FlexRay Configuration

Features

For dSPACE FlexRay Configuration Package 4.7

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

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

How to Contact dSPACE Support

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

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

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

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

Software Updates and Patches

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

Important Notice

Contents

About This Document

Contents

This document provides feature-oriented access to the information you need to connect your real-time model to a FlexRay bus. It shows how to work with the RTI model generated by the FlexRay Configuration Tool.

Required knowledge

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.
📖	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.

Introduction

Introduction

The following topics give a short introduction to the RTI FlexRay Configuration Blockset and describe how a real-time system is connected to a FlexRay bus.

Where to go from here

Information in this section

Features of the RTI FlexRay Configuration Blockset

Features

The main features of the RTI FlexRay Configuration Blockset are:

- Generating RTI FlexRay communication blocks configured for your FlexRay network. The necessary configuration data is created by the FlexRay Configuration Tool based on a FIBEX file or AUTOSAR system description file.
- Supporting single-channel and dual-channel FlexRay systems
- Providing configured Simulink blocks for
  - Time-triggered task execution
  - Sending and receiving signals
  - Accessing raw data of send or receive frames
  - Controlling the checksum calculation for frames
  - Reading the status of frames
  - Configuring the controller

  - Synchronizing the FlexRay cluster and real-time system
  - Enabling or disabling the communication of an ECU
  - Updating the configured Simulink blocks if the configuration was changed
- Providing Simulink blocks for
  - Controlling the communication layer and timetable
  - Handling bus errors
  - Handling deadline violations
  - Reading status information
  - Handling interrupts
  - Resetting the controller
- Simulating the FlexRay node on a dSPACE real-time system
- Supporting PDU-based modeling
- Simulating several FlexRay buses on one dSPACE real-time system, for example, to simulate a gateway

Related topics

Basics

Connecting Real-Time Systems to the FlexRay Bus.. 8

Connecting Real-Time Systems to the FlexRay Bus

Introduction

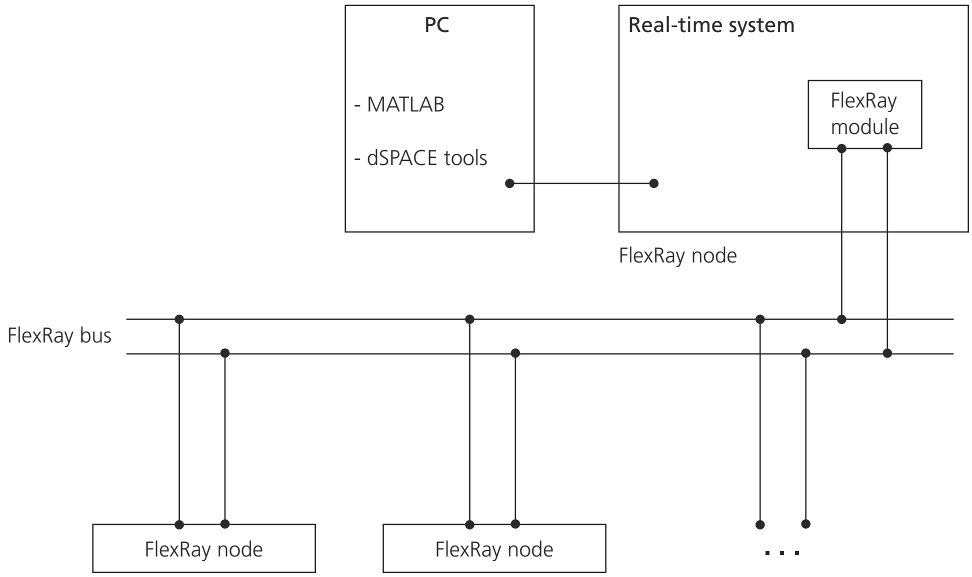
To configure a real-time system based on DS1006/DS1007 or a MicroAutoBox II as a simulation node in a FlexRay network, the FlexRay Configuration Tool and the RTI FlexRay Configuration Blockset are required. Various real-time platforms and FlexRay modules are supported.

Note

The *RTI FlexRay Configuration Blockset* does not support MicroAutoBox III and SCALEXIO systems. If you work with a MicroAutoBox III or a SCALEXIO system, you must use the *FlexRay Configuration Blockset* instead.

Connecting real-time systems to the FlexRay bus

A real-time system (based on DS1006/DS1007 or the MicroAutoBox II) can be connected to a FlexRay bus. The system must be equipped with a FlexRay module, which contains the FlexRay controller.



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)

Hardware installation

DS1006, DS1007 For details on configuring and installing the hardware of a modular system based on a DS1006 or DS1007, refer to:

- Setting Up I/O Boards (DS1006 Hardware Installation and Configuration Guide 🕮)
- Setting Up I/O Boards (DS1007 Hardware Installation and Configuration Guide 🕮)

For details on the I/O boards and the FlexRay modules, refer to:

- DS4501 IP Carrier Board (PHS Bus System Hardware Reference 🕮)
- DS4505 Interface Board (PHS Bus System Hardware Reference 🕮)

MicroAutoBox II For a description of the installation procedure on a MicroAutoBox II, refer to Connecting to a FlexRay Bus (MicroAutoBox II Hardware Installation and Configuration Guide 🕮).

Related topics

Basics

Features of the RTI FlexRay Configuration Blockset.. 7

Setting up a FlexRay Network

Introduction

The following topics show how you can work with the RTI FlexRay Configuration Blockset.

Where to go from here

Information in this section

Working with the RTI FlexRay Configuration Blockset

Introduction

Some of the RTI blocks must be configured by the FlexRay Configuration Tool before they can be used. If you change the FlexRay configuration, you can update them using an RTI block.

Where to go from here

Information in this section

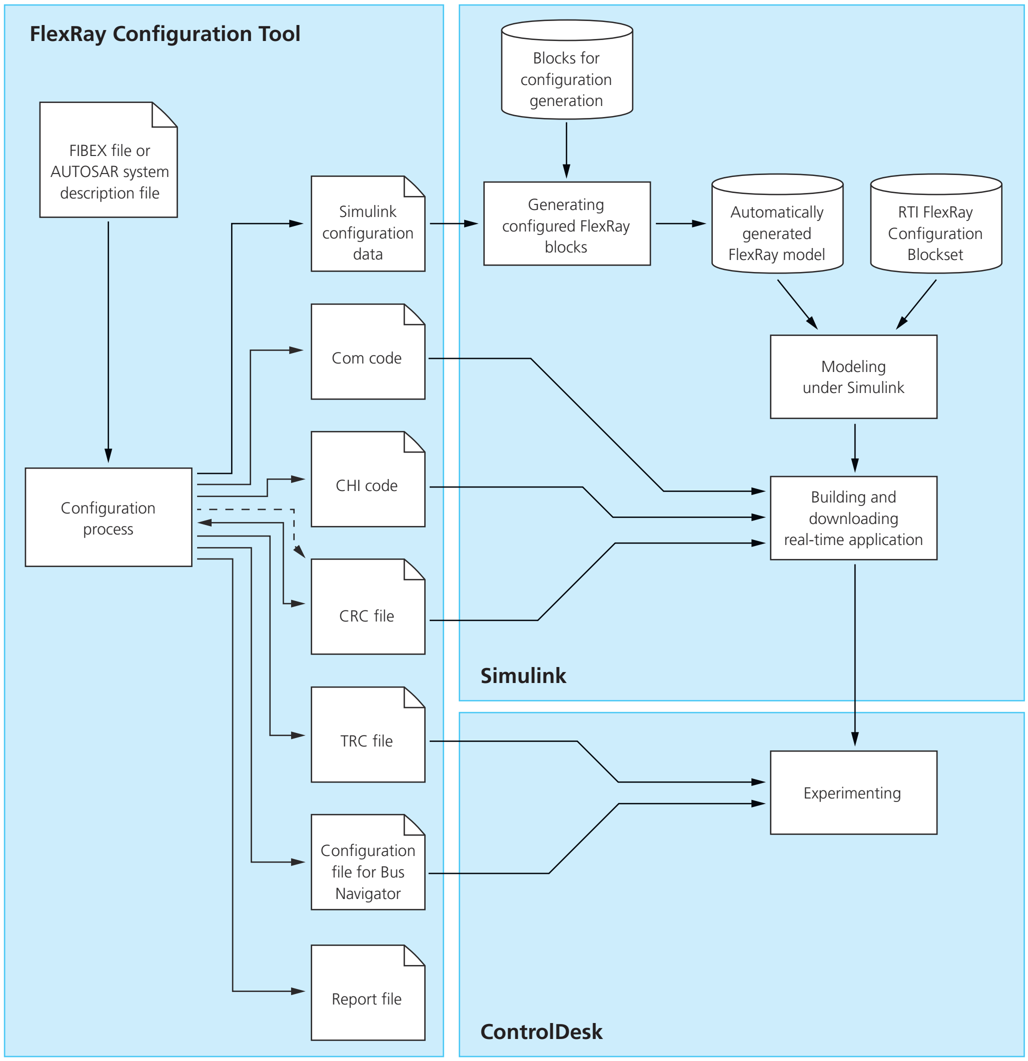
Overview of the Workflow

Introduction

The workflow for setting up a FlexRay network for real-time simulation is described below. Some of the RTI blocks must be configured with the FlexRay Configuration Tool before they can be used.

Workflow

The following illustration gives an overview of the workflow.



The workflow is divided into several processes:

Configuration process The configuration process is performed in the FlexRay Configuration Tool. It creates Simulink configuration data, communication code, and CHI code based on a FIBEX file or AUTOSAR system description file. For information on the workflow, refer to FlexRay Configuration with the FlexRay Configuration Tool (FlexRay Configuration Tool Guide 🕮).

Tip

You can perform the configuration process and the model design at different PCs. If you want to do this, you must copy the generated files (Simulink configuration data, Com, CHI code, and TRC file) to the PC where the RTI FlexRay Configuration Blockset is installed.

Generating configured FlexRay blocks The Simulink configuration data has all the necessary parameters for building the RTI blocks needed to model the FlexRay node. The RTI FlexRay Configuration Blockset contains a MATLAB command which generates the RTI blocks. The automatically generated FlexRay model has RTI blocks for each configured PDU that contains several signals. Refer to How to Generate RTI Blocks for Designing a FlexRay Node on page 15.

Modeling under Simulink When modeling the FlexRay communication, you work with two different blockset parts:

- The RTI FlexRay Configuration Blockset, which provides RTI blocks independently of your FlexRay configuration.
- The automatically generated FlexRay model, which provides RTI blocks generated for your FlexRay configuration. For instructions on using the RTI blocks for modeling, refer to PDU-Based Modeling on page 20.

Updating the configured FlexRay blocks If you change the FlexRay configuration, it is not necessary to replace the configured FlexRay blocks in your Simulink model manually. You can update them using an RTI block. Refer to How to Update the FlexRay Blocks in Simulink Models on page 17.

To update an existing FlexRay configuration under Simulink, you can use the update mode of the RTIFLEXRAYCONFIG model generation function. Refer to How to Update the Automatically Generated FlexRay Model on page 16.

Build and download real-time application When the design of your real-time model is finished, you can build the code for the real-time application and download it to the real-time system. Refer to Real-Time Simulation with FlexRay Networks on page 57.

Experimenting When the real-time application is downloaded to your real-time hardware, you can start the simulation. Refer to How to Specify Simulation Parameters on page 58.

Related topics

Basics

Introduction to the FlexRay Configuration Tool (FlexRay Configuration Tool Guide 🕮)

References

General Information on the RTI FlexRay Configuration Blockset (RTI FlexRay Configuration Blockset Reference 🕮)

How to Generate RTI Blocks for Designing a FlexRay Node

Objective

The FlexRay Configuration Tool writes a file containing all the necessary parameters for the real-time model (Simulink configuration data). Before you can start to design the real-time model of your FlexRay node, you must start a generation process via a MATLAB command. In this process the RTI blocks are configured for your FlexRay node based on the Simulink configuration data.

Automatically generated FlexRay model

An automatically generated FlexRay model contains all the RTI blocks configured for your FlexRay network. The following method describes how you can generate such a PDU-based model. You can use the generated model as a library for designing the real-time application for your FlexRay node. The RTI blocks are clustered in subsystems according to their types. Refer to Structure of the Automatically Generated FlexRay Model (RTI FlexRay Configuration Blockset Reference 📖).

Preconditions

The Simulink configuration data must be generated using the FlexRay Configuration Tool (see Basics of Code Generation (FlexRay Configuration Tool Guide 📖)).

Method

To generate RTI blocks for designing a FlexRay node

1 Open MATLAB.

2 In the MATLAB Command Window, enter

```
rtiflexrayconfig_modelgenerate('FileName')
```

or

```
rtiflexrayconfig_modelgenerate('FileName', 'BlockSet', 'PDU')
```

to start the model generation.

`FileName` is the file name of an M file containing the Simulink configuration data. It is generated by the FlexRay Configuration Tool. If the M file is not located in the current folder, you must additionally specify the whole path, for example, 'c:\mydir\filename.m'. You can specify further optional parameters, see rtiflexrayconfig_modelgenerate (RTI FlexRay Configuration Blockset Reference 📖).

Result

The FlexRay model is generated with the name specified in the FlexRay Configuration Tool. After the generation process, the model is opened.

A configured RTIFLEXRAYCONFIG UPDATE block is in the root level of the generated model. The block can be used for updating your Simulink application model if the FlexRay configuration changes. It contains information on the configuration and the FIBEX file or AUTOSAR system description file.

Related topics

HowTos

References

RTIFLEXRAYCONFIG UPDATE (RTI FlexRay Configuration Blockset Reference 📖)
rtiflexrayconfig_modelgenerate (RTI FlexRay Configuration Blockset Reference 📖)

How to Update the Automatically Generated FlexRay Model

Objective

You can use the following method if you want to extend the automatically generated FlexRay model. The method should only be used if you have added signals to the configuration after generating the FlexRay model. It is not recommended to use this method if you changed the configuration parameters of existing RTI blocks.

Basics

The **`rtiflexrayconfig_modelgenerate`** command modifies the automatically generated FlexRay model. The modified model contains all the configured blocks. You can use it, for example, to add a new configured frame or signal. It does not change or delete blocks which are already used in your Simulink model. You can replace the changed FlexRay blocks in your Simulink model manually or automatically via the RTIFLEXRAYCONFIG UPDATE block. Using the block may be time-consuming (specially in large models) but it ensures that all FlexRay blocks are replaced (see How to Update the FlexRay Blocks in Simulink Models on page 17).

Preconditions

The automatically generated FlexRay model must be generated beforehand.

Method

To update the automatically generated FlexRay model

1 Open MATLAB.

2 In the MATLAB Command Window, enter

```
rtiflexrayconfig_modelgenerate('FileName','GenerationMode','Update')
```

or

```
rtiflexrayconfig_modelgenerate('FileName', 'Blockset', 'PDU',
 'GenerationMode', 'Update')
```

to update the model.

`FileName` is the file name of an M file containing the Simulink configuration data. The `GenerationMode` and `Update` parameters start the command in the update mode.

Result

The automatically generated FlexRay model is generated and overwrites the previous model. It contains all the configured FlexRay blocks. The configured FlexRay blocks in your Simulink model are not updated. This can be done using the RTIFLEXRAYCONFIG UPDATE block (see below).

Related topics

References

RTIFLEXRAYCONFIG UPDATE (RTI FlexRay Configuration Blockset Reference 📖)
rtiflexrayconfig_modelgenerate (RTI FlexRay Configuration Blockset Reference 📖)

How to Update the FlexRay Blocks in Simulink Models

Objective

When you have used the blocks of the automatically generated FlexRay model in your Simulink model and then changed the FlexRay configuration, you must update your Simulink model. You can use the RTIFLEXRAYCONFIG UPDATE block to update your Simulink model automatically. This is useful when you have a lot of changes in a large model.

Basics

The automatically generated FlexRay model contains a RTIFLEXRAYCONFIG UPDATE block configured for your FlexRay configuration. If you change the FlexRay configuration afterwards, you can use this block to update all the FlexRay blocks used in your Simulink model.

Note

You must use the RTIFLEXRAYCONFIG UPDATE block which is configured first for your configuration. Do not replace this block by the new generated RTIFLEXRAYCONFIG UPDATE block. This block cannot update the RTI blocks used before.

The block updates all the FlexRay blocks used in your Simulink model. Additionally, it generates a reduced automatically generated FlexRay model, named `<ModelName>_diff.mdl`. The model contains only the Simulink blocks of the FlexRay configuration which are not used in your FlexRay model. All the RTI blocks which are used in your Simulink model are updated. Obsolete blocks from your Simulink model can be deleted.

Note

The mapping subsystems which belong to the RTIFLEXRAYCONFIG PDU RX and RTIFLEXRAYCONFIG PDU TX blocks are not updated. You must update the mapping subsystems manually.

Tip

It is recommended to use the update function of the RTIFLEXRAYCONFIG UPDATE block even if the FlexRay configuration has not changed. The reduced automatically generated FlexRay model is specially useful for keeping track of large Simulink models.

