properties view, among other container-relevant information.

Below are two examples. The first shows the properties of a contained IPDU that is in a container IPDU with a dynamic container layout.



The second example shows the properties of a contained IPDU in a static container IPDU (identifiable by **PDU header type** = No header). For this contained IPDU, an update bit was defined in the communication matrix.



Tip

In the **Configuration** and **Monitoring** views, the FlexRay Configuration Tool provides the **Contained IPDU type is** property filter, which allows you to display only the contained IPDUs that are in containers with a static container layout or those contained in dynamic containers. Refer to Filter (FlexRay Configuration Tool Reference 📖).

You can get send and trigger status information on the contained IPDUs and the associated container IPDUs at run time via the Simulink model or in ControlDesk. To include suitable variables in the TRC file, you must select the **Frame Container Control** feature in the **Element Selection** dialog.

When you work with container IPDUs and contained IPDUs, some limitations apply. Refer to Limitations on page 141.

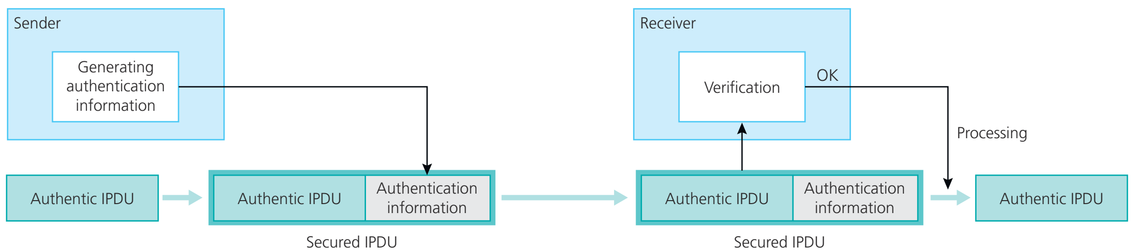
Secure onboard communication for PDUs

The FlexRay Configuration Tool supports *secure onboard communication (SecOC)* according to AUTOSAR. Secure onboard communication provides authentication mechanisms to identify PDU data that was modified or transmitted without authorization, e.g., due to a replay attack.

PDUs for SecOC In secure onboard communication scenarios, the following PDUs are used:

- *Authentic IPDUs*: Authentic IPDUs contain the payload, which will be secured by the authentication mechanisms of SecOC. The required authentication information itself is not included in authentic IPDUs but in secured IPDUs. According to AUTOSAR, authentic IPDUs can be ISignal IPDUs, container IPDUs, or multiplexed IPDUs, for example.

- *Secured IPDUs* and *cryptographic IPDUs*: Secured IPDUs are used to secure the payload of authentic IPDUs, i.e., they contain the required authentication information. The sender includes the authentication information in the secured IPDU and the receiver verifies the received authentication information, as shown in the following example.



  The handling of the related authentic IPDUs depends on whether secured IPDUs are configured as cryptographic IPDUs:

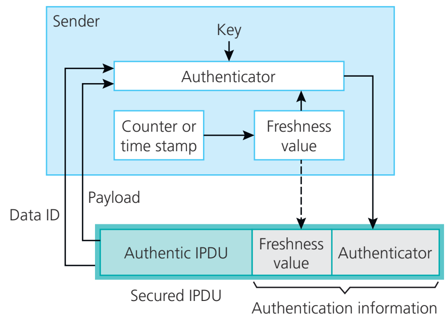
  - If a secured IPDU is configured as a cryptographic IPDU, the related authentic IPDU is not included in the secured IPDU. In this case, the authentic information is contained in the cryptographic IPDU. The authentic IPDU and the cryptographic IPDU are transmitted separately via the bus. The receiver verifies the authentication information each time the authentic IPDU or cryptographic IPDU is received.
  - If a secured IPDU is not configured as a cryptographic IPDU, the related authentic IPDU is directly included in the secured IPDU. In this case, only one PDU is exchanged via the bus.

Authentication information According to AUTOSAR, there are various ways to generate and verify authentication information. To generate and verify specific authentication information, OEM-specific implementations are required. In general, authentication information that is included in a secured IPDU consists of a freshness value and an authenticator:

- *Freshness value*: The freshness value is a monotonously increasing value. Depending on the OEM-specific implementation, the freshness value can be a counter or time stamp value. The freshness value is required for calculating the authenticator. Additionally, the freshness value can directly be included in the secured IPDU. If it is, it can be included completely or in part, i.e., as a truncated freshness value. The truncated freshness value contains the lower bits of the complete freshness value (the more significant bits are discarded). The size of the truncated value is specified in the communication cluster file.
- *Authenticator*: The authenticator is calculated according to OEM-specific algorithms and keys. For calculating the authenticator, the following data is required:
  - Data identifier of the secured IPDU
  - Payload of the authentic IPDU
  - Freshness value

The authenticator is included in the secured IPDU either completely or in part, i.e., as a truncated authenticator. The truncated authenticator value contains the higher bits of the complete authenticator value (the less significant bits are discarded). The size of the truncated authenticator is specified in the communication cluster file.

The following illustration is an example of authentication information in a secured IPDU.

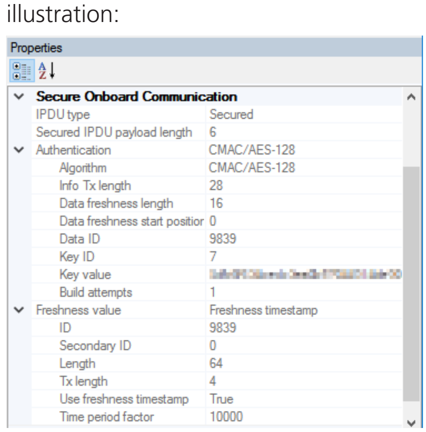


Secured PDU header A secured IPDU can contain the optional *secured PDU header*. The secured PDU header indicates the length of the authentic IPDU. Different header types are possible: `noHeader`, `securedPduHeader08Bit`, `securedPduHeader16Bit`, and `securedPduHeader32Bit`.

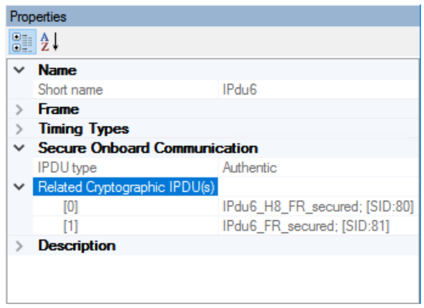
The following illustration shows the structure of a secured IPDU with a secured PDU header:

'Secured IPDU' is a term according to AUTOSAR.

Working with secured IPDUs in the FlexRay Configuration Tool After you import a communication cluster file in the FlexRay Configuration Tool, the authentic IPDUs and cryptographic IPDUs are displayed in the Communication Cluster view. Secured IPDUs containing the authentication information are not displayed. For authentic IPDUs and cryptographic IPDUs, the FlexRay Configuration Tool displays the corresponding secure onboard communication information in the Properties view, as shown for a TX IPDU in the following illustration:



The **IPDU type** property indicates whether the selected element is a secured IPDU, authentic IPDU, or cryptographic IPDU. Secured IPDUs and cryptographic IPDUs are represented by icons featuring a small lock symbol in the **Communication Cluster** view. For authentic IPDUs, the **Related Cryptographic IPDU(s)** property displays the associated cryptographic IPDUs, as shown in the following illustration.



For a cryptographic IPDU, the related authentic IPDUs are listed accordingly.

To implement secure onboard communication, you must enable SecOC support for your project and provide the OEM-specific implementation for generating authentication information via user code. Refer to General Page (FlexRay Configuration Tool Reference).

You get access to the authentication data and status information on the verification of secured IPDUs at run time in ControlDesk. To include suitable variables in the TRC file, you must select the **Frame Authentication Control** feature in the **Element Selection** dialog.

When you work with secure onboard communication, some limitations apply. Refer to Limitations on page 141.

Global time synchronization

The FlexRay Configuration Package supports *global time synchronization (GTS)* according to AUTOSAR. Global time synchronization means providing and distributing synchronized times in a bus network across all ECUs.

Time bases A synchronized time is called a *time base*. A time base is a unique time entity characterized by the progression of time, the ownership, and the reference to the physical world. There are two types of time bases:

- *Synchronized time base*: A synchronized time base is a time base that is synchronized with other time bases at different processing entities. These synchronized time bases constitute a global time. A synchronized time base does not depend on the existence of another time base.
- *Offset time base*: An offset time base is a time base that depends on a particular synchronized time base and holds an offset value with respect to that time base.

Time bases can provide relative times (e.g., time after power-up of an ECU or an operating hours counter) or absolute times (e.g., a UTC time). Because there can be more than just one time inside a vehicle, multiple time bases can exist in parallel. Time bases can be based on other time bases. Per ECU, there can be up to 16 synchronized time bases and offset time bases each.

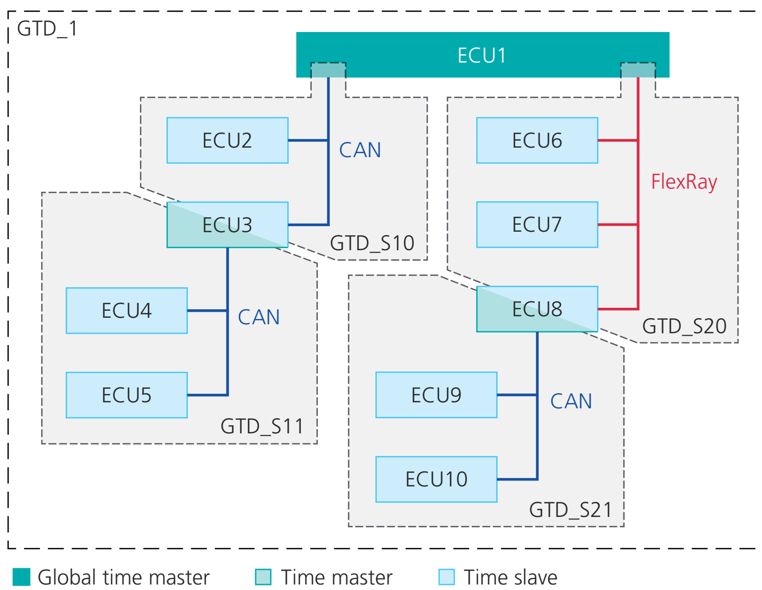
The FlexRay Configuration Tool does not support offset time bases.

Global time domain All the components (e.g., nodes and communication systems) that are linked to the same synchronized time base constitute a *global time domain*. There can be several global time domains in a bus network. Global time domains are distinguished by *domain identifiers* as follows:

- Global time domains that use different synchronized time bases have different domain identifiers.
- Global time domains that use the same synchronized time base have the same domain identifier.

Global time domains that use the same synchronized time base are connected by the means of *subdomains*. A subdomain denotes which components (e.g., nodes) are linked to the related time base where the scope is limited to exactly one communication cluster. A global time domain can declare several other global time domains as its subdomains.

The following illustration shows an example for the global time distribution in a network topology with four communication clusters and five global time domains. *GTD_S10* and *GTD_S20* are subdomains of *GTD_1*. *GTD_S11* and *GTD_S21* are subdomains of *GTD_S10* and *GTD_S20*, respectively. All global time domains use the same domain identifier.



Depending on its role in the network, an ECU can be a time master and/or a time slave.

Time master, time slave, and time gateway Global time synchronization is based on a master-slave system. An ECU can be both time master and time slave at the same time. Additionally, if synchronized times are processed via multiple communication clusters, an ECU can act as a time gateway

- A *time master* is an entity that is the master for a certain time base and distributes this time base to a set of time slaves within a certain cluster of a bus network. If a time master is also the global owner and origin of the time base, it is called the *global time master*.

- A *time slave* is an entity that is the recipient of a certain time base within a certain cluster of a bus network. Each time slave has its own time base instance, i.e., a local instance of the time. Whenever a time slave receives a new time from the time master, the time slave interpolates the time until the next synchronization with the time master.
- A *time gateway* is a time base instance that receives the synchronized time as a time slave from a time master on one cluster and then forwards it to another cluster as a time master. A time gateway typically consists of one time slave and one or more time masters.

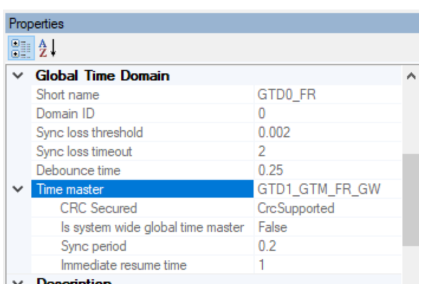
Time synchronization messages The messages that distribute the time information are called *time synchronization messages*. There are CRC-secured and unsecured time synchronization messages. However, the FlexRay Configuration Package only supports secured messages.

The following illustration shows the structure of a secured time synchronization message that is used for time synchronization over FlexRay:

Message type	CRC	Time domain	E2E sequence counter	FlexRay cycle counter	SGW (*1 bit*)	Reserved (*1 bit*)	User byte 0	User byte 1	Seconds	Nano seconds
8 bits	*8 bits*	*4 bits*	*4 bits*	*6 bits*			*8 bits*	*8 bits*	*48 bits*	*32 bits*
Byte 0	Byte 1	Byte 2		Byte 3			Byte 4	Byte 5	Byte 6 ... Byte 11	Byte 12 ... Byte 15

Working with global time synchronization in the FlexRay Configuration Package After you import a communication cluster file in the FlexRay Configuration Tool, the global time synchronization elements are displayed in the Communication Cluster view. GTS master and slave elements are represented by icons featuring a small clock symbol (example: time master).

For time master and time slave PDUs, the FlexRay Configuration Tool displays relevant GTS parameter values in the Properties view, as shown in the following illustration for a time master PDU:



To configure time synchronization via FlexRay, you must add the corresponding time master and time slave elements to the simulation configuration. When generating a Simulink model based on the Simulink configuration data from the FlexRay Configuration Tool, a PDU block is created in the automatically generated FlexRay model for each time synchronization message that was selected for sending or receiving by an ECU. You can drag the blocks to your model to access the time synchronization messages from the model.

You get access to information on global time synchronization PDUs/frames at run time via the Simulink model or in ControlDesk. To include suitable variables in the TRC file, you must select the Frame Global Time Sync Access feature in the Element Selection dialog.

When you work with global time synchronization, some limitations apply. Refer to Limitations on page 141.

End-to-end protection for ISignal groups

The FlexRay Configuration Tool supports end-to-end communication protection for ISignal groups according to the following AUTOSAR end-to-end protection profiles:

- Profile 01
- Profile 02
- Profile 05
- Profile 06

The end-to-end protection of an ISignal group must be specified in the communication matrix. Each end-to-end-protected ISignal group must be mapped to exactly one ISignal IPDU. An ISignal IPDU can contain one or more end-to-end-protected ISignal groups.

For more information, refer to Implementing Checksum Algorithms for Signal Groups Protected via End-To-End Communication Protection on page 104.

Limitations

When you work with bus communication according to AUTOSAR, some limitations apply. Refer to Limitations on page 141.

Related topics

References

Element Selection (FlexRay Configuration Tool Reference 🕮)

Limitations With Opaque Byte Order Format

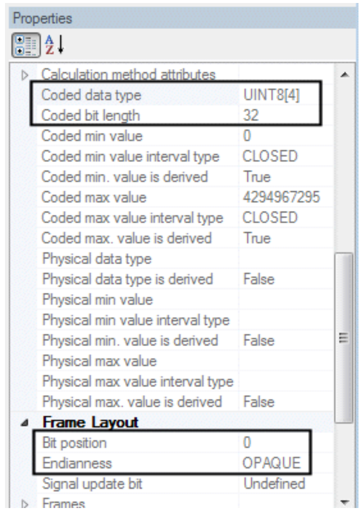
Introduction

In addition to the Motorola format (big endian) and Intel format (little endian), AUTOSAR system description files also offer the opaque byte order format. The FlexRay Configuration Tool supports the opaque byte order format, but there are some limitations to note.

Opaque byte order format

The FlexRay Configuration Tool supports signals with opaque byte order format. Data of signal instances with the opaque byte order is interpreted as dynamic uint8 arrays (`uint8[n]`), where n depends on the signal length. Unlike with big endian or little endian format, the data with opaque byte order is transmitted without endianness conversion. The start bits of the opaque signal instances must be byte-aligned, which means that their start bit positions must be 0 or multiples of 8.

The **Properties** view displays relevant information on signals with opaque byte order. Besides the endianness, the parameter values for items such as the coded data type of the signal, the signal length in bits, and the start bit position of the signal instance are displayed.



There are some limitations for working with the opaque byte order format (see below).

Limitations

The following limitations apply when working with AUTOSAR system description files containing data with the opaque byte order format:

- The FlexRay Configuration Tool supports the opaque byte order format for signal instances only.
- The opaque byte order format is supported only for signals with a maximum size of 254 bytes whose signal length is a multiple of 8 bits.
- For signals with the opaque byte order format, only the IDENTICAL, TEXTTABLE, and BITFIELD-TEXTTABLE computation methods are supported. The TEXTTABLE and BITFIELD-TEXTTABLE computation methods are interpreted as IDENTICAL during import.
- Signals with the opaque byte order format cannot be used as alive counters.
- For array signals with a length of 8 bytes (`uint8[8]`), only default values that can be represented by doubles are supported, not the maximum default values.
- For signals with the opaque byte order format whose signal length exceeds 64 bits, no validity port can be generated in Simulink. This means that you cannot read or write the signal validity status in Simulink in the **Signals** subsystem of the corresponding PDU RX or PDU TX block.
- For signals with the opaque byte order format whose signal length exceeds 64 bits, the physical port data type is not supported.
- For signals with the opaque byte order format whose signal length exceeds 64 bits, you cannot configure the **Physical default value** property.

How to Create a New Project

Objective	To start another configuration, you must create a new configuration project.
Project	A project is the container for one FIBEX or AUTOSAR system description file and the configuration based on it.
Method	**To create a new project** **1** From the menu bar, choose Project – New. If another project is currently open, it is closed. If you made any changes to the open project, you are prompted to save them before exiting.
Result	You created a new project.
Next steps	You can now import a communication cluster file.
Related topics	HowTos How to Import a Communication Cluster File.. 40 References New (FlexRay Configuration Tool Reference 📖)

How to Import a Communication Cluster File

Objective	Before you can start your configuration, you must import a FIBEX or AUTOSAR system description file. This visualizes the FlexRay network information in the Communication Cluster view and is the basis for your configuration.
Validity of communication cluster files	The imported communication cluster file must be valid according to a validation schema. ▪ When a FIBEX file is imported, the FlexRay Configuration Tool itself does not validate it against a FIBEX schema. Using an invalid FIBEX file can result in faulty configuration.

- For AUTOSAR system description files, the FlexRay Configuration Tool can perform a validation check during the import process. You can enable or disable this validation.

For further information on communication cluster files, refer to Communication Cluster Files Usable for Configuration on page 26 and Limitations Applying to the Communication Cluster File on page 27.

Project

A project is the container for *one* communication cluster file and the configuration based on it. If a project is currently open, you must first create a new project before you can import a FIBEX or AUTOSAR system description file. Refer to How to Create a New Project on page 40. If no other configuration project is currently open, you automatically create a new project while importing the communication cluster file.

Restrictions

Some restrictions apply to the communication cluster file:

- An invalid FIBEX or AUTOSAR system description file cannot be imported. A file is invalid, for example, if the timing does not fit one of the supported dSPACE platforms.
- Unsupported frames and signals are not imported.
- Some rules and limitations apply to the FIBEX or AUTOSAR system description file you want to import. Refer to Limitations Applying to the Communication Cluster File on page 27.

Preconditions

If a project is open, it must have no FIBEX or AUTOSAR system description file.

Method

To import a communication cluster file

1. From the menu bar, choose Project – Import Communication Cluster to open the Select a Communication Cluster dialog.
2. Specify the source folder and select the file to be added.
3. If you selected an AUTOSAR system description file to be imported, you can activate a file validation check during the import process by selecting the Schema Validation checkbox.
4. Click Select.

Result

The FIBEX or AUTOSAR system description file is imported and displayed in the Communication Cluster view.

Next steps

You can now start your configuration.

Related topics

Basics

HowTos

References

Import Communication Cluster (FlexRay Configuration Tool Reference 📖)

How to Update the Communication Cluster File

Objective

If the communication cluster file which you created your configuration with is changed, you can update the configuration using the new communication cluster file.

Basics

The FlexRay Configuration Tool updates the current FlexRay configuration automatically. Inconsistencies and configuration changes which occur during the update are reported as log messages.

Limitations of the communication cluster update

The following limitations apply if you update the communication cluster file in your configuration:

- Updating is allowed only within the following groups:
  - *FIBEX 1.0*: 1.1.5a, 1.2.0, 1.2.0a
  - *FIBEX 2.0*: 2.0.0b, 2.0.0d, 2.0.0e, 2.0.1
  - *FIBEX 3.0*: 3.0.0, 3.1.0, 3.1.1
  - *FIBEX 4.0*: 4.1.0, 4.1.1, 4.1.2
  - *FIBEX+*
  - *AUTOSAR 3*: AUTOSAR 3.1.4, AUTOSAR 3.2.1, AUTOSAR 3.2.2
  - *AUTOSAR 4*: AUTOSAR 4.0.3, AUTOSAR 4.1.1, AUTOSAR 4.1.2, AUTOSAR 4.2.1, AUTOSAR 4.2.2, AUTOSAR 4.3.0, AUTOSAR 4.3.1, AUTOSAR 4.4.0
  - *AUTOSAR Classic Platform*: AUTOSAR Classic Release R19-11, R20-11
- The communication cluster files must have the same critical cluster parameters, for example, speed, cycle, sample clock period.

Method

To update the communication cluster file

1 From the menu bar, choose Project – Update Communication Cluster to open the Select a Communication Cluster dialog.

2 Specify the source folder and select the new communication cluster file.

3 If you selected an AUTOSAR system description file as the new file, you can activate a file validation check during the update process by selecting the Schema Validation checkbox.

4 Click Select.

The FlexRay Configuration Tool reads the new communication cluster file and updates the FlexRay configuration.

5 Read the log messages in the Log view to view the inconsistencies.

> **Note**
>
> The messages displayed in the view are also written to the `FlexRayConfigurationTool.log` file.

Result

Your FlexRay configuration is updated according to the new communication cluster file.

Related topics

References

Update Communication Cluster (FlexRay Configuration Tool Reference 🕮)

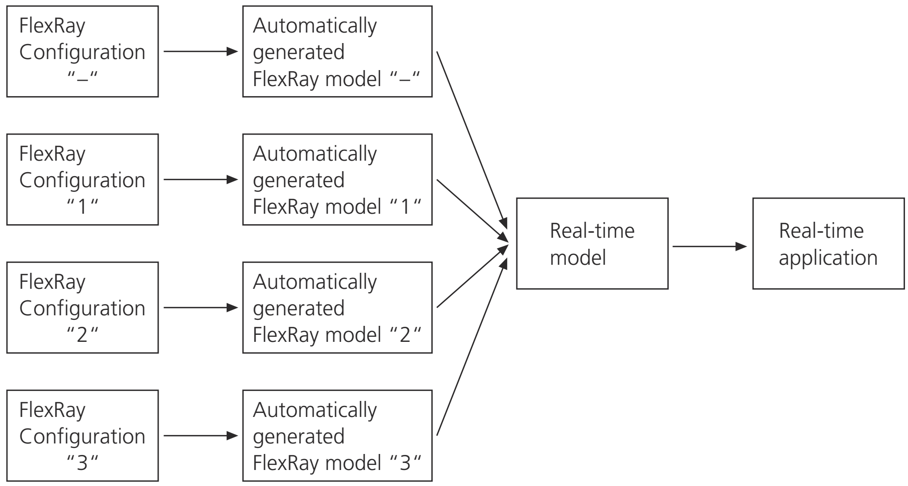
How to Create Configurations for Multiple Buses

Objective

One FlexRay configuration project can contain the configuration of only one FlexRay bus. If you want to use several buses in one Simulink model, for example, to model a gateway, you must create several FlexRay projects with several configurations.

