FlexRay Configuration Blockset

Reference

For FlexRay Configuration Blockset 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

This publication and the contents hereof are subject to change without notice.

Contents

About This Document

Contents

This reference describes the FlexRay Configuration Blockset. The blockset is used to connect SCALEXIO systems with a DS2671 Bus Board, DS2672 Bus Module, and/or DS6311 FlexRay Board or the MicroAutoBox III to a FlexRay bus.

> **Note**
>
> The *FlexRay Configuration Blockset* supports only MicroAutoBox III and SCALEXIO systems. If you work with a different dSPACE real-time system, you must use the *RTI FlexRay Configuration Blockset* instead.

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.

General Information on the FlexRay Configuration Blockset

Where to go from here

Information in this section

Features of the FlexRay Configuration Blockset

Features

The main features of the FlexRay Configuration Blockset are:

- Generating 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 Template.
- Supporting single-channel and dual-channel FlexRay systems
- Simulating the FlexRay node on a MicroAutoBox III or a SCALEXIO system
- Supporting PDU-based modeling

- Simulating several FlexRay buses on one dSPACE real-time system, for example, to simulate a gateway
- Supporting task-driven modeling

Library layout / model

The FlexRay Configuration Blockset works in conjunction with a FIBEX file or AUTOSAR System Template. The FlexRay Configuration Tool reads the imported FIBEX file or AUTOSAR system description file and generates Simulink configuration data. A MATLAB command generates blocks which are configured according to the configuration data. The generated blocks can be used for real-time simulation.

The blockset consists of different parts.

FlexRay Configuration Blockset The blockset contains blocks to model a FlexRay node under Simulink. The block library contains the blocks for the FlexRay communication such as PDU TX and PDU RX blocks. Refer to Overview of the FlexRay Configuration Blockset on page 9.

The blockset provides the basic blocks for the automatically generated FlexRay model. It cannot be used in Simulink directly.

Automatically generated FlexRay model Automatically generated FlexRay models are generated according to configuration data created by the FlexRay Configuration Tool. The blocks of the model are configured for your FlexRay network, which is described by a FIBEX file or AUTOSAR system description file. Refer to Overview of an Automatically Generated FlexRay Model on page 11.

Configuring FlexRay communication

The automatically generated FlexRay models provide the basic blocks for FlexRay communication. To analyze and handle the FlexRay communication model, you must configure it in ConfigurationDesk. For information on the entire workflow, refer to:

- FlexRay Bus Connection (SCALEXIO – Hardware and Software Overview 📖)
- Overview of the Workflow (Model Interface Package for Simulink - Modeling Guide 📖)

Speeding up TRC file generation

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 📖).

Hardware support

The FlexRay Configuration Blockset supports the following dSPACE hardware:

- SCALEXIO systems with a DS2671 Bus Board, DS2672 Bus Module, and/or DS6311 FlexRay Board

- MicroAutoBox III in the following variants:
  - 1403/1511/1514
  - 1403/1513/1514

  The DS1514 must be equipped with a DS4340 FlexRay Interface Module.

Demo model

For a Simulink model that shows how to use the FlexRay blocks, refer to the demo which comes with ConfigurationDesk. You can find the model files in the `FlexRayConfigDemo` folder in the Documents folder (`%USERPROFILE%\My Documents\dSPACE\ConfigurationDesk\<VersionNumber>`).

Related topics

References

Overview of the FlexRay Configuration Blockset..9

Overview of the FlexRay Configuration Blockset

Introduction

This topic gives you a short description of the FlexRay Configuration Blockset and its block library.