Preconditions

The automatically generated FlexRay model must be generated beforehand. The RTIFLEXRAYCONFIG UPDATE block must be in your Simulink model.

Method

To update the FlexRay blocks in Simulink models

1. Open the RTIFLEXRAYCONFIG UPDATE block in your Simulink model.
2. Specify the basis RTI FlexRay configuration.
3. Select or unselect the Delete obsolete block(s) option. If you select the option all the obsolete FlexRay blocks are deleted in your Simulink model. Obsolete blocks are blocks which were dragged from the previous automatically generated FlexRay configuration and do not exist in the updated FlexRay configuration.
4. Click Update.

Result

All the configured FlexRay blocks in your Simulink model are updated. A reduced FlexRay model is generated automatically, containing only blocks which are not used in your Simulink model. Two log files are created in the working directory. When the update process is completed, you can find links to these two log files in the MATLAB workspace:

- `<ModelName>_diff.mdl` containing all the updated blocks.
- `<ModelName>_UpdateSummary.log` lists the unused blocks from the library and old blocks in the model. The log file also displays the number of updated blocks.

Related topics

HowTos

How to Update the Automatically Generated FlexRay Model... 16

References

RTIFLEXRAYCONFIG UPDATE (RTI FlexRay Configuration Blockset Reference 📖)

PDU-Based Modeling

Introduction

With PDU-based modeling, the automatically generated FlexRay model has RTI blocks for each configured PDU that comprises several signals, i.e., PDU-based modeling handles several signals with one Simulink block. You can access the single signals via the PDU blocks.

Where to go from here

Information in this section

Information in other sections

Blocks for PDU-Based Bus Simulation (RTI FlexRay Configuration Blockset Reference 📖)
Describing the blocks that are used for generating a configuration for PDU-based modeling.

How to Send or Receive Signals of PDUs

Objective

The following instructions are on using the blocks and mapping subsystems for sending and receiving PDUs and their signals.

RTI blocks

PDUs and their signals are sent using the RTIFLEXRAYCONFIG PDU TX block. PDUs and their signals are received using the RTIFLEXRAYCONFIG PDU RX block. During the configuration process, these blocks are generated for each PDU which is sent or received. Several mapping subsystems are also generated and connected to the blocks to simplify their handling. The blocks and the mapping subsystems are stored in the automatically generated FlexRay model.

The block names of the generated blocks have information on the PDU:

`<PDU-/Frame-Name>_ID_BC_CR_PDU_TX/RX` (for single-channel FIBEX or AUTOSAR system description files)
`<PDU-/Frame-Name>_ID_BC_CR_PDU_TX/RX_Ch` (for dual-channel FIBEX or AUTOSAR system description files)

- ID: Slot ID
- BC: Base cycle
- CR: Cycle repetition
- PDU_TX: Send PDU
  PDU_RX: Receive PDU
- Ch: Used channel (A, B, AB)

Note

RTIFLEXRAYCONFIG PDU TX and RTIFLEXRAYCONFIG PDU RX blocks can be used for PDUs. PDUs are defined in FIBEX+, FIBEX 3.x and FIBEX 4.x versions and AUTOSAR System Templates.

In FIBEX 2.0 and lower versions there are no PDU elements available, only frames. The RTIFLEXRAYCONFIG PDU TX and RTIFLEXRAYCONFIG PDU RX blocks can also be used for frames, with one frame containing one PDU.

The block dialog (PDU page) shows whether the data of the FIBEX file is interpreted as a PDU or a frame.

Method

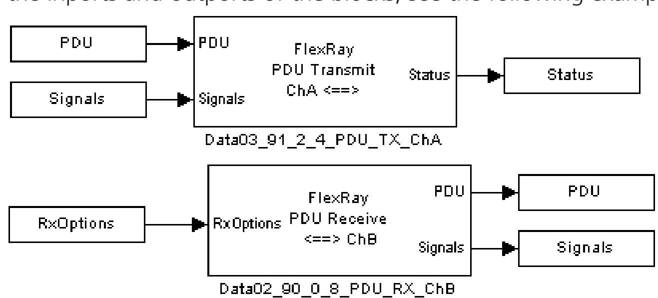
To send or receive signals of PDUs

1 Open the automatically generated FlexRay model.

2 Open the subsystems to access the RTIFLEXRAYCONFIG PDU RX or RTIFLEXRAYCONFIG PDU TX blocks. The blocks are nested in several subsystems. Open the subsystems in the following order:

1. RTIFLEXRAYCONFIG PDU Blocks subsystem
2. <ECU_Name> subsystem (<ECU_Name> is the short name of the ECU which sends or receives the signal)
3. Static Transmission, Dynamic Transmission, Network Management or <Application_Name> subsystem (<Application_Name> is the name of an application task that PDUs and their signals are assigned to)

The subsystems contain all the PDUs and their signals which are configured for sending or receiving. Several mapping subsystems are used to structure the inports and outports of the blocks, see the following examples.



For details on the structure of the subsystems, refer to RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference) and RTIFLEXRAYCONFIG PDU RX (RTI FlexRay Configuration Blockset Reference).

3 To send a PDU, drag its RTIFLEXRAYCONFIG PDU TX block and the connected mapping subsystems (PDU, Signals, and Status), if available, to the Simulink model.

To receive a PDU, drag its RTIFLEXRAYCONFIG PDU RX block and the connected mapping subsystems (RXOptions, PDU, and Signals), if available, to the Simulink model.

Result

The Simulink model is prepared for sending or receiving the signals of the PDU. The signals are available in the **Signals** mapping subsystem. You can connect each signal to further Simulink blocks in the Simulink model.

Related topics

HowTos

How to Generate RTI Blocks for Designing a FlexRay Node.. 15

References

RTIFLEXRAYCONFIG PDU RX (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 📖)
Structure of the Automatically Generated FlexRay Model (RTI FlexRay Configuration Blockset Reference 📖)

Sending Static PDUs and Sub-PDUs

Introduction

You can enable or disable the sending of static PDUs and sub-PDUs in PDU-based modeling.

Enabling or disabling static PDUs

You can enable or disable the sending of static PDUs using the **RTIFLEXRAYCONFIG PDU TX** block. The block has two inports in the PDU system which you can use:

HWEnable The **HWEnable** inport enables or disables the sending of static TX PDUs via hardware: that is, by enabling or disabling the controller TX buffer which is reserved for the relevant bus slot. As a consequence, all the PDUs which share the same bus slot and therefore are assigned to the same controller TX buffer are enabled or disabled together. Some PDUs cannot be enabled or disabled in this way, for example, startup and sync PDUs.

Note

- If the Simulink model contains several PDU TX blocks whose PDUs share the same bus slot, their **HWEnable** inports manipulate the same slot. In this case, the setting of the most recently calculated **HWEnable** port automatically becomes the valid setting. To safely enable or disable the sending of static PDUs which share the same slot, you must either enable or disable all the **HWEnable** inputs consistently at the same time, or use only one PDU TX block in your model, if possible.
- Enabling or disabling the sending of static TX PDUs via hardware has a higher priority than via software. As a consequence, all TX PDUs which share the same bus slot are automatically enabled by hardware *and* software, if one of them is enabled by its **HWEnable** inport.

SWEnable The **SWEnable** inport enables or disables the sending of static TX PDUs via software. This is possible for each static PDU.

Sending null frames or regular data

The FlexRay Configuration Tool allows explicit sending of null frames for static PDUs. When a static PDU is disabled or enabled via software, either regular data or a null frame is sent, according to the database version and settings made in the FlexRay Configuration Tool. The following describes which data is sent in each case.

Static PDU is disabled via software If a static PDU is *disabled via software* (`SWEnable` = 0), the data that is sent depends on the following factors:

- Database version
- Setting of the **SW Enable Configuration** property (see General Page (FlexRay Configuration Tool Reference 📖))
- Settings of the CHI Code Generator (see Generators Page (FlexRay Configuration Tool Reference 📖))

The following table shows which data is sent in a specific case.

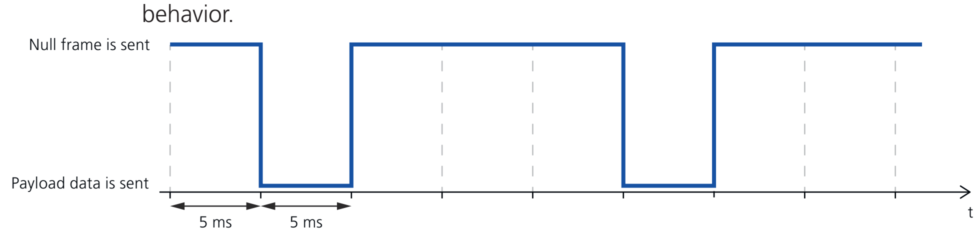
Database Version	SW Enable Configuration (General Properties)	Static TX Buffer Transmission Mode (CHI Code Generator)	Behavior
FIBEX version ≤ 2.0	Control of L-PDU commit to FlexRay buffer[1)]	Event (null frame used)	Null frames are sent.
		State (old value used)	Old data is sent.

Database Version	SW Enable Configuration (General Properties)	Static TX Buffer Transmission Mode (CHI Code Generator)	Behavior
FIBEX+, FIBEX 3.x, FIBEX 4.1.x, or AUTOSAR System Template	Control of L-PDU commit to FlexRay buffer	Event (null frame used)	Null frames are sent.
		State (old value used)	Old data is sent. The update bit has the value that was set before SWEnable was set to 0.
	Control of I-PDU payload data update	Event (null frame used)	The update bit is set to 0 and old data is sent. **Note** An update bit value of 0 due to the SWEnable setting can be overruled. This can happen when the update bit of a PDU is manipulable. If the UpdateBitEnable variable is 1 (which means that automatic calculation of the update bit is disabled), the update bit of the PDU is set to the value specified by the UpdateBitValue variable. If SWEnable is 0 and the update bit value fed to the UpdateBitValue inport is 1, the PDU still sends old data. If UpdateBitEnable is 0, the SWEnable setting is used as specified to enable/disable the sending of static TX PDUs.
		State (old value used)	The update bit is set to 0 and old data is sent.

1) This setting cannot be changed.

Static PDU is enabled via software If a static PDU is *enabled via software* (`SWEnable` = 1), usually the PDU payload data is sent. However, there is one exception to this rule: An LPDU that contains exactly one PDU that neither has been updated nor has a PDU update bit, has a null frame sent instead of the LPDU with old payload data. This transmission behavior enables you to detect whether a received PDU contains updated payload data, even if no PDU update bit exists.

The following illustration shows an example for this null frame transmission behavior.



In the example, the LPDU is sent cyclically every 5 ms. Its cycle repetition (defined in absolutely scheduled timing) is 1. The only PDU contained in the LPDU has a cyclic timing of 20 ms. The PDU is updated every 20 ms, which means in every fourth transmission cycle of the LPDU. When the PDU is updated, no null frame is sent. Instead, the new PDU data is packed into the LPDU and the new LPDU data is committed to the FlexRay controller. In the cycles in which the PDU is not updated but has old data, a null frame is sent.

> **Note**
>
> Timings of PDUs can change during run time. For example, this can happen when you use a static PDU with sub-PDUs that have their own timings or if you work with different transmission modes for a PDU.

If a static PDU is *enabled via software*, the data which is sent depends on the following factors:

- Database version
- Setting of the **SW Enable Configuration** property (see General Page (FlexRay Configuration Tool Reference 🕮))
- Settings of the CHI Code Generator (see Generators Page (FlexRay Configuration Tool Reference 🕮))

Sending null frames instead of old payload data is possible if the following conditions are met:

- Database version = FIBEX+, FIBEX 3.x, FIBEX 4.1.x, or AUTOSAR System Template
- **SW Enable Configuration** = 'Control of L-PDU commit to FlexRay buffer'
- **Static TX buffer transmission mode** = 'Event (null frame used)'

In all other cases, old payload data is sent as a rule.

Triggering sub-PDUs

A sub-PDU is subordinate to a PDU. It contains signals only. Sub-PDUs of a static PDU can have their own timing (CT parameter), but use the same slot of the superordinate PDU (specified by ID, BC, and CR parameters). A sub-PDU of a static PDU can be triggered by triggering the corresponding static PDU with a switch code. The trigger type SSC is used for this.

The **RTIFLEXRAYCONFIG PDU TX** block provides the **EnableMultiplexerSwitchCode** inport in the PDU system, which lets you specify the switch code for static sub-PDUs.

Related topics

HowTos

How to Manipulate the Update Bit of a PDU 35

References

RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮)

Sending Dynamic PDUs or Sub-PDUs

Introduction

Dynamic PDUs or sub-PDUs must be sent by using the RTIFLEXRAYCONFIG PDU TX block, see Triggering of Dynamic Frames and Subframes (FlexRay Configuration Tool Guide 📖).

Tip

The following section describes how you can trigger the dynamic PDUs or sub-PDUs via the Simulink model. You can also trigger them using the TRC file, see How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File (FlexRay Configuration Tool Guide 📖) and Using the Generated TRC File of PDU-Based Modeling on page 63.

Enabling or disabling dynamic PDUs

Cyclic PDUs (DC and DEC types) are enabled by default if they contain no sub-PDUs. You can use the TxEnable inport in the PDU mapping subsystem which is connected to the RTIFLEXRAYCONFIG PDU TX block to enable or disable the sending of a dynamic PDU.

Triggering PDUs

The timing of a PDU is specified by several parameters (ID, BC, CR, and CT). Whether a dynamic PDU can be sent cyclically or event-triggered depends on the PDU trigger type DE, DC, or DEC (see Triggering of Dynamic Frames and Subframes (FlexRay Configuration Tool Guide 📖)). A PDU can contain signals and/or sub-PDUs.

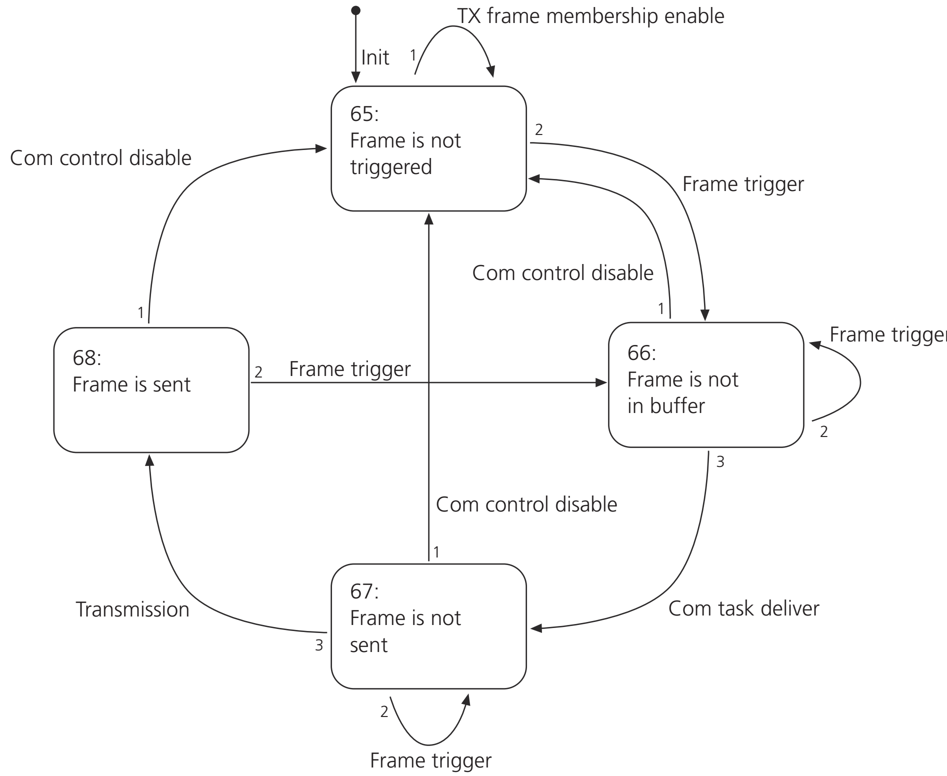
Triggering sub-PDUs

A sub-PDU is subordinate to a PDU. Sub-PDUs of a PDU can have their own timing (CT parameter) but use the same slot of the superordinate PDU (specified by ID, BC, and CR parameters). A switch code specifies the sub-PDU to trigger. Whether a dynamic sub-PDU can be sent cyclically or be event-triggered depends on the trigger type SDE, SDC, or SDEC (see Triggering of Dynamic Frames and Subframes (FlexRay Configuration Tool Guide 📖)). A sub-PDU contains signals only.

If a sub-PDU does not have its own timing, it inherits the timing of the (parent) PDU in FIBEX version lower than 3.0. Since FIBEX version 3.0 and for AUTOSAR System Templates, sub-PDUs cannot inherit the timing of the parent PDU. These sub-PDUs are interpreted as dynamic event sub-PDUs. Cyclic transmission of the sub-PDUs is not possible, they must be triggered.

Monitoring the transmit status

You can monitor the transmit status of dynamic event-based PDUs using the RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS block. You can use the block in tasks other than the task where the RTIFLEXRAYCONFIG PDU TX block is used. The RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS block has an outport where you can read the status, see the following illustration.