Basics

If your real-time hardware is connected to more than one FlexRay bus, you must create a configuration for each FlexRay bus. The FlexRay blocks must be created for all the configurations and implemented in one real-time model.



Configuration ID

If a real-time model contains multiple FlexRay buses, most of the RTIFLEXRAYCONFIG blocks belong to one FlexRay bus only. A configuration ID is used to identify blocks from different FlexRay buses. All blocks with the same configuration ID belong to the same bus. The configuration ID must be unique for each FlexRay bus.

The configuration ID also specifies the priority. Four values are allowed for the configuration ID: -, 1, 2, 3. Configuration ID '-' has the highest priority, configuration ID 3 has the lowest priority.

It is recommended to implement always the FlexRay bus with configuration ID '-' in the real-time model. It is possible to set the ID to 1,2, or 3 instead of '-', but FlexRay configurations 1, 2, and 3 have a different synchronization task. For more information, refer to:

- Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖)
- Modeling Several FlexRay Buses on One dSPACE Real-Time System (Model Interface Package for Simulink - Modeling Guide 📖)

Task offset shift

As tasks cannot be executed in parallel on one node, a time offset shift can be specified for FlexRay configurations with IDs 1, 2, 3.

Limitations of multiple bus support

The following limitations apply to multiple bus support (working with several configurations for one model):

- The FlexRay Configuration Tool supports a maximum of four configurations for one processor board or SCALEXIO Processing Unit. Only two configurations are supported for a MicroAutoBox II/III. The maximum number of possible configurations might be lower due to limited processing power.

- It cannot be guaranteed that the tasks of the lower-priority FlexRay configurations are executed at the scheduled times. This can lead to the following problems:
  - Temporary buffer lock error
  - Dynamic TX PDUs might be omitted.
  - Static TX PDUs contain old data or are sent as null frames.
- Only one XCP configuration is allowed for a real-time system.
- It is not possible to send static TX PDUs which share the same communication slot on the same channel via different FlexRay nodes or different bus controllers, even if their cycle counter filtering (defined by base cycle and cycle repetition in absolute scheduled timing) is different. If the controller of a FlexRay node is configured to send a static TX PDU within a specific communication slot with the given cycle counter filtering of the TX PDU, it automatically sends null frames in the slot each time the PDU is not sent. Any attempt by another bus node to send a static TX PDU in the slot results in invalid frames on the FlexRay bus.

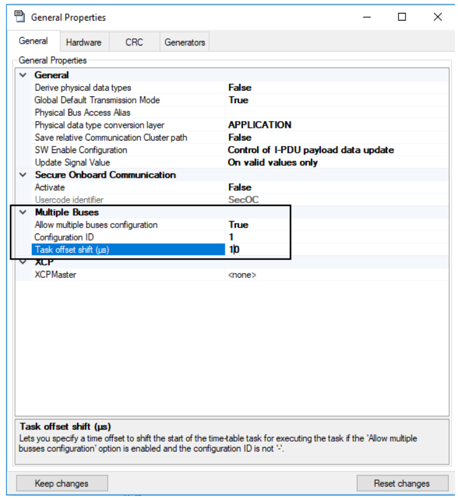
  For further information, refer to *FlexRay Communications System Protocol Specification Version 2.1*.
- Only one CRC C file is allowed for all the FlexRay configurations used.
- The RTIFLEXRAYCONFIG ERROR HOOK STATUS block cannot distinguish between different FlexRay configurations.
- The RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT block might be triggered for the wrong controller if several FlexRay configurations are used on one core of a processor board and the corresponding controllers are on different I/O board types in the same slot position. To avoid this, place the controller modules in different slot positions.
- After migrating FlexRay multiprocessor models based on several FlexRay configurations from an earlier FlexRay Configuration Package version to dSPACE FlexRay Configuration Package 2.4 or later, you must assign some RTIFLEXRAYCONFIG blocks to the RTIFLEXRAYCONFIG UPDATE block manually.
- The tasks are sorted according to their priorities during the first build process. If you add a FlexRay configuration afterwards, its tasks cannot be inserted with the correct priority. In this case you can
  - Sort the tasks manually.
  - Delete the RTI Data block. If no RTI Data block is in the model, the tasks are sorted again in the next build process.
- If no FlexRay configuration with configuration ID `"-"` is in the real-time model, the FlexRay-based capture synchronization in ControlDesk does not work.

Method

To create configurations for multiple buses

1 From the menu bar of the dSPACE FlexRay Configuration Tool, choose Tools - General Properties.

The **General Properties** dialog opens at the **General** page.



2 Set the **Allow multiple buses configuration** property to True.

When the value is True, the **Configuration ID** property can be edited.

3 Select the value for the **Configuration ID** property.

If the configuration ID is 1, 2, or 3, the **Task offset shift** property can be edited.

4 Specify the value for the **Task offset shift** property. It shifts the time-table task start of configuration 1, 2 or 3. They will be interrupted by the tasks of configuration ID "-", which have the higher priority. You can use this to fulfill the timing constraints and deadlines of the FlexRay slots for configuration ID 1, 2, 3.

5 Click **Keep changes** to confirm your settings and close the dialog.

Result

The configuration project can be used to generate the blocks for modeling one FlexRay bus. For details, refer to:

- Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖)
- Modeling Several FlexRay Buses on One dSPACE Real-Time System (Model Interface Package for Simulink - Modeling Guide 📖)

Related topics

References

General Page (FlexRay Configuration Tool Reference 📖)
RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG ERROR HOOK STATUS (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG UPDATE (RTI FlexRay Configuration Blockset Reference 📖)

Managing Configuration Projects

Introduction

Managing configuration projects involves saving and opening projects.

Saving projects

If you want to save your project, the FlexRay Configuration Tool provides Save and Save As.

Save You can save the current project. In the Save Project dialog, you can specify the path and file name. The project is saved with your whole configuration. The location of the imported FIBEX or AUTOSAR system description file is referenced to the project file.

Save As You can save the current project under a new name.

> **Tip**
>
> If you change the location of a referenced communication cluster file after saving the project, the FlexRay Configuration Tool opens a standard dialog for you to enter the new path. You cannot import a different communication cluster file than the referenced one. You can save the path of the communication cluster file as a full path or relative to the working folder. For detailed information, see the Save relative communication cluster path option on General Page (FlexRay Configuration Tool Reference 📖).

Opening projects

You can open an existing project via the Project - Open command. A standard Open dialog lets you search for a project file.

Updating changed FIBEX files

If the current imported FIBEX or AUTOSAR system description file is changed, you can update your project. The FlexRay Configuration Tool provides a communication cluster update and checks the new file and the current project for consistency. Refer to How to Update the Communication Cluster File on page 42.

Related topics

References

Open (FlexRay Configuration Tool Reference 📖)
Save (FlexRay Configuration Tool Reference 📖)
Save As (FlexRay Configuration Tool Reference 📖)

Handling the Elements of a Communication Cluster File

Introduction

The FlexRay Configuration Tool lets you handle the elements of the communication cluster, display their parameters in the **Properties** view, and search for elements.

Communication cluster file elements

After you import a FIBEX or AUTOSAR system description file, the FlexRay Configuration Tool displays the following FlexRay network information in the **Communication Cluster** view:

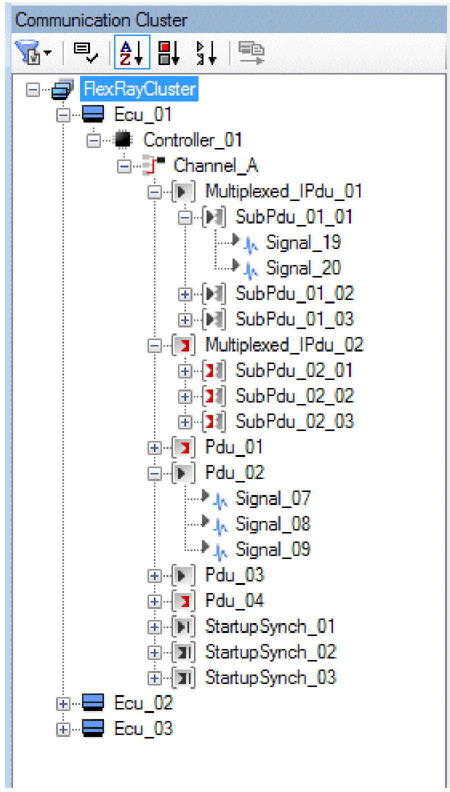
- FlexRay cluster and ECUs belonging to it
- Controllers and channel(s)
- Frames belonging to the channels including communication scheduling and the corresponding signals

Communication cluster element parameters

Each communication cluster element includes several parameters. The FlexRay Configuration Tool displays the FlexRay-relevant parameters in the **Properties** view.

Communication Cluster view

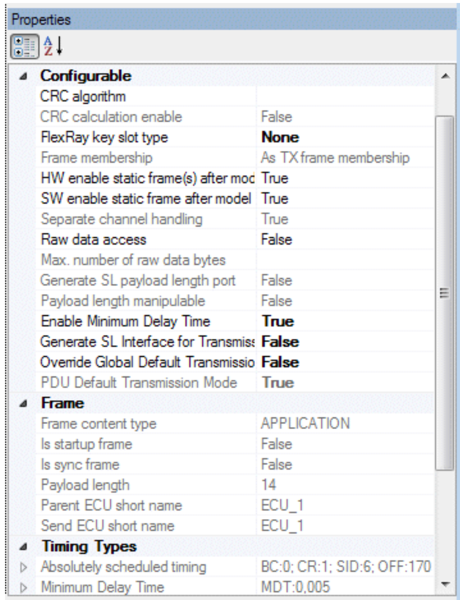
A FIBEX or AUTOSAR system description file is represented by a hierarchical graphical structure in the **Communication Cluster** view. The illustration below shows the **Communication Cluster** view with an imported communication cluster file.



Element types The FlexRay Configuration Tool uses icons to indicate the types of communication cluster elements. For a list of all the icons and their descriptions, refer to Icons for Communication Cluster Elements and FlexRay Task Types (FlexRay Configuration Tool Reference 📖).

Properties view

If you select a communication cluster element in one of the views, you can see the most significant parameters in the **Properties** view. The FlexRay Configuration Tool displays the parameters in categories, with complex data nesting. See the example of frame parameters below.



Parameters which are imported from a FIBEX or AUTOSAR system description file are read-only.

Configurable parameters You can configure some parameters for frames, signals, and tasks. Refer to:

- Configurable Properties for ECUs, Signals, and Frames on page 58
- Tunable Properties of Tasks on page 124

Searching for elements

The FlexRay Configuration Tool provides a search function to find a communication cluster element in the hierarchical structures of the views. You can enter a search string in a drop-down field and start searching by clicking . Wildcards are supported. The FlexRay Configuration Tool searches in the view you have currently activated. The matching element is highlighted. It stores search strings already entered in the current session for you to reuse.

Filtering and sorting elements

You can filter the communication cluster elements displayed in the different views of the FlexRay Configuration Tool, and you can change the sort order of the displayed elements. Refer to Sorting and Filtering Elements in the Views on page 65.

Related topics

Basics

Sorting and Filtering Elements in the Views.. 65

References

Find (FlexRay Configuration Tool Reference 📖)

Creating Configurations

Introduction

The configuration is the basis for a blockset which will be generated for your Simulink model afterwards. The following sections provide information on different configuration tasks.

Where to go from here

Information in this section

Information in other sections

Basics on Creating Configurations

Where to go from here

Information in this section

Basics of Simulation

Introduction

The FlexRay Configuration Tool allows you to create configurations for simulating one FlexRay node or several FlexRay nodes (restbus simulation).

Simulation configuration with the FlexRay Configuration Tool

The FlexRay Configuration Tool has the Configuration view for creating your simulation configuration.

Configuration view The Configuration view allows you to create and manage your configurations. Refer to Basics on Sending and Receiving Signals on page 71.

Selecting FlexRay network elements for simulation configuration After a communication cluster file is imported, all the FlexRay network elements are displayed in the Communication Cluster view. From there, you can select and place them in the Configuration view for simulation configuration via drag & drop. Refer to Selection of ECUs, Frames and Signals on page 64.

Configurable properties for signals and frames The FlexRay Configuration Tool provides some configurable properties for frames and signals to be connected with the simulation. You can influence some block parameters of the automatically generated FlexRay model blocks with the configurable properties. Refer to Configurable Properties for ECUs, Signals, and Frames on page 58.

Simulation types

You can create configurations for the following simulation types:

Simulation of FlexRay nodes To simulate an entire ECU, you can select an ECU and place it in the Configuration view. You can also select several ECUs at once. Simulink configuration data can be generated for the selected ECU(s).

Simulation of the restbus If you have a real ECU and want the restbus to be simulated, you can create all the frames and signals transmitted by the other ECUs via the FlexRay network. When you select an ECU, the corresponding send frames are placed on the Configuration view automatically. Simulink configuration data can be generated for the restbus.

For instructions on doing this, refer to Basics on Sending and Receiving Signals on page 71.

Preparing simulation

Task creation After creating your simulation configuration, the FlexRay Configuration Tool lets you create tasks. Refer to How to Create Tasks on page 126.

Controller configuration The FlexRay Configuration Tool lets you view the required controller and buffer count for your simulation configuration. Refer to How to View the Controller Configuration on page 128.

Simulink configuration data The FlexRay Configuration Tool lets you generate a Simulink configuration data file containing all the necessary parameters for the real-time model. Refer to How to Generate Code on page 137.

Automatically generated FlexRay model You can generate the corresponding RTI blocks from the Simulink configuration data using MATLAB. An automatically generated FlexRay model collects all the RTI blocks configured for your FlexRay network. You can use the generated model as a library for designing the real-time application for your FlexRay node. Refer to Using the Generated Code on page 139.

Limitations for using the automatically generated FlexRay model

You can use only one FlexRay library for a dSPACE node:

- In a single-processor system, you can use only the blocks from one automatically generated FlexRay model.

- In a multiprocessor system, each processor can have its own FlexRay model. After separation using RTI-MP, there is also one automatically generated FlexRay model for each processor.

Related topics

Basics

Modeling Concept

Introduction

The dSPACE FlexRay Configuration Package provides PDU-based modeling (PDU = protocol data unit), i.e., the model is based on PDUs.

The FlexRay Configuration Tool lets you create a configuration for PDU-based modeling. The generated Simulink configuration data can be used together with the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset to generate the 'automatically generated FlexRay model'. The automatically generated FlexRay model has RTI blocks for each configured PDU that contains several signals.

PDU-based modeling can be used for dual-channel FlexRay systems.

PDU-based modeling using the RTI FlexRay Configuration Blockset

If you work with a PHS-bus-based system or a MicroAutoBox II, the automatically generated FlexRay model contains a block from the *RTI FlexRay Configuration Blockset* for each PDU. PDUs comprise several signals. The blocks combine several features, so that only two block types must be used for transferring PDU frames:

- RTIFLEXRAYCONFIG PDU RX
- RTIFLEXRAYCONFIG PDU TX

The following additional block is also configured. You can use it to get the status of TX frames independently of the PDU TX block, for example, in another task.

- RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS

PDU-based modeling using the FlexRay Configuration Blockset

If you work with a MicroAutoBox III or a SCALEXIO system, the automatically generated FlexRay model contains a block from the *FlexRay Configuration Blockset* for each PDU. PDUs comprise several signals. The blocks combine several features, so that only two block types must be used for transferring PDU frames:

- FLEXRAYCONFIG PDU RX
- FLEXRAYCONFIG PDU TX

Related topics

References

Structure of the Automatically Generated FlexRay Model (RTI FlexRay Configuration Blockset Reference 📖)
Structure of the Automatically Generated FlexRay Model (FlexRay Configuration Blockset Reference 📖)

Configurable Properties for ECUs, Signals, and Frames

Introduction

The FlexRay Configuration Tool provides some configurable properties for configuring ECUs, signals, and frames in the **Monitoring**, **Configuration**, or **Task** view. You can use the configurable properties to influence some parameters of the read and write blocks in the automatically generated FlexRay model.

Configurable properties

You can configure some properties for ECUs, signals, and frames. If you activate an ECU, frame or signal in the **Configuration**, **Monitoring**, or **Task** view, the corresponding configurable properties are displayed in the **Properties** view.

Configurable ECU properties You can configure the following properties for ECUs.

Configurable Property	Default	Description
ECU disable mode	Controller	Specifies whether disabling/enabling the communication of the simulated ECU is to be done by disabling/enabling the controller(s) or the buffer(s) which transmit the ECU's frames. ▪ For ECUs with sync frame or startup-sync frame, only the 'Controller' mode is available. If you want to activate the 'Buffer' mode for these ECUs, you must set the **FlexRay key slot type** parameter for the sync or startup-sync frame to 'None'. As an alternative, you can specify 'Global RX pool' as the frame membership for the sync or startup-sync frame. In this case, the sync or startup-sync frame will not be disabled. ▪ If the **ECU disable mode** is configured differently for the ECUs belonging to a TX frame membership group (some ECUs use the 'Controller' mode, others use the 'Buffer' mode), the FlexRay Configuration Tool sets the 'Controller' mode for the TX frame membership. This is done for time optimization purposes.
Monitoring frame membership	As TX frame membership	Selects the membership type for frames in the **Monitoring** view.
RX frame membership	As TX frame membership	Selects the membership type for RX frames in the **Configuration** view.
TX frame membership	1	Selects a frame membership group[1)] for the TX frames of this ECU.

[1)] For information on frame membership groups, refer to Building Frame Membership Groups on page 90.

Configurable signal properties You can configure the following properties for signals.

Configurable Property	Signal Type	Default	Description
Alive counter start[1)]	Send signal	0	Specifies the start value of the alive counter. The start value must be lower than or equal to the stop value.
Alive counter step[1)]	Send signal	1	Specifies the increment value of the alive counter. The increment value must be greater or equal to 0 and it must be lower than or equal to the difference of stop and start value.
Alive counter stop[1)]	Send signal	1	Specifies the stop value of the alive counter. The stop value must be greater than or equal to the start value.
Coded default value	Send signal	0 or default value if defined in the communication cluster file	Specifies the usage of default values for send signals if the application task did not yet calculate the values. If you change the coded default value, the physical default value is adapted.
Coded initial value	Receive signal	0 or default value if defined in the communication cluster file	Specifies the usage of default values in the application task for receive signals that were not yet received. If you change the coded initial value, the physical initial value is adapted.
Generate SL interface	Send and receive signal	True	Activates the generation of a Simulink interface. If the value is False, no Simulink interface for writing or reading the signal will be generated at the corresponding PDU RX or PDU TX block. The default value for monitored signals is False.
Generate SL read status port	Receive signal	False	Activates the read status port. If it is activated, you can read the status from the outport in Simulink in the **Signals** subsystem of the corresponding PDU block.
Generate SL validity port	Send and receive signal	False	Activates the validity port (available only if a value with validity status ≠ 'VALID' is specified in the imported communication cluster file). If it is activated, you can read and write the signal validity status in Simulink in the **Signals** subsystem of the corresponding PDU RX or PDU TX block.
Physical default value	Send signal	0 or default value if defined in the communication cluster file	Specifies the usage of default values for send signals if the application task did not yet calculate the values. If you change the physical default value, the coded default value is adapted.
Physical initial value	Receive signal	0 or default value if defined in the communication cluster file	Specifies the usage of default values in the application task for receive signals that were not yet received. If you change the physical initial value, the coded initial value is adapted.
Port data type[2)]	Send and receive signal	Coded data type	Specifies whether the coded or physical port data type is used for encoding or decoding the signal. For signals with the SCALE_LINEAR_TEXTTABLE computation method, the 'Coded and Physical' port data type can be selected.[3)]
Used for alive counter[1)]	Send signal	False	Specifies the usage of the signal as alive counter.

Configurable Property	Signal Type	Default	Description
Used for CRC calculation	Send and receive signal	False	Marks the signal as a CRC signal (available only if the CRC algorithm is defined for the frame (see Configurable frame properties on page 60)).

[1] For details on configuring an alive counter, refer to How to Configure an Alive Counter on page 79.
[2] The specified value is used for the configured RTI block. It cannot be changed in the model.
[3] The 'Coded and Physical' port data type can be set individually for each signal via the **Properties** view, or globally for all relevant signals via the **Activate support for text tables** setting for the SL Data Generator in the **General Properties** dialog.

Configurable frame properties You can configure the following properties for frames.

Configurable Property	Frame Type	Default	Description
CRC algorithm	Static and dynamic frames	Empty	Specifies which CRC algorithm is used for the frame (see CRC Page (FlexRay Configuration Tool Reference 📖)).
CRC calculation enable	Static and dynamic frames	False	Specifies whether the checksum calculation is initially enabled or disabled. It is available only if a CRC algorithm is defined for the frame. Checksum calculation can also be controlled under Simulink, refer to How to Handle Checksum Calculation for a PDU (FlexRay Configuration Features 📖).
Default cyclic switch code	Static and dynamic PDUs with sub-PDUs	Smallest available switch code value	Specifies the switch code for the cyclic sub-PDUs.
Default event switch code	Dynamic PDUs with sub-PDUs	Smallest available switch code value	Specifies the switch code for the event-triggered sub-PDUs.
Enable Minimum Delay Time	TX frames	False	Specifies whether minimum delay time support is enabled or disabled for the PDU. The property exists only if minimum delay time support is possible for the PDU. Refer to How to Configure PDUs for Minimum Delay Time Support on page 85.
FlexRay key slot type	Static TX frames		Specifies the type of a static frame: ▪ StartupSync: startup-sync frame ▪ Sync: sync frame ▪ None: "normal" frame You can change the FlexRay key slot type only for static TX frames. If you configure a frame/PDU for a dual channel configuration, the value of this property is valid for both channels.
Frame membership	Static TX frames whose key slot type is StartupSync or Sync	As TX frame membership	Specifies the frame membership of the selected Send-Startup-Sync frame: ▪ As TX frame membership ▪ Global RX pool For detailed information on frame memberships, refer to Building Frame Membership Groups on page 90.
Generate SL Interface for Transmission Modes[1]	Static and dynamic PDUs	False	Specifies whether to add a control element to the PDU mapping subsystem connected to the **RTIFLEXRAYCONFIG PDU TX** or **FLEXRAYCONFIG PDU TX** block for you to switch between the transmission modes during run time. For further information on transmission modes, refer to How to Configure PDU Transmission Modes on page 81.