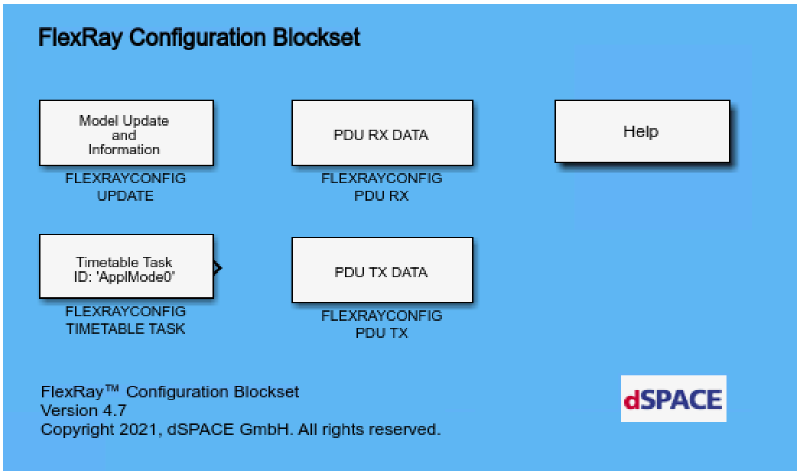
FlexRay Configuration Blockset

The FlexRay Configuration Blockset is a Simulink blockset for simulating a FlexRay node with a MicroAutoBox III or SCALEXIO system.

Library access

In the MATLAB Command Window, enter **`flexrayconfig`** or **`frconf`**. The block library of the FlexRay Configuration Blockset is displayed.

> **Note**
>
> The block library provides the basic blocks for the automatically generated FlexRay model. Do not use it directly in the Simulink model. Instead, use the automatically generated FlexRay model.


FlexRay Configuration Blockset
Model Update and Information
FLEXRAYCONFIG UPDATE
PDU RX DATA
FLEXRAYCONFIG PDU RX
Help
Timetable Task ID: 'ApplMode0'
FLEXRAYCONFIG TIMETABLE TASK
PDU TX DATA
FLEXRAYCONFIG PDU TX
dSPACE

FlexRay™ Configuration Blockset
Version 4.7
Copyright 2021, dSPACE GmbH. All rights reserved.

FlexRay Configuration blocks For details on the blocks of the FlexRay Configuration Blockset, refer to FlexRay Configuration Blockset on page 15.

Help Displays this reference information.

Related topics

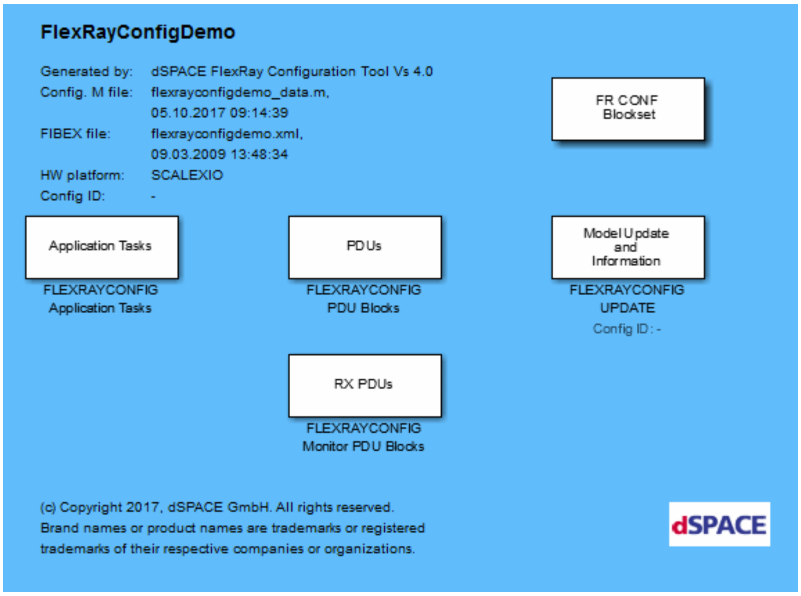
References

- Features of the FlexRay Configuration Blockset..7
- FLEXRAYCONFIG PDU RX..38
- FLEXRAYCONFIG PDU TX..22
- FLEXRAYCONFIG TIMETABLE TASK..53
- FLEXRAYCONFIG UPDATE...16

Overview of an Automatically Generated FlexRay Model

Automatically generated FlexRay model

The following illustration shows an example of an automatically generated FlexRay model.