The block's output values are as follows:

1. After initialization, the block output value is 65 (PDU not triggered).
2. When you have triggered the dynamic PDU, the output value is 66 (PDU was triggered but PDU is not in the send buffer).
3. When the Com task has written the PDU data to the send buffer, the output value is 67 (PDU is in the send buffer but it was not yet sent).
4. When the FlexRay controller has sent the PDU, the output value is 68 (PDU was sent).

Related topics

References

RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮)

How to Handle Checksum Calculation for a PDU

Objective

If you have assigned a CRC algorithm to a PDU in the FlexRay Configuration Tool, you can enable or disable the checksum calculation or select another CRC algorithm.

Basics

You can implement your own checksum algorithm (CRC algorithm). You must implement the CRC algorithms in a C-coded source file (CRC C file) using a special template. You can assign the CRC algorithm to TX or RX PDUs in the FlexRay Configuration Tool. The RTIFLEXRAYCONFIG PDU RX and RTIFLEXRAYCONFIG PDU TX blocks have inports that enable or disable checksum calculation and inports that select another CRC algorithm. For details, refer to Basics on Implementing Checksum Algorithms (FlexRay Configuration Tool Guide 📖).

The method described below must be executed for each PDU whose checksum calculation you want to change.

Changing CRC C file

You can use the FLEXRAYCONFIG UPDATE block to select another CRC C file than that specified with the FlexRay Configuration Tool, refer to CRC Settings Page (RTIFLEXRAYCONFIG UPDATE) (RTI FlexRay Configuration Blockset Reference 📖).

CRC manipulation via TRC file

If CRC calculation is enabled and it is selected for the TRC file in the FlexRay Configuration Tool (see Basics on Implementing Checksum Algorithms (FlexRay Configuration Tool Guide 📖)), corresponding variables are also written to the TRC file. Thus, you can manipulate CRC calculation in ControlDesk. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Preconditions

- You must have assigned a CRC algorithm to the PDUs. For details, refer to How to Assign a Checksum Algorithm to Frames (FlexRay Configuration Tool Guide 📖).
- The RTIFLEXRAYCONFIG PDU RX or RTIFLEXRAYCONFIG PDU TX block and the connected mapping subsystems of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To handle checksum calculation for a PDU-based modeling

1. If the PDU is transmitted, open the PDU mapping subsystem connected to its RTIFLEXRAYCONFIG PDU TX block.

   If the PDU is received, open the RXOptions mapping subsystem connected to its RTIFLEXRAYCONFIG PDU RX block.

2 Connect the inports of the block to appropriate Simulink blocks.

Port	Description
CRCEnable	Enables or disables checksum calculation.
CRCType	Selects the CRC algorithm. The value must be one of the ID values that were specified in the FlexRay Configuration Tool.

3 Repeat the steps above for all the PDUs whose CRC algorithm you want to change.

Result

You have changed the checksum calculation for PDUs.

Related topics

References

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

How to Send PDUs in Raw Format

Objective

The following instructions are on using the blocks for sending PDUs in raw format.

Raw data access

When a PDU is configured for raw data access, the RTIFLEXRAYCONFIG PDU TX block has additional ports. The ports can be used to access each bit or byte of a PDU, ignoring the specified signals.

Raw data in TRC file

If raw data access is enabled for PDUs and selected for the TRC file in the FlexRay Configuration Tool (see How to Configure a Frame for Raw Data Access (FlexRay Configuration Tool Guide 📖)), access variables are also written to the TRC file. Thus, you can manipulate raw data in ControlDesk. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Preconditions

- The PDU must be configured for raw data access in the FlexRay Configuration Tool, refer to How to Configure a Frame for Raw Data Access (FlexRay Configuration Tool Guide 📖).
- The RTIFLEXRAYCONFIG PDU TX block of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To send PDUs in raw format

1 Open the PDU mapping subsystem that is connected to the RTIFLEXRAY PDU TX block.

2 Connect the inports of the subsystem to appropriate Simulink blocks:

Port	Description
RawDataEnable	Enables raw data access
RawDataTxBytes	Provides a vector of bytes containing data to be written to the TX PDU. The number of bytes is specified by the Max number of raw data bytes property, which can be specified in the FlexRay Configuration Tool.
RawDataStartPosition	Specifies the start position within the TX PDU in bits. The value must be specified in bits and is therefore not limited to bytes. A value of 0 starts with the first bit position. The maximum value depends on the payload length of the TX PDU, which is stated in bytes.
RawDataLength	Specifies the number of bits which are accessed. If the value is 0, no data is written. The maximum value depends on the Max number of raw data bytes property which can be specified in the FlexRay Configuration Tool. It is stated in bytes.

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

3 Open the Status mapping subsystem that is connected to the RTIFLEXRAY PDU TX block.

4 Connect the outports of the subsystem to appropriate Simulink blocks:

Port	Description
RawDataAccessStatus	Displays status information of raw data access.

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

Result

Your model is prepared for raw data access. You can change each bit of the PDU to be sent.

Related topics

HowTos

References

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

How to Receive PDUs in Raw Format

Objective

The following instructions are on using the blocks for receiving PDUs in raw format.

Raw data access

When a PDU is configured for raw data access, the RTIFLEXRAYCONFIG PDU RX block has additional ports. The ports can be used to access each bit or byte of a PDU, ignoring the specified signals.

Raw data in TRC file

If raw data access is enabled for PDUs and selected for the TRC file in the FlexRay Configuration Tool (see How to Configure a Frame for Raw Data Access (FlexRay Configuration Tool Guide 📖)), variables for accessing are also written to the TRC file. Thus, you can monitor raw data in ControlDesk. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Preconditions

- The PDU must be configured for raw data access in the FlexRay Configuration Tool, refer to How to Configure a Frame for Raw Data Access (FlexRay Configuration Tool Guide 📖).
- The RTIFLEXRAYCONFIG PDU RX block of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To receive PDUs in raw format

1 Open the RXOptions mapping subsystem that is connected to the RTIFLEXRAY PDU RX block.

2 Connect the inports of the block to appropriate Simulink blocks.

Port	Description
RawDataStartPos	Specifies the start position within the RX PDU in bits. The value must be specified in bits and is therefore not limited to bytes. The value 0 starts with the first bit position.

Port	Description
RawDataLength	The maximum value depends on the payload length of the RX PDU, which is stated in bytes. Specifies the number of bits which are accessed. If the value is 0, no data is read and the RawDataRxBytes outport contains the previously read data. The maximum value depends on the Max number of raw data bytes property which can be specified in the FlexRay Configuration Tool. It is stated in bytes.

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖).

3 Connect the outports of the block to appropriate Simulink blocks.

Port	Description
RawDataRxBytes	Provides a vector of bytes which contains the data to be written to the RX PDU.
RawDataAccessStatus	Displays status information of raw data access

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖).

Result

Your model is prepared for raw data access. You can read the bits of the received PDU.

Related topics

HowTos

How to Generate RTI Blocks for Designing a FlexRay Node........ 15
How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File (FlexRay Configuration Tool Guide 📖)
How to Send PDUs in Raw Format........ 30

References

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

How to Manipulate the Payload Length of a PDU

Objective

You can manipulate the payload length of the TX PDUs or read the payload length of RX PDUs when you use raw data access.

Preconditions

- The PDU must be configured for raw data access, and payload length manipulation or reading must be enabled in the FlexRay Configuration Tool, refer to How to Configure a Frame for Raw Data Access (FlexRay Configuration Tool Guide 🕮).
- The RTIFLEXRAYCONFIG PDU TX block of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To manipulate the payload length of a PDU

1 Open the PDU mapping subsystem connected to the RTIFLEXRAY PDU TX block.

2 Connect the inports of the block to appropriate Simulink blocks.

Port	Description
PayloadLengthEnable	Enables or disables payload length manipulation.
PayloadLengthValue	Sets the PDU's payload length value in bytes.

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

3 Open the Status mapping subsystem that is connected to the RTIFLEXRAY PDU TX block.

4 Connect the outports of the subsystem to appropriate Simulink blocks:

Port	Description
PayloadLengthStatus	Displays status information with payload length manipulation.

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

Result

You can manipulate the payload length of the TX PDU.

Tip

You can read the payload length of RX PDUs. The PDU mapping subsystem connected to the RTIFLEXRAY PDU RX block is given a PayloadLengthValue outport if the PDU is configured to read the payload length.

Related topics

References

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

How to Manipulate the Update Bit of a PDU

Objective

If an update bit of a PDU is specified in a FIBEX file or AUTOSAR system description file, you can manipulate its values. An update bit is a Boolean value which is true each time that the PDU is sent. It is false when the PDU is not sent, but the frame which contains this PDU is sent.

Tip

You can also manipulate the update bit via the TRC file if it is configured accordingly. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Preconditions

- The update bit of a PDU must be specified in the FIBEX or AUTOSAR system description file.
- The RTIFLEXRAYCONFIG PDU TX block of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To manipulate update bits

1 Open the PDU mapping subsystem connected to the RTIFLEXRAY PDU TX block.

2 Connect the inports of the block to appropriate Simulink blocks.

Port	Description
UpdateBitEnable	Enables or disables manual update bit manipulation.
UpdateBitValue	Sets the value of the update bit.

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖).

Result

You can set the update bit manually.

Tip

You can read the update bit of RX PDUs. The Status mapping subsystem connected to RTIFLEXRAY PDU RX block is given a UpdateBitValue outport. The value is updated only when a PDU is received.

Related topics

Basics

Sending and Receiving Signal- and Signal-Group-Specific Update Bits

Introduction

Communication cluster files can contain signal-specific and signal-group-specific update bits. The update bits of signals/signal groups are set when the associated TX PDU is sent. You can evaluate the received signal/signal group update bits for RX PDUs.

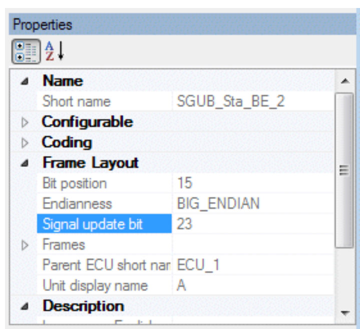
Sending update bits of signals/signal groups

The update bit of a signal/signal group is set to 1 each time that the TX PDU that the signal/signal group belongs to is sent. The update bit value is 0 if the TX PDU is not sent.

Receiving update bits of signals/signal groups

You can evaluate the received signal/signal group update bits for RX PDUs. The value of a signal group update bit is assigned to the signals belonging to that signal group.

The FlexRay Configuration Tool displays the signal update bit positions for signals of TX or RX PDUs in the **Properties** view. The bit positions are specified in the communication cluster file.



The signal update bit values of signals of an RX PDU are part of the **Signals** bus of the **RTIFLEXRAYCONFIG PDU RX** block. Refer to RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖).

Signal update bit values can also be made available in the TRC file. If they are selected for the TRC file in the FlexRay Configuration Tool, the <Signal_x> groups within the Receiving and Monitoring groups contain the Update Bit Value variable. For further information, refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Preconditions

Whether update bits of signals and/or signal groups can be received by RX PDUs depends on the database your FlexRay configuration is based on. The following table shows which signal- or signal-group-specific update bits the dSPACE FlexRay Configuration Package supports for the different database types:

Configuration Based On ...	Reception of Signal-Specific Update Bits	Reception of Signal-Group-Specific Update Bits
FIBEX 2.0 or earlier	–	–
FIBEX 3.0 or 3.1	✓	–
FIBEX 4.1, 4.1.1, or 4.1.2	✓	–
FIBEX+	✓	–
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 system description file	✓	✓

Related topics

Basics

Using the Generated TRC File of PDU-Based Modeling.. 63

References

RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖)

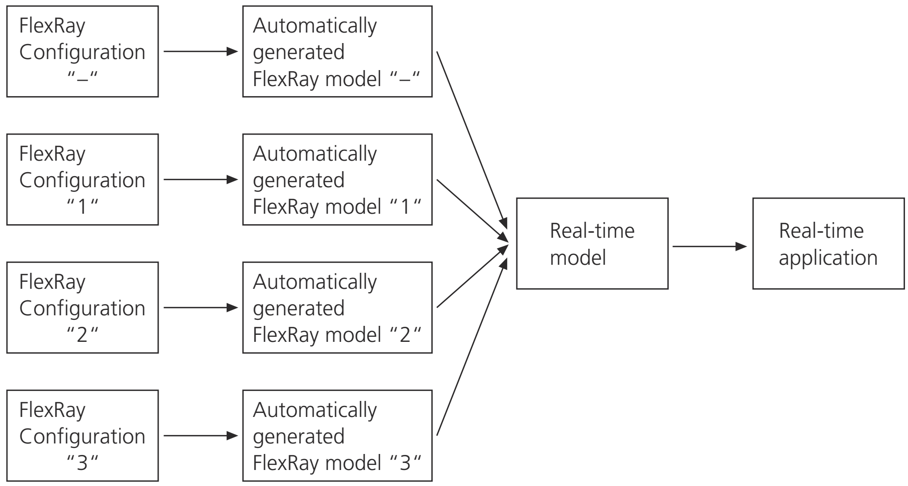
Modeling Several FlexRay Buses on One dSPACE Real-Time System

Introduction

You can use RTIFLEXRAYCONFIG blocks from different FlexRay configurations in one real-time model, for example, to model a gateway.

Overview

The following illustration shows the principle of generating a real-time model with several FlexRay configurations. Depending on the platform type you can use up to four FlexRay configurations in one real-time model.



Creating configurations

You must create a FlexRay configuration for each FlexRay bus which you want to implement in your real-time model, using the FlexRay Configuration Tool. The FlexRay configurations are the basis for the automatically generated FlexRay models. A configuration ID is used to identify blocks from different FlexRay configurations. It must therefore be unique for each FlexRay bus. The configuration ID also specifies the priority of the FlexRay configurations. For details, refer to How to Create Configurations for Multiple Buses (FlexRay Configuration Tool Guide 🕮).

Modeling

Before you can start modeling, you must generate the FlexRay model for each FlexRay configuration. For details, refer to How to Generate RTI Blocks for Designing a FlexRay Node on page 15.

After generating the FlexRay models (one for each configuration), you can use the generated and automatically configured RTI blocks to implement the FlexRay communication in your real-time model. Note the following points:

- The configuration ID is added as read-only information beneath the block name of the configured RTI blocks.
- It is recommended to implement always the FlexRay bus with configuration ID '-' in the real-time model.
- Some blocks of the RTI FlexRay Configuration Blockset must be assigned to a FlexRay configuration during modeling. You must select the configuration ID in a block parameter in these blocks.
- FlexRay configurations 1, 2, and 3 have a different synchronization task. For details, refer to TTE Synchronization Task Subsystem (RTI FlexRay Configuration Blockset Reference 🕮).
- The synchronization task of FlexRay configurations 1, 2, 3 are driven by the cycle count start interrupt of the FlexRay controller. It is therefore not possible to use the Generate Interrupt at start of every communication option for this controller. However, you can use the cycle count start interrupt using the Generate Interrupt for specific communication cycle option.

- One RTIFLEXRAYCONFIG UPDATE block is generated for each configuration. If you update the model with these blocks, you can also replace the basic configuration subsystem automatically. This is specially helpful if you changed the configuration ID, because the synchronization task subsystem is placed inside the basic configuration subsystem and configuration `"-"` has a different synchronization task than the other configurations.
- Only one CRC C file is allowed for all the FlexRay configurations used. However, you can implement different checksum algorithms in this CRC C file.
- The task belonging to FlexRay configuration`"-"` must have a higher priority than the task of the other configurations. You can check the priority during the build process.

  The tasks must have the following order, for example:
  1. `<Group 'Timetable ID: FRSyncTask'>`
  2. `<Group 'Timetable ID: ApplMode0'>`
  3. `C1_RTIFLEXRAYCONFIG_CC_START`
  4. `<Group 'Timetable ID: ApplMode0_C1'>`
  5. `C2_RTIFLEXRAYCONFIG_CC_START`
  6. `<Group 'Timetable ID: ApplMode0_C2'>`
  7. `C3_RTIFLEXRAYCONFIG_CC_START`
  8. `<Group 'Timetable ID: ApplMode0_C3'>`
  9. `Timer Task 1 [0.001] s`
- 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 the following blocks to an RTIFLEXRAYCONFIG UPDATE block manually:
  - RTIFLEXRAYCONFIG STATUS
  - RTIFLEXRAYCONFIG INTERRUPT
  - RTIFLEXRAYCONFIG CONTROLLER STOP
  - RTIFLEXRAYCONFIG CONTROLLER RESTART
  - RTIFLEXRAYCONFIG TX WAKEUP
  - RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT
  - RTIFLEXRAYCONFIG COM EVENT CONTROL
  - RTIFLEXRAYCONFIG COM CYCLIC CONTROL
  - RTIFLEXRAYCONFIG TIMETABLE CONTROL

Real-time simulation

In the real-time simulation, you want to observe the communication between the FlexRay buses. This can be done via variables which are available in the TRC file or via the Bus Navigator.