Configurable Property	Frame Type	Default	Description
Generate SL payload length port	Dynamic frames	False	Specifies whether to generate a port for raw data access for you to set the payload length of send frames or read the payload length of receive frames. A payload length port can only be generated for dynamic frames and when raw data access is enabled (see below). For dynamic RX PDUs, the **Payload length readable** property must also be set to 'True' (see below). Each TX and RX PDU must be assigned to a frame to which no other PDU is assigned.
HW enable static frame(s) after model start[2)]	Static TX frames	True	Specifies whether the static frame is enabled or disabled after model start via hardware. All frames which share the same slot are enabled or disabled together. Some frames cannot be enabled or disabled by hardware, for example, startup or sync frames.
Ignore PDU update bit[2)]	RX frames with PDUs	True	Specifies whether the PDU update bit is to be ignored or evaluated when a frame/PDU is received. The Ignore update bit property can be configured for a PDU only if the PDU update bit functionality is defined for it in the FIBEX or AUTOSAR system description file.
Max number of raw data bytes[2)]	RX frames	Payload length of frame	Specifies the number of bytes which are read by the configured block. This corresponds to the width of the **RxBytes** outport in Simulink. The maximum value is limited by the payload length of the RX frame or PDU.
Override Global Default Transmission Mode[1)]	Static and dynamic PDUs	False[3)]	Specifies whether the global default transmission mode specified on the **General** page of the **General Properties** dialog must be taken as the default transmission mode for the PDU, or whether another default transmission mode can be specified for it. If allowed, you can select an individual default transmission mode for the PDU via the **PDU Default Transmission Mode** property.
Payload length readable	Dynamic RX frames	False	Specifies whether the payload length of the RX frame shall be actively read during run time each time when the frame is received. A payload length readable port can only be generated for dynamic RX frames and when raw data access is enabled (see below).
Payload length manipulable	Dynamic TX frames	False	Specifies whether the payload length of the TX frame can be manipulated during run time. A payload length manipulation port can only be generated for dynamic TX frames and when raw data access is enabled (see below).
PDU Default Transmission Mode[1)]	Static and dynamic PDUs	True[4)]	Specifies the transmission mode to be used as the default transmission mode for the selected PDU. You can configure this property only if the **Override Global Default Transmission Mode** property is set to `True`.
Raw data access	Static and dynamic frames	False	Specifies whether raw data access is enabled in Simulink. The default value for frames without signals is True.
Separate channel handling	TX frames	True	Specifies whether channel handling is separated in dual channel configuration. For details, refer to Dual Channel Configurations on page 67.

Configurable Property	Frame Type	Default	Description

SW enable static frame after model start[2)]	Static frames	Enabled	Specifies whether the static frame is enabled or disabled after model start via software. This is also possible for startup-sync frames.
			If a static PDU is *enabled* via software (SWEnable = 1), the data which is sent depends on the setting of the SW Enable Configuration property (see General Page (FlexRay Configuration Tool Reference 🕮)) and on the settings of the CHI Code Generator (see Generators Page (FlexRay Configuration Tool Reference 🕮)):
			▪ The SW Enable Configuration property in the General Properties dialog is set to Control of L-PDU commit to FlexRay buffer and the Static TX buffer transmission mode property of the CHI Code Generator is set to Event (null frame used):
			  ▪ A null frame is sent if the LPDU to be sent contains exactly one PDU, and this PDU has not been updated and does not have a PDU update bit.
			  ▪ Payload data is sent for all other LPDUs.
			▪ Other settings:
			  ▪ Payload data is sent.
			If a static PDU is *disabled* via software (SWEnable = 0), the data which is sent depends on the database version, on the setting of the SW Enable Configuration property (see General Page (FlexRay Configuration Tool Reference 🕮)), and on the settings of the CHI Code Generator (see Generators Page (FlexRay Configuration Tool Reference 🕮)):
			▪ FIBEX version ≤ 2.0:
			The SW Enable Configuration property in the General Properties dialog is always set to Control of L-PDU commit to FlexRay buffer. You cannot change this setting.
			  ▪ Null frames are sent if the Static TX buffer transmission mode property of the CHI Code Generator is set to Event (null frame used).
			  ▪ Old data is sent if the Static TX buffer transmission mode property is set to State (old value used).
			▪ FIBEX+, FIBEX 3.0, FIBEX 3.1, FIBEX 4.1.x, or AUTOSAR System Template:
			If the SW Enable Configuration property in the General Properties dialog is set to Control of L-PDU commit to FlexRay buffer:
			  ▪ Null frames are sent if the Static TX buffer transmission mode property of the CHI Code Generator is set to Event (null frame used).
			  ▪ Old data is sent if the Static TX buffer transmission mode property is set to State (old value used). The update bit has the value that was set before SWEnable was set to 0.

<table>
<tr><th>Configurable Property</th><th>Frame Type</th><th>Default</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>If the SW Enable Configuration property in the General Properties dialog is set to Control of I-PDU payload data update:
▪ The update bit is set to 0 and old data is sent, independently of the settings of the Static TX buffer transmission mode property.
If the Static TX buffer transmission mode property is set to Event (null frame used), note the following information:

Note

In some cases, the <code>SW Enable</code> variable is overruled. This can happen if the update bit of a PDU is manipulable (see <code>Update Bit</code> group below). If the <code>Update Bit Enable</code> variable is 1, automatic update bit calculation is disabled. The update bit of the PDU is set to the value specified by the <code>Update Bit Value</code> variable. If <code>SW Enable</code> is 0 and the update bit value is 1, the PDU still sends old data. If the <code>Update Bit Enable</code> variable is not used, <code>SW Enable</code> is used to enable or disable the sending of static TX PDUs.</td></tr>
</table>

[1] If you create a dual-channel configuration, the specified value is used for both channels. You cannot specify different property values for channel A and channel B.

[2] The specified value is used for the configured RTI block and cannot be changed in the model.

[3] For PDUs that do not have any timing information specified in the underlying FIBEX or AUTOSAR system description file, the default value is 'True'.

[4] For PDUs that do not have any timing information specified in the underlying FIBEX or AUTOSAR system description file, the default transmission mode is 'LPDU timing triggered'.

Handling properties

Properties filter The Properties filter lets you display ECUs, signals, and frames which meet specific conditions to be connected with the configurable properties. Refer to Sorting and Filtering Elements in the Views on page 65.

Tip

You can select several ECUs, signals, or frames in one view at once using the **`Shift`** or **`Ctrl`** key. This allows you to change common properties all at once.

Related topics

Basics

Selection of ECUs, Frames and Signals

Introduction

The FlexRay Configuration Tool allows you to select ECUs, frames, and signals via drag & drop.

Selecting FlexRay network elements for simulation or monitoring purposes

After a FIBEX or AUTOSAR system description file is imported, all the FlexRay network elements are displayed in the Communication Cluster view. From there, you can select and place them in the other views for simulation configuration and/or monitoring via drag & drop. For instructions, refer to:

- Basics on Sending and Receiving Signals on page 71
- How to Select and Configure Signals for Monitoring on page 114

Limitations of FlexRay element selection

The following limitations apply to FlexRay element selection:

Dragging elements

- You can drag each receiving element to the Configuration view only once. Receiving elements already selected are grayed out in the Communication Cluster view.
- You can drag each sending element to the Configuration and Monitoring view only once. A sending element dropped to the Configuration view can additionally be dropped to the Monitoring view. Sending elements already selected in both configurations are grayed out in the Communication Cluster view.
- A frame of an ECU can contain signals which are not received by the ECU itself. These signals cannot be dropped.

Send frames for simulation

- You can select only complete send frames for simulation. Even if only a single signal or subframe is selected, the whole send frame with all the signals and subframes is assigned.
- You can remove the subframes inside a frame except for one. If you delete signals of a subframe or a frame, the whole frame or subframe is removed from your configuration.
- If an XCP master node is not selected in the FlexRay Configuration Tool, XCP_RUNTIME_CONFIGURED frames cannot be used for simulation and monitoring.
- If an XCP master node is selected in the FlexRay Configuration Tool, XCP_RUNTIME_CONFIGURED and XCP_PRE_CONFIGURED frames from the XCP master node can only be simulated (not monitored) and you cannot select XCP frames for the TRC file.

Network Management frames (NM frames) You can select only complete Network Management (NM) frames for simulation or monitoring. A corresponding Simulink block with all the relevant signals is generated for the entire NM frame.

Signals for multiple receivers You cannot select signals or subframes which are not received by an ECU. If a frame is sent to multiple receivers and these receivers receive only a few signals or subframes of the frame, the remaining signals or subframes of the frame are blocked for the receivers. The remaining signals or subframes are also blocked for restbus simulation.

Signals for monitoring You can select single send signals (except for signals of NM frames) only for monitoring. If you select an ECU or a cluster, only their send signals are used for monitoring.

Time synchronization PDUs for monitoring You can select time master PDUs for monitoring, but you cannot select time slave PDUs for monitoring.

Selecting the XCP master You cannot select an XCP master node if a dual-channel communication cluster file is imported.

Related topics

Basics

Graphical User Interface of the FlexRay Configuration Tool.. 18

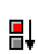
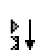
Sorting and Filtering Elements in the Views

Introduction

You can filter and sort the communication cluster elements displayed in the different views according to your needs.

Sorting

You can sort the project items alphabetically, or according to direction, type or category. The FlexRay Configuration Tool provides sorting functions via tool buttons at the top of each view.

Tool Button	Description
	To sort the displayed communication cluster elements alphabetically (default).
	To sort the displayed communication cluster elements by communication type.
	To sort the displayed communication cluster elements by the direction of communication.
	To sort the parameters by category. The function is available only in the **Properties** view.

For further information, refer to Sorting (FlexRay Configuration Tool Reference 📖).

Filtering

You can filter the project items displayed in the views. You can enable or disable the following filters via tool buttons at the top of each view.

Tool Button	Filter	Description
	Element filter	Provides multiple element-specific filter options that can be used in combination: ▪ ECU filter: To show one or all of the ECUs with their communication cluster elements. ▪ Redundant frames in logical channel: To show redundant frames in an additional node. If the filter is active, all the frames which are sent or received via channels A *and* B are displayed under a Logical_Channel_AB node. All the frames which are sent or received via *one* channel (A or B) only remain in the Channel_A or Channel_B node. ▪ Only the sending frames: To show only the sending frames with their signals. ▪ Only ECUs/tasks and the signals: To show only the signals of the ECUs/tasks.
	Short name filter	Lets you display only the signals whose short names contain a specified string. Wildcards ("*" and "?") are supported. **Tip** The FlexRay Configuration Tool also provides a global short name filter which applies to all views. If you want to specify a global filter, click the toolbar button at the top of the FlexRay Configuration Tool. 
	Property filter	Lets you display signals which meet a specific condition. The Property filter is available only in the Configuration and Monitoring views.

For details on the available filters, refer to Filter (FlexRay Configuration Tool Reference 📖).

Related topics

References

Filter (FlexRay Configuration Tool Reference 📖)
Sorting (FlexRay Configuration Tool Reference 📖)

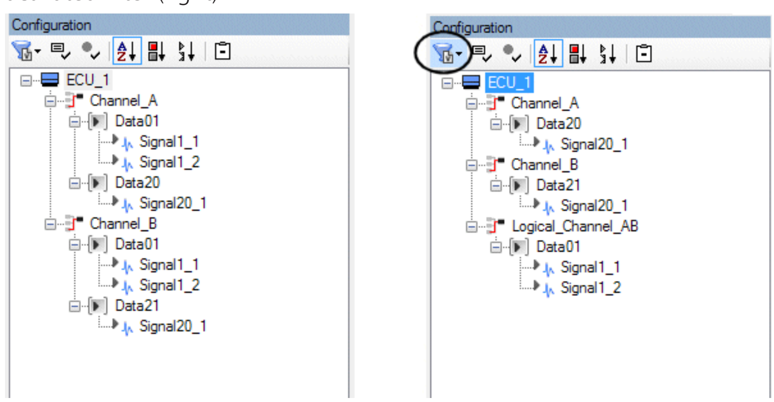
Dual Channel Configurations

Introduction

The FlexRay Configuration Tool can be used to create a configuration using two channels (dual channel configuration) if it is described by the communication cluster file. There are some features that simplify your work.

Filtering redundant frames

The views have a filter that is useful for dual channel configuration. If you activate the redundant frames in logical channel filter ( ), all the PDUs which are sent or received via both channels are displayed under the Logical_Channel_AB node. The PDUs which are sent or received via one channel only are still displayed under the Channel_A or Channel_B node. The following illustration shows an example of a deactivated filter (left) and an activated filter (right).



Although a PDU is displayed under the Logical_Channel_AB node, some parameter values of channel A and channel B can be different. This can happen, for example, if you use separate channel handling for a sending PDU (see below). In this case the fields of the parameter values which are different are empty in the Properties view.

Separate channel handling

If a PDU is sent via both channels, you can specify whether channel handling is separated. If channel handling is separated, RTIFLEXRAYCONFIG PDU TX or FLEXRAYCONFIG PDU TX blocks are generated for each channel and you can specify different property values for the PDU and its signals. If channel handling is not separated, one RTIFLEXRAYCONFIG PDU TX or FLEXRAYCONFIG PDU TX block is generated for both channels and you cannot specify different property values for the PDU and its signals.

Deactivating separate channel handling If separate channel handling was set and you have specified different property values for the PDUs or their signals, the property values are synchronized when you deactivate the separate channel handling. The FlexRay Configuration Tool copies the values specified for channel A to the properties of channel B.

- The values of the following PDU properties are synchronized:
  - HW enable static frame(s) after model start
  - SW enable static frame after model start
  - Payload length manipulable
  - Generate SL payload length port
  - Max number of raw data bytes
  - CRC algorithm
  - CRC calculation enable
  - Raw data access
  - Frame membership
  - Generate SL Interface for Transmission Modes
  - Override Global Default Transmission Mode
  - PDU Default Transmission Mode
- The values of the following signal properties are synchronized:
  - Used for CRC calculation
  - Coded default value
  - Physical default value
  - Port data type
  - Generate SL validity port
  - Alive counter step
  - Alive counter start
  - Alive counter stop
  - Used for alive counter
  - Generate SL interface

Required hardware resources

If you use an application containing a dual-channel FlexRay configuration, the assigned dSPACE real-time hardware must have a sufficient number of channels. A transceiver can work with only one channel (channel A or B), but not with both channels, at the same time.

When you create a FlexRay configuration based on a dual-channel communication cluster file, the number of required hardware resources depends on the number of channels that are actually used in the FlexRay configuration. If there is only one channel used within the FlexRay configuration, only one real controller is required. If there are two FlexRay channels within the FlexRay configuration, one or two real controllers are required, depending on the hardware you use. For example:

- If you work with a SCALEXIO system two real controllers are always required.
- If you work with a modular system based on a DS1007 with DS4501 IP Carrier Boards, only one controller is required, because the controller has two FlexRay channels.

In the generated M file, the `usedChannels` parameter can have the value 'A', 'B', or 'AB', depending on which channels are used within the FlexRay configuration.

Note

You can assign channels of several bus boards and/or modules to a real-time application. However, if you work with a dual-channel FlexRay configuration, there is one restriction to note: Channel A and channel B must be assigned to adjacent hardware channels on the same bus board or module.

Related topics

References

Filter (FlexRay Configuration Tool Reference 📖)
FLEXRAYCONFIG PDU TX (FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 📖)

Sending and Receiving Signals

Introduction

To send or receive frames and signals, you must create a configuration. The configuration is the base for the automatically generated Simulink model which contains all the configured blocks for your Simulink model.

Where to go from here

Information in this section

Triggering of Dynamic Frames and Subframes

Introduction

You can trigger the dynamic frames and subframes (since FIBEX 2.0).

Basics

The sending of dynamic frames must be triggered. It can be triggered by an event or cyclically. Since FIBEX 2.0, a dynamic subframe can also be triggered by triggering the corresponding frame with a switch code. The frame trigger type shows the possible trigger type.

Frame Trigger Type	Description
Dynamic event (DE)	The dynamic frame is triggered by an event. The property value for event-controlled timing is specified.
Dynamic cyclic (DC)	The dynamic frame is cyclically triggered. The property value for cyclic timing is specified.
Dynamic event cyclic (DEC)	The dynamic frame is triggered cyclically or by an event. The property values for event-controlled timing and cyclic timing are specified.
Dynamic event subframe (SDE)	The dynamic frame is triggered by triggering one or several subframes. The subframes are triggered by events (subframe triggering type: subframe dynamic event). The property value of the subframes for event-controlled timing is specified. The corresponding property value of the dynamic frame is empty.
Dynamic cyclic subframe (SDC)	The dynamic frame is triggered by triggering one or several subframes. The subframes are cyclically triggered (subframe triggering type: subframe dynamic cyclic). The property value of the subframes for cyclic timing is specified. The corresponding property value of the dynamic frame is empty.
Dynamic event cyclic subframe (SDEC)	The dynamic frame is triggered by triggering one or several subframes. The subframes are triggered by events and/or cyclically (subframe triggering type: subframe dynamic event cyclic). The property values of the subframes for event-controlled timing and cycle timing are specified. The corresponding property values of the dynamic frame are empty.

RTI block for modeling

Frame triggering and subframe triggering are done by the same RTI block, RTIFLEXRAYCONFIG PDU TX or FLEXRAYCONFIG PDU TX, respectively. This block can be used for event and cyclically triggered frames/PDUs.

Related topics

References

FLEXRAYCONFIG PDU TX (FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)

Basics on Sending and Receiving Signals

Introduction

The configuration of the signals specifies the RTI blocks of the automatically generated Simulink model.

Basics of creating configurations for simulation

For basic information on simulation, refer to Basics of Simulation on page 55.

Deriving the physical data type

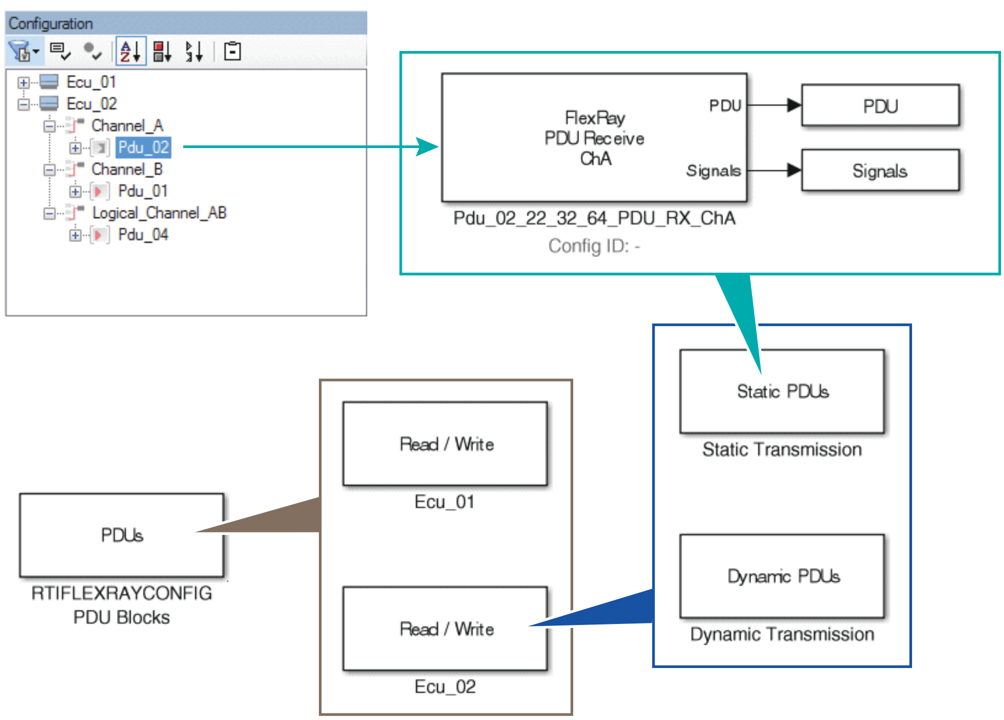
If the physical data type is not defined in the communication cluster file, you can let the FlexRay Configuration Tool set the physical data types of signals automatically. In addition, minimum and maximum physical values are derived for all signals. Refer to General Page (FlexRay Configuration Tool Reference 📖).

Limitations of selecting single frames and signals

There are a few limitations in connection with frame and signal selection. Refer to Limitations on page 141.

Example

After configuration a Simulink block is generated for each PDU and its signals used in the configuration. The Simulink blocks are structured in several subsystems. The following illustration shows an example of selected signals for configuration in the Configuration view, and the corresponding RTIFLEXRAYCONFIG PDU RX and RTIFLEXRAYCONFIG PDU TX blocks of the FlexRay model after automatic generation in MATLAB.



Possible methods

To create a configuration for simulation, you can

- Select single frames and signals to be simulated. Refer to How to Select Single Frames and Signals for Simulation on page 73.

- Select all the signals of an ECU for simulation. Refer to How to Select All Signals of an ECU for Simulation on page 75.
- Create restbus simulation for selected ECU(s) automatically. Refer to How to Configure a Restbus Simulation on page 76.

Related topics

References

RTIFLEXRAYCONFIG PDU RX (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)

How to Select Single Frames and Signals for Simulation

Objective

You can create configurations for single frames and signals.

Basics

- For basic information on simulation, refer to Basics of Simulation on page 55.
- For information on selecting network elements for simulation, refer to Basics on Sending and Receiving Signals on page 71.

Icons

The signal and frame types are represented as different icons.

Icon	Frame Type
	Static receiving frame[1), 2)]
	Static transmitting frame[1), 2), 3)]
	Static receiving sync frame[1), 2)]
	Static transmitting sync frame[1), 2), 3)]
	Static receiving startup-sync frame[1), 2)]
	Static transmitting startup-sync frame[1), 2), 3)]
	Static receiving subframe
	Static transmitting subframe[3)]
	Dynamic receiving frame[2)]
	Dynamic transmitting frame[2), 3)]
	Dynamic receiving subframe
	Dynamic transmitting subframe[3)]

1) The icon contains a clock if the element is a time synchronization message.

Icon	Frame Type

[2] The icon contains a lock symbol if the element is a secured or cryptographic IPDU.
[3] The icon contains a monitor if the element is monitored.

Icon	Signal Type
	Static receiving signal[1]
	Static transmitting signal[2, 3]
	Static receiving signal, which belongs to a sync frame[1]
	Static transmitting signal, which belongs to a sync frame[2, 3]
	Static receiving signal, which belongs to a startup-sync frame[1]
	Static transmitting signal, which belongs to a startup-sync frame[2, 3]
	Dynamic receiving signal[1]
	Dynamic transmitting signal[2, 3]

[1] The icon is crossed out if the ECU does not receive the signal, and the signal therefore cannot be configured in the FlexRay Configuration Tool. The communication cluster file defines that the signal is part of the PDU, but it does not instruct the ECU to unpack the signal from the PDU.
[2] The icon contains a monitor if the element is monitored.
[3] The icon is crossed out if the ECU does not send the signal, and the signal therefore cannot be configured in the FlexRay Configuration Tool. The communication cluster file defines that the signal is part of the PDU, but it does not instruct the ECU to pack the signal into the PDU. For the signal, the PDU default bit value will be used.

Restrictions

There are a few limitations in connection with frame and signal selection. Refer to Limitations on page 141.

Preconditions

You must first import a FIBEX or AUTOSAR system description file. Refer to How to Import a Communication Cluster File on page 40.

Method

To select single frames and signals for simulation

1. If the Configuration view is not open, click Views - Configuration.
2. From the Communication Cluster view, drag a frame or signal to the Configuration view.

Tip

- You can sort and filter the displayed communication cluster elements by signal or frame. Refer to Sorting and Filtering Elements in the Views on page 65.
- You can select several frames and signals at once using the `Shift` or `Ctrl` key.

The FlexRay Configuration Tool adds the ECU with selected frames and signals to the **Configuration** view.

3 In the **Properties** view, specify the configurable properties for frames and signals (see Configurable Properties for ECUs, Signals, and Frames on page 58).

4 If you want to remove an element from the **Configuration** view, select it and click .

Result

Frames and/or signals are selected for simulation.

Next steps

- You can select send frames of ECUs for monitoring purposes. Refer to How to Select and Configure Signals for Monitoring on page 114.
- You can create tasks. Refer to How to Create Tasks on page 126.

Related topics

HowTos

References

Configuration View (FlexRay Configuration Tool Reference)

How to Select All Signals of an ECU for Simulation

Objective

You can create configurations for all the frames and signals of an ECU.