TRC file A TRC file is generated for the simulation. The generated TRC file contains variables for controlling the PDUs, their signals and raw data. To visualize the variables, you can build a data connection with instruments. If you have enabled the multiple bus option in the FlexRay Configuration Tool, configuration IDs are added to the structure of the TRC file:

```
FlexRay
   ConfigId -
      ...
   ConfigId 1
      ...
```

Below the `ConfigId` nodes, the structure is the same as for a single bus. For details, refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

The TRC file is supported by ControlDesk.

Bus Navigator ControlDesk's Bus Navigator is a graphical environment for handling bus configurations, for example, FlexRay configurations. It has a Bus Navigator tree which lists all the elements of the FlexRay configurations. The tree has a FlexRay node for configuration '-' and FlexRay_c*n* nodes (*n* is the configuration ID) for configurations 1, 2, 3. Under these nodes, the FlexRay variables are sorted by ECU and channel. For details, refer to Introduction to the Bus Navigator (ControlDesk Bus Navigator 📖).

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.

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 📖)

How to Switch the Transmission Mode of a PDU

Objective

If different transmission modes or timings are specified for PDUs, you can switch between them. Only one transmission mode can be active for a PDU at a time.

Tip

You can also switch the PDU transmission mode via the TRC file if it is configured accordingly. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

PDU transmission modes

The timing of PDUs is specified in the FIBEX or AUTOSAR system description file. A FIBEX or AUTOSAR system description file can contain different timings for one PDU. To create FlexRay configurations that use different timings for each PDU, the FlexRay Configuration Package provides transmission modes. The transmission modes are assigned to each PDU. Each transmission mode is assigned to a specific timing defined in the FIBEX or AUTOSAR system description file.

Besides the transmission modes with timings from the underlying FIBEX or AUTOSAR system description file, the FlexRay Configuration Package also provides transmission modes that allow you to create additional timings based on the corresponding LPDU timing.

For more information on configuring transmission modes, refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 🕮).

Preconditions

- Different transmission modes or timings of a PDU must be specified in the FIBEX or AUTOSAR system description file.
- The PDU must be configured for switching between its transmission modes during run time in the FlexRay Configuration Tool. Refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 🕮).
- The RTIFLEXRAYCONFIG PDU TX block of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To switch the transmission mode of a PDU

1 Open the PDU mapping subsystem connected to the RTIFLEXRAYCONFIG PDU TX block.

2 Connect the inports of the block to appropriate Simulink blocks.

Port	Description
TransmissionModeSelector	Selects the transmission mode.
TxTrigger	Triggers event-based sending of the PDU. This port is available only in connection with event-based and mixed timings. If the value is equal to or greater than 1, the event-based timing is activated. Otherwise, cyclic timing is used (in the case of mixed timings).

For more details on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

Result

You can switch the transmission mode of a PDU.

Related topics

HowTos

How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 🕮)

References

RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)

How to Switch the Minimum Delay Time Support of a PDU

Objective

If minimum delay time information is specified for PDUs in an AUTOSAR system description file, you can enable or disable minimum delay time support for them.

> **Tip**
>
> You can also enable or disable minimum delay time support via the TRC file if it is configured accordingly. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Minimum delay time support

Minimum delay time information is defined in the AUTOSAR system description file. This includes the definition of minimum delay time values for individual IPDUs and the ECU-specific definition on whether the defined minimum delay times are to be applied to event-based PDU transmissions only or also to cyclic PDU transmissions.

The minimum delay time of an IPDU specifies the minimum delay time between successive transmissions of this IPDU. It determines the time span (in seconds) that must elapse before new IPDU data can be packed into the LPDU, i.e., before a new transmission of this IPDU is possible.

For more information on configuring PDUs for minimum delay time support, refer to How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 📖).

Preconditions

- Minimum delay time information must be specified for the PDU in the AUTOSAR system description file.
- The minimum delay time feature must be enabled for the PDU in the FlexRay Configuration Tool. Refer to How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 📖).
- The RTIFLEXRAYCONFIG PDU TX block of the PDU must be in the Simulink model (see How to Send or Receive Signals of PDUs on page 21).

Method

To switch the minimum delay time support of a PDU

1 Open the PDU mapping subsystem connected to the RTIFLEXRAYCONFIG PDU TX block.

2 Connect the inports of the block to appropriate Simulink blocks.

Port	Description
MDTEnable	Enables or disables minimum delay time support.

For more information on the port, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖).

3 Open the **Status** mapping subsystem connected to the RTIFLEXRAYCONFIG PDU TX block.

4 Connect the outports of the block to appropriate Simulink blocks.

Port	Description
MDTStatus	Displays information on the last captured trigger and on whether and how it was considered for triggering.
MDTTime	Displays the remaining minimum delay time.

For more information on the ports, refer to RTIFLEXRAYCONFIG PDU TX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

Result

You can enable or disable the minimum delay time support of a PDU and get status information on the minimum delay time feature.

Related topics

HowTos

How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 🕮)

References

RTIFLEXRAYCONFIG PDU TX (RTI FlexRay Configuration Blockset Reference 🕮)

Controlling a FlexRay Network

Introduction

You can reset the FlexRay controller, read its status, use FlexRay-specific interrupts, control the communication layer, implement error handling and deadline violations, and change the standard synchronization task.

Where to go from here

Information in this section

How to Enable or Disable the Communication of an ECU

Objective

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

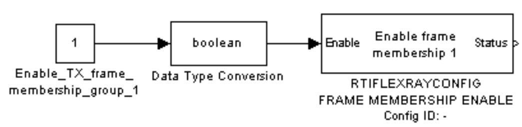
Preconditions

- The frame membership group must be set for the ECU, refer to Building Frame Membership Groups (FlexRay Configuration Tool Guide 🕮).
- The automatically generated FlexRay model must be generated beforehand.

Method

To select frame membership groups for enabling or disabling ECU communication

1. In the automatically generated FlexRay model, double-click the RTIFLEXRAYCONFIG Extras Blocks subsystem.
2. From the subsystem, drag an RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block to the model for each frame membership group you want to enable or disable.
3. Double-click the RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block to open its block dialog.
4. In the block dialog, select the settings for the frame membership group.
5. Click OK.
6. Connect the inport of the RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block with a constant and a data conversion block from the Simulink library.



Result

The frame membership groups are selected to enable or disable ECU communication via ControlDesk.

Related topics

References

RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE (RTI FlexRay Configuration Blockset Reference 🕮)

How to Reset FlexRay Controllers

Objective

You can define a soft reset for a FlexRay controller, for example, to be triggered by loss of synchronization. You can specify whether it then acts as a following coldstart node or may act as the leading coldstart node.

Cold start inhibit flag

You can set the cold start inhibit flag of the FlexRay controller. If the flag is set, the FlexRay controller can only act as a following coldstart node. This means that the node can either integrate itself into a running cluster or transmit startup frames after another coldstart node (the leading coldstart node) started the initialization of cluster communication. If the flag is not set, the FlexRay controller can act as a leading coldstart node, which can impair the communication of a running FlexRay cluster.

Restrictions

It is strongly recommended that you do not reset the FlexRay controller while it is in `NORMAL` state. Protocol errors could be generated if the FlexRay controller enters the `RESET` state while the FlexRay cluster is running.

Method

To reset FlexRay controllers

1. Open the RTI FlexRay Configuration Blockset.
2. Drag an RTIFLEXRAYCONFIG CONTROLLER RESET block to your Simulink model.
3. Open the RTIFLEXRAYCONFIG CONTROLLER RESET block and select the controller and trigger type.
4. To set the Coldstart Inhibit flag for the FlexRay controller, click the Options page and select the Enable cold start mode inport option.

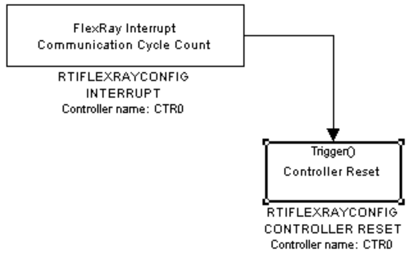
   The RTIFLEXRAYCONFIG CONTROLLER RESET block gets a block inport which you can use to set the cold start inhibit flag.
5. Connect the new input to a signal which triggers the reset.

Result

The controller is set to the soft reset state when the selected trigger signal arrives.

Example

The following example shows an RTIFLEXRAYCONFIG CONTROLLER RESET block which is triggered from an RTIFLEXRAYCONFIG INTERRUPT block.



Next steps

The trigger signal can be generated by an RTIFLEXRAYCONFIG INTERRUPT block. Refer to How to Use FlexRay-Specific Interrupts on page 48.

Related topics

HowTos

References

RTIFLEXRAYCONFIG CONTROLLER RESET (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG INTERRUPT (RTI FlexRay Configuration Blockset Reference 📖)

How to Use FlexRay-Specific Interrupts

Objective

You can use FlexRay-specific interrupts in your model, for example, to set a controller to the soft-reset state.

Trigger signals

Several states of a FlexRay communication can be used to generate a trigger signal, refer to RTIFLEXRAYCONFIG INTERRUPT (RTI FlexRay Configuration Blockset Reference 📖). For example, a signal can be triggered when the wakeup pattern is received on channel A or channel B.

Method

To use FlexRay-specific interrupts

1. Open the RTI FlexRay Configuration Blockset.
2. Drag an RTIFLEXRAYCONFIG INTERRUPT block to your Simulink model.
3. Open the RTIFLEXRAYCONFIG INTERRUPT block and select the controller and interrupt type.

Result	The trigger signal is available in the model and can be connected to another block or any other function-call subsystem, for example, an RTIFLEXRAYCONFIG CONTROLLER RESET block.
Example	For an example, refer to How to Reset FlexRay Controllers on page 47.
Next steps	You can use the trigger signal to reset the FlexRay controller. Refer to How to Reset FlexRay Controllers on page 47.
Related topics	References RTIFLEXRAYCONFIG INTERRUPT (RTI FlexRay Configuration Blockset Reference 📖)

How to Monitor the Status of FlexRay Controllers

Objective

You can monitor the status of a FlexRay controller.

Status information

Several items of status information are available for a FlexRay controller. The status information which is available depends on the controller type used. For more information, refer to RTIFLEXRAYCONFIG STATUS (RTI FlexRay Configuration Blockset Reference 📖).

Method

To monitor the status of FlexRay controllers

1. Drag an RTIFLEXRAYCONFIG STATUS block to your Simulink model.
2. Open the RTIFLEXRAYCONFIG STATUS block, click the Unit tab and select the controller.
3. Click the Options tab and select the outports which you want to monitor.
4. Enter the sample time. The sample time defines the time steps when the values are updated.
5. Connect the outports to a Simulink block, for example, a Gain block.

Result

The selected outports are available in your Simulink model.

Example

For an example, refer to the RTI demo library of the RTI FlexRay Configuration Blockset.

Related topics

References

RTIFLEXRAYCONFIG STATUS (RTI FlexRay Configuration Blockset Reference 🕮)

How to Send Wakeup Patterns

Objective

You can select a controller which transmits the wakeup patterns.

Basics

You can select the controller to transmit wakeup patterns using the RTIFLEXRAYCONFIG TX WAKEUP block.

Method

To send wakeup patterns

1. Open the RTI FlexRay Configuration Blockset.
2. Drag an RTIFLEXRAYCONFIG TX WAKEUP block to your Simulink model.
3. Open the RTIFLEXRAYCONFIG TX WAKEUP block and select the controller, channel type, and trigger type.
4. Connect the block to a signal which triggers the wakeup.

Result

A wakeup pattern is transmitted on the FlexRay bus by the selected controller.

Related topics

References

RTIFLEXRAYCONFIG TX WAKEUP (RTI FlexRay Configuration Blockset Reference 🕮)

How to Control the Communication Layer

Objective

You can control the execution of event-based and cyclic frames of the communication layer.

Controlling the communication layer

During the configuration process, a synchronization task is created in the automatic generated FlexRay model. The synchronization task contains an RTIFLEXRAYCONFIG COM ADMIN block which is triggered according to the synchronization state. If the host is synchronized, the communication layer is switched to an operation mode specified in RTIFLEXRAYCONFIG COM EVENT CONTROL and/or RTIFLEXRAYCONFIG COM CYCLIC CONTROL blocks. If the model does not contain the control blocks, the whole communication layer is switched online.

TX PDUs in dual-channel FlexRay systems

If you use TX PDUs in a dual-channel FlexRay system and that only one buffer to send data on both channels, and you disable the communication layer of one channel, the COM layer of the other channel is disabled automatically for these TX PDUs.

For further information on the operation mode of a communication layer, refer to Unit Page (RTIFLEXRAYCONFIG COM CYCLIC CONTROL) (RTI FlexRay Configuration Blockset Reference 🕮).

Method

To control the communication layer

1. Drag a triggered subsystem to your Simulink model and open it.
2. Open the RTI FlexRay Configuration Blockset.
3. Drag an RTIFLEXRAYCONFIG COM EVENT CONTROL block (for event-based frames) or an RTIFLEXRAYCONFIG COM CYCLIC CONTROL block (for cyclic frames) from the blockset to your Simulink model.
4. Open the dragged block and select an operation mode.

Result

When the host is synchronized, the communication layer is switched according to the specified operation mode.

Related topics

References

RTIFLEXRAYCONFIG COM ADMIN (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG COM CYCLIC CONTROL (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG COM EVENT CONTROL (RTI FlexRay Configuration Blockset Reference 🕮)
RTIFLEXRAYCONFIG Synchronization Task (RTI FlexRay Configuration Blockset Reference 🕮)

How to Implement Error Handling

Objective

During execution of the communication code, several error conditions may occur. RTI FlexRay Configuration Blockset provides error hooks to handle most of them. You can use the error hooks to implement error handling.

Error Handling

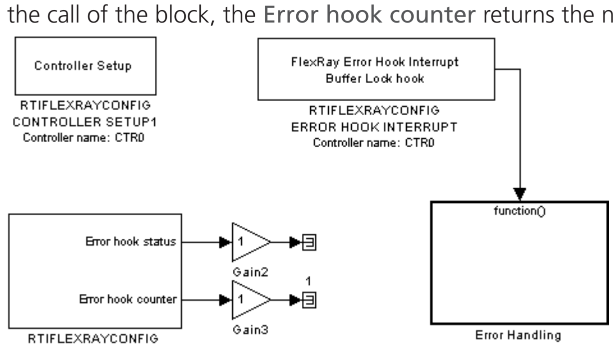
The RTI FlexRay Configuration blockset has two blocks for error handling:

- The RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT block allows you to use the error hooks to call a subsystem for error handling.
- The RTIFLEXRAYCONFIG ERROR HOOK STATUS block gives you status information on the FlexRay errors.

Method

To implement error handling

1. Open the RTI FlexRay Configuration Blockset.
2. Drag the RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT block to your Simulink model.
3. Open the dragged block and specify its parameters.
4. Drag a function-call subsystem to your Simulink model.
5. Connect the subsystem to the outport of the RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT block. This subsystem is executed when the specified error occurs.
6. Open the subsystem and design the error handling routine.
   In addition to the error handling, you can monitor the status of the error hooks.
7. Drag the RTIFLEXRAYCONFIG ERROR HOOK STATUS block to your Simulink model.
8. Open the RTIFLEXRAYCONFIG ERROR HOOK STATUS block, click the Options tab, and specify the sample time.
9. Connect the outports to other blocks, for example, Gain blocks so that values can be read during run time.
   The Error hook status outport returns the last error type that occurred since the call of the block, the Error hook counter returns the number of errors.



Result

Error handling is implemented.

Related topics

References

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

Advanced Deadline Violation Handling

Introduction

You can specify the behavior of a FlexRay application after a deadline violation (DLV) has occurred. By default, a FlexRay application is terminated if a deadline violation occurs. You can change the default behavior so that only a warning is issued.

Termination or warning

You can use the RTIFLEXRAYCONFIG DLV HANDLING block to specify whether the tasks are terminated or a warning is issued when a deadline violation occurs. You can define the behavior of the synchronization task or the application and communication task separately. Only the application task or communication task executed directly before the synchronization task is checked for a deadline violation.

Interrupt

If a deadline violation occurs in an application task or communication task, an interrupt can be issued using the RTIFLEXRAYCONFIG DLV INTERRUPT block. You can use the interrupt to trigger a function-call subsystem. If you use an RTIFLEXRAYCONFIG DLV INTERRUPT block, an RTIFLEXRAYCONFIG DLV HANDLING block is not necessary. The deadline violation handling is automatically set to "warning".

Increasing the WCET

The WCET adjustment factor can be used to increase the default worst-case execution time (WCET) of application tasks and Com tasks in the FlexRay Configuration Tool (see Tunable Properties of Tasks (FlexRay Configuration Tool Guide 📖)). If you use the multiple bus feature and a deadline violation occurs in a Com task, it is recommended to modify the WCET factor in all Com tasks and not only in the Com task which triggered the deadline violation. The timing of the tasks are more relaxed than, so that the competing task of the buses could be handled without deadline violations.