Basics

- For basic information on simulation, refer to Basics of Simulation on page 55.
- For information on selecting network elements for simulation, refer to Basics on Sending and Receiving Signals on page 71.

Restrictions

There are a few limitations in connection with frame and signal selection. Refer to Limitations on page 141.

Preconditions

You must first import a FIBEX or AUTOSAR system description file. Refer to How to Import a Communication Cluster File on page 40.

Method

To select all signals of an ECU for simulation

1. If the Configuration view is not open, click Views - Configuration.
2. From the Communication Cluster view, drag an ECU to the Configuration view.

   Tip

   You can select several ECUs at once using the **`Shift`** or **`Ctrl`** key.

   FlexRay Configuration Tool adds the selected ECU with corresponding frames and signals to the Configuration view.
3. In the Properties view, specify the configurable properties for frames and signals (see Configurable Properties for ECUs, Signals, and Frames on page 58).
4. If you want to remove an element from the Configuration view, select it and click .

Result

All the frames and signals of an ECU are selected for simulation.

Next steps

- You can select send frames of ECUs for monitoring purposes. Refer to How to Select and Configure Signals for Monitoring on page 114.
- You can create tasks. Refer to How to Create Tasks on page 126.

Related topics

HowTos

References

Configuration View (FlexRay Configuration Tool Reference)

How to Configure a Restbus Simulation

Objective

You can create configurations for restbus simulation. In a restbus simulation all contra frames and signals of an ECU are simulated.

Basics

- For basic information on simulation, refer to Basics of Simulation on page 55.

- For information on selecting network elements for simulation, refer to Basics on Sending and Receiving Signals on page 71.

Restrictions

- There are a few limitations in connection with frame and signal selection. Refer to Limitations on page 141.
- You cannot start the automatic configuration features if send frame(s) of the ECU(s) you want to simulate the restbus for are still in the **Configuration** view. The automatically generated configuration is not valid if you add further send frames of the restbus-simulated ECU(s) to the **Configuration** view afterwards.

Preconditions

You must first import a FIBEX or AUTOSAR system description file. Refer to How to Import a Communication Cluster File on page 40.

Method

To configure a restbus simulation

1 In the **Communication Cluster** view, select the ECU you want to simulate the restbus for.

> **Tip**
>
> You can select several ECUs at once using the `Shift` or `Ctrl` key.

2 If the **Configuration** view is not open, click **Views - Configuration**.

3 On the **Communication Cluster** view's toolbar, click .
FlexRay Configuration Tool adds the ECUs to the **Configuration** view and selects the corresponding send frames automatically.

4 In the **Properties** view, specify the configurable properties for frames and signals (see Configurable Properties for ECUs, Signals, and Frames on page 58).

5 If you want to remove an element from the **Configuration** view, select it and click .

Result

Your configuration is prepared for restbus simulation.

Next steps

- You can select send frames of ECUs for monitoring purposes. Refer to How to Select and Configure Signals for Monitoring on page 114.
- You can create tasks. Refer to How to Create Tasks on page 126.

Related topics

HowTos

References

Configuration View (FlexRay Configuration Tool Reference 🕮)
Create Corresponding ECU (FlexRay Configuration Tool Reference 🕮)

How to Configure a Frame for Raw Data Access

Objective

You can access raw data of a frame/PDU to manipulate it in your Simulink model or in ControlDesk using the TRC file.

Raw data access

You can access the raw data of frames whose Raw data access property is set to True. The Raw data access property is automatically set to True if an imported frame has no signals.

The RTIFLEXRAYCONFIG PDU TX and RTIFLEXRAYCONFIG PDU RX blocks or the FLEXRAYCONFIG PDU TX and FLEXRAYCONFIG PDU RX blocks of the PDUs/frames are given additional ports that you can connect to other Simulink blocks.

Manipulating the payload length

If raw data access is enabled for a receive or send dynamic frame, you can manipulate or read its payload length if the frame contains one PDU. To enable manipulation or read, you must specify the frame's configurable properties:

- Payload length readable: To read the payload during run time
- Generate SL payload length port: To get a port to set or read the payload length

Preconditions

The frame must be selected for the FlexRay model.

Method

To configure a frame for raw data access

1 In the Configuration, Monitoring, or Task view, select the frame whose raw data you want to access.

> **Tip**
>
> You can select several frames at once using the **`Shift`** or **`Ctrl`** key.

2 In the Properties view, set the Configurable - Raw data access property to True.

3 Specify the manipulation options:
 - If you want to manipulate the payload length of a TX frame/PDU, set Configurable - Payload length manipulable to True.
 - If you want to read the payload length of an RX frame/PDU, set Configurable - Payload length readable to True.
 - If you want to access the payload length within the Simulink model, set Configurable - Generate SL payload length port to True.

Result

RTI blocks for raw data access are created in the generation process.

> **Note**
>
> You can access raw data of the frames or PDUs from variables in the TRC file, refer to How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File on page 134.

Related topics

HowTos

How to Send PDUs in Raw Format (FlexRay Configuration Features 📖)
How to Send PDUs in Raw Format (Model Interface Package for Simulink - Modeling Guide 📖)

References

FLEXRAYCONFIG PDU RX (FlexRay Configuration Blockset Reference 📖)
FLEXRAYCONFIG PDU TX (FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU RX (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 📖)

How to Configure an Alive Counter

Objective

You can use a send signal as alive counter.

Alive counter

An alive counter is a counter which is incremented by a defined value each time the signal is sent. This counter can be sent and then it can be evaluated by the receivers.

Limitations

The following signals cannot be used as alive counter:

- Signals that are contained in more than one TX frame (except for TX frames which are sent more than once per cycle, for example, 2.5 ms frames)
- Signals with one of the following coded data type:
  - UInt64
  - Int64
  - ASCIISTRING
  - FLOAT32
  - FLOAT64
  - OTHER
  - UInt8[n]
- Signals in frames that are of type XCP_RUNTIME_CONFIGURED and XCP_PRE_CONFIGURED in projects with selected XCP master node

Preconditions

The send signal which should be used as alive counter must be selected for the FlexRay model.

Method

To configure an alive counter

1. In the Configuration or Task view, select the send signal which shall be used as alive counter.
2. In the Properties view, set the Configurable - Used for alive counter property to True.

   When this option is set, the properties for specifying the alive counter are enabled.
3. In the Properties view, specify the properties for the alive counter:
   - Alive counter start specifies the start value
   - Alive counter step specifies the increment value
   - Alive counter stop specifies the stop value
4. To manipulate the alive counter during run time, select the Signal Alive Counter Control feature of the signal for the TRC file. Refer to How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File on page 134.

Result

The signal is configured and used as alive counter at the beginning of the simulation (`Source switch` = 2). An entry for the specified alive counter is generated into the TRC file. You can use ControlDesk to control the alive counter. Refer to:

- Using an Alive Counter in the Simulation (FlexRay Configuration Features 🕮)
- (MicroAutoBox III, SCALEXIO) Using an Alive Counter in the Simulation (Model Interface Package for Simulink - Modeling Guide 🕮)

Related topics

Basics

References

Element Selection (FlexRay Configuration Tool Reference 🕮)

How to Configure PDU Transmission Modes

Objective

You can create configurations with different transmission modes (that is, with different possible timing specifications) for each PDU. You can switch between the transmission modes later on during run time.

Transmission modes

The timing of PDUs is specified in the communication matrix. However, the way this information is specified depends on the database type. In AUTOSAR system description files, up to two different transmission modes can be defined for each IPDU. Each transmission mode is assigned one specific timing, at maximum. FIBEX communication cluster files do not use transmission modes, but any number of timings can be defined for each IPDU.

Transmission modes 'True' and 'False' The FlexRay Configuration Package allows you to work with PDUs that have up to two different transmission modes or timings. It provides the transmission modes *True* and *False* for this. For each transmission mode, there is a timing specification. The timings can have the following types:

- Cyclic
- Event-based
- Mixed (consists of a cyclic and an event-based timing)
- None (no timing specified)

The transmission modes are assigned to each PDU, and the timing information specified for the PDU in the communication cluster file is assigned to the respective transmission modes as follows:

- AUTOSAR system description files: The transmission mode and timing information can be taken directly since AUTOSAR also uses the two transmission modes.
- FIBEX files: For both the cyclic and the event-based timing type, the FlexRay Configuration Tool assigns the first timing definition each (as found in the FIBEX file) to the transmission mode *True*, and the second timing definition to the transmission mode *False*. Further timing specifications are ignored. If the FIBEX file contains both cyclic and event-based timings, the FlexRay Configuration Tool combines them to mixed timings. There are some limitations to note, see Limitations of transmission mode support on page 146.

The transmission mode *False* is always represented by the integer value 0, the transmission mode *True* by the integer value 1.

Transmission modes 'User-Defined' and 'LPDU timing triggered' Besides the two transmission modes with timings from the underlying FIBEX or AUTOSAR system description file (described above), the FlexRay Configuration Package also provides the transmission modes *User-Defined* and *LPDU timing triggered*, which can also be assigned to PDUs. These two transmission modes are always based on the corresponding LPDU timing.

- The transmission mode *User-Defined* allows you to create an additional event-based timing for a PDU, using the corresponding LPDU timing. You can use this mode to simulate errors in PDU timings, for example. The transmission mode *User-Defined* is always represented by the integer value 99.
- The transmission mode *LPDU timing triggered* is always cyclic. It can only be assigned to PDUs for which no timing information is specified in the underlying FIBEX or AUTOSAR system description file, i.e., for which neither the *True* nor the *False* transmission mode is supported. It allows cyclic transmission of PDUs, using the corresponding LPDU timing. The transmission mode *LPDU timing triggered* is always represented by the integer value 98.

  During the import of a communication cluster file, the transmission mode *LPDU timing triggered* is set as the default transmission mode for all the PDUs without timing specification in the FIBEX or AUTOSAR system description file.

Default transmission mode

For each TX PDU in a configuration, a *PDU default transmission mode* must be specified. It determines the transmission mode and thus implicitly the timing that should be active by default for the PDU after download.

In the FlexRay Configuration Tool, you can select the *global default transmission mode*. Via this project-global setting, you set the default transmission mode for all the PDUs in the project in one step. If a PDU should use a different default transmission mode, you can individually override the global setting by specifying a PDU-specific default transmission mode.

Switching the transmission mode during run time

You can switch between the PDU transmission modes at run time via the Simulink model or in ControlDesk. However, this is not possible by default.

- To get Simulink interfaces in order to switch the PDU transmission modes via the model, you must enable interface generation for each individual PDU in the **Properties** view.
- To switch between the transmission modes by using the TRC file, suitable variables must be included in the TRC file. You must use the **Element Selection** dialog for this.

Restrictions

- Creating configurations with different transmission modes for each PDU is possible only in connection with communication matrix information based on:
  - FIBEX 3.0 files
  - FIBEX 3.1 files
  - FIBEX 3.1.1 files

- FIBEX 4.1.0 files
- FIBEX 4.1.1 files
- FIBEX 4.1.2 files
- AUTOSAR system description files based on AUTOSAR 3.1.4, 3.2.1, 3.2.2, 4.0.3, 4.1.1, 4.1.2, 4.2.1, 4.2.2, 4.3.0, 4.3.1, 4.4.0, or AUTOSAR Classic Platform Release R19-11 or R20-11
- There are a few limitations on the PDU transmission modes. Refer to Limitations on page 141.

Preconditions

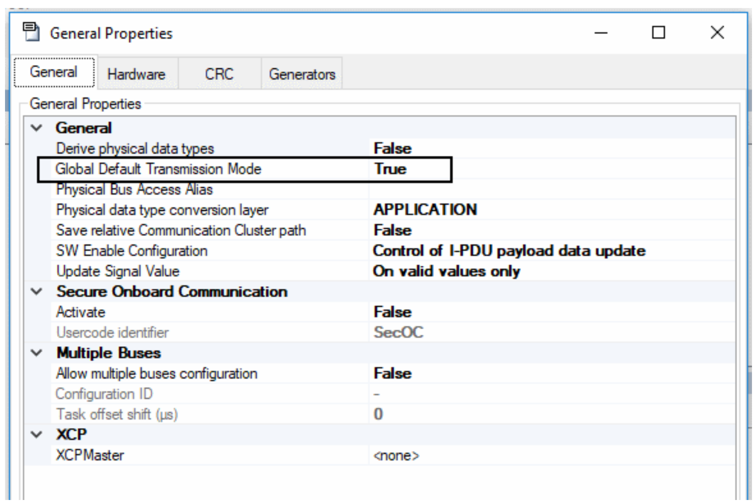
You must first import a FIBEX or AUTOSAR system description file containing different transmission mode or timing definitions for PDUs. Refer to How to Import a Communication Cluster File on page 40.

Method

To configure PDU transmission modes

1 From the menu bar of the dSPACE FlexRay Configuration Tool, choose Tools - General Properties.

The General Properties dialog opens at the General page.



2 Select the value for the Global Default Transmission Mode property. It specifies the default transmission mode for all the PDUs in the project.

The *LPDU timing triggered* transmission mode is not available for selection, because it can only be used with PDUs to which neither the *True* nor the *False* transmission mode is assigned. The *LPDU timing triggered* transmission mode can only be specified as the PDU-specific default transmission mode (see below).

3 Click Keep changes to confirm your settings and close the dialog.

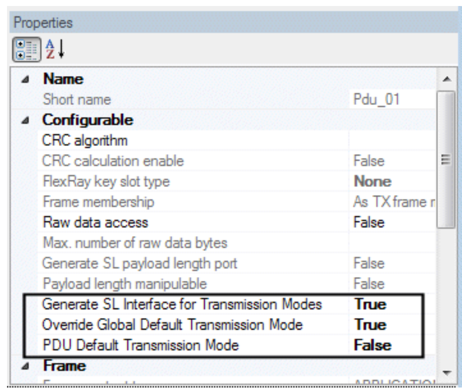
You can now specify PDU-specific transmission mode settings.

4 In the Configuration view, select a PDU.

5 In the **Properties** view, specify the transmission mode properties for the selected PDU:

- If you want to make the PDU use a default transmission mode different from the global default transmission mode, set **Configurable - Override Global Default Transmission Mode** to True. This enables the property for specifying the PDU-specific default transmission mode.

  PDUs without timing specification in the underlying FIBEX or AUTOSAR system description file do not use the global default transmission mode. Therefore, the **Override Global Default Transmission Mode** property is always set to True for them and cannot be changed.
- Select the default transmission mode to be used for the selected PDU via the **Configurable - PDU Default Transmission Mode** property.
- If you want to get a Simulink interface for switching the transmission mode of the selected PDU via the model, set **Configurable - Generate SL Interface for Transmission Modes** to True.



6 Repeat steps 4 ... 5 for all the PDUs you want to specify PDU-specific transmission mode settings for.

7 If you want to switch the transmission mode of PDUs during run time in ControlDesk, suitable variables must be generated in the TRC file. To include the variables to the TRC file, perform the following steps:

- From the menu bar, choose **Tools – Element Selection** to open the **Element Selection** dialog.
- In the features list, select the **Frame Dynamic Control** or **Frame Static Control** feature, depending on the PDU to be configured.

  Available Elements lists all the PDUs that are configured or available for the selected feature and not selected for the TRC file yet.
- Drag the PDU onto the **TRC Interface** item in **Selected Elements**.
- Repeat these steps for all the PDUs whose transmission mode you want to control via TRC file.
- Click **OK**.

Result

The transmission modes for the TX PDUs are configured. If enabled, Simulink interfaces for switching between the transmission modes are created in the model generation process, and entries for controlling the PDU transmission modes are generated into the TRC file.

Related topics

Basics

Using the Generated TRC File of PDU-Based Modeling (FlexRay Configuration Features 🕮)
Using the Generated TRC File of PDU-Based Modeling in ControlDesk (Model Interface Package for Simulink - Modeling Guide 🕮)

HowTos

How to Switch the Transmission Mode of a PDU (FlexRay Configuration Features 🕮)
How to Switch the Transmission Mode of a PDU (Model Interface Package for Simulink - Modeling Guide 🕮)

References

FLEXRAYCONFIG PDU TX (FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)

How to Configure PDUs for Minimum Delay Time Support

Objective

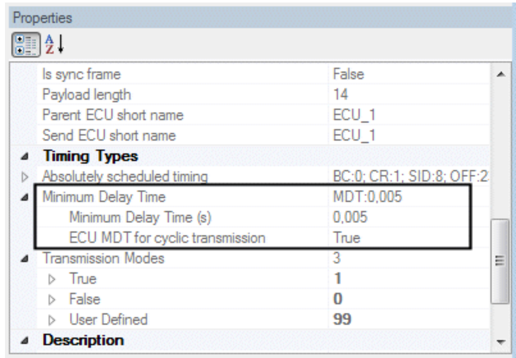
AUTOSAR system description files can contain minimum delay time information for TX PDUs. The FlexRay Configuration Tool allows configurations with minimum delay time support. You can disable and enable minimum delay time support later on during run time.

Minimum delay time

The minimum delay time of an IPDU specifies the minimum delay time between successive transmissions of this IPDU. It determines the time span that must elapse before new IPDU data can be packed into the LPDU.

Minimum delay time (MDT) information is defined in the AUTOSAR communication cluster files. This includes the definition of minimum delay time values for individual IPDUs as well as the ECU-specific definition of whether the defined minimum delay times are to be applied to event-based PDU transmissions only or also to cyclic PDU transmissions.

The FlexRay Configuration Tool displays the minimum delay time information in the Properties view.



Basics on minimum delay time support

You can enable minimum delay time support for the IPDUs for which minimum delay time information is defined in the AUTOSAR system description file. If minimum delay time support is enabled for an IPDU, the current delay time is calculated when the IPDU is committed. A new transmission of this IPDU is not possible until the minimum delay time expires.

Using the minimum delay time is useful for TX IPDUs when a debounce time is to be considered. Enabled minimum delay time support prevents the IPDU from being sent in too quick succession. However, in cases where transmission cannot be paused (such as for a static FlexRay bus segment), at least the sending of new information is guaranteed.

Enabling minimum delay time support In the FlexRay Configuration Tool, for each TX IPDU in a configuration for which minimum delay time information is available, you must specify whether to use this information or not. To include the minimum delay time, you must enable minimum delay time support for the IPDU. If you disable minimum delay time support for an IPDU, its minimum delay time information specified in the AUTOSAR system description file is always ignored.

The FlexRay Configuration Tool lets you enable or disable minimum delay time support for all the possible IPDUs in the project in one step. If a TX IPDU should use a different setting, you can individually specify a PDU-specific setting.

Switching minimum delay time support during run time For all the IPDUs, for which minimum delay time support is enabled in the FlexRay Configuration Tool, you can disable or enable the minimum delay time support at run time via the Simulink model or in ControlDesk.

- By default, Simulink interfaces to switch the minimum delay time support via the model are generated during model generation. You do not have to enable interface generation specifically.
- To switch the minimum delay time support in ControlDesk, suitable variables must be included in the TRC file. You must use the Element Selection dialog for this. Refer to How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File on page 134.

Minimum delay time support and transmission modes The availability and applicability of the minimum delay time feature depends on the PDU transmission mode used. The minimum delay time feature is available and can be enabled if the 'True' (1) or 'False' (0) transmission mode is used. The minimum delay time feature cannot be used with PDUs for which the transmission mode 'User-Defined' (99) or 'LPDU timing triggered' (98) is currently selected.

Restrictions

- Creating FlexRay configurations with minimum delay time support is possible only in connection with AUTOSAR communication cluster files based on a FlexRay Configuration Tool supported AUTOSAR version ≥ 4.0.3 or AUTOSAR Classic Platform Release R19-11 or R20-11. For information on the supported AUTOSAR Releases, refer to Communication Cluster Files Usable for Configuration on page 26.
- The minimum delay time feature is supported only for PDUs for which a minimum delay time > 0.0 s and at least one IPDU timing (cyclic, event-controlled) is defined in the communication matrix.
- Minimum delay time support can be configured only for PDUs with 'Application' frame content type.
- There are a few limitations on minimum delay time support. Refer to Limitations on page 141.

Preconditions

You must first import an AUTOSAR system description file containing minimum delay time definitions for PDUs. Refer to How to Import a Communication Cluster File on page 40.

Method

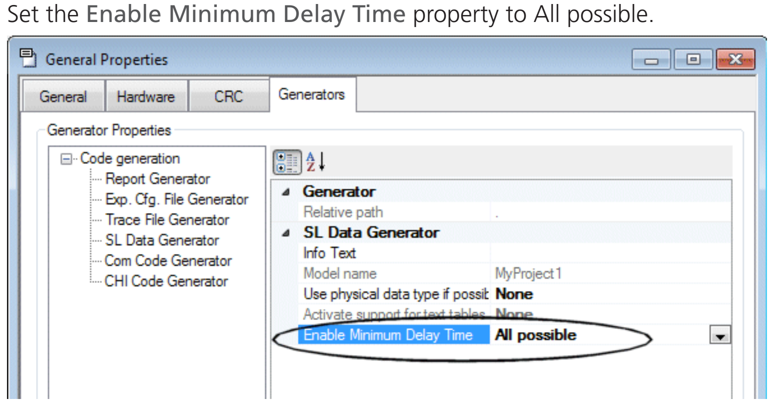
To configure PDUs for minimum delay time support

1 To enable minimum delay time support simultaneously for all the possible PDUs in the current FlexRay configuration at one time, choose Tools - General Properties from the menu bar of the dSPACE FlexRay Configuration Tool.

2 In the General Properties dialog, select the Generators page.

3 Select SL Data Generator.

4 Set the Enable Minimum Delay Time property to All possible.



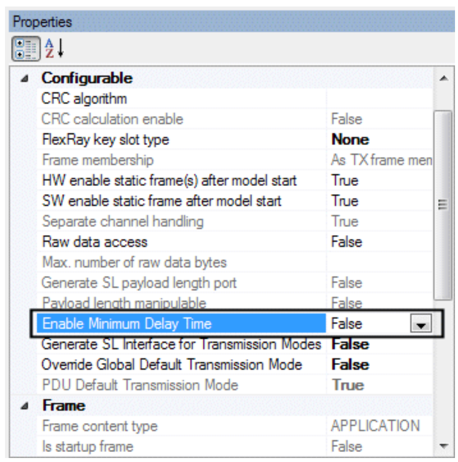
Note

Make this selection only if minimum delay time support is to be enabled for most of the possible PDUs. If minimum delay time support is to be enabled for just a few PDUs in the current FlexRay configuration, you should enable minimum delay time support individually for these PDUs (refer to step 6). The property filter helps you locate them.

5 Click Keep changes to confirm your settings and close the dialog.
   You can now specify PDU-specific minimum delay time support settings.

6 In the Configuration view, select a PDU.

7 In the Properties view, specify the minimum delay time support setting to be used for the selected PDU via the Configurable - Enable Minimum Delay Time property.



The Configurable - Enable Minimum Delay Time property exists only for PDUs for which minimum delay time support could be enabled.

8 Repeat steps 6 ... 7 for all the PDUs you want to specify PDU-specific minimum delay time support settings for.

9 If you want to enable or disable the minimum delay time support of TX PDUs during run time in ControlDesk, suitable variables must be generated in the TRC file. To include the variables to the TRC file, perform the following steps:
   - From the menu bar, choose Tools – Element Selection to open the Element Selection dialog.
   - In the features list, select the Frame Minimum Delay Time Control feature.

     Available Elements lists all the PDUs that are configured or available for the selected feature and not selected for the TRC file yet.
   - Drag the PDU onto the TRC Interface item in Selected Elements.
   - Repeat these steps for all the TX PDUs whose minimum delay time support you want to control via TRC file.
   - Click OK.