Related topics

Basics

References

RTIFLEXRAYCONFIG DLV HANDLING (RTI FlexRay Configuration Blockset Reference 📖)
RTIFLEXRAYCONFIG DLV INTERRUPT (RTI FlexRay Configuration Blockset Reference 📖)

How to Change the Standard Synchronization Task

Objective

In some special cases, the synchronization task which is automatically generated does not fulfill your requirements.

Restrictions

If you change the standard synchronization task and then trigger the configuration and update process, all the changes are lost. Only the values for simulation parameters and specific parameters for task handling (priorities, overrun behavior) remain unchanged.

Method

To change the standard synchronization task

Note

- Be careful when you change the synchronization task. Its start time is based on its worst-case execution time (WCET). Extending the synchronization task may cause a deadline violation.
- The existing synchronization task is deleted if the Exchange the Basic Configuration subsystem option is checked in the RTIFLEXRAYCONFIG UPDATE block.

1 Open the Synchronization State subsystem.

2 Perform your changes. For information on the RTI blocks, refer to Blocks for Configuration Generation (RTI FlexRay Configuration Blockset Reference 📖).

Result

The synchronization task was changed.

Related topics

References

RTIFLEXRAYCONFIG UPDATE (RTI FlexRay Configuration Blockset Reference 📖)

Real-Time Simulation with FlexRay Networks

Introduction

When the Simulink model is finished, you can build the real-time application, download the code to the real-time system, and start the simulation.

Where to go from here

Information in this section

Information in other sections

RTI Task Configuration Dialog - Task Groups (RTI and RTI-MP Implementation Reference 📖)
If your model contains blocks of the RTI FlexRay Configuration Blockset or of the RTI LIN MultiMessage Blockset, the RTI Task Configuration dialog is extended.

How to Specify Simulation Parameters

Objective

Before you can simulate the model, some parameters have to be specified.

Default priorities

The time-triggered tasks have the highest priorities. If you change the priority order, for example, if the priority of an interrupt-driven task is higher than the priority of a time-triggered task, the time characteristic may change. For information on how to change priorities, refer to How to Change the Task Priorities (RTI and RTI-MP Implementation Guide 📖).

Step size and timer task

The FlexRay model uses a fixed step size together with a fixed-step solver. The fixed step size determines the sample period of the timer task. In multiple timer task mode, there may be more than one timer task. All timer tasks must be sampled with a multiple of the specified fixed step size. Timer tasks are not time-triggered. They have a lower priority than the timetable tasks. By default, timer tasks are active when no timetable task is running.

Method

To specify the simulation parameters for a single-processor system

1. On the Simulation ribbon, click Prepare – Model settings.
   The Configuration Parameters dialog opens
2. In the Configuration Parameters dialog, select the Solver page.
3. In the Type drop-down list, select Fixed-step and an appropriate solver type.

Result

The model is set up for simulation.

Next steps

When the simulation parameters are specified, you can build the code. Refer to How to Build and Download Code on page 59.

Related topics

References

Solver Dialog (Model Configuration Parameters Dialogs) (RTI and RTI-MP Implementation Reference 🕮)

How to Build and Download Code

Objective

The build procedure is started in the same way as for any other single-processor model.

RTI Task configuration

The RTI Task Configuration dialog is extended for the model containing RTI FlexRay Configuration blocks, refer to RTI Task Configuration Dialog - Task Groups (RTI and RTI-MP Implementation Reference 🕮).

Building and downloading code

For general information, refer to Building and Downloading the Model (RTI and RTI-MP Implementation Guide 📖).

Tip

In mapping subsystems, virtual Simulink blocks such as Bus Creator blocks and Bus Selector blocks are used. Entries for these blocks are generated to the TRC file. To reduce the number of generated unnecessary TRC file entries and to reduce the time needed for the code generation process, you can suppress the generation of entries for the blocks. Refer to Code Generation Dialog (Model Configuration Parameters Dialogs) (RTI and RTI-MP Implementation Reference 📖).

Preconditions

The simulation parameters are specified.

Method

To build and download the code

1 On the C Code ribbon, select Generate Code - Build. If the C Code ribbon is hidden, go to the APPS ribbon and select Simulink Coder to display it.

   The build process starts. When you build the code for the first time, the RTI Task Configuration dialog opens.

2 In the RTI Task Configuration dialog, set the priorities of the tasks as required.

Result

The code is built and downloaded to the real-time system (according to your settings) for simulation.

Next steps

If you have used the standard ground state task, you have some options to control the simulation. Refer to Simulating with the Standard Synchronization Task on page 61.

Related topics

HowTos

Simulating with the Standard Synchronization Task

Introduction

If you use the standard synchronization task in your Simulink model, you have some options to control the simulation while it is running.

Switching the synchronization service

You can switch the synchronization service on or off during simulation. For this purpose you must modify the parameter of the RTIFLEXRAYCONFIG_SyncSwitch Constant block using ControlDesk.

Stopping and starting application tasks

You can stop and start time-triggered application tasks during simulation. For this purpose you must modify the parameter of the RTIFLEXRAYCONFIG_EnableTaskExecution Constant block using ControlDesk.

Application task and synchronization state

You can execute application tasks according to the synchronization state or independently of it. In the default configuration, application tasks are executed only if the host and the FlexRay network are synchronized (syncState = 1). You can execute an application task independently of the synchronization state. For this purpose you must set the parameter of the RTIFLEXRAYCONFIG_ExecuteTasksWithSyncOnly Constant block to 0 using ControlDesk. If no connection to a FlexRay network exists but application tasks need to be executed, you can execute the time-triggered application tasks independently of the synchronization state.

Setting the synchronization mode

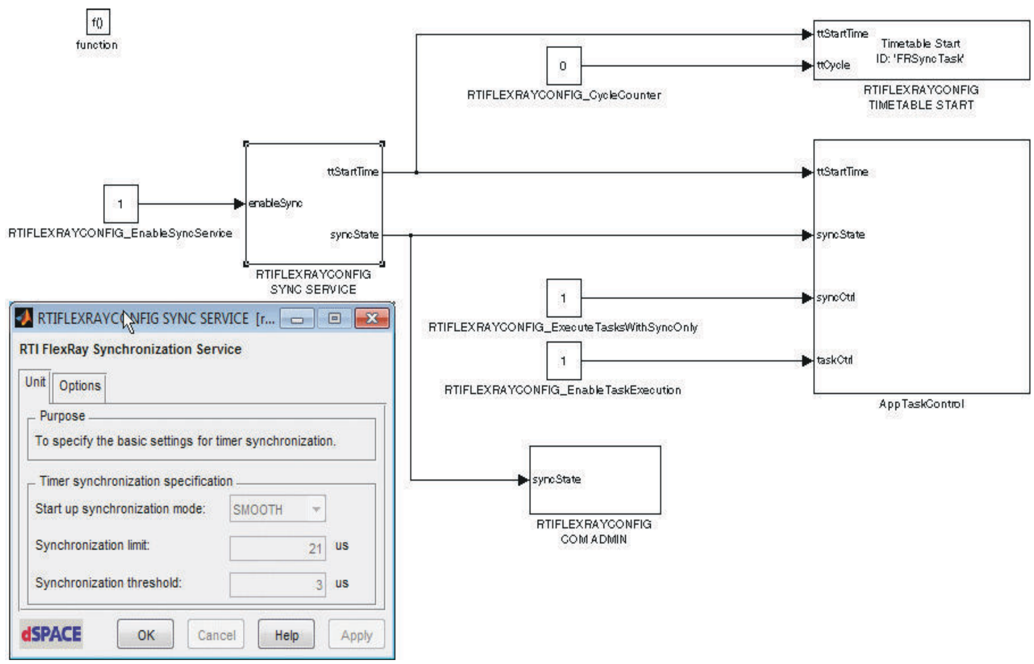
The time-triggered task execution of a FlexRay host can be hard- or smooth-synchronized. If you select smooth synchronization, the host is synchronized if the difference between the local and global times exceeds a specified value. If you select hard synchronization, the host is synchronized after the timetable has started.

Smooth synchronization The difference between the local and global time is compensated in steps. The maximum value of the steps is specified by the `Synchronization Limit` parameter. This avoids too-fast changing between the asynchronized and synchronized task execution. The disadvantage is that task synchronization takes several communication cycles.

Hard synchronization The difference between the local and global time is compensated in a few communication cycles. The time difference is not limited.

This may lead to an undefined period between the last asynchronized task execution and the first synchronized task execution.

The synchronization mode is selected in the dialog of the **RTIFLEXRAYCONFIG SYNC SERVICE** block, see the following illustration.



Related topics

References

RTIFLEXRAYCONFIG SYNC SERVICE (RTI FlexRay Configuration Blockset Reference 🕮)

Preventing a Deadline Violation

Deadline violation (DLV)

If the worst-case execution time of the application task or the communication task directly before the synchronization task code is exceeded, a deadline violation (DLV) occurs. This also occurs if the synchronization task starts and the application or communication task is still running. A deadline violation always leads to program termination if you do not use DLV Handling blocks (see Error Hook Blocks and Deadline Violation Handling (RTI FlexRay Configuration Blockset Reference 🕮)).

Preventing a DLV

To prevent a deadline violation, you can specify a new worst-case execution time (WCET) in the FlexRay Configuration Tool.

- For manually created tasks, you can specify larger values for the WCET.
- For automatic generated tasks, you can increase the **WCET Adjustment Factor**.

Related topics

Basics

Using the Generated TRC File of PDU-Based Modeling

Introduction

The generated TRC file contains variables for controlling the PDUs, their signals, and raw data. To visualize the variables, you can establish a data connection with instruments in ControlDesk. After you built and downloaded the real-time application, all signals from the TRC file are displayed in the **Variables** controlbar in ControlDesk.

Structure of the TRC file

The variables are structured in groups as follows ("<>" marks placeholders):

```
FlexRay
   ConfigId <Config Id>   (optional)
      Sending
         <ECU_Name>
            <Channel>
               <PDU_Name>
                  Signals
                     <Signal_1>
                     ...
                     <Signal_n>
                  Enable
                  Trigger
                  CRC Data
                  Update Bit
                  Raw Data
                  Transmission Mode
                  Minimum Delay Time
                  Contained PDU Send Status
                  Secure Onboard Communication
                  Global Time Synchronization
      Receiving
         <ECU_Name>
            <Channel>
               <PDU_Name>
                  Signals
                     <Signal_1>
                     ...
                     <Signal_n>
                  CRC Data
                  Update Bit
                  Status
                  Raw Data
                  Update Contained PDU
                  Secure Onboard Communication
                  Global Time Synchronization
      Monitoring
         <ECU_Name>
            <Channel>
               <PDU_Name>
                  Signals
                     <Signal_1>
                     ...
                     <Signal_n>
                  CRC Data
                  Update Bit
                  Status
                  Raw Data
                  Update Contained PDU
                  Secure Onboard Communication
                  Global Time Synchronization
```

The group names are derived from the short names of the elements which are specified in the database (FIBEX file or AUTOSAR system description file). In some cases, the group names differ, see TRC File with Several PDUs Having the Same Short Name on page 81.

The ConfigId <Config Id> group is optional (<ConfigId> is '-', 1, 2, or 3). The group is generated if you use the multiple bus option. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System on page 37.

The variables that are included in the groups are described below.

Sending group

The Sending group contains all the variables to control the TX PDUs and their signals. PDUs that are included in the TRC file must be selected in the FlexRay Configuration Tool, see How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File (FlexRay Configuration Tool Guide 🕮).

<Signal_x> group A <Signal_x> group contains the variables to control one signal of the PDU.

Variable	Description
Value	Value for the signal to be sent. This variable is optional. To generate it, you must have selected the **Signal Value Access** feature.
Dynamic Value	Value to be sent during dynamic signal manipulation[1]. The value must be specified as a coded data type. This variable is optional. To generate it, you must have selected the **Signal Value Access** feature.
Physical Value	Physical value for the signal to be sent. This variable is optional. To generate it, you must have selected the **Signal Value Access** feature and must have enabled the **Use physical data type** option of the Trace File Generator, and conversion must be possible.
Dynamic Physical Value	Value to be sent during dynamic signal manipulation[1], specified as a physical data type. This variable is optional. To generate it, you must have selected the **Signal Value Access** feature and must have enabled the **Use physical data type** of the Trace File Generator, and conversion must be possible.
Coded Physical Value Source Switch	To specify whether the coded or physical signal value is to be sent. The following values are possible: ▪ 0: Coded signal value is sent: i.e., a text table value is sent via the `Value` variable. ▪ 1: Physical signal value is sent: i.e., the `Physical Value` variable is used to sent the signal value.. This variable is optional. It is available for signals with the SCALE_LINEAR_TEXTTABLE computation method only. To generate it, the following conditions must be met: ▪ You must have selected the **Signal Value Access** feature. ▪ You must have enabled the **Use physical data type** and **Activate support for text tables** options of the Trace File Generator. ▪ The **Physical data type is usable** property must be set to True. ▪ The **Physical data type conversion layer** option must be set to COMMUNICATION on the **General** page of the **General Properties** dialog.
Validity	To specify the validity status for the signal to be sent over the bus: ▪ 0: NOT VALID ▪ 1: VALID ▪ 2: ERROR ▪ 4: NOT AVAILABLE

Variable	Description
	▪ 8: NOT DEFINED ▪ 16: OTHER The variable is essential only if data to be sent is specified in ControlDesk (`Source Switch = 1`). This variable is optional. To generate it, you must have selected the Signal Validity Control feature. This is possible only if a non-valid value (coded and/or physical) exists for the respective signal: i.e., if at least one value with validity ≠ 'VALID' is defined in the database file for the signal.
Dynamic Validity	To specify the validity status for the signal to be sent during dynamic signal manipulation[1]: ▪ 0: NOT VALID ▪ 1: VALID ▪ 2: ERROR ▪ 4: NOT AVAILABLE ▪ 8: NOT DEFINED ▪ 16: OTHER This variable is optional. To generate it, you must have selected the Signal Validity Control feature. This is possible only if a value (coded and/or physical) with validity status ≠ 'VALID' exists for the respective signal.
Countdown Value	Time span for dynamic signal manipulation[1]. It specifies how often the dynamic values are sent.
Source Switch	To switch the source for the send signal: ▪ 0: Data of the Simulink model is sent ▪ 1: Data which is specified via the TRC file is sent ▪ 2: The current value of the alive counter is sent ▪ 12: The alive counter is stopped for the time span specified for dynamic signal manipulation (`Countdown Value`). The last value of the alive counter is sent. When `Countdown Value` reaches 0, the source is switched back to the last `Source Switch` value.

AliveCounter	The `AliveCounter` group is added if you have specified a signal as an alive counter in the FlexRay Configuration Tool and activated the Signal Alive Counter Control feature in the Element Selection dialog. The group contains the following variables:
	▪ `Value`: Current value of the alive counter.
	▪ `Runtime Behavior`: To specify the behavior of the alive counter at run time.
	  ▪ 0: The alive counter runs independently of the selected source for the signal.
	Note
	There is one exception: If `Source Switch` is 12, the alive counter stops independently of the `Runtime Behavior`.
	  ▪ 1: The alive counter runs only if its value is really sent. If the source for the signal is switched to another source than the alive counter, for example, to SL (a value from the Simulink model), the alive counter stops.
	  ▪ 2: The alive counter stops at the current value.
	▪ `Offset`: Offset value that is added to the alive counter during run time.

Variable	Description
Tx inspect	The `Tx inspect` group is added if the **Physical data type conversion layer** option is set to **COMMUNICATION** on the **General** page of the **General Properties** dialog in the FlexRay Configuration Tool. The group can contain the following variables: ▪ `Tx Value` or `Tx Physical Value` The value shows the transmitted coded or physical value of a signal via TRC file. A signal is sent as physical data type if it has a usable physical data type and the **Use physical data type** option is set to True. The option is set for the Trace File Generator on the **Generators** page of the **General Properties** dialog in the FlexRay Configuration Tool. ▪ `Available SL Value` or `Available SL Physical Value` The value shows the coded or physical value that is set in the Simulink model for the corresponding signal. The **Port data type** property of the signal specifies the data type which is used in the Simulink model.

[1] For details on dynamic signal manipulation, refer to Dynamic Signal Manipulation on page 84.

Enable group The variables of the `Enable` group control the sending of cyclic PDUs.

The following variables are available for dynamic PDUs when you have selected them for the **Frame Dynamic Control** feature:

Variable	Description
Tx Enable	To enable the sending of dynamic cyclic PDUs.
Multiplexer Switch Code	To select the switch code of a sub-PDU. It is available only for dynamic PDUs that contain sub-PDUs. You can only enter valid values which are documented in the variable's description. Switch codes are available only for sub-PDUs that have cyclic timing.
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is sent. ▪ 1: Data which is specified via the TRC file is sent.

The following variables are available for static PDUs if you have selected them for the **Frame Static Control** feature:

Variable	Description
SW Enable	To enable the sending of static PDUs. It is only available for static PDUs. If a static PDU is *enabled* via software (SWEnable = 1), payload data or null frames can be sent. The data which is sent depends on the configuration of SWEnable (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 frames is sent if the LPDU to be sent contains only PDUs that have not been updated and that do not contain a PDU update bit.

<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td></td><td>
Payload data is sent for all other LPDUs.
Other settings:
Payload data is sent.
If a static PDU is disabled via software (SWEnable = 0), old data or null frames can be sent. The data which is sent depends on the database version, on the configuration of SWEnable (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 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 SW Enable was set to 0.
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>
<tr><td>Multiplexer Switch Code</td><td>To select the switch code of a sub-PDU. It is available only for static PDUs that contain sub-PDUs. You can only enter valid values which are documented in the variable's description.</td></tr>
<tr><td>Source Switch</td><td>To switch the source for the TX PDU:
0: Data of the Simulink model is sent.
1: Data which is specified via the TRC file is sent.</td></tr>
</table>

Trigger group The variables of the `Trigger` group control the sending of dynamic PDUs to. The following variables are available when you have selected a PDU for the Frame Dynamic Control feature:

Variable	Description
Tx Trigger	To trigger the dynamic PDU.
Multiplexer Switch Code	To select the switch code of a sub-PDU. It is available only for dynamic PDUs that contain sub-PDUs. You can only enter valid values which are documented in the variable's description. There are different switch codes for event-triggered and cyclic sub-PDUs.

Variable	Description
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is sent. ▪ 1: Data which is specified via the TRC file is sent.

CRC Data group The variables of the `CRC Data` group control the CRC calculation of TX PDUs. The following variables are available if you have selected a PDU for the Frame CRC Control feature:

Variable	Description
CRC Enable	To enable or disable the CRC calculation of the PDU: ▪ 0: CRC calculation is disabled. ▪ 1: CRC calculation is enabled.
Dynamic CRC Enable	To enable or disable the CRC calculation of the PDU during dynamic checksum calculation manipulation.[1)]
Type	To select the algorithm that is used for CRC calculation. The CRC algorithms which are specified in the FlexRay Configuration Tool are documented in the variable's description. For details, refer to Using User-Defined Checksum Algorithms (FlexRay Configuration Tool Guide 📖).
Dynamic Type	To select the algorithm that is used for CRC calculation during dynamic checksum calculation manipulation.[1)]
Countdown Value	Number of transmission cycles for which dynamic checksum calculation manipulation is performed.[1)]
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is used for checksum calculation manipulation. ▪ 1: Data which is specified via the TRC file is used for checksum calculation manipulation. ▪ 8: For the specified number of transmission cycles (`Countdown Value`), the dynamic CRC manipulation data specified via the TRC file is used for checksum calculation manipulation. When `Countdown Value` reaches 0, the source is automatically switched back to the switch state that was set before.

[1)] For further information, refer to Dynamic Checksum Calculation Manipulation on page 86.

Update Bit group The variables of the `Update Bit` group manipulate the update bit of a TX PDU. The following variables are available if you have selected a PDU for the Frame Update Bit Control feature:

Variable	Description
Value	Value of the update bit.
Dynamic Value	Update bit value to be sent during dynamic update bit manipulation.[1)]
Update Bit Enable	To enable or disable the automatically update bit calculation in the Com task when the `Source Switch` variable is 1: ▪ 0: Automatic calculation is enabled. This overwrites the value specified by the `Value` variable. ▪ 1: Automatic calculation is disabled. The value specified by the `Value` variable is used.
Dynamic Update Bit Enable	To enable or disable the automatic update bit calculation during dynamic update bit manipulation.[1)]
Countdown Value	Number of transmission cycles for which dynamic update bit manipulation is performed.[1)]

Variable	Description
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is used for update bit manipulation. ▪ 1: Data which is specified via the TRC file (`Value` variable) is used for update bit manipulation. ▪ 8: For the specified number of transmission cycles (`Countdown Value`), the dynamic update bit manipulation data specified via the TRC file is used for update bit manipulation. When `Countdown Value` reaches 0, the source is automatically switched back to the switch state that was set before.

[1] For further information, refer to Dynamic Update Bit Manipulation on page 85.

Raw Data group The variables of the `Raw Data` group manipulate raw data of a PDU. The following variables are available if you have selected a PDU for the Frame Raw Data Access feature:

Variable	Description
Tx Bytes	Raw data to be sent.
Raw Data Enable	To select raw data which is sent: ▪ 0: Raw data specified in the Simulink model is sent. If it is not available, signal data specified using the TRC file (if the `Source Switch` variable is not 0) or signal data specified in the Simulink model (if the `Source Switch` variable is 0) is sent. If signal data is not available, the data sent last or the initial values are sent. ▪ 1: Raw data specified by the `Tx Bytes` variable is sent. The variable is effective only if the `Source Switch` variable is not 0.
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is sent. ▪ 1: Data which is specified via the TRC file (`Tx Bytes` variable) is sent.

Transmission Mode group The variables of the `Transmission Mode` group control the transmission mode of a PDU. The following variables are available if you have selected a PDU for the Frame Dynamic Control or Frame Static Control feature:

Variable	Description
Value	Transmission mode to be used: ▪ 0: Transmission mode 'False' ▪ 1: Transmission mode 'True' ▪ 98: Transmission mode 'LPDU timing triggered' ▪ 99: Transmission mode 'User-Defined' For more information on PDU transmission modes, refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 🕮).
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is sent. ▪ 1: Data which is specified via the TRC file (`Value` variable) is sent.

Minimum Delay Time group The variables of the `Minimum Delay Time` group allow you to control the minimum delay time feature of TX PDUs. The following variables are available if you have selected a PDU for the Frame Minimum Delay Time Control feature:

Variable	Description
Enable MDT	To enable or disable minimum delay time support for the PDU: ▪ 0: The minimum delay time feature is disabled. There is no delay time between successive transmissions of the PDU. ▪ 1: The minimum delay time feature is enabled. The minimum delay time specified for the PDU in the AUTOSAR system description file must be complied with between two transmissions of the PDU. For more information on minimum delay time support, refer to How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 📖).
Source Switch	To switch the source for the TX PDU: ▪ 0: Data of the Simulink model is used to enable or disable the minimum delay time feature. ▪ 1: Data which is specified via the TRC file (`Enable MDT` variable) is used to enable or disable the minimum delay time feature.
MDT Triggering Status	Information on the last captured trigger and on whether and how the trigger was considered for triggering (if the minimum delay time feature is enabled for the PDU). ▪ 0: No recent trigger. ▪ 1: Accepted event trigger. The event-based triggering occurred outside of an active minimum delay time. ▪ 2: Accepted cyclic trigger. The cyclic triggering occurred outside of an active minimum delay time. ▪ 3: Accepted event trigger, discarded cyclic trigger. The simultaneous event-based and cyclic triggering occurred outside of an active minimum delay time. ▪ 4: Discarded event trigger. The event-based triggering occurred during the active minimum delay time. ▪ 8: Discarded cyclic trigger. The cyclic triggering occurred during the active minimum delay time. ▪ 12: Discarded event trigger, discarded cyclic trigger. The simultaneous event-based and cyclic triggering occurred during the active minimum delay time.
Time	Remaining minimum delay time.

Contained PDU Send Status The variables of the `Contained PDU Send Status` group control the sending of contained IPDUs. The following variables are available if you have selected a contained IPDU for the Frame Container Control feature:

Variable	Description
Contained PDU Send Status	Send status of the contained IPDU and the associated container: ▪ 0: Contained IPDU was not triggered. ▪ 1: Contained IPDU was triggered but not added to the container IPDU yet. ▪ 2: Contained IPDU was triggered and packed into the container IPDU. ▪ 4: Container IPDU (and thus the contained IPDU) was sent.
Container PDU Trigger Status	Trigger status of the container IPDU: ▪ 0: Container IPDU was not triggered. ▪ 1: Container IPDU was triggered because the threshold was exceeded. ▪ 2: Container IPDU was triggered because it was full. ▪ 4: Container IPDU was triggered by the first added contained IPDU (`FirstContainedTrigger`). ▪ 8: Container IPDU was triggered because this contained IPDU was added to the container IPDU (`TRIGGER_ALWAYS`) ▪ 16: Container IPDU was triggered by the container IPDU's timeout.

Variable	Description
	For container IPDUs with static container layout, only 0 and 8 are valid trigger status values. This is because a static container IPDU can only be triggered by adding a contained IPDU (`TRIGGER_ALWAYS`).

For more information on container IPDUs and contained IPDUs, refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 📖).

Secure Onboard Communication group The variables of the `Secure Onboard Communication` group control the sending of secured IPDUs. The following variables are available if you have selected an authentic IPDU for the Frame Authentication Control feature:

Variable	Description

Authenticator Value	The `Authenticator Value` group contains the following groups and variables:
	▪ The `Control` group contains the following variables:
	  ▪ `Type`: To invalidate the authenticator of a secured IPDU. The meaning of this value is user-code-dependent. Proposed values are:
	    ▪ 0: Do not invalidate the authenticator
	    ▪ 1: Invalidate the authenticator
	  This variable corresponds to the `AuthenticationType` user code variable.

Tx Inspect	The `Tx Inspect` group contains the following variables:
	▪ `Authenticator Value`: The value shows the actually transmitted (truncated) authenticator value.
	  This variable does not correspond to any user code variable.
	▪ `Freshness Value`: The value shows the actually transmitted (truncated) freshness value.
	  This variable does not correspond to any user code variable.

For more information on secure onboard communication, refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 📖).

Global Time Synchronization group The variables of the `Global Time Synchronization` group let you control the sending of time synchronization messages. The following variables are available if you have selected frames for the Frame Global Time Sync Access feature:

Variable	Description

CRC	The `CRC` group contains the following variables:
	▪ `Value`: CRC value for the time synchronization message to be sent.
	▪ `Dynamic Value`: CRC value to be sent during dynamic global time synchronization manipulation.[1)]
	▪ `Countdown Value`: Number of transmission cycles for which dynamic CRC manipulation is performed.[1)]
	▪ `Source Switch`: To switch the source for the TX PDU:
	  ▪ 0: Data of a fixed E2E_P02 algorithm is sent.
	  ▪ 1: Data which is specified via the TRC file is sent.

<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td></td><td>8: For the specified number of transmission cycles (<code>Countdown Value</code>), the dynamic CRC manipulation data specified via the TRC file is used for GTS manipulation. When <code>Countdown Value</code> reaches 0, the source is automatically switched back to the switch state that was set before</td></tr>
<tr><td>E2E Sequence Counter</td><td>The <code>E2E Sequence Counter</code> group contains the following variables:
<code>Value</code>: Value of the end-to-end protection sequence counter (SC).
<code>Dynamic Value</code>: End-to-end protection sequence counter (SC) value to be sent during dynamic global time synchronization manipulation.[1)]
<code>Countdown Value</code>: Number of transmission cycles for which dynamic E2E sequence counter manipulation is performed.[1)]
<code>Source Switch</code>: To switch the source for the TX PDU:
0: Data of a fixed E2E_P02 algorithm is sent.
1: Data specified via the TRC file is sent.
8: For the specified number of transmission cycles (<code>Countdown Value</code>), the dynamic E2E sequence counter manipulation data specified via the TRC file is used for GTS manipulation. When <code>Countdown Value</code> reaches 0, the source is automatically switched back to the switch state that was set before</td></tr>
<tr><td>Time Gateway Synchronization Status</td><td>The <code>Time Gateway Synchronization Status</code> group contains the following variables:
<code>Value</code>: Value of the SYNC_TO_GATEWAY (SGW) bit from the Time Base status of the time base manager instance.
<code>Dynamic Value</code>: Value of the SYNC_TO_GATEWAY (SGW) bit to be sent during dynamic global time synchronization manipulation.[1)]
<code>Countdown Value</code>: Number of transmission cycles for which dynamic time gateway synchronization manipulation is performed.[1)]
<code>Source Switch</code>: To switch the source for the TX PDU:
0: Data of the time base manager is sent.
1: Data specified via the TRC file is sent.
8: For the specified number of transmission cycles (<code>Countdown Value</code>), the manipulation data of the time gateway synchronization status specified via the TRC file is used for GTS manipulation. When <code>Countdown Value</code> reaches 0, the source is automatically switched back to the switch state that was set before</td></tr>
<tr><td>User Byte 0</td><td>The <code>User Byte 0</code> group contains the following variables:
<code>Value</code>: Value of the User Byte 0.
<code>Dynamic Value</code>: User Byte 0 value to be sent during dynamic global time synchronization manipulation.[1)]
<code>Countdown Value</code>: Number of transmission cycles for which dynamic user byte 0 manipulation is performed.[1)]
<code>Source Switch</code>: To switch the source for the TX PDU:
0: Data of the time base manager is sent.
1: Data specified via the TRC file is sent.
8: For the specified number of transmission cycles (<code>Countdown Value</code>), the dynamic user byte 0 manipulation data specified via the TRC file is used. When <code>Countdown Value</code> reaches 0, the source is automatically switched back to the switch state that was set before</td></tr>
<tr><td>User Byte 1</td><td>The <code>User Byte 1</code> group contains the following variables:
<code>Value</code>: Value of the User Byte 1.
<code>Dynamic Value</code>: User Byte 1 value to be sent during dynamic global time synchronization manipulation.[1)]</td></tr>
</table>

Variable	Description

	▪ `Countdown Value`: Number of transmission cycles for which dynamic user byte 1 manipulation is performed.[1]
	▪ `Source Switch`: To switch the source for the TX PDU:
	  ▪ 0: Data of the time base manager is sent.
	  ▪ 1: Data specified via the TRC file is sent.
	  ▪ 8: For the specified number of transmission cycles (`Countdown Value`), the dynamic user byte 1 manipulation data specified via the TRC file is used. When `Countdown Value` reaches 0, the source is automatically switched back to the switch state that was set before

Total Time	The `Total Time` group contains the following variables:
	▪ `Value`: Value of the total time.
	▪ `Dynamic Value`: Total time value to be sent during dynamic global time synchronization manipulation.[1]
	▪ `Countdown Value`: Number of transmission cycles for which dynamic total time manipulation is performed.[1]
	▪ `Source Switch`: To switch the source for the TX PDU:
	  ▪ 0: Data of the time base manager is sent.
	  ▪ 1: Data specified via the TRC file is sent.
	  ▪ 8: For the specified number of transmission cycles (`Countdown Value`), the dynamic total time manipulation data specified via the TRC file is used for GTS manipulation. When `Countdown Value` reaches 0, the source is automatically switched back to the switch state that was set before

1) For further information, refer to Dynamic Manipulation of GTS Communication on page 88.

Sent Values The `Sent Values` group contains the following variables showing the actually transmitted GTS-relevant values:

Variable	Description
CRC	Actually transmitted CRC value for the time synchronization message.
Time Domain Id	Identifier of the global time domain that is transferred in the time synchronization message.
E2E Sequence Counter	Actually transmitted value of the end-to-end protection sequence counter (SC).
FlexRay Cycle Counter	Value of the FlexRay cycle counter (FCNT) at the time the global time synchronization PDU is built.
Time Gateway Synchronization Status	Value of the SYNC_TO_GATEWAY (SGW) bit from the Time Base status of the time base manager instance.
User Byte 0	Actually transmitted value of the User Byte 0.
User Byte 1	Actually transmitted value of the User Byte 1.
Seconds	Displays the seconds that are transferred via the FlexRay bus in the time synchronization message.
Nanoseconds	Displays the nanoseconds that are transferred via the FlexRay bus in the time synchronization message.
Total Time	Actually transmitted value of the total time.

For more information on global time synchronization, refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 🕮).

Receiving or monitoring group

The `Receiving` and `Monitoring` groups contain all the variables to receive or monitor RX PDUs. PDUs that are included in the TRC file must be selected in the FlexRay Configuration Tool, see How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File (FlexRay Configuration Tool Guide 📖).

<Signal_x> group A `<Signal_x>` group contains the variables to read one signal of the PDU.

Variable	Description
Value	Value of the received or monitored signal. This variable is optional. To generate it, you must have selected the Signal Value Access feature.
Physical Value	Physical value of the received or monitored signal. This variable is optional. To generate it, you must have selected the Signal Value Access feature and must have enabled the Use physical data type of the TRC File Generator, and conversion must be possible.
Text Table Status	Indicates whether the received value is a text table value. ▪ 0: The received value is not a text table value. The received value can be read via the `Physical Value` variable. ▪ 1: The received value is a text table value. The received value can be read via the `Value` variable. This variable is optional. It is available for signals with the SCALE_LINEAR_TEXTTABLE computation method only. To generate it, the following conditions must be met: ▪ You must have selected the Signal Value Access feature. ▪ You must have enabled the Use physical data type and Activate support for text tables options of the Trace File Generator. ▪ The Physical data type is usable property must be set to True.
Status	Status information of the received or monitored signal: ▪ 0: No error, signal is valid (signal validity status = 'VALID') ▪ 1: Access error ▪ 2: Signal is not received ▪ 4: Signal is not valid (signal validity status ≠ 'VALID') ▪ 8: CRC is incorrect It is available only if the signal is selected for the Signal RX Status Access feature in the FlexRay Configuration Tool (see How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File (FlexRay Configuration Tool Guide 📖))
Update Bit Value	Signal update bit value of the received or monitored signal. For further information, refer to Sending and Receiving Signal- and Signal-Group-Specific Update Bits on page 36. This variable is optional. To generate it, you must have selected the Signal Value Access feature.