Result

Minimum delay time support for the TX PDUs is configured. Simulink interfaces for enabling or disabling minimum delay time support will be created in the model generation process, and entries for controlling the minimum delay time support will be generated into the TRC file.

Related topics

Basics

Using the Generated TRC File of PDU-Based Modeling (FlexRay Configuration Features 🕮)
Using the Generated TRC File of PDU-Based Modeling in ControlDesk (Model Interface Package for Simulink - Modeling Guide 🕮)

References

FLEXRAYCONFIG PDU TX (FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)

Disabling the Communication of Simulated ECUs

Introduction

To simulate an ECU malfunction or to replace them by the real ECUs, the communication of simulated ECUs must be disabled. This means that the controllers or buffers which transmit an ECU's frames are disabled.

Where to go from here

Information in this section

Information in other sections

Building Frame Membership Groups

Introduction

The communication of simulated ECUs can be disabled to simulate an ECU malfunction or to replace them by the real ECUs.

Basics

Disabling the communication of simulated ECUs can be useful for:

- Simulating ECU malfunction
- Disabling the ECU simulation and connecting the real ECU to the FlexRay network

Disabling the communication of a simulated ECU is possible at controller or buffer level. When disabling the ECU communication, you must stop the sending of frames by disabling either the controller(s) or the buffer(s) they are assigned to. You can specify the disable mode (controller, buffer) in the FlexRay Configuration Tool for each ECU.

You can monitor the frames of disabled controllers if you set the monitor frame membership accordingly (see below).

Building frame membership groups

Normally, the FlexRay Configuration Tool minimizes the number of controllers needed for the simulation by assigning as many frames as possible to a controller. It is possible that TX frames of different ECUs are assigned to the same controller. In that case, you cannot disable communication of only a specific ECU by disabling the controller.

To avoid affecting other ECUs, the controller must transmit only frames of the ECU to be disabled. You can group the frames by assigning frame memberships. You can specify different frame memberships for single ECUs or ECU groups, which means that the FlexRay Configuration Tool then assigns only frames of ECUs with the same frame membership group to a controller.

Additionally, you can specify whether to disable the communication of the simulated ECU via controller or buffer. In controller mode, the different frame memberships are assigned to different controllers or controller groups. In buffer mode, the different memberships are assigned to the same controller or controller group. The controller mode is more run-time-efficient, and the buffer mode is more resource-efficient.

ECUs can have the following frame memberships:

- TX frame membership
- RX frame membership

TX frame membership The TX frame membership is used to assign a membership value to an ECU. This value is used for the frame-to-buffer assignment. All TX frames of the ECU will be assigned to the same controller or controller group if one controller is not sufficient.

RX frame membership The RX frames of an ECU can be assigned to the TX frame membership or a global RX pool which is independent and can be assigned to a different controller. If the RX frames are assigned to the TX frame membership, no frames can be received when the controller is disabled. If they are assigned to the global RX pool, frames can be received even if the controller sending the TX frames is disabled. The receive frames assigned to the global RX pool cannot be disabled.

Note

The behavior described above applies to using the **ECU disable mode** set to 'Controller'. If the **ECU disable mode** is set to 'Buffer', the behavior differs: The reception of RX PDUs is not disabled if the corresponding TX frame membership is disabled, i.e., RX PDUs can always be received.

As a consequence, no **RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE** block will be generated if all the ECUs of a TX frame membership receive RX PDUs and/or Monitoring PDUs only.

If a frame is configured to be transmitted and received, and the RX frame is assigned to the TX frame membership, the frame will not be received from the bus but from the internal TX frame memory. External reception can be guaranteed only if this RX frame is not assigned to the TX frame membership.

The same applies to frames of a monitored ECU where **Monitoring frame membership** replaces **RX frame membership**.

Frame memberships for frames The RX pool controller is used for receiving frames, but it could also be used to send a sync or startup-sync frame if needed (e.g. when no more controllers are available or no other ECU in the network sends a second startup-sync frame). You can therefore also assign frame membership values to the frames. However, this can only be done with the Send-Sync or Startup-Sync frames. The **frame membership** property of the other frames is not configurable. You can assign the frame membership **As TX frame membership** or **Global RX pool** to the Send-Sync or Startup-Sync frames.

Example

On a bus with four ECUs (A, B, C, and D), the frames of the ECUs A and B are selected in the FlexRay Configuration Tool. The tool minimizes the number of controllers needed for the simulation by assigning as many frames as possible to a controller. Since all frames have membership 1 by default, they can be assigned to the same controller. For this example, it is assumed that ECU communication is disabled at controller level, and that all frames can be assigned to one controller.

If you want to simulate the failure of ECU B during simulation, you cannot disable the communication of ECU B by disabling the controller because this would disable the communication of ECU A as well.

If the frames of ECU B are assigned to a different membership, the frames are assigned to a different controller than the frames of ECU A. Disabling the communication of ECU B separately is possible.

Enabling and disabling frame membership groups

The frame membership groups are enabled or disabled in the **RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE** block, or (for MicroAutoBox III or SCALEXIO) via the **FlexRay** function block in ConfigurationDesk.

Related topics

HowTos

How to Specify Frame Memberships..93

References

FlexRay (ConfigurationDesk Function Block Properties 🕮)
RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE (RTI FlexRay Configuration Blockset Reference 🕮)

How to Specify Frame Memberships

Objective

The communication of simulated ECUs can be disabled to simulate an ECU malfunction or to replace them by the real ECUs.

Precondition

The Configuration/Monitoring and the Properties views are open.

Method

To specify frame memberships

1. In the Configuration or Monitoring view, select the ECU you want to set the frame membership for.
2. In the Properties view, select the ECU disable mode for the simulated ECU:
   - Controller: The communication of the ECU is disabled by disabling the controller(s).
   - Buffer: The communication of the ECU is disabled by disabling the buffer(s).

   Note

   - For ECUs with sync frame or startup-sync frame, only the 'Controller' mode is available.
     If you want to activate the 'Buffer' mode for these ECUs, you must set the FlexRay key slot type parameter for the sync or startup-sync frame to 'None'. As an alternative, you can specify 'Global RX pool' as the frame membership for the sync or startup-sync frame. In this case, the sync or startup-sync frame will not be disabled.
   - If the ECU disable mode is configured differently for the ECUs belonging to a TX frame membership group (some ECUs use the 'Controller' mode, others use the 'Buffer' mode), the FlexRay Configuration Tool sets the 'Controller' mode for the TX frame membership. This is done for time optimization purposes.

3. Select the TX frame membership group.

   Note

   To allow ECU communication to be enabled/disabled during run time, you must assign a membership group ID to the ECU. Selecting "NONE" as the TX frame membership group indicates that the communication of the selected ECU cannot be disabled during run time. In that case, neither an RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block (for Simulink models) nor Membership and Membership State function ports for the FlexRay function block in ConfigurationDesk (for MicroAutoBox III or SCALEXIO) will be generated for the membership group.

4 Select the RX frame membership group or Monitoring frame membership group:
   - **As TX frame membership**: When the membership is disabled, the RX frames and TX frames are disabled together.
   - **Global RX pool**: When the membership is disabled, only TX frames are disabled.

Result

The frame membership group for the ECU is set. You can disable the communication of the selected ECU in your Simulink model (refer to How to Enable or Disable the Communication of an ECU (FlexRay Configuration Features 📖)) or via ConfigurationDesk (refer to Common FlexRay Functions (Model Interface Package for Simulink - Modeling Guide 📖)).

Related topics

Basics

Building Frame Membership Groups.. 90

References

Configuration View (FlexRay Configuration Tool Reference 📖)
FlexRay (ConfigurationDesk Function Block Properties 📖)
Monitoring View (FlexRay Configuration Tool Reference 📖)
Properties View (FlexRay Configuration Tool Reference 📖)
RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE (RTI FlexRay Configuration Blockset Reference 📖)

Using User-Defined Checksum Algorithms

Introduction

You can implement your own checksum algorithms in the real-time application. Several checksum algorithms can be included as C code in a special template. The checksum algorithm which is used is selected in a dialog of the FlexRay Configuration Tool.

Where to go from here

Information in this section

Basics on Implementing Checksum Algorithms

Basics

You can implement your own checksum algorithms (CRC algorithms). You must implement the CRC algorithms in a C-coded source file (CRC C file) using a special template. In the FlexRay Configuration Tool, you can assign the CRC algorithm to TX or RX frames. When the FlexRay model has been automatically generated, it contains RTI blocks which can be used to enable or disable checksum calculation or to select another CRC algorithm.

Implementing a CRC C file

The CRC algorithms must be implemented in the CRC C file. You can implement several CRC algorithms in this file. To select one CRC algorithm for checksum calculation, a switch-case construct is used. The value which switches between the CRC algorithms is defined in the FlexRay Configuration Tool. For implementation details, refer to Implementing Checksum Algorithms on page 100.

Assigning CRC algorithm to frames

You can specify names for the CRC algorithms and assign them to TX or RX frames in the FlexRay Configuration Tool. The assigned CRC algorithms are the default CRC algorithm used for the frames. It can be changed in the Simulink model afterwards. You cannot enable CRC calculation for a frame in the Simulink model if no CRC algorithm was assigned to it. For details on assigning CRC algorithm, refer to How to Assign a Checksum Algorithm to Frames on page 97.

CRC algorithm in the model

If you have not assigned CRC algorithms to frames in the FlexRay Configuration Tool, it is not possible to use a CRC algorithm for these frames in your Simulink model. If you have assigned CRC algorithms to frames, you can change the CRC algorithms used for these frames in your Simulink model.

The automatically generated FlexRay model contains the RTIFLEXRAYCONFIG PDU TX and RTIFLEXRAYCONFIG PDU RX blocks (or FLEXRAYCONFIG PDU TX and FLEXRAYCONFIG PDU RX blocks) for sending and receiving PDUs. These blocks have inports that handle the CRC algorithms. You can enable or disable the calculation of the CRC algorithms. You can also select another CRC algorithm using an inport or a block parameter. For details, refer to:

- How to Handle Checksum Calculation for a PDU (FlexRay Configuration Features 🕮)
- (MicroAutoBox III, SCALEXIO) How to Handle Checksum Calculation for a PDU (Model Interface Package for Simulink - Modeling Guide 🕮)

Basics of frame ID

If you want to include frame-specific frame IDs in the CRC algorithm, you can use a frame ID file. This file associates frames with user-defined integer IDs. It is selected on the **CRC** page in the **General Properties** dialog of the FlexRay Configuration Tool. For detailed information on its structure, refer to Structure of a Frame ID File on page 110.

For detailed information on accessing the frame IDs in the CRC algorithm, refer to Implementing Checksum Algorithms on page 100.

Related topics

References

CRC Page (FlexRay Configuration Tool Reference 📖)
FLEXRAYCONFIG PDU RX (FlexRay Configuration Blockset Reference 📖)
FLEXRAYCONFIG PDU TX (FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU RX (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 📖)

How to Assign a Checksum Algorithm to Frames

Objective

When you implement your own checksum algorithms, you can select one of them for transmitting frames from and to your Simulink model.

Basics

To enable checksum calculation for a frame, you must assign a checksum algorithm to it. Additionally, you must select a signal of the frame to carry the checksum (CRC signal).

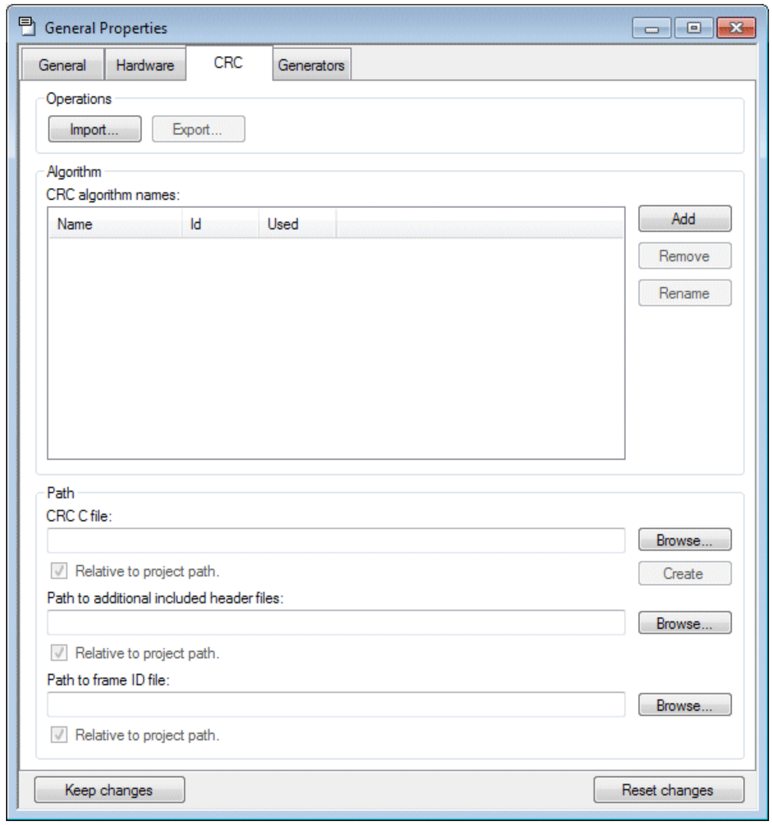
The checksum algorithms must be implemented in a C file. The FlexRay Configuration Tool can create a template in which you can implement your checksum algorithms. For details, refer to Implementing Checksum Algorithms on page 100 and Implementing Checksum Algorithms for Signal Groups Protected via End-To-End Communication Protection on page 104.

Method

To assign a checksum algorithm to frames

1 From the menu bar of the dSPACE FlexRay Configuration Tool, choose Tools - General Properties.

2 In the **General Properties** dialog, select the **CRC** page.



3 Click **Add** for each algorithm which you have in the CRC C file.

The dSPACE FlexRay Configuration Tool adds a row for each added algorithm to the **CRC algorithm names** list.

4 In the **CRC algorithm names** list, select each algorithm and click **Rename** to specify proper names for them.

5 If the CRC C file already exists, click **Browse** and select your CRC C file.

6 If the CRC C file does not yet exist, specify the path and a file name for it in the **CRC C file** edit field and click **Create**.

The FlexRay Configuration Tool creates a template for the CRC C file with the specified file name. You can implement the checksum algorithms and select the CRC signal in this file. For details, refer to Implementing Checksum Algorithms on page 100.

> **Tip**
>
> If you want to work with a relative path, select the **Relative to project path** checkbox for the CRC C file. If selected, the CRC file is saved in a path relative to the configuration project file.

7 If you have included additional header files in your CRC C file, specify the path to them. If the **Relative to project path** checkbox is selected, the path to the files is relative to the project file path. If the checkbox is cleared, the path to the additional header files is absolute.

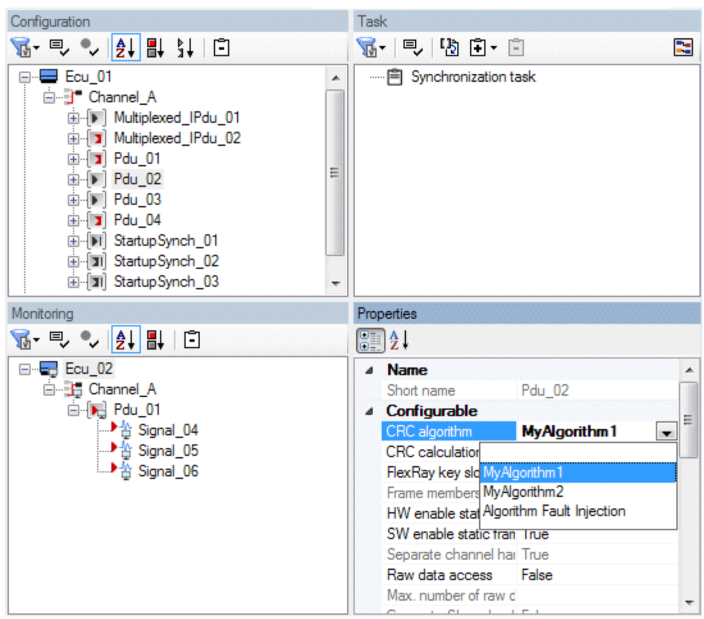
8 If you want to include frame IDs in your CRC file, specify the path to the frame ID file. Click **Browse** and select your frame ID file. If the **Relative to project path** checkbox is selected, the path to the file is relative to the project file. If the checkbox is cleared, the path to the frame ID file is absolute. For details on a frame ID file, refer to Structure of a Frame ID File on page 110.

9 Click **Keep changes** to confirm your settings and close the dialog.

You can now assign the CRC algorithm to frames.

10 In the **Configuration** view, select a frame.

11 In the **Properties** view, select a CRC algorithm in the **CRC algorithm** field.



The CRC algorithm is selected for the frame. The **CRC calculation enable** field is set to true.

12 In the **Configuration** view, select a signal of the frame to carry the checksum.

13 In the **Properties** view of the CRC signal, set the **Used for CRC calculation** property to True.

14 Repeat the previous steps for all frames.

Result

The CRC algorithms are assigned to the frames. These are the default CRC algorithms for the frames. You can select other CRC algorithms in the model afterwards.

Related topics

Basics

HowTos

References

Implementing Checksum Algorithms

Introduction

You can implement several checksum algorithms and select one of them afterwards. The checksum algorithms must be implemented in the C programming language, the CRC C file.

A CRC algorithm can be applied to frames and PDUs. This means that the term 'frame' relates to a frame or PDU, depending on the used communication cluster. In FIBEX 2.0 and lower versions, the CRC algorithm is applied to frames. In FIBEX+, FIBEX 3.0 or later, and AUTOSAR System Templates, the CRC algorithm is applied to PDUs.

This topic describes the CRC C file, which is delivered with the dSPACE FlexRay Configuration Package or is created by the FlexRay Configuration Tool on the **CRC** page in the **General Properties** dialog (see How to Assign a Checksum Algorithm to Frames on page 97). The code is clearly organized in several parts (see below).

You can find the complete CRC C file template (`crc_template.c`) in the following folders:

- <RCP_HIL_InstallationPath>\MATLAB\RTIFLEXRAYCONFIG\FlexRayAL after installation of the RTI FlexRay Configuration Blockset
- <RCP_HIL_InstallationPath>\MATLAB\FLEXRAYCONFIG\C after installation of the FlexRay Configuration Blockset

The template is not ready to use, but you can use it as a basis for implementing your own checksum algorithms.

Tip

If you want to implement checksum algorithms for signal groups that are protected via AUTOSAR end-to-end communication protection, use the `crc_template_e2e_protection.c` template provided by the dSPACE FlexRay Configuration Package instead. Refer to Implementing Checksum Algorithms for Signal Groups Protected via End-To-End Communication Protection on page 104.

Data types

The following data types are defined in the `dsftcom_crc.h` header file.

dsftcomFrameCrcStatus The dsftcomFrameCrcStatus data type is used for the return value of the checksum function. It can have the following values:

- DSFTCOM_CRC_OK: Checksum of the received frame is correct or checksum is added to the frame to be sent.
- DSFTCOM_CRC_NOT_OK: Checksum of the received frame is not correct or frame has no checksum.

dsftcomCRCFrameData The dsftcomCRCFrameData structure has all the data of the received frame or frame to be sent. It has the following elements:

Name	Data Type	Description
data	UInt8*	Pointer to raw data of the frame
payload_length	UInt8	Frame length
slot_id	UInt32	Frame slot ID
base_cycle	UInt32	Frame base cycle
cycle_repetition	UInt32	Frame cycle repetition
short_name	const char*	Pointer to short name of the frame
user_defined_id	UInt32	Frame ID from the frame ID file

1st part

```
#include "dsftcom_crc.h"
dsftcomFrameCrcStatus dsftcom_crc(
   UInt32 crcOption,
   dsftcomCRCFrameData* pFrame,
   UInt32 crcType,
   UInt32* pCsBitPos,
   UInt32* pCsLength,
   UInt32 signalCount)
{
```

Description To get all the necessary definitions, you must include the `dsftcom_crc.h` header file. You can find it in the following folders:

- `<RCP_HIL_InstallationPath>\MATLAB\RTIFLEXRAYCONFIG\FlexRayAL` after installation of the RTI FlexRay Configuration Blockset

- <RCP_HIL_InstallationPath>\SCALEXIO\Include and <RCP_HIL_InstallationPath>\DS1403\Include after installation of the FlexRay Configuration Blockset

The CRC function must be called `dsftcom_crc` and has the following parameters:

Parameter	Description
`crcOption`	Indicates whether the frame is sent (crcOption == 0) or received (crcOption == 1).
`pFrame`	Pointer to a structure containing data of the frame
`crcType`	Specifies the checksum algorithm which is used. The value is specified on the **CRC** page of the FlexRay Configuration Tool (see CRC Page (FlexRay Configuration Tool Reference)).
`pCsBitPos`	Specifies an array containing the bit positions of the signals where you can save the CRC values to or read the CRC values from. The start bit positions of the CRC signals are always specified relative to the frame. This also applies to CRC signals that are included in subframes.
`pCsLength`	Specifies an array containing the lengths of the signals where you can save the CRC values to or read the CRC values from.
`signalCount`	Specifies the number of CRC signals belonging to the PDU. It is also the length of the `pCsBitPos` and `pCsLength` arrays. The arrays are sorted by start bit positions in ascending order, i.e., the arrays start with the CRC signal with the lowest start bit position.

2nd part

```
UInt32 FrameLength = pFrame->payload_length;
/* Example Calculate byte position of CRC signal in frame */
UInt32 I;
UInt32 CheckSum = 0;
UInt32 CsBitRem = (pCsBitPos[0] % 8);
UInt32 CsBytePos = (pCsBitPos[0] - CsBitRem)/8;

// Contains the frame a CRC signal?
if (signalCount == 0)
{
  // if not: return with error
  return(DSFTCOM_CRC_NOT_OK);
}
```

Description The bit and byte positions of the first CRC signal (index 0) are calculated. Only one CRC signal is used in this template. The `CheckSum` and `FrameLength` parameters are initialized.

A check is made on whether the frame has a CRC signal. If it has no CRC signal (signalCount == 0), the function is aborted and the DSFTCOM_CRC_NOT_OK value is returned.

3rd part

```
// Calculate CRC in dependency of CRC type
switch(crcType)
{
   case 0: // Algorithm 0
   {
      // Insert code for algorithm 0
      /* Example: XOR */
      for (UInt32 I = 0; I < FrameLength; I++)
      {
         if (I != CsBytePos)
         {
            CheckSum = CheckSum ^ pFrame->data[i];
         }
      }
   }; break;
   case 1: // Algorithm 1
   {
      // Insert code for algorithm 1
      /* Example: XOR with Slot Id */
      for (I = 0; I < FrameLength; I++)
      {
         if (I != CsBytePos)
         {
            CheckSum = CheckSum ^ pFrame->data[i];
         }
      }
      CheckSum = CheckSum ^ pFrame->slot_id;
   }; break;
   case 2: // Algorithm 2
   {
      // Insert code for algorithm 2
      /* Example: Write wrong value to CRC signal (fault injection) */
      CheckSum = pFrame->data[CsBytePos] + 1;
   }; break;
   default: // Undefined CRC type
   {
      return(DSFTOM_CRC_NOT_OK);
   };
}
```

Description In this part the checksum is calculated. The switch-case construct selects the algorithm which is used for calculation. The value for the selection (`crcType`) is specified on the CRC page of FlexRay Configuration Tool (see CRC Page (FlexRay Configuration Tool Reference 📖)). The value for the `crcType` corresponds to the `ID` on the CRC page. The value of `ID` is set automatically by the FlexRay Configuration Tool. It starts with 0 and is incremented by 1 for each CRC algorithm. In this example, you can switch between three CRC algorithms. If none of the implemented algorithms is selected, the DSFTCOM_CRC_NOT_OK value is returned.

4th part

```
    // Is frame sent?
    if(crcOption == 0)
    {
        // Insert code for writing CRC into frame
        /* Example: Write CRC into frame */
        pFrame->data[CsBytePos] = CheckSum;
        return(DSFTCOM_CRC_OK);
    }
    else
    {
        // Insert code for CRC check
        /* Example: Compare calculated sum with value of CRC signal in frame */
        if (pFrame->data[CsBytePos] != CheckSum)
        {
            return(DSFTCOM_CRC_NOT_OK);
        }
        return(DSFTCOM_CRC_OK);
    }
}
```

Description In the last part, the checksum is added to the frame if the frame is sent, or the calculated checksum is compared with the received checksum if the frame is received. If it is correct, the DSFTCOM_CRC_OK value is returned. If it is incorrect, the DSFTCOM_CRC_NOT_OK value is returned.

Related topics

Basics

Implementing Checksum Algorithms for Signal Groups Protected via End-To-End Communication Protection

Introduction

You can implement checksum algorithms for signal groups that are protected via AUTOSAR end-to-end communication protection (E2E protection). The dSPACE FlexRay Configuration Package supports run-time access to E2E protection parameters from AUTOSAR communication matrices. The parameters are required to calculate signal group CRCs.

You can implement several checksum algorithms for signal groups and select one of them during run time.

Note

Signal group CRCs are supported in connection with communication matrix information based on AUTOSAR 3.2.1, 3.2.2, 4.0.3, 4.1.1, 4.1.2, 4.2.1, 4.2.2, 4.3.0, 4.3.1, 4.4.0, and AUTOSAR Classic Platform Release R19-11 and R20-11 system description files.

The dSPACE FlexRay Configuration Package provides a CRC file (`crc_template_e2e_protection.c`), which you can use as a template to implement your own checksum algorithms based on E2E protection parameters in the real-time application. The template is not ready to use, but you can use it as a basis for implementing your own checksum algorithms. You can find the template in the following folders:

- <RCP_HIL_InstallationPath>\MATLAB\RTIFLEXRAYCONFIG\FlexRayAL after installation of the RTI FlexRay Configuration Blockset
- <RCP_HIL_InstallationPath>\MATLAB\FLEXRAYCONFIG\C after installation of the FlexRay Configuration Blockset

The template's code is clearly organized in several parts.

The template for the CRC C file, which is created by the FlexRay Configuration Tool on the CRC page (see How to Assign a Checksum Algorithm to Frames on page 97), contains sections for implementing CRCs for signal groups (see below).

Tip

To implement checksum algorithms without E2E protection, use the `crc_template.c` file instead. Refer to Implementing Checksum Algorithms on page 100.

Data types

The following data types are defined in the `dsftcom_crc.h` header file.

dsftcomFrameCrcStatus The dsftcomFrameCrcStatus data type is used for the return value of the checksum function. It can have the following values:

- DSFTCOM_CRC_OK: Checksum of the received PDU is correct or checksum is added to the PDU to be sent.
- DSFTCOM_CRC_NOT_OK: Checksum of the received PDU is not correct or PDU has no checksum.

dsftcomCRCFrameData The dsftcomCRCFrameData structure has all the data of the received PDU or PDU to be sent. It has the following elements:

Name	Data Type	Description
data	UInt8*	Pointer to raw data of the PDU
payload_length	UInt8	PDU length
slot_id	UInt32	Frame slot ID

Name	Data Type	Description
base_cycle	UInt32	Frame base cycle
cycle_repetition	UInt32	Frame cycle repetition
num_crc_signal_groups	UInt32	Number of signal groups with E2E protection
crc_signal_groups	dsftcomCRCSignalGroupData*	Pointer to signal groups of the PDU that are protected via E2E protection
short_name	const char*	Pointer to short name of the PDU
user_defined_id	UInt32	Frame ID from the frame ID file

dsftcomCRCSignalGroupData The dsftcomCRCSignalGroupData structure has all the data of the transmitted signal group protected via AUTOSAR E2E protection. It has the following elements:

Name	Data Type	Description
data_offset	UInt32	Bit offset of array representing the signal group
counter_offset	UInt32	Bit offset of counter
crc_offset	UInt32	Bit offset of CRC byte in the signal group
max_delta_counter_init	UInt32	Maximum counter incrementation. The value describes the allowed gap between two counter values of consecutively received data.
data_length	UInt32	Length (in bits) of array representation of signal group (including CRC and counter bits) The value must be ≤ 240 and a multiple of 8.
data_id_array[16]	UInt8	Array with data IDs. A data ID is a unique identifier used for protection against masquerading.
category	const char*	Pointer to category of the signal group. The category represents the identification of a concrete E2E protection profile.
data_id_mode	UInt32	Inclusion mode of data ID in CRC computation (two bytes, one byte (alternating high and low byte), or only low byte)

1st part

```
#include "dsftcom_crc.h"
dsftcomFrameCrcStatus dsftcom_crc(
   UInt32 crcOption,
   dsftcomCRCFrameData* pFrame,
   UInt32 crcType,
   UInt32* pCsBitPos,
   UInt32* pCsLength,
   UInt32 signalCount)
{
```

Description To get all the necessary definitions, you must include the `dsftcom_crc.h` header file. You can find it in the following folders:

- `<RCP_HIL_InstallationPath>\MATLAB\RTIFLEXRAYCONFIG\FlexRayAL` after installation of the RTI FlexRay Configuration Blockset
- `<RCP_HIL_InstallationPath>\SCALEXIO\Include` and `<RCP_HIL_InstallationPath>\DS1403\Include` after installation of the FlexRay Configuration Blockset

The CRC function must be called **dsftcom_crc** and has the following parameters:

Parameter	Description
`crcOption`	Indicates whether the PDU is sent (crcOption == 0) or received (crcOption == 1).
`pFrame`	Pointer to a structure containing data of the PDU.
`crcType`	Specifies the checksum algorithm which is used. The value is specified on the **CRC** page of the FlexRay Configuration Tool (see CRC Page (FlexRay Configuration Tool Reference)).
`pCsBitPos`	Specifies an array containing the bit positions of the signals where you can save the CRC values to or read the CRC values from. The start bit positions of the CRC signals are always specified relative to the PDU. This also applies to CRC signals that are included in sub-PDUs.
`pCsLength`	Specifies an array containing the lengths of the signals where you can save the CRC values to or read the CRC values from.
`signalCount`	Specifies the number of CRC signals belonging to the PDU. It is also the length of the **pCsBitPos** and **pCsLength** arrays. The arrays are sorted by start bit positions in ascending order, i.e., the arrays start with the CRC signal with the lowest start bit position.

2nd part

```
/* Get number of signal groups with end2end protection */
UInt32 NumCRCSignalGroups = pFrame->num_crc_signal_groups;
/* Get pointer on signal groups CRC data from end2end protection description */
/* For definition of dsftcomCRCSignalGroupData see dsftcom_crc.h */
dsftcomCRCSignalGroupData *crcSignalGroupDataPointer = pFrame->crc_signal_groups;
/* Example Calculate byte position of CRC signal in Pdu */
UInt32 i;
UInt8  CheckSum;
// Contains the Pdu a CRC signal group?
if (NumCRCSignalGroups == 0)
{
  // if not: return with error
  return(DSFTCOM_CRC_NOT_OK);
}
```

Description The number of signal groups with E2E protection is determined, and the E2E protection description data is retrieved.

A check is made on whether the PDU has a CRC signal group. If it has no CRC signal group (num_crc_signal_groups == 0), the function is aborted and the DSFTCOM_CRC_NOT_OK value is returned.

3rd part

```
// Calculate CRC in dependency of CRC type
switch(crcType)
{
   case 0: // Algorithm 0
   {
      // Insert code for algorithm 0
      for (i = 0; i < NumCRCSignalGroups; i++)
      {
         if (strcmp("PROFILE_01", crcSignalGroupDataPointer[i].category) == 0)
         {
            CheckSum = CalculateCheckSumProfile1(pFrame->data,
                           crcSignalGroupDataPointer[i].data_length,
                           crcSignalGroupDataPointer[i].data_offset,
                           crcSignalGroupDataPointer[i].data_id_array,
                           crcSignalGroupDataPointer[i].crc_offset);
         }
         else if (strcmp("USER_SPECIFIC_PROFILE_01", crcSignalGroupDataPointer[i].category) == 0)
         {
            CheckSum = CalculateCheckSumProfileUser(pFrame->data,
                           crcSignalGroupDataPointer[i].data_length,
                           crcSignalGroupDataPointer[i].data_offset,
                           crcSignalGroupDataPointer[i].data_id_array,
                           crcSignalGroupDataPointer[i].data_id_mode,
                           crcSignalGroupDataPointer[i].crc_offset);
         }
         // Is Pdu sent?
         if(crcOption == 0)
         {
            // Insert code for writing CRC into Pdu
            pFrame->data[crcSignalGroupDataPointer[i].crc_offset/8] = CheckSum;
         }
         else
         {
            // Insert code for CRC check
            if (pFrame->data[crcSignalGroupDataPointer[i].crc_offset/8] != CheckSum)
            {
               return(DSFTCOM_CRC_NOT_OK);
            }
         }
      }; break;
   }
```

```
    case 1: // Algorithm 1
    {
      // Insert code for algorithm 1
      // error injection
      for (i = 0; i < NumCRCSignalGroups; i++)
      {
        if (strcmp("PROFILE_01", crcSignalGroupDataPointer[i].category) == 0)
        {
          CheckSum = CalculateCheckSumProfile1(pFrame->data,
                      crcSignalGroupDataPointer[i].data_length,
                      crcSignalGroupDataPointer[i].data_offset,
                      crcSignalGroupDataPointer[i].data_id_array,
                      crcSignalGroupDataPointer[i].crc_offset);
        }
        else if (strcmp("USER_SPECIFIC_PROFILE_01", crcSignalGroupDataPointer[i].category) == 0)
        {
          CheckSum = CalculateCheckSumProfileUser(pFrame->data,
                      crcSignalGroupDataPointer[i].data_length,
                      crcSignalGroupDataPointer[i].data_offset,
                      crcSignalGroupDataPointer[i].data_id_array,
                      crcSignalGroupDataPointer[i].data_id_mode,
                      crcSignalGroupDataPointer[i].crc_offset);
        }
        // Is Pdu sent?
        if(crcOption == 0)
        {
          // Insert code for writing CRC into Pdu
          // Fault injection
          CheckSum = CheckSum + 1;
          pFrame->data[crcSignalGroupDataPointer[i].crc_offset/8] = CheckSum;
        }
        else
        {
          return(DSFTCOM_CRC_NOT_OK);
        }
      }; break;
      default: // Undefined CRC type
      {
        return(DSFTCOM_CRC_NOT_OK);
      };
    }
    return(DSFTCOM_CRC_OK);
  }
```

Description In this part the checksum is calculated. The switch-case construct selects the algorithm which is used for calculation. The value for the selection (`crcType`) is specified on the CRC page of the FlexRay Configuration Tool (see CRC Page (FlexRay Configuration Tool Reference 📖)). The value for the `crcType` corresponds to the `ID` on the CRC page. The value of `ID` is set automatically by the FlexRay Configuration Tool. It starts with 0 and is incremented by 1 for each CRC algorithm. In this example, you can switch between two CRC algorithms. If neither of the implemented algorithms is selected, the DSFTCOM_CRC_NOT_OK value is returned.

The check against the `category` parameter is performed to check the used E2E protection profile. The `category` parameter represents the identification of a concrete E2E protection profile. AUTOSAR E2E protection provides several profiles. A profile determines certain CRC criteria, such as the CRC function and the position of the CRC signal. There is also a user-defined profile, which allows you to specify these CRC settings yourself. For further information, refer to the AUTOSAR specification.

The actual checksum generation is done in special functions such as `CalculateCheckSumProfile1()`. These functions are used in the template's code to ensure that the code does not become too complicated. This part does not focus on the concrete implementation of the checksum algorithms, because the CRC function is determined by a concrete AUTOSAR E2E protection profile.

The checksum is added to the PDU if the PDU is sent, or the calculated checksum is compared with the received checksum if the PDU is received. If it is correct, the DSFTCOM_CRC_OK value is returned. If it is incorrect, the DSFTCOM_CRC_NOT_OK value is returned.

Related topics

Basics

Aspects of Miscellaneous Supported AUTOSAR Features..29
Basics on Implementing Checksum Algorithms..95
Implementing Checksum Algorithms..100
Using User-Defined Checksum Algorithms...95

Structure of a Frame ID File

Introduction

You can define a frame ID file which can be included in the CRC configuration. It can also be used for PDUs. PDUs are defined in FIBEX+, FIBEX 3.x, and FIBEX 4.x versions, and in the AUTOSAR System Template.

Basics

A frame ID file is an XML file in which you can assign an ID to each frame or PDU. A frame ID file has the following structure:

```
<frame-config>
  <frames>
    <frame>
       <name>frame name</name>
       <id>frame ID</id>
    </frame>
    ... further frames
  </frames>
</frame-config>
```

Input	Description
frame name	The frame name must be the short name of the frame/PDU in the FIBEX or AUTOSAR system description file. Each frame name can only exist once in the frame ID file.

Input	Description
frame ID	The frame ID is the ID to be assigned to the frame. Each ID can be assigned to more than one frame/PDU. The default ID of the frames without an assigned ID is *0*.

The XML file can be included in the CRC algorithm on the **CRC** page. For instructions, refer to How to Use the Same Checksum Algorithms in Several Configuration Projects on page 111.

Example of a frame ID file

Below is an example of a simple frame ID file defining the IDs for four frames/PDUs.

```
<frame-config>
  <frames>
    <frame>
       <name>Frame_1</name>
       <id>2</id>
    </frame>
    <frame>
       <name>Frame_2</name>
       <id>3</id>
    </frame>
    <frame>
       <name>Frame_3</name>
       <id>3</id>
    </frame>
    <frame>
       <name>Frame_4</name>
       <id>0</id>
    </frame>
  </frames>
</frame-config>
```

Related topics

Basics

HowTos

How to Use the Same Checksum Algorithms in Several Configuration Projects

Objective

When you have specified checksum algorithms in one configuration project, you can reuse the settings in the FlexRay Configuration Tool in other configuration projects. You can export the settings to an XML file and import them into the other configuration projects.

> **Note**
>
> When you import the settings from a file, the existing settings on the page are overwritten.

Preconditions

You have specified the settings of the checksum algorithms in a configuration project.

Method

To use the same checksum algorithms in several configuration projects

1. Open the configuration project where you have specified the settings for the checksum algorithms.
2. From the menu bar of the dSPACE FlexRay Configuration Tool, choose Tools - General Properties.

   The General Properties dialog opens.
3. In the General Properties dialog, select the CRC page.
4. On the CRC page, click Export.

   The Save As dialog opens.
5. Specify a file name and click Save.

   All the settings of the CRC page are saved in the specified file and can be used in other configuration projects.
6. Open the configuration project where you want to use the settings for the checksum algorithms.
7. From the menu bar of the dSPACE FlexRay Configuration Tool, choose Tools - General Properties.
8. In the General Properties dialog, select the CRC page.
9. On the CRC page, click Import.

Result

The settings for the checksum algorithms are imported. You can use the checksum algorithms in the configuration project.

Related topics

References

CRC Page (FlexRay Configuration Tool Reference 📖)

Monitoring Signals

Introduction

You can monitor signals which are transmitted via the FlexRay bus. When you select a signal for monitoring, you can read it via an RTI block in the Simulink model or the TRC file in ControlDesk.

Where to go from here

Information in this section

Basics of Monitoring

Introduction

The FlexRay Configuration Tool allows you to select FlexRay send signals for monitoring.

Monitoring

The FlexRay Configuration Tool has the **Monitoring** view for creating your monitoring configuration.

Monitoring view The **Monitoring** view allows you to create and manage your monitoring configuration. You can select send signals to generate FlexRay Configuration blocks in MATLAB for monitoring the FlexRay communication. Refer to How to Select and Configure Signals for Monitoring on page 114.

Selecting FlexRay network elements for monitoring After a FIBEX or AUTOSAR system description file is imported, all the FlexRay network elements are displayed in the **Communication Cluster** view. From there, you can drag & drop them in the **Monitoring** view. For monitoring purposes, you can only select send signals. It is possible to select a send signal for monitoring *and* configuration. This is useful for restbus simulation, where frames can be read back by the same simulation. To read back frames, the **Monitoring frame membership group** must be set to **RX pool**. Refer to Selection of ECUs, Frames and Signals on page 64.

Simulink block No RTI blocks are generated for monitored signals by default. You can use the generated TRC file to read the signals in ControlDesk (see Basics of Code Generation on page 131). However, if you want to get a Simulink interface for a signal, you must set the **Generate SL port** property to True. The

RTIFLEXRAYCONFIG PDU RX or FLEXRAYCONFIG PDU RX block of the selected signal gets an outport for the signal.

Tunable properties for signals and frames The FlexRay Configuration Tool provides some tunable properties for frames and signals to be connected with monitoring. Refer to Configurable Properties for ECUs, Signals, and Frames on page 58.

Related topics

References

FLEXRAYCONFIG PDU RX (FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU RX (RTI FlexRay Configuration Blockset Reference 📖)

How to Select and Configure Signals for Monitoring

Objective

You can select and configure signals to monitor the FlexRay bus communication.

Basics of monitoring

You can select ECUs, frames, or single send signals for monitoring. If you select a single send signal, the FlexRay Configuration Tool adds the ECU to the Monitoring view, but only with the selected signal and the frame belonging to it. For more basic information, refer to Basics of Monitoring on page 113.

Limitations

There are a few limitations in connection with frame and signal selection for monitoring. Refer to Limitations on page 141.

Preconditions

You must first import a communication cluster file. Refer to How to Import a Communication Cluster File on page 40.

Possible methods

To select and configure signals for monitoring, you can

- Select single send signals or send frames. Refer to Method 1.
- Select all the send signals of an ECU at once. Refer to Method 2.

Method 1

To select and configure single signals or frames for monitoring

1 If the Monitoring view is not open, click Views - Monitoring.

2 From the **Communication Cluster** view, drag a send signal or send frame to the **Monitoring** view.

Icon	Frame/Signal Type
	Static transmitting frame[1)]
	Static transmitting sync frame[1)]
	Static transmitting startup-sync frame[1)]
	Static transmitting subframe
	Dynamic transmitting frame
	Dynamic transmitting subframe
	Static transmitting signal
	Static transmitting signal, which belongs to a sync frame
	Static transmitting signal, which belongs to a startup-sync frame
	Dynamic transmitting signal

1) The icon contains a clock if the element is a time synchronization message.

Tip

You can select several frames and signals at once using the `Shift` or `Ctrl` key.

FlexRay Configuration Tool adds only the selected send signal or send frame with the corresponding ECU to the **Monitoring** view. The icons in the **Monitoring** view contain a monitor symbol indicating that the elements are monitored.

3 In the **Properties** view, specify the tunable properties for the frames and signals. Especially, set **Generate SL interface** to True if Simulink blocks for the signals should be generated. By default, you can read the signal in ControlDesk using the TRC file only.

4 If you want to remove an element from the **Monitoring** view, select it and click .

Tip

As an alternative, you can also press `Delete` to remove the selected element from the view.

Method 2

To select and configure all the signals of an ECU for monitoring

1 If the Monitoring view is not open, click Views - Monitoring.

2 From the Communication Cluster view, drag an ECU to the Monitoring view.

> **Tip**
>
> You can select several ECUs at once using the `Shift` or `Ctrl` key.

FlexRay Configuration Tool adds the selected ECU with all the corresponding send frames and send signals to the Monitoring view. The icons in the Monitoring view contain a monitor symbol indicating that the elements are monitored.

3 In the Properties view, specify the tunable properties for the frames and signals. Especially, set Generate SL interface to True if Simulink blocks for the signals should be generated. By default, you can read the signal in ControlDesk using the TRC file only.

Result

The signals or frames are selected for monitoring. In the Monitoring view, a monitor is added to the icons of the signals and frames.

Next step

You can create tasks. Refer to How to Create Tasks on page 126.

Related topics

References

Monitoring View (FlexRay Configuration Tool Reference)
Views (FlexRay Configuration Tool Reference)

Hardware Configuration

How to Configure Hardware

Objective

Before you can generate the code for the FlexRay model, you must select the board types your real-time system is based on.

Basics of real-time systems

After generating RTI FlexRay communication blocks configured for your FlexRay network, you can simulate the FlexRay node on a dSPACE real-time system. The FlexRay Configuration Package supports several real-time systems. For details on the supported real-time systems, refer to Supported Platforms on page 14. The **Hardware** page of the **General Properties** dialog displays the supported processor or controller board types, I/O board types, and controller modules, and lets you choose only the processor boards and controller modules which are compatible with the imported communication cluster file.

Limitations of dSPACE FlexRay hardware

- It is not possible to mix DS4501 and DS4505 boards equipped with FlexRay modules within one dSPACE single processor system.
- It is not possible to mix different FlexRay IP-Core modules on a DS4501 I/O board.
- A DS4505 I/O board could only be used with DS4340 FlexRay modules.
- DS4340 FlexRay modules can be used on DS1514 (MicroAutoBox III), or DS4501 or DS4505 I/O boards. The DS4505 board is faster than the DS4501 board with the DS4340 FlexRay modules.
- If you work with a MicroAutoBox III system with both the DS1521 and the DS1514, two different controller types are used in one project configuration:
  - The DS1521 supports two Bosch ERay IP-Core controllers.
  - The DS1514 supports two DS4340 controllers.

  In this case, the FlexRay Configuration Tool can generate code for different controller types, but with limitation: It always generates code for two subsequent controllers of the same controller type, i.e., the controllers of a board form a pair.

The following table shows two examples of supported controller type combinations. The first example (option 1) describes a case where 'DS1521 + DS1514' is selected as the I/O board type and Bosch ERay IP-Core is selected as the controller module on the **Hardware** page in the **General Properties** dialog. As a consequence, first the code for the Bosch ERay IP-Core controllers is generated, then the code for the DS4340 controllers. The second example (option 2) describes an example where 'DS1521 + DS1514' is selected as the I/O board type and the DS4340 is selected as the controller module on the **Hardware** page in the **General Properties** dialog, i.e., first the code for the DS4340 controllers is generated, then the code for the Bosch ERay IP-Core controllers.