CRC Data group The variables of the `CRC Data` group control the CRC calculation of RX PDUs. The following variables are available if you have selected a PDU for the Frame CRC Control feature:

Variable	Description
CRC Enable	To enable or disable the CRC calculation of the PDU: ▪ 0: CRC calculation is disabled. ▪ 1: CRC calculation is enabled.

Variable	Description
Dynamic CRC Enable	To enable or disable the CRC calculation of the PDU during dynamic checksum calculation manipulation.[1)]
Type	To select the algorithm that is used for CRC calculation. The CRC algorithms which are specified in the FlexRay Configuration Tool are documented in the variable's description. For details, refer to Using User-Defined Checksum Algorithms (FlexRay Configuration Tool Guide 📖).
Dynamic Type	To select the algorithm that is used for CRC calculation during dynamic checksum calculation manipulation.[1)]
Countdown Value	Number of transmission cycles for which dynamic checksum calculation manipulation is performed.[1)]
Status	To get the status of the CRC calculation. ▪ 0: CRC calculation OK. ▪ 1: CRC calculation not OK. ▪ 2: CRC not calculated.
Source Switch	To switch the source for the RX PDU: ▪ 0: Data of the Simulink model is used. ▪ 1: Data which is specified via the TRC file is used. ▪ 8: Dynamic CRC manipulation data specified via the TRC file is used for the specified number of transmission cycles (`Countdown Value`). When `Countdown Value` reaches 0, the source is automatically switched back to the switch state that was set before.

1) For further information, refer to Dynamic Checksum Calculation Manipulation on page 86.

Update Bit group The variables of the `Update Bit` group are for reading the values of the PDU's update bit and the **Ignore Update Bit** property. The following variables are available if you have selected a PDU for the **Frame Update Bit Control** feature:

Variable	Description
Value	Value of the received update bit.
Ignore Update Bit	Value of the **Ignore Update Bit** property which is specified in the FlexRay Configuration Tool.

Status group The variables of the `Status` group are for reading status information of an RX PDU. The following variables are available if you have selected a PDU for the **Frame Receive Status Access** feature:

Variable	Description
Error	Indicates various statuses. Each status has its own bit, so parallel statuses are possible. In that case the error values are added. For information on the values, refer to RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 📖).
Data Receive Counter	Counts the number of received FlexRay frames. The counter also counts null frames.

Variable	Description
Nullframe	Indicates whether the received frame was a null frame (only for static PDUs): ▪ 0: The last received frame was not a null frame. ▪ 1: The last received frame was a null frame.
Multiplexer Switch Code	Indicates the switch code that was used for a send PDU (only for PDUs containing sub-PDUs).

Raw Data group The variables of the `Raw Data` group fetch information on the raw data and the corresponding payload length. The following variables are available if you have selected a PDU for the Frame Raw Data Access feature:

Variable	Description
Rx Bytes	Raw data of the RX PDU.
Payload Length Value	Payload length of the RX raw data. It is only available if the Payload length readable property of the PDU is enabled.

Update Contained PDU The variables of the `Update Contained PDU` group are for reading status information on whether the contained IPDU was contained in the container IPDU. The following variables are available if you have selected a contained IPDU for the Frame Container Control feature:

Variable	Description
Value	Indicates whether the contained IPDU was contained in the container IPDU: ▪ 0: Contained IPDU was not contained in the container IPDU. ▪ 1: Contained IPDU was contained in the container IPDU.

For more information on container IPDUs and contained IPDUs, refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 📖).

Secure Onboard Communication group The variables of the `Secure Onboard Communication` group are used to get information on received secured IPDUs. The following variables are available if you have selected an authentic IPDU for the Frame Authentication Control feature:

Variable	Description
Authenticator Value	The `Authenticator Value` group contains the following groups and variables: ▪ `Value`: The value shows the actually received (truncated) authenticator value. This variable does not correspond to any user code variable. ▪ The `Control` group contains the following variables: ▪ `Enable`: The meaning of this value is user-code-dependent. Proposed values are: ▪ 0: Verification is disabled. ▪ 1: Verification is enabled. This variable corresponds to the `EnableVerification` user code variable. ▪ `Status`: The meaning of this value is user-code-dependent. Proposed values are: ▪ 0: Verification successful. ▪ 1: Verification not successful. ▪ 2: Verification not successful due to wrong freshness value. ▪ 0x3F: Verification not executed. This variable corresponds to the `VerificationResult` user code variable.

Variable	Description
Freshness Value	The `Freshness Value` group contains the following groups and variables: ▪ `Value`: The value shows the actually received (truncated) freshness value. This variable does not correspond to any user code variable. ▪ The `Control` group contains the following variables: ▪ `Calculated Value`: The variable contains the calculated freshness value, i.e., the full value reconstructed in the user code. This variable corresponds to the `CalculatedFreshnessValue` user code variable.

For more information on secure onboard communication, refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 🕮).

Global Time Synchronization group The variables of the `Global Time Synchronization` group provide information on a received global time synchronization PDU. The following variables are available if you have selected global time synchronization messages for the Frame Global Time Sync Access feature:

Variable	Description
CRC	CRC value that was transferred with the global time synchronization message via the FlexRay bus.
E2E Protection Status	Result of the CRC calculation, and, if applicable, an error state when the E2E profile is calculated: ▪ 0x00: E2E_P02STATUS_OK and E2E_E_OK OK: New data has been received. CRC is correct. No data has been lost since the last correct data reception. ▪ 0x01: E2E_P02STATUS_NONEWDATA Error: Check function has been invoked, but no new data is available since the last call. As a result, no E2E checks have been consequently executed. ▪ 0x02: E2E_P02STATUS_WRONGCRC Error: Data has been received, but the CRC is incorrect. ▪ 0x03: E2E_P02STATUS_SYNC NOT VALID: New data has been received after detection of an unexpected behavior of the counter. Data has a correct CRC and a counter within the expected range with respect to the

Variable	Description
	most recent data received, but the determined continuity check for the counter is not finalized yet. ▪ 0x04: E2E_P02STATUS_INITIAL Initial: New data has been received. CRC is correct, but this is the first data since the receiver's initialization or reinitialization, so the counter cannot be verified yet. ▪ 0x08: E2E_P02STATUS_REPEATED Error: New data has been received. CRC is correct, but the counter is identical to the most recent data received with Status _INITIAL, _OK, or _OKSOMELOST. ▪ 0x20: E2E_P02STATUS_OKSOMELOST OK: New data has been received. CRC is correct. Some data in the sequence has probably been lost since the last correct/initial reception, but this is within the configured tolerance range. ▪ 0x40: E2E_P02STATUS_WRONGSEQUENCE Error: New data has been received. CRC is correct, but too many data in the sequence has probably been lost since the last correct/initial reception. ▪ 0x13: E2E_E_INPUTERR_NULL At least one pointer parameter is a NULL pointer. ▪ 0x17: E2E_E_INPUTERR_WRONG At least one input parameter is erroneous, e.g., out of range. ▪ 0x19: E2E_E_INTERR An internal library error has occurred. ▪ 0x1A: E2E_E_WRONGSTATE Function executed in wrong state.
Time Domain Id	Identifier of the global time domain that is transferred in the global time synchronization message.
E2E Sequence Counter	End-to-end protection sequence counter that is transferred in the global time synchronization message.
FlexRay Cycle Counter	FlexRay cycle counter (FCNT) that was current at the time when the global time synchronization PDU was built.
Time Gateway Synchronization Status	Displays the value of the SYNC_TO_GATEWAY (SGW) bit from the Time Base status of the time base manager instance: ▪ 0: Sync to gateway. ▪ 1: Sync to time subdomain.
User Byte 0	User byte 0 that is transferred in the global time synchronization message.
User Byte 1	User byte 1 that is transferred in the global time synchronization message.
Seconds	Value for seconds that is transferred via the FlexRay bus in the global time synchronization message. Because the FlexRay bus is used to always transfer the time of the next FlexRay Cycle 0 start, the second ratio can deviate from the integer part of `Total Time`.
Nanoseconds	Value for nanoseconds that is transferred via the FlexRay bus in the global time synchronization message. Because the FlexRay bus is used to always transfer the time of the next FlexRay Cycle 0 start, the nanosecond ratio almost always deviates from the decimal places of `Total Time`.
Total Time	Time value that is transferred via the FlexRay bus. This is the time that was read when the global time synchronization message was coded.

<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td>Time Base Status</td><td>Time base status information of the global time synchronization PDU (name, bit position, value):

TIMEOUT, Bit 0 (LSB),
0: No synchronization timeout
1: Synchronization timeout. The time base has not been synchronized for a longer period than specified by the Loss Timeout, if a time slave is connected to the time base.
Reserved, Bit 1, 0
SYNC_TO_GATEWAY, Bit 2,
0: Time base and global time master are synchronous.
1: The forwarding of the global time is interrupted and the time base is synchronized with an ECU that is located on a sublevel of the global time master.
GLOBAL TIME_BASE, Bit 3,
0: The time base has never been synchronized with the global time master and is executed based on a local time.
1: The time base has been synchronized with the global time master at least once since the start.
TIMELEAP_FUTURE, Bit 4,
0: The time did not leap further into the future than specified in the Time leap future parameter.
1: The time has leapt further into the future than specified by the Time leap future threshold parameter and has not been synchronized properly in the time leap future/past threshold intervals as specified in the time leap healing counter.
TIMELEAP_PAST, Bit 5,
0: The time did not leap further into the past than specified in the Time leap past parameter.
1: The time has leapt further into the past than specified by the Time leap past threshold parameter and has not been synchronized properly in the time leap future/past threshold intervals as specified in the Time leap healing counter parameter.
Reserved, Bit 6-31, 0
</td></tr>
<tr><td>Status</td><td>Status information on the global time synchronization PDU (name, bit position, description):

MSG_TYPE_UNSUPPORTED, Bit 0, The first eight bits of the messages do not have the value 0x20, which is required for GTS messages.
Reserved, Bit 1, –
Reserved, Bit 2, –
MSG_TD_WRONG, Bit 3, The bits 3-0 of byte 2 contain an unexpected Time Domain ID.
MSG_NANOSECONDS_INVALID, Bit 4, The value for nanoseconds in the bytes 12-15 is larger than 1,000,000,000.
Reserved, Bit 5, –
E2E_PROTECTION_API_ERROR, Bit 6, The value for E2E_Protection does not match E2E_P02STATUS_OK.
Reserved, Bit 7, –
Reserved, Bit 8, –
TBM_UPDATED_BUT_SC_NOT_OK, Bit 9, The value of the E2E_Protection does not match the E2E_P02STATUS_OK, but the time base was synchronized because the return value of the E2E_Protection was E2E_P02STATUS_OKSOMELOST or E2E_P02STATUS_INITIAL, for example.
TBM_UPDATED_BUT_CRC_NOT_OK, Bit 10, The CRC protection was negative but the time base was still synchronized because CRC-IGNORED is specified in the database for the CRC, for example.
</td></tr>
</table>

Variable	Description
	▪ TBM_UPDATED_ALL_OK, Bit 11, The time base was synchronized and both the CRC and the sequence counter check were completed successfully.

For more information on global time synchronization, refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 📖).

Related topics

Basics

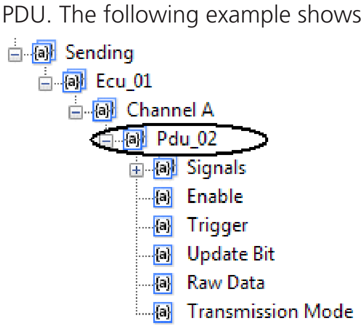
TRC File with Several PDUs Having the Same Short Name

Introduction

If several PDUs are sent or received by the same ECU, the structure of the TRC file is modified to identify individual PDU instances. The group name of the PDUs is extended, as shown in the examples below.

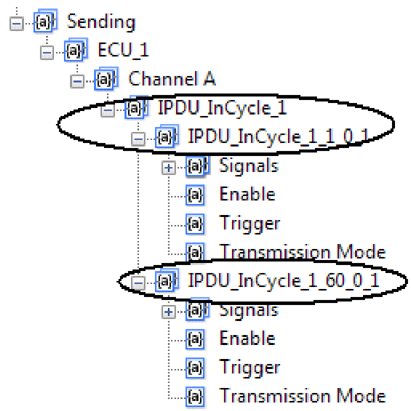
PDU is sent or monitored once

If a PDU is sent or monitored only once, the group name is the short name of the PDU. The following example shows the `Pdu_02` PDU.



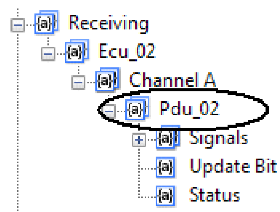
PDU is sent or monitored several times

If a PDU is sent or monitored several times by an ECU, several groups are created. The group names are the short name of the PDU plus the different timings (SlotID_BaseCycle_CycleRepetition). These groups are subordinate to a group whose name is the short name of the PDU. The following example shows the `IPDU_InCycle_1` PDU, which is sent with two different timings (`IPDU_InCycle_1_1_0_1` and `IPDU_InCycle_1_60_0_1`).



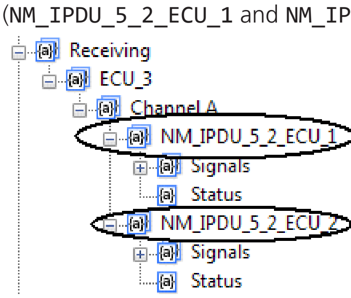
PDU is received once

If a PDU is received once and sent by one ECU, the group name is the short name of the PDU. The following example shows the `Pdu_02` PDU.



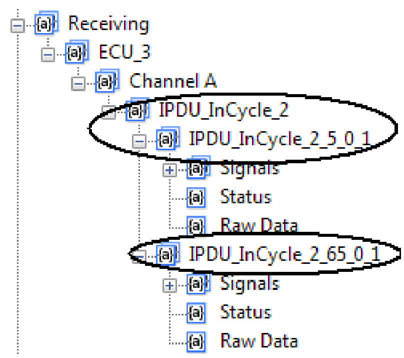
PDU is received once but was sent by several ECUs

If a PDU is received once but sent by several ECUs, the group name is the short name of the PDU plus the short name of the sending ECU. The following example shows the `NM_IPDU_5_2` PDU which is sent by ECU_1 and ECU_2 (`NM_IPDU_5_2_ECU_1` and `NM_IPDU_5_2_ECU_2`).



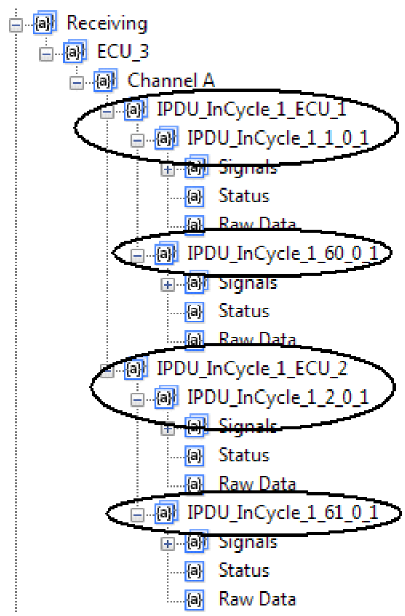
PDU is received several times and was sent by one ECU

If a PDU is received several times and sent by one ECU, the group name is the short name of the PDU plus its timing (SlotID_BaseCycle_CycleRepetition). These groups are subordinate to a group whose name is the short name of the PDU. The following example shows the `IPDU_InCycle_2` PDU, which is sent with different timings (`IPDU_InCycle_2_5_0_1` and `IPDU_InCycle_2_65_0_1`).



PDU is received several times and was sent by several ECUs

If a PDU is received several times and sent by several ECUs, several groups are created. The group names are the short name of the PDU plus the different timings (SlotID_BaseCycle_CycleRepetition). These groups are subordinate to a group whose name is the short name of the PDU plus the short name of the ECU. The following example shows the `IPDU_InCycle_1` PDU, which is sent by ECU_1 with different timings (`IPDU_InCycle_1_1_0_1` and `IPDU_InCycle_1_60_0_1`), and which is also sent by ECU_2 with different timings (`IPDU_InCycle_1_2_0_1` and `IPDU_InCycle_1_61_0_1`).



Related topics

Basics

Dynamic Signal Manipulation

Introduction

You can manipulate a send signal for a specified number of transmissions. After the manipulation, the signal has the value of the signal source that was selected before.

Basics

It is possible to modify data of a TX file. In dynamic signal manipulation, the modification is valid only for a specified time span. The time span can be specified by a parameter. After this time has elapsed, the real-time application uses the signal source that was selected before. Additionally, you can set the validity status of the signal to a specific value during this time.