	Option 1	**Option 2**
Controller 0	Bosch ERay IP-Core	DS4340
Controller 1	Bosch ERay IP-Core	DS4340
Controller 2	DS4340	Bosch ERay IP-Core
Controller 3	DS4340	Bosch ERay IP-Core

The FlexRay Configuration Tool does not generate code for pairs of different controller types. The following table shows examples of controller type combinations that are *not* supported:

Controller 0	DS4340	Bosch ERay IP-Core
Controller 1	Bosch ERay IP-Core	DS4340
Controller 2	DS4340	DS4340
Controller 3	Bosch ERay IP-Core	Bosch ERay IP-Core

Method

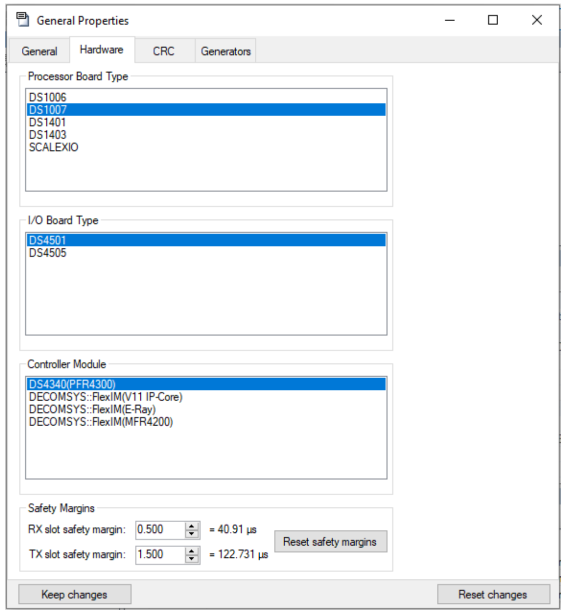
To configure the hardware

Note

If tasks are already configured and you change the hardware and the controller module, the FlexRay Configuration Tool checks the compatibility of your hardware configuration with the task timing. If they are not compatible, the FlexRay Configuration Tool deletes the current task configuration. A warning prompts you to decide whether to continue changing the hardware and lose the configuration.
To restore the current hardware configuration at the time of opening the **General Properties** dialog, click **Reset changes**.

1 From the menu bar, choose Tools – General Properties.

2 In the General Properties dialog, select the Hardware page.



3 Select the processor board type.

The selected processor board is immediately set as the simulation hardware if it is suitable for the rest of the current hardware configuration. If it is not, the FlexRay Configuration Tool displays ❗ symbols indicating that further settings are required. The tool tips of the symbols provide further information.

Tip

The Log View displays the currently set hardware configuration.

4 Select the I/O board type.

The selected I/O board is immediately set as the simulation hardware if it is suitable for the rest of the current hardware configuration. If it is not, the FlexRay Configuration Tool displays ❗ symbols indicating that further settings are required. The tool tips of the symbols provide further information.

5 Select a controller module.

The selected controller module is immediately set as the simulation hardware if it is suitable for the rest of the current hardware configuration.

6 Specify the safety margins. For details, refer to Hardware Page (FlexRay Configuration Tool Reference 📖).

7 Click Keep changes to confirm your settings and close the dialog.

Result

You configured the FlexRay hardware.

Related topics

References

Hardware Page (FlexRay Configuration Tool Reference 📖)

Creating Tasks

Introduction

You can configure and create communication tasks, application tasks, and synchronization tasks.

Where to go from here

Information in this section

Basics of Task Creation

Introduction

The FlexRay Configuration Tool has a Task view for managing communication and application tasks. Default task generation schemes for simulations are available. You can also create tasks manually.

Task types

With the FlexRay Configuration Tool, you can configure and create the following task types:

Communication tasks Communication tasks handle static or dynamic send and receive frames. You can create communication tasks for static and dynamic frames manually or let the FlexRay Configuration Tool configure them.

Application tasks Application tasks can handle static send and receive frames. You can use them for fast cycle response. An application task has an entry point for the receive frames and an exit point for the send frames. You can create application tasks only manually.

Application tasks run synchronously with the FlexRay bus. For this reason, you can use them to synchronize parts of your model with the FlexRay bus.

Synchronization tasks The synchronization tasks synchronize the application with the FlexRay bus. The FlexRay Configuration Tool creates and configures the synchronization task automatically.

Task configuration with the FlexRay Configuration Tool

The FlexRay Configuration Tool provides automatic and manual task creation.

Task view The Task view allows you to manage tasks and start automatic task configuration. You can also create tasks manually. For instructions on the two methods, refer to How to Create Tasks on page 126.

Tunable properties for tasks The FlexRay Configuration Tool provides some tunable properties for configuring user-defined tasks. Refer to Tunable Properties of Tasks on page 124.

Automatic task configuration

The FlexRay Configuration Tool provides automatic creation of time-triggered communication tasks. The default tasks are placed in the Task view and the frames and signals are assigned automatically. The FlexRay Configuration Tool displays the worst-case execution times (WCETs) of the tasks in the Properties view.

Limitations of automatic task configuration

The following limitations apply to automatic task configuration:

Dynamic communication part length If the dynamic part of the communication cycle is too short for automatic task creation, the FlexRay Configuration Tool generates no fetch and deliver tasks for static frames. You must create these tasks manually.

Static communication part length If the static part of the communication cycle is too short for automatic task creation, the FlexRay Configuration Tool generates no fetch and deliver tasks for dynamic frames. You must create these tasks manually.

Number of assigned signals The FlexRay Configuration Tool assigns as many signals as possible to the tasks. The rest of the signals are not used. The limit is reached when the WCET of a communication task reaches the start time of a manually created time-triggered task. This cannot be moved automatically. You must solve this conflict manually.

Manual task creation

You can create communication and application tasks manually and add signals or frames from the Configuration view via drag & drop. You can specify the task timing and position. For detailed information, refer to Tunable Properties of Tasks on page 124. If you add signals or frames to communication tasks, the order in which the signals or frames are executed by the application are controlled by the schedule. No general rules are required, but some limitations apply. See below.

Limitations of manual task creation

The following limitations apply to manual task creation:

Task timing Manual assignment of frames/signals to tasks requires:

- The base cycle and cycle repetition of tasks and frames have to be compatible.

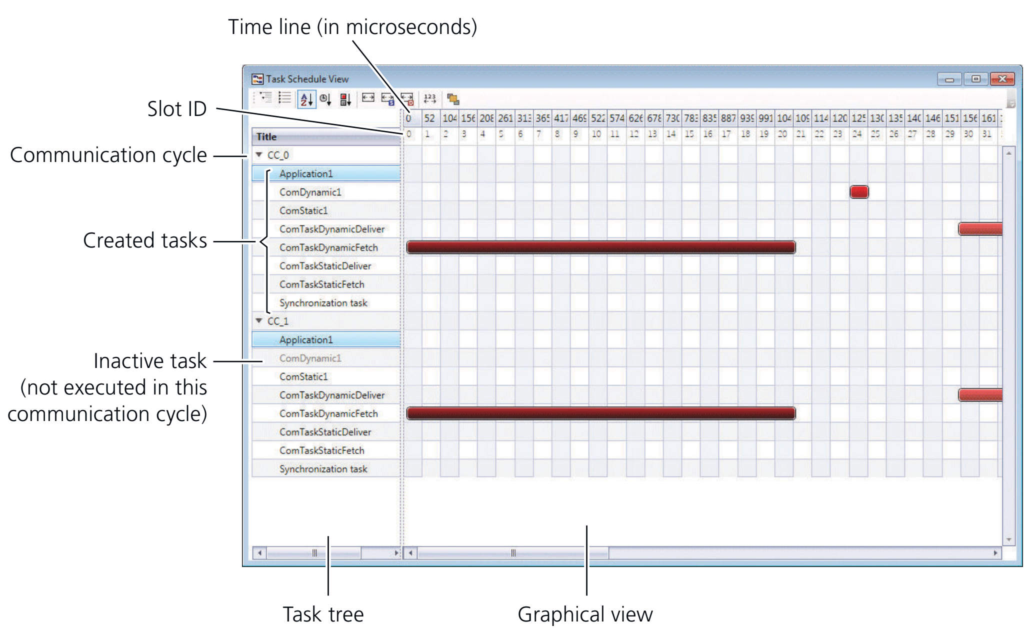
Task length and position Manual assignment of frames/signals to tasks requires:

- Tasks cannot cross the communication cycle end.
- Tasks cannot overlap.
- Communication tasks for static frames cannot overlap the frame-relevant slot.
- Communication tasks for dynamic frames cannot be placed in the dynamic part of the cycle.

Task Schedule View

The FlexRay Configuration Tool provides the Task Schedule View, which displays all the communication, application and synchronization tasks you configured in the FlexRay Configuration Tool in their time sequence within an application cycle. You can see which task is executed in which communication cycle, and where tasks depend on each other.

The Task Schedule View helps you plan communication and application tasks exactly.



Related topics

HowTos

References

Show Task Schedule View (FlexRay Configuration Tool Reference 📖)

Tunable Properties of Tasks

Introduction

The FlexRay Configuration Tool provides some tunable properties for configuring user-defined tasks. Tunable properties influence the parameters of the task blocks of the automatically generated FlexRay model.

Tunable properties of tasks

You can configure some tunable properties for communication, application and synchronization tasks. The properties of automatically created communication tasks are read-only (except WCET adjustment factor, see the table below). If you select a task in the Task view, the corresponding properties are displayed in the Properties view.

Communication and application tasks You can configure the following properties for communication and application tasks.

Configurable Property	Default	Description
Short name	String	Specifies a unique short name. The property is not configurable for automatically created tasks.
Base cycle	0	Specifies the first communication cycle in which the task is executed. The valid values are in the range 0 ... 63. The base cycle must match the base cycle of the added frames. The property is not configurable for automatically created tasks.
Cycle repetition	1	Specifies a constant repetition. The length of the cycle and the cycle repetition determine the period of the task. The cycle repetition must match the cycle repetition of the added frames. The valid values are 1, 2, 4, 8, 16, 32 and 64. The property is not configurable for automatically created tasks.
Offset Offsets (application tasks only)	Numerical value	Specifies a time offset in the communication cycle. This is the time between the communication cycle starting and the task executing. The maximum range of the offset is 0 ... 65535. Additionally, the range is restricted by the length of the specific communication cycle. For application tasks, you can specify several semicolon separated offsets. The property is not configurable for automatically created tasks.

Configurable Property	Default	Description
Function WCET (application tasks only)	0	Specifies the function WCET.
WCET adjustment factor	1	Specifies an adjustment factor of the worst-case execution time (WCET). The factor lets you change the calculated WCET manually if this is necessary.
Used for real-time test triggering (application tasks only)	False	Specifies whether the application task is to be used to replay data. If the Real-Time Testing option is enabled, the task is used to execute the corresponding RTT replay script. If you work with a MicroAutoBox III or SCALEXIO system, this property is not available. You must enable the Real-Time Testing option for the replay task in ConfigurationDesk instead. Refer to Configuring Tasks in ConfigurationDesk (ConfigurationDesk Real-Time Implementation Guide 📖). Real-Time Testing does not support the DS1007, which means that data replay is not possible for the DS1007. For this reason, the Real-Time Testing option is always disabled if you work with a DS1007. You cannot change this setting.
Data capture block ID (application tasks only)	Empty	Specifies a unique ID for the task. If an ID is specified, you can use the application task to log data in ControlDesk. If you work with a MicroAutoBox III or SCALEXIO system, this property is not available. You must specify the DAQ raster name to be used for the logging task in ConfigurationDesk instead. Refer to Configuring Tasks in ConfigurationDesk (ConfigurationDesk Real-Time Implementation Guide 📖).

Synchronization task You can configure the following properties for the synchronization task.

Configurable Property	Default	Description
WCET adjustment factor	1	Specifies an adjustment factor of the worst-case execution time (WCET). The factor lets you change the calculated WCET manually if this is necessary.
Startup sync mode	HARD	Selects the hard or smooth start-up synchronization mode. ▪ In the smooth mode, the synchronization algorithm does not use more macroticks for correction than specified by the SyncLimit parameter. ▪ In the hard mode, synchronization is performed after the first communication cycle. The SyncLimit parameter is ignored. For details on the SyncLimit parameter, refer to Unit Page (RTIFLEXRAYCONFIG SYNC SERVICE) (RTI FlexRay Configuration Blockset Reference 📖).

Tip

You can select several tasks at once using the `Shift` or `Ctrl` key.

Related topics

HowTos

How to Create Tasks

Objective

You can let the FlexRay Configuration Tool create default tasks and create communication and application tasks manually.

Basics of task creation

For basic information, refer to Basics of Task Creation on page 121.

Limitations of creating tasks

There are a few limitations in connection with task creation. Refer to Limitations on page 141.

Preconditions

The following preconditions apply:

- You must first create a configuration. Refer to Basics on Sending and Receiving Signals on page 71.
- The Task view must be open. Refer to Views (FlexRay Configuration Tool Reference 🕮).

Possible methods

To create tasks, you can

- Start automatic task creation. Refer to Method 1.
- Create tasks manually. Refer to Method 2.

Method 1

To create tasks automatically

1 On the Task view's toolbar, click .
The FlexRay Configuration Tool creates communication tasks and assigns the signals listed in the Configuration view and Monitoring view to the tasks automatically.

2 In the Properties view, specify the tunable task properties.

Tip

You can open the Task Schedule View, which visualizes all the created tasks in their time sequence within an application cycle. It tells you at a glance which task is executed in which communication cycle and where tasks depend on each other, and helps you plan communication and application tasks exactly. Refer to Show Task Schedule View (FlexRay Configuration Tool Reference).

Method 2

To create tasks manually

1 On the Task view´s toolbar, click to open the task types list:

Icon	Task Type
	Com Task Static
	Com Task Dynamic
	Application Task

2 From the task type list, choose the task type you want to create.
The FlexRay Configuration Tool creates a new task and adds the corresponding icon to the Task view.

3 In the Properties view, specify the tunable task properties.

4 To assign signals to the task, drag the signals from the Configuration or Monitoring view to the task in the Task view.

Note

You can see where the signals were dropped from by looking at the icons. Signals dropped from the Monitoring view have a monitor in their icons.

5 If you want to remove an element from the Task view, select it and click .

Result

You created tasks.

Next step

You can generate code. Refer to How to Generate Code on page 137.

Related topics

Basics

References

Automatic Task Creation (FlexRay Configuration Tool Reference 🕮)
Create Task (FlexRay Configuration Tool Reference 🕮)
Show Task Schedule View (FlexRay Configuration Tool Reference 🕮)
Task View (FlexRay Configuration Tool Reference 🕮)

How to View the Controller Configuration

Objective

You can view the available controllers and the required buffer count for your simulation configuration.

Basics of controllers

A controller is responsible for implementing the protocol aspects of the FlexRay communications system. The controller host interface (CHI) manages the data and control flow between the host processor and the controller in each node. For details, refer to *FlexRay Communications System Protocol Specification Version 2.1*.

Controller configuration

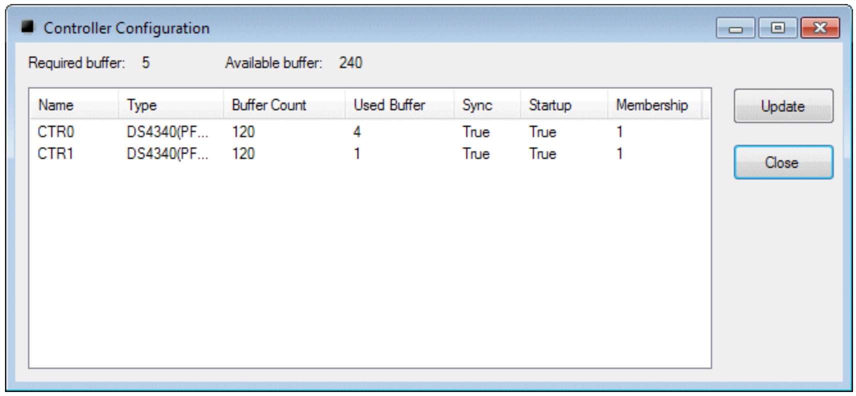
The FlexRay Configuration Tool lets you view the controller configuration in the Controller Configuration dialog. You can see the required controller count for your current configuration. Available and required buffers are also displayed, with information on whether each controller is used for synchronization.

For a full description of the information in the Controller Configuration dialog, refer to Controller Configuration (FlexRay Configuration Tool Reference 🕮).

Method

To view the controller configuration

1 From the menu bar, choose Tools – Controller Configuration to open the Controller Configuration dialog.



2 If you changed your configuration for simulation, you can update the controller configuration. Click Update.

Result

You viewed the available and required controller and buffer count.

Related topics

References

Controller Configuration (FlexRay Configuration Tool Reference 📖)

Generating Code

Where to go from here

Information in this section

Basics of Code Generation

Introduction

After you have finished your configuration, you can generate Simulink configuration data, Com code, CHI code, a TRC file, a configuration file for the Bus Navigator in ControlDesk, and optionally a report file to use for building your real-time application.

Code generation with the FlexRay Configuration Tool

The FlexRay Configuration Tool provides the Generators page in the General Properties dialog, which lets you make the necessary settings. Refer to Generators Page (FlexRay Configuration Tool Reference 🕮).

Simulink configuration data

You can generate Simulink configuration data stored in an M file. The M file contains generated code using data from the imported FIBEX or AUTOSAR system description file and your settings in the FlexRay Configuration Tool. The name of the M file is the file name of the project plus "_data".

Code generation

Com code You can generate code files for the FlexRay communication layer, called Com code.

CHI code You can generate code files for the FlexRay controller host interface (CHI), called CHI code.

TRC file generation

A TRC file is used by ControlDesk to access the signals and frames if they are configured for TRC file access. If you want to read a signal in ControlDesk only, it is not necessary to generate RTI blocks, drag them to your model and read the outports. The FlexRay Configuration Tool generates a TRC file for signals and frames with the following properties:

- Monitored signals and frames (signals and frames dragged to the Monitoring view). Monitored signals are automatically selected for the TRC file if they are connected to a task. You can deselect them in the Element Selection dialog, refer to How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File on page 134.
- Receiving signals and frames of the Configuration view (optional)
- Sending signals and frames of the Configuration view (optional)

If a signal is contained in the TRC file, you can also get its status information via the TRC file. To do so, select Signal RX Status Access in the Element Selection dialog for the signal. Note that the RTIFLEXRAYCONFIG_LASTCORRECT_DATA_NOT_READ status cannot be read using the TRC file. The validity information is available as a status variable (name: Status) and has the following meanings.

Bit	Value	Description
No bit set	0	No error, signal is valid
1st bit set	1	Access error
2nd bit set	2	Signal is not received
3rd bit set	4	Signal is not valid (signal validity status ≠ 'VALID')
4th bit set	8	CRC is incorrect

Note

If you read or write the validity status of a signal via the **Validity** port, the block input/output can have the following values:

- 0: NOT VALID
- 1: VALID
- 2: ERROR
- 4: NOT AVAILABLE
- 8: NOT DEFINED
- 16: OTHER

For information on how to work with the TRC file, refer to:

- Using the Generated TRC File of PDU-Based Modeling (FlexRay Configuration Features 📖)
- (MicroAutoBox III, SCALEXIO) Using the Generated TRC File of PDU-Based Modeling in ControlDesk (Model Interface Package for Simulink - Modeling Guide 📖)

Report generation

When generating code, you can also generate an HTML file about the current configuration. The file can contain messages from the log view and information on frame-buffer assignment. Refer to Generators Page (FlexRay Configuration Tool Reference 📖).

Bus Navigator

ControlDesk has a Bus Navigator which makes it easy to experiment with the FlexRay bus. The Bus Navigator requires a configuration file that describes the FlexRay configuration. This configuration file is always generated during code generation. For details on the Bus Navigator, refer to Introduction to the Bus Navigator (ControlDesk Bus Navigator 📖).

Using generated code

You can use the generated Simulink configuration data, Com code, and CHI code for simulation purposes with the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset. Refer to Using the Generated Code on page 139.

Related topics

Basics

HowTos

How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File

Objective

You can manipulate or monitor frames/PDUs and signals via ControlDesk. This is not possible by default. You must use the **Element Selection** dialog to enable manipulation or monitoring for the relevant frames or signals.

Basics

There are several features that you can use to manipulate or monitor frames/PDUs and signals. These features can be configured for the frames/PDUs and signals in the FlexRay Configuration Tool. For controlling the features in the experiment, the corresponding variables must be included in the TRC file so that they can be connected to instruments in ControlDesk. When an RTI block is used to control the features, the variables are automatically included in the TRC file. You can also use the code generator in the FlexRay Configuration Tool to generate the variables directly, without using RTI blocks. Simply select the features and the frames/PDUs or signals whose variables you want to include in the TRC file. For further information on the features that can be selected, refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

If features are selected for RTI blocks and the TRC file, you can switch between the corresponding variables during run time via the corresponding TRC variable (`Source Switch`).

Some features are already configured to be included in the TRC file, refer to *TRC file generation* in Basics of Code Generation on page 131.

You can use all the features listed in the features list in the **Element Selection** dialog. Select a feature and read the short information on the elements it supports at the bottom of the features list.

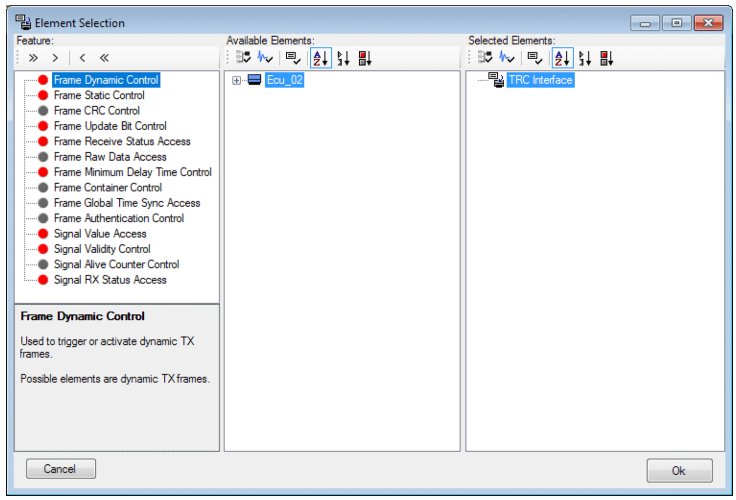
Preconditions

- The frames/PDUs and signals must be configured for the necessary feature.
- The frames/PDUs and signals must be assigned to a task.

Method

To prepare the manipulation or monitoring of frames/PDUs and signals via experiment software

1 From the menu bar, choose Tools – Element Selection to open the Element Selection dialog.



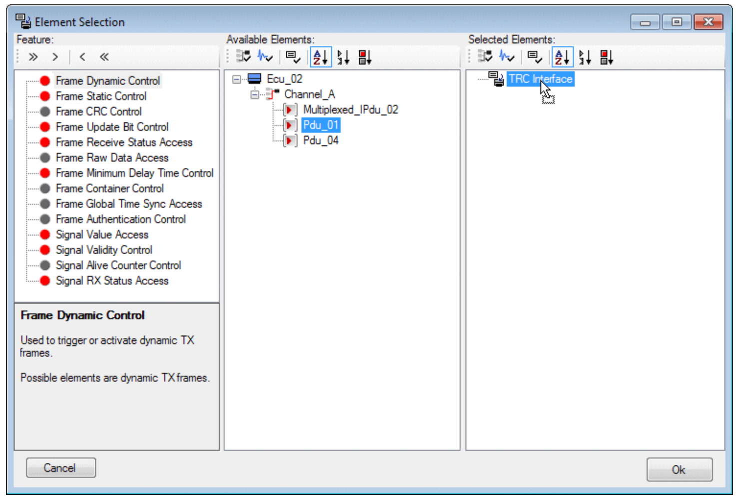
All the features are listed in the features list. A colored state icon next to a feature indicates whether elements are configured for the feature and, if they are, whether the elements are already selected for the TRC file.

State Icon	Description
	A gray state icon indicates that no elements are configured for the selected feature.
	A red state icon indicates that elements are configured for the selected feature, but no elements are selected for the TRC file yet.
	A green state icon indicates that all available elements that are configured for the selected feature are already selected for the TRC file.
	A yellow state icon indicates that elements are configured for the selected feature, and a subset of them is already selected for the TRC file.

2 In the features list, select the feature that you want to control via TRC file.

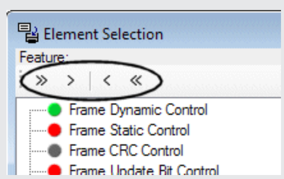
Available Elements lists all the frames/PDUs and signals that are configured or available for the selected feature. If the list is empty, no element is configured or available for the selected feature. If you think some elements are missing, check their configurations in the Properties view of the FlexRay Configuration Tool.

3 Drag an element (frame/PDU or signal) from Available Elements exactly onto the TRC Interface item in Selected Elements.



Tip

The FlexRay Configuration Tool provides buttons to select/deselect all the available elements for one or all features for/from control via TRC file with a single click. You can find the buttons above the features list.



4 Repeat the previous steps for all the elements that you want to control via TRC file.

5 Click Ok to close the dialog.

Result

When the code is generated, the variables for manipulating or monitoring all the selected and the configured frames/PDUs and signals are included in the TRC file. In addition, the TRC file contains variables for the configured features.

Related topics

Basics

Using the Generated TRC File of PDU-Based Modeling (FlexRay Configuration Features 📖)
Using the Generated TRC File of PDU-Based Modeling in ControlDesk (Model Interface Package for Simulink - Modeling Guide 📖)

References

Element Selection (FlexRay Configuration Tool Reference 📖)

How to Generate Code

Objective

You can generate configuration data and code to use for building your real-time application.

Basics of Generating Code

Generating code covers generating Simulink configuration data, Com code, CHI code, a TRC file, a configuration file for the Bus Navigator, and optionally a report file. For detailed information on code generation, refer to Basics of Code Generation on page 131.

You can select the elements that are included in the TRC file, refer to How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File on page 134.

The FlexRay Configuration Tool provides the following generator forms in the Code Generation dialog:

- Report Generator
- Exp. Cfg. File Generator
- Trace File generator
- Simulink Data Generator
- COM Code Generator
- CHI Code Generator

You can specify several properties for the generators. For details on the properties, refer to Generators Page (FlexRay Configuration Tool Reference 📖).

Limitations of Bus Navigator support

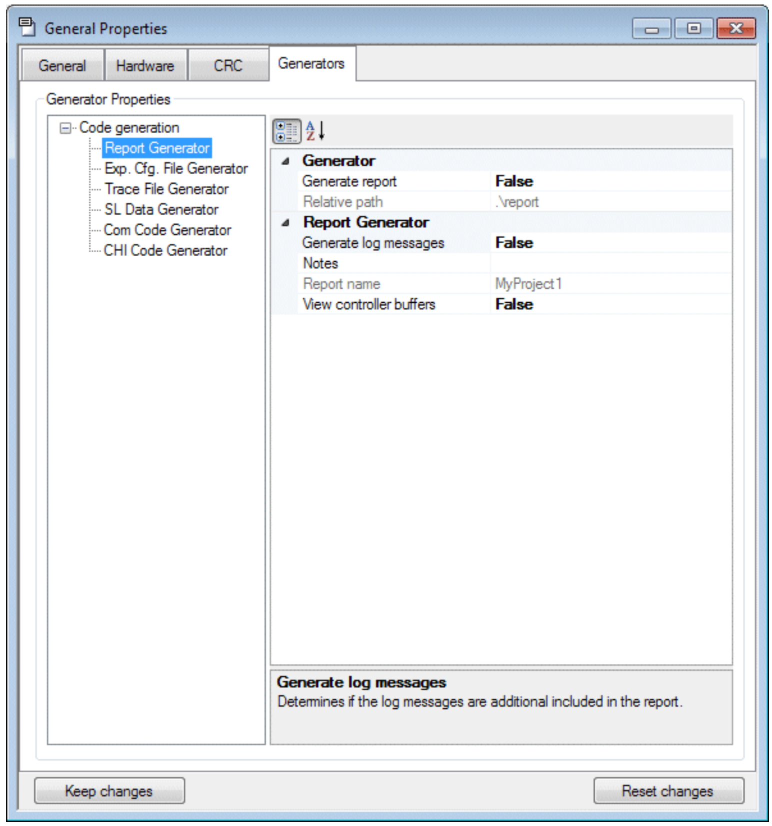
Configurations that are based on a FIBEX 1.2 file do not support the Bus Navigator in ControlDesk. The configuration files generated by the Exp. Cfg. File Generator cannot be loaded in ControlDesk.

Method

To generate code

1 From the menu bar, choose Tools – General Properties.

2 In the General Properties dialog, select the Generators page.



3 Click Code generation if you want to change the absolute path. All the generated files are stored on relative paths under the absolute path. Alternatively, you can specify to use a code generation path that is relative to the project file.

4 Select the generator forms (Report Generator ... CHI Code Generator) from the list and specify the settings for each generator.

5 Click Keep changes to confirm your settings and close the General Properties dialog.

6 From the toolbar, click  to start code generation.

Result

The FlexRay Configuration Tool generates the code files and a report file and stores them in the specified folder.

Related topics

Basics

References

Generators Page (FlexRay Configuration Tool Reference 🕮)

Using the Generated Code

Introduction

You can use the generated Simulink configuration data, Com code, and CHI code for simulation purposes with the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset.

Simulink configuration data

The Simulink configuration data is stored in an M file. In MATLAB, you can read this M file to automatically parameterize blocks of the RTI FlexRay Configuration Blockset or FlexRay Configuration Blockset and generate a configuration model. You can use this model to create your Simulink FlexRay network model for simulation. For instruction and further information on the workflow, refer to:

- Overview of the Workflow (FlexRay Configuration Features 🕮)
- (MicroAutoBox III, SCALEXIO) Overview of the Workflow (Model Interface Package for Simulink - Modeling Guide 🕮)

Com and CHI code

For real-time simulation, the Com and CHI code is used for the build process of the real-time application, which can be downloaded to the dSPACE hardware. For detailed information, refer to:

- How to Build and Download Code (FlexRay Configuration Features 🕮)
- (MicroAutoBox III, SCALEXIO) Building Real-Time Applications (ConfigurationDesk Real-Time Implementation Guide 🕮)

Related topics

HowTos

Appendix

Where to go from here

Information in this section

Limitations

Introduction

The following limitations apply to this version of the FlexRay Configuration Tool.

Limitations of dSPACE FlexRay hardware

- It is not possible to mix DS4501 and DS4505 boards equipped with FlexRay modules within one dSPACE single processor system.
- It is not possible to mix different FlexRay IP-Core modules on a DS4501 I/O board.
- A DS4505 I/O board could only be used with DS4340 FlexRay modules.
- DS4340 FlexRay modules can be used on DS1514 (MicroAutoBox III), or DS4501 or DS4505 I/O boards. The DS4505 board is faster than the DS4501 board with the DS4340 FlexRay modules.
- If you work with a MicroAutoBox III system with both the DS1521 and the DS1514, two different controller types are used in one project configuration:
  - The DS1521 supports two Bosch ERay IP-Core controllers.
  - The DS1514 supports two DS4340 controllers.

In this case, the FlexRay Configuration Tool can generate code for different controller types, but with limitation: It always generates code for two subsequent controllers of the same controller type, i.e., the controllers of a board form a pair.

The following table shows two examples of supported controller type combinations. The first example (option 1) describes a case where 'DS1521 + DS1514' is selected as the I/O board type and Bosch ERay IP-Core is selected as the controller module on the **Hardware** page in the **General Properties** dialog. As a consequence, first the code for the Bosch ERay IP-Core controllers is generated, then the code for the DS4340 controllers. The second example (option 2) describes an example where 'DS1521 + DS1514' is selected as the I/O board type and the DS4340 is selected as the controller module on the **Hardware** page in the **General Properties** dialog, i.e., first the code for the DS4340 controllers is generated, then the code for the Bosch ERay IP-Core controllers.

	Option 1	**Option 2**
Controller 0	Bosch ERay IP-Core	DS4340
Controller 1	Bosch ERay IP-Core	DS4340
Controller 2	DS4340	Bosch ERay IP-Core
Controller 3	DS4340	Bosch ERay IP-Core

The FlexRay Configuration Tool does not generate code for pairs of different controller types. The following table shows examples of controller type combinations that are *not* supported:

Controller 0	DS4340	Bosch ERay IP-Core
Controller 1	Bosch ERay IP-Core	DS4340
Controller 2	DS4340	DS4340
Controller 3	Bosch ERay IP-Core	Bosch ERay IP-Core

Limitations of communication cluster file import

Refer to Limitations Applying to the Communication Cluster File on page 27.

Limitations of the communication cluster update

The following limitations apply if you update the communication cluster file in your configuration:

- Updating is allowed only within the following groups:
  - *FIBEX 1.0*: 1.1.5a, 1.2.0, 1.2.0a
  - *FIBEX 2.0*: 2.0.0b, 2.0.0d, 2.0.0e, 2.0.1
  - *FIBEX 3.0*: 3.0.0, 3.1.0, 3.1.1
  - *FIBEX 4.0*: 4.1.0, 4.1.1, 4.1.2
  - *FIBEX+*
  - *AUTOSAR 3*: AUTOSAR 3.1.4, AUTOSAR 3.2.1, AUTOSAR 3.2.2

  - *AUTOSAR 4*: AUTOSAR 4.0.3, AUTOSAR 4.1.1, AUTOSAR 4.1.2, AUTOSAR 4.2.1, AUTOSAR 4.2.2, AUTOSAR 4.3.0, AUTOSAR 4.3.1, AUTOSAR 4.4.0
  - *AUTOSAR Classic Platform*: AUTOSAR Classic Release R19-11, R20-11
- The communication cluster files must have the same critical cluster parameters, for example, speed, cycle, sample clock period.

Limitations of FlexRay element selection

The following limitations apply to FlexRay element selection:

Dragging elements

- You can drag each receiving element to the Configuration view only once. Receiving elements already selected are grayed out in the Communication Cluster view.
- You can drag each sending element to the Configuration and Monitoring view only once. A sending element dropped to the Configuration view can additionally be dropped to the Monitoring view. Sending elements already selected in both configurations are grayed out in the Communication Cluster view.
- A frame of an ECU can contain signals which are not received by the ECU itself. These signals cannot be dropped.

Send frames for simulation

- You can select only complete send frames for simulation. Even if only a single signal or subframe is selected, the whole send frame with all the signals and subframes is assigned.
- You can remove the subframes inside a frame except for one. If you delete signals of a subframe or a frame, the whole frame or subframe is removed from your configuration.
- If an XCP master node is not selected in the FlexRay Configuration Tool, XCP_RUNTIME_CONFIGURED frames cannot be used for simulation and monitoring.
- If an XCP master node is selected in the FlexRay Configuration Tool, XCP_RUNTIME_CONFIGURED and XCP_PRE_CONFIGURED frames from the XCP master node can only be simulated (not monitored) and you cannot select XCP frames for the TRC file.

Network Management frames (NM frames) You can select only complete Network Management (NM) frames for simulation or monitoring. A corresponding Simulink block with all the relevant signals is generated for the entire NM frame.

Signals for multiple receivers You cannot select signals or subframes which are not received by an ECU. If a frame is sent to multiple receivers and these receivers receive only a few signals or subframes of the frame, the remaining signals or subframes of the frame are blocked for the receivers. The remaining signals or subframes are also blocked for restbus simulation.

Signals for monitoring You can select single send signals (except for signals of NM frames) only for monitoring. If you select an ECU or a cluster, only their send signals are used for monitoring.

Time synchronization PDUs for monitoring You can select time master PDUs for monitoring, but you cannot select time slave PDUs for monitoring.

Selecting the XCP master You cannot select an XCP master node if a dual-channel communication cluster file is imported.

Limitations of automatic task configuration

The following limitations apply to automatic task configuration:

Dynamic communication part length If the dynamic part of the communication cycle is too short for automatic task creation, the FlexRay Configuration Tool generates no fetch and deliver tasks for static frames. You must create these tasks manually.

Static communication part length If the static part of the communication cycle is too short for automatic task creation, the FlexRay Configuration Tool generates no fetch and deliver tasks for dynamic frames. You must create these tasks manually.

Number of assigned signals The FlexRay Configuration Tool assigns as many signals as possible to the tasks. The rest of the signals are not used. The limit is reached when the WCET of a communication task reaches the start time of a manually created time-triggered task. This cannot be moved automatically. You must solve this conflict manually.

Limitations of manual task creation

The following limitations apply to manual task creation:

Task timing Manual assignment of frames/signals to tasks requires:

- The base cycle and cycle repetition of tasks and frames have to be compatible.

Task length and position Manual assignment of frames/signals to tasks requires:

- Tasks cannot cross the communication cycle end.
- Tasks cannot overlap.
- Communication tasks for static frames cannot overlap the frame-relevant slot.
- Communication tasks for dynamic frames cannot be placed in the dynamic part of the cycle.

Limitations for using the automatically generated FlexRay model

You can use only one FlexRay library for a dSPACE node:

- In a single-processor system, you can use only the blocks from one automatically generated FlexRay model.

- In a multiprocessor system, each processor can have its own FlexRay model. After separation using RTI-MP, there is also one automatically generated FlexRay model for each processor.

Limitations of multiple bus support

The following limitations apply to multiple bus support (working with several configurations for one model):

- The FlexRay Configuration Tool supports a maximum of four configurations for one processor board or SCALEXIO Processing Unit. Only two configurations are supported for a MicroAutoBox II/III. The maximum number of possible configurations might be lower due to limited processing power.
- It cannot be guaranteed that the tasks of the lower-priority FlexRay configurations are executed at the scheduled times. This can lead to the following problems:
  - Temporary buffer lock error
  - Dynamic TX PDUs might be omitted.
  - Static TX PDUs contain old data or are sent as null frames.
- Only one XCP configuration is allowed for a real-time system.
- It is not possible to send static TX PDUs which share the same communication slot on the same channel via different FlexRay nodes or different bus controllers, even if their cycle counter filtering (defined by base cycle and cycle repetition in absolute scheduled timing) is different. If the controller of a FlexRay node is configured to send a static TX PDU within a specific communication slot with the given cycle counter filtering of the TX PDU, it automatically sends null frames in the slot each time the PDU is not sent. Any attempt by another bus node to send a static TX PDU in the slot results in invalid frames on the FlexRay bus.

  For further information, refer to *FlexRay Communications System Protocol Specification Version 2.1*.
- Only one CRC C file is allowed for all the FlexRay configurations used.
- The **RTIFLEXRAYCONFIG ERROR HOOK STATUS** block cannot distinguish between different FlexRay configurations.
- The **RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT** block might be triggered for the wrong controller if several FlexRay configurations are used on one core of a processor board and the corresponding controllers are on different I/O board types in the same slot position. To avoid this, place the controller modules in different slot positions.
- After migrating FlexRay multiprocessor models based on several FlexRay configurations from an earlier FlexRay Configuration Package version to dSPACE FlexRay Configuration Package 2.4 or later, you must assign some **RTIFLEXRAYCONFIG** blocks to the **RTIFLEXRAYCONFIG UPDATE** block manually.
- The tasks are sorted according to their priorities during the first build process. If you add a FlexRay configuration afterwards, its tasks cannot be inserted with the correct priority. In this case you can
  - Sort the tasks manually.
  - Delete the **RTI Data** block. If no **RTI Data** block is in the model, the tasks are sorted again in the next build process.

- If no FlexRay configuration with configuration ID `"- "` is in the real-time model, the FlexRay-based capture synchronization in ControlDesk does not work.

Limitations of transmission mode support

The following limitations apply to transmission mode support (working with different transmission modes/timings for each PDU):

- In contrast to AUTOSAR, any number of timings can be assigned to each PDU in FIBEX files. The FlexRay Configuration Tool only supports two of these PDU timings at maximum and assigns them to the transmission modes `True` and `False`:
  - Whether a cyclic or event-based timing is assigned to the transmission mode `True` or `False` depends on where it is defined in the FIBEX file. For both the cyclic and the event-based timing type, the FlexRay Configuration Tool assigns the first timing definition each to the transmission mode `True`, and the second timing definition to the transmission mode `False`. Further timing specifications are ignored.
  - If the FIBEX file contains both cyclic and event-based timings, the FlexRay Configuration Tool combines them to mixed timings. The first cyclic timing and the first event-based timing are both assigned to the transmission mode `True`. The second cyclic timing and the second event-based timing are both assigned to the transmission mode `False`. The assigned PDU is of timing type static event cyclic (SEC) or dynamic event cyclic (DEC), and can be triggered by events and/or cyclically.

    There are two special cases:
    - If the FIBEX file contains *two* (or more) cyclic timings but only *one* event-based timing for a PDU, the event-based timing is combined with the two cyclic timings. That is, the first cyclic timing and the event-based timing are assigned to the transmission mode `True`, and the second cyclic timing and the event-based timing are assigned to the transmission mode `False`.
    - If the FIBEX file contains *one* cyclic timing but *two* (or more) event-based timings for a PDU, the cyclic timing is combined with the two event-based timings. The cyclic timing and the first event-based timing are combined and assigned to the transmission mode `True`. The cyclic timing and the second event-based timing are assigned to the transmission mode `False`.
- The `LPDU timing triggered` transmission mode cannot be selected as the global default transmission mode.
- Final repetitions of event-based timings of TX PDUs are not supported.
- The `False` transmission mode of PDUs with sub-PDUs is not supported. The FlexRay Configuration Tool always assigns the default transmission mode `True` to the sub-PDUs. You cannot specify a default transmission mode for sub-PDUs manually.
- The `LPDU timing triggered` transmission mode is not supported for PDUs for which timing information is specified in the underlying FIBEX or AUTOSAR system description file, i.e., to which the transmission mode `False` or `True` is assigned.
- The `LPDU timing triggered` transmission mode is not supported for multiplexed PDUs and sub-PDUs.

- Automated switching between transmission modes during run time (for example, based on transfer properties or using data filters) is not possible.
- The FlexRay Configuration Tool does not support transmission modes for cryptographic IPDUs.

Limitations of minimum delay time support

The following limitations apply to minimum delay time support:

- The minimum delay time feature is supported only for AUTOSAR system description files based on a FlexRay Configuration Tool supported AUTOSAR version ≥ 4.0.3 or AUTOSAR Classic Platform Release R19-11 or R20-11. For information on the supported AUTOSAR Releases, refer to Communication Cluster Files Usable for Configuration on page 26.
- The minimum delay time feature is not supported for static sub-PDUs.
- The debounce time property, which can be defined for event-based timings in a FIBEX file, is not supported. A specified debounce time value is imported and displayed in the **Properties** view. However, in contrast to the AUTOSAR minimum delay time, the debounce time is not evaluated for transmission.

Limitations for container IPDUs

The following limitations apply to container IPDUs:

- The FlexRay Configuration Tool does not support nested container IPDUs.
- The FlexRay Configuration Tool does not support container IPDUs for dual-channel FlexRay systems.
- The FlexRay Configuration Tool does not support container IPDUs as in-cycle response PDUs.
- For the `RxAcceptContainedIpdu` AUTOSAR attribute, the FlexRay Configuration Tool does not support the `ACCEPT_CONFIGURED` value for container IPDUs, which allows only a certain set of contained IPDUs in a container IPDU.

Limitations for secure onboard communication

The following limitations apply to secure onboard communication:

- You cannot get information on the authentication information and status information on the verification of secured IPDUs at run time via the Simulink model. However, you can get information at run time via TRC variables.
- When a secured IPDU is configured as a cryptographic IPDU, neither the cryptographic IPDU nor the associated authentic IPDU can be configured as a multiplexed sub-IPDU. The cryptographic IPDU cannot be a multiplexed IPDU either.
- When a secured IPDU is configured as a cryptographic IPDU, the cryptographic IPDU cannot be a static container IPDU or a dynamic container IPDU. The same applies to the associated authentic IPDU.
- For cryptographic IPDUs, you cannot change the FlexRay key slot type in the FlexRay Configuration Tool.
- The FlexRay Configuration Tool does not support secured PDU headers for cryptographic IPDUs.

- The FlexRay Configuration Tool does not support the import of the `SecuredAreaLength` and `SecuredAreaOffset` attributes. Instead, the entire cryptographic IPDU is used to calculate the authenticator.
- The FlexRay Configuration Tool does not support raw data access for cryptographic IPDUs. If raw data support is enabled, you can enable raw data access to a cryptographic IPDU and manipulate its raw data at run time, but these values are overwritten by the SecOC user code. This means that the raw data access is ineffective.

Limitations for global time synchronization

The following limitations apply to global time synchronization:

- The FlexRay Configuration Tool does not support dynamic GTS PDUs. Only static GTS PDUs are imported.
- The FlexRay Configuration Tool does not support multiple time masters per time domain. If more than one time master is defined for a time domain in the communication cluster file, both the time domain and time master data is ignored during import.
- The FlexRay Configuration Tool does not support offset time bases.
- The FlexRay Configuration Tool supports only secured time synchronization PDUs (message type 0x20), i.e., CRC-secured time master PDUs and CRC-validated time slave PDUs. Other time synchronization PDUs are not imported.
- The FlexRay Configuration Tool does not support Global Time Precision Measurement.
- The FlexRay Configuration Tool does not support immediate time synchronization.
- The FlexRay Configuration Tool does not support triggering customers.
- The FlexRay Configuration Tool does not support raw data access for time synchronization messages.
- You cannot select time slave PDUs for monitoring.

Limitations of SCALE_LINEAR_TEXTTABLE computation method support

The following limitations apply to working with the SCALE_LINEAR_TEXTTABLE computation method:

- If the **Physical data type conversion layer** option is set to **APPLICATION** on the **General** page of the **General Properties** dialog, no access to switch between coded and physical value is available for sending signals with the 'SCALE_LINEAR_TEXTTABLE' computation method. This means that no `Coded Physical Value Source Switch` variable is added to the TRC file.

Limitations concerning sending null frames when SWEnable = 1

If sending of static TX PDUs is enabled via software, LPDUs with non-updated PDU data have a null frame sent instead of the old PDU payload data. However, sending null frames is not supported for the following LPDUs:

- LPDUs that contain more than one PDU
- LPDUs that contain at least one PDU with a PDU update bit

In these cases, old PDU payload data is sent instead of null frames.

Limitation of the RX Timestamp feature

The following limitations apply to the RX Timestamp feature:

- The RX timestamp information is available only for SCALEXIO systems. RX Timestamp records the time at which an LPDU was received.
- The RX timestamp value is updated only if the PDU was received on the FlexRay wire. If the PDU was received internally, i.e., if its value was copied from the controller memory, the RX timestamp value is not updated.

Limitations regarding working with the opaque byte order

Refer to Limitations With Opaque Byte Order Format on page 38.

Limitations of Bus Navigator support

Configurations that are based on a FIBEX 1.2 file do not support the Bus Navigator in ControlDesk. The configuration files generated by the Exp. Cfg. File Generator cannot be loaded in ControlDesk.

Related topics

References

RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG ERROR HOOK STATUS (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG UPDATE (RTI FlexRay Configuration Blockset Reference 🕮)

Troubleshooting

Introduction

If a problem related to FlexRay bus configuration comes up, this topic provides a collection of possible malfunctioning scenarios and solutions.

RX PDU not received when using DS4340 FlexRay Interface Modules

Problem You are working with DS4340 FlexRay Interface Modules, and an RX PDU is not received.

Reason This problem is caused by a hardware limitation of the FlexRay controller of the DS4340 FlexRay module. It occurs if an RX PDU and a TX PDU use the same FlexRay communication slot with different FlexRay cycles.

The following table shows a FlexRay configuration that leads to the problem:

PDU	Slot ID	Base Cycle	Cycle Repetition
RX PDU	98	0	4
TX PDU	98	2	4

Workaround Assign the RX PDU to a different FlexRay controller. This can be done via the **Frame membership** property of the RX PDU in the FlexRay Configuration Tool. Select **Global RX pool** as the RX frame membership for the ECU that receives the RX PDU.