TRC entries

To manage dynamic signal manipulation, the following TRC entries are inserted in the TRC file for each signal.

Name	Description	Optional
Dynamic Value	Value to be sent during dynamic signal manipulation. The value must be specified as a coded data type.	No
Dynamic Physical Value	Value to be sent during dynamic signal manipulation, specified as a physical data type.	Yes
Dynamic Validity	Signal validity to be sent during dynamic signal manipulation.	Yes
Countdown Value	Time span for dynamic signal manipulation. It specifies how often the dynamic physical value and dynamic validity status are sent.	No

You can connect the variables to ControlDesk instruments to set their values.

Selecting the data source

To select the data source, the TRC file contains the `Source Switch` variable. The following values are defined for the variable.

Value	Description
0	The value of the Simulink model is sent. If there is no corresponding signal block, the coded default value or the last value that was sent is used.
1	The value of the already used `Value` variable is used.
2[1)]	The current value of the alive counter is used.
8	The value of the `Dynamic Value` variable is sent together with the `Dynamic Validity` value for the specified time span (`Countdown Value`). When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.
12[1)]	The alive counter is stopped for the time span specified by the `Countdown Value`. The last value of the alive counter is sent. When `Countdown Value` reaches 0, it is switched back to the last `Source Switch` value.

[1)] This value must only be set if you have configured the signal as an alive counter, see Using an Alive Counter in the Simulation on page 89.

If you use invalid values for the `Source Switch` variable, it is not changed.

Related topics

Basics

Dynamic Update Bit Manipulation

Introduction

You can manipulate the update bit of a TX PDU for a specified number of transmission cycles. After the manipulation, the update bit has the value of the data source that was selected before.

Basics

It is possible to modify the update bit of a TX PDU. In *dynamic* update bit manipulation, the modification is valid only for a specified number of transmission cycles. You can specify this countdown value. After that the update bit manipulation behaves as before: that is, the update bit value and the update bit enable values are reset to the values of the data source that was selected before.

TRC entries

To manage dynamic update bit manipulation, the following TRC entries are inserted in the TRC file for each TX PDU.

Name	Description	Optional
Dynamic Value	Update bit value to be sent during dynamic update bit manipulation.	No
Dynamic Update Bit Enable	Update bit enable value to be used during dynamic update bit manipulation.	No
Countdown Value	Number of transmission cycles the dynamic update bit value and the dynamic update bit enable value are used for update bit manipulation. After that the values are reset to the previous update bit and enable update bit values.	No

You can connect the variables to ControlDesk instruments to set their values.

Selecting the data source

To select the data source, the TRC file contains the `Source Switch` variable. The following values are defined for the `Source Switch` variable.

Value	Description
0	The values of the Simulink model are used for update bit manipulation.
1	The values of the already used `Value` and `Update Bit Enable` variables are used.
8	The values of the `Dynamic Value` and `Dynamic Update Bit Enable` variables are used for the specified number of transmission cycles (`Countdown Value`). When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.

If you use invalid values for the `Source Switch` variable, it is not changed.

Related topics

Basics

Dynamic Checksum Calculation Manipulation

Introduction

You can manipulate checksum calculation of TX PDUs for a specified number of transmission cycles and checksum calculation of RX PDUs for a specified number of receive cycles. After the manipulation, checksum calculation returns to the previous mode, i. e., the real-time application uses the data sources for CRC type and CRC enable that were selected before.

Basics

It is possible to modify checksum calculation for TX and RX PDUs. In *dynamic* checksum calculation manipulation, the modification is valid only for a specified number of transmission or receive cycles. You can specify this number by a parameter. After that the checksum calculation manipulation behaves as before: that is, the real-time application uses the data sources for CRC type and CRC enable that were selected before.

TRC entries

If CRC calculation is enabled and it is selected for the TRC file in the FlexRay Configuration Tool, corresponding variables to manage dynamic checksum calculation manipulation are also written to the TRC file. The following TRC entries are inserted for each TX PDU and RX PDU.

Name	Description	Optional
Dynamic Type	Type value to be sent during dynamic checksum calculation manipulation.	No
Dynamic CRC Enable	CRC enable value to be used during dynamic checksum calculation manipulation.	No
Countdown Value	Number of transmission cycles the dynamic type value and the dynamic CRC enable value are used for checksum calculation manipulation. After that the values are reset to the previous type and enable CRC values.	No

You can connect the variables to ControlDesk instruments to set their values.

Selecting the data source

To select the data source, the TRC file contains the `Source Switch` variable. The following values are defined for the `Source Switch` variable.

Value	Description
0	The values of the Simulink model are used for checksum calculation manipulation.
1	The values of the already used `Type` and `CRC Enable` variables are used for checksum calculation manipulation.

Value	Description
8	The values of the `Dynamic Type` and `Dynamic CRC Enable` variables are used for checksum calculation manipulation for the specified number of transmission cycles (`Countdown Value`). When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.

If you use invalid values for the `Source Switch` variable, it is not changed.

Related topics

Basics

Using the Generated TRC File of PDU-Based Modeling... 63

Manipulating the Authenticator Value of a Secured IPDU

Introduction

The FlexRay Configuration Tool supports secure onboard communication (SecOC) according to AUTOSAR. Secure onboard communication means securing the payload of authentic IPDUs by authentication mechanisms. Authentication information that is included in a secured IPDU consists of an authenticator and a freshness value.

Authenticator invalidation of TX secured IPDUs

You can manipulate the authenticator value of a secured IPDU that is to be transmitted during run time. You cannot specify an explicit authenticator value, but you can determine whether the authenticator value should be interpreted as a correct or an incorrect value.

Access to the invalidation activation is via the TRC variable `Authenticator Value`. Refer to Using the Generated TRC File of PDU-Based Modeling on page 63.

Related topics

Basics

Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 📖)

Dynamic Manipulation of GTS Communication

Introduction

You can manipulate the GTS communication of a TX PDU for a specified number of transmission cycles. After the manipulation, the time synchronization PDUs have the values of the data source that was selected before.

Basics

It is possible to modify some GTS-relevant data of a time synchronization PDU to be sent. In *dynamic* GTS communication manipulation, the modification is valid only for a specified number of transmission cycles. You can specify this countdown value. After this the GTS communication manipulation behaves as before, that is, the GTS values are reset to the values of the data source that was selected before.

TRC entries

To manage dynamic manipulation of GTS communication, the following TRC entries are inserted in the TRC file for each global time synchronization PDU that is sent.

Name	Description	Optional
Dynamic Value	Value of the GTS parameter to be sent during dynamic manipulation.	No
Countdown Value	Number of transmission cycles the dynamic value is used for GTS communication manipulation. After that the values are reset to the previous values.	No

You can connect the variables to ControlDesk instruments to set their values.

Selecting the data source

To select the data source, the TRC file contains the `Source Switch` variable. The following values are defined for the `Source Switch` variable.

Value	Description
0	The value of the time base manager (time gateway synchronization status, user byte 0, user byte 1, total time) or the fixed E2E_P02 algorithm (CRC, E2E sequence counter) is used for GTS communication manipulation.
1	The value of the already used `Value` variable is used.
8	The value of the `Dynamic Value` variable is used for the specified number of transmission cycles (`Countdown Value`). When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.

If you use invalid values for the `Source Switch` variable, it is not changed.

Related topics

Basics

Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 📖)
Using the Generated TRC File of PDU-Based Modeling.. 63

Using an Alive Counter in the Simulation

Introduction

You can activate and monitor an alive counter in ControlDesk when you have specified a signal as an alive counter in the FlexRay Configuration Tool and activated the Signal Alive Counter Control feature in the Element Selection dialog. Refer to How to Configure an Alive Counter (FlexRay Configuration Tool Guide 📖).

Configuring an alive counter at run time

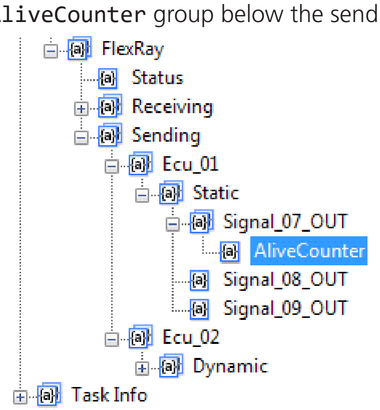
To configure the alive counter during run time, the `Source Switch` variable created for dynamic signal manipulation has two additional states. For more information on the variable and its states, refer to Dynamic Signal Manipulation on page 84.

Value	Description
2	The current value of the alive counter is used for sending.
12	The alive counter is stopped for the time span specified by the `Countdown Value`. The last value of the alive counter is sent. When `Countdown Value` reaches 0, it is switched back to the last `Source Switch` value.

If a signal is specified as an alive counter, the initial value of the `Source Switch` variable is 2.

Manipulating an alive counter at run time

The TRC file contains the variables for controlling the alive counter in an `AliveCounter` group below the send signal.



The `AliveCounter` group comprises the following variables:

Variable Name	Description	Range
Value	The current value of the alive counter	Coded data type
Runtime Behavior	Specifies the behavior of the alive counter: ▪ 0: The alive counter runs independently of the selected source for the signal (`Source Switch` variable) **Note** There is one exception: If `Source Switch` is 12, the alive counter stops independently of the `Runtime Behavior`. ▪ 1: The alive counter runs only if its value is really sent (`Source Switch` is 2 or 12). If it is switched to another source than the alive counter, for example, a value from the Simulink model, the alive counter stops. ▪ 2: The alive counter stops at the current value.	0, 1, 2
Offset	Allows you to add an offset to the value of the alive counter at run time.	Coded data type

The variables can be connected to appropriate ControlDesk instruments to control the alive counter.

Related topics

Examples

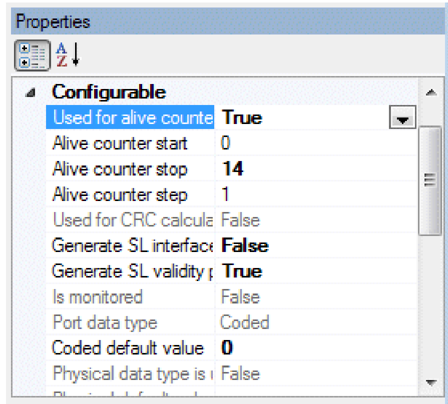
Example of Using an Alive Counter

Introduction

The example demonstrates the configuration and manipulation of an alive counter.

Configuring the TX node

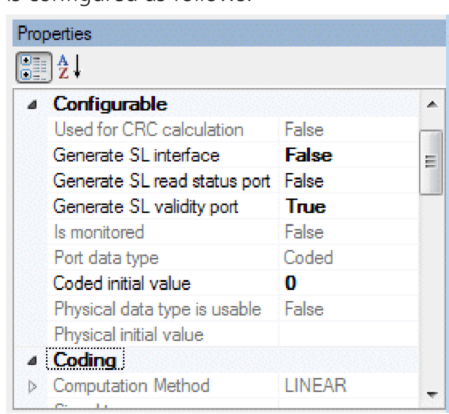
The alive counter must be configured for the TX node. After importing the FIBEX file or AUTOSAR system description file, you must drag the send signal to be used as the alive counter into the Configuration view. Then set the Used for alive counter property to True and specify the parameters of the alive counter, see the following illustration.



The alive counter starts at 0 at the beginning of the simulation. It is increased by 1 every time the signal is sent. When the value 14 is reached, the alive counter starts at 0 again. Additionally, the **Generate SL validity port** parameter is set to True. To generate a variable for the alive counter in the TRC file, the **Signal Alive Counter Control** feature must be selected in the **Element Selection** dialog (see How to Prepare the Manipulation or Monitoring of Frames/PDUs and Signals via TRC File (FlexRay Configuration Tool Guide 📖)). After the automatic task generation has been executed, the code for the TX node can be generated.

Configuring an RX node

RX nodes are the nodes that receive the alive counter. To configure an RX node, drag the RX signal which contains the alive counter into the **Configuration** view after importing the FIBEX file or AUTOSAR system description file. The RX signal is configured as follows.



After automatic task generation has been executed, the code for the RX node can be generated.

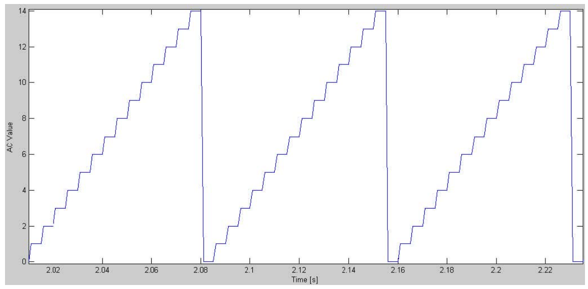
Connecting the variables to instruments

To control the alive counter, you must set some variables. This can be done as usual in ControlDesk. You must connect the variables of the `AliveCounter` group to suitable instruments to view and configure the variables of the alive counter.

To monitor the alive counter, you must create a simple layout for the RX node. It is sufficient to add one Display instrument that is connected to the `Value` variable of the `AliveCounter` group to the layout.

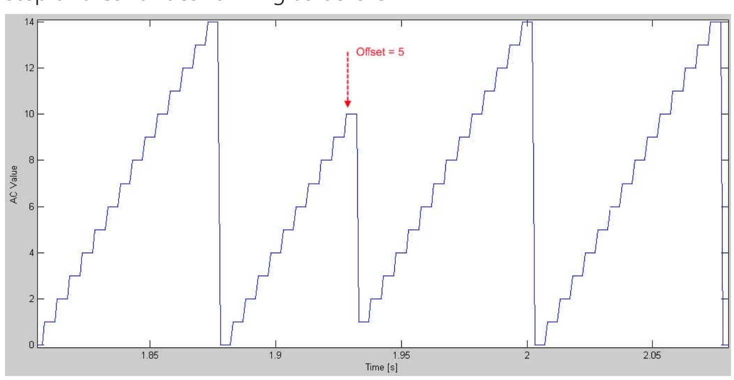
Default behavior

As the send signal has been configured as an alive counter, the `Source Switch` variable is configured to the default value 2. This means that the alive counter value is sent from the beginning of the simulation, see the following illustration.



Adding an offset to the alive counter

You can add an offset value to the alive counter during run time. This is done via the `Offset` variable which is in the `AliveCounter` group of the signal. As soon as a value for the offset is defined and confirmed, the value is added. The following illustration shows an example. At the time the alive counter has the value 10, an offset of 5 is added. The value of the alive counter is 1 in the next step and continues running as before.

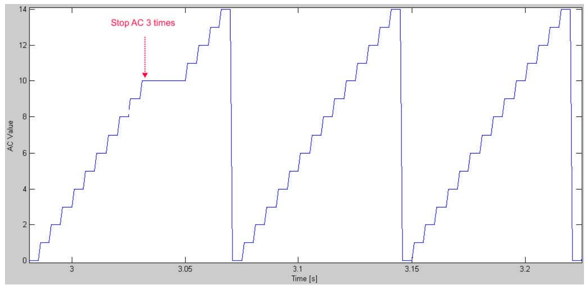


Stopping the alive counter

You can stop the alive counter for a certain time or permanently.

A permanent stop can be achieved using the `Runtime Behavior` variable which is contained in the `AliveCounter` group. As soon as this variable is set to 2, the alive counter stops at its current value.

To stop the alive counter for a certain time only, the dynamic signal manipulation feature is used (see Dynamic Signal Manipulation on page 84). The `Countdown Value` parameter must be set first. In this example, the value is set to 3 to stop the alive counter for 3 transmissions. After the value was set, the `Source Switch` parameter is set to 12. The alive counter is stopped for 3 transmissions and continues counting afterwards.



Switching to a constant value

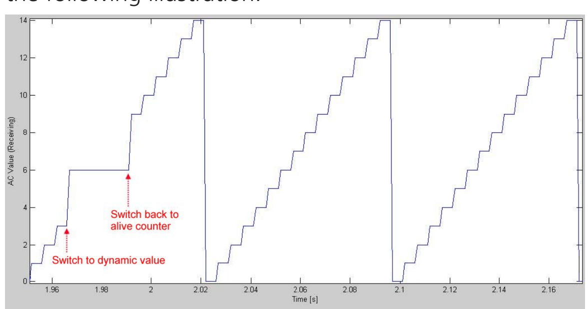
While the alive counter is transmitted, you can switch to a predefined value for a certain time or permanently. You can also switch to the Simulink model value (`Source Switch` = 0). You can use the `Runtime Behavior` variable to set the behavior of the alive counter during that time.

- 0: The alive counter continues running in the background.
- 1: The alive counter stops at the current value.

In the following example, the value 6 is transmitted for 5 transmissions. During this time, the alive counter continues running in the background. To achieve this behavior, the parameter must be set to the following values:

- `Countdown Value`: 5
- `Dynamic Value`: 6
- `Runtime Behavior`: 0

To switch to the dynamic value, set the **`Source Switch`** variable to 8. After 5 transmissions, the **`Source Switch`** variable is automatically reset to the previous value (2) and the value of the alive counter is used for transmission again. See the following illustration.



Related topics

Basics

W