FlexRayConfigDemo

Generated by: dSPACE FlexRay Configuration Tool Vs 4.0
Config. M file: flexrayconfigdemo_data.m, 05.10.2017 09:14:39
FIBEX file: flexrayconfigdemo.xml, 09.03.2009 13:48:34
HW platform: SCALEXIO
Config ID: -

FR CONF Blockset

Application Tasks — FLEXRAYCONFIG Application Tasks

PDUs — FLEXRAYCONFIG PDU Blocks

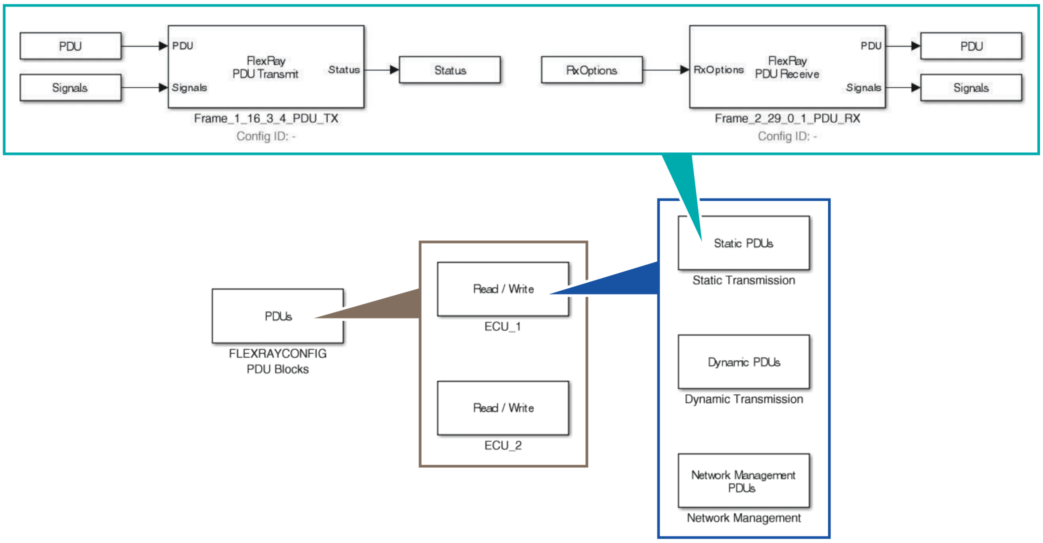
Model Update and Information — FLEXRAYCONFIG UPDATE — Config ID: -

RX PDUs — FLEXRAYCONFIG Monitor PDU Blocks

(c) Copyright 2017, dSPACE GmbH. All rights reserved.
Brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

dSPACE

The model contains subsystems with the configured FlexRay blocks. All the generated FlexRay blocks are organized in subsystems such as static and dynamic subsystems. This allows you to assign the single subsystems to different tasks. The following illustration shows an example of the hierarchy of generated subsystems.



Related topics

HowTos

How to Generate Blocks for Modeling a FlexRay Communication (Model Interface Package for Simulink - Modeling Guide 🕮)

References

dsfr_modelgenerate.......... 55
Structure of the Automatically Generated FlexRay Model.......... 12

Structure of the Automatically Generated FlexRay Model

Structure

The following table shows the structure of a model, with the names of the blocks and subsystems (the FLEXRAYCONFIG string is omitted). Some names used for subsystems or blocks are defined in the FIBEX file or AUTOSAR system description file. Angle brackets contain placeholders.

Level	Description / Related FlexRay Configuration Block
PDU Blocks	This subsystem contains all the PDUs which can be used in the real-time model.
<ECU_Name>	Short name of the ECU which sends/receives PDUs. This subsystem exists for each ECU which sends/receives signals.
Static Transmission	This subsystem contains all the blocks for sending/receiving static signals in a PDU.
<PDU-Name>_X_Y_Z_PDU_TX_Ch	PDU block for sending static signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU TX on page 22.
<PDU-Name>_X_Y_Z_PDU_RX_Ch	PDU block for receiving static signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU RX on page 38.
Dynamic Transmission	This subsystem contains all the blocks for sending/receiving dynamic signals in a PDU.
<PDU-Name>_X_Y_Z_PDU_TX_Ch	PDU block for sending dynamic signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle)

Level	Description / Related FlexRay Configuration Block
	▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU TX on page 22.
<PDU-Name>_X_Y_Z_PDU_RX_Ch	PDU block for receiving dynamic signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU RX on page 38.
Network Management	This subsystem contains all the PDUs for sending/receiving signals for network management.
<PDU-Name>_X_Y_Z_PDU_TX_Ch	PDU block for sending signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU TX on page 22.
<PDU-Name>_X_Y_Z_PDU_RX_Ch	PDU block for receiving signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU RX on page 38.
<Application_Task_Name> (SIM_<ECU_Name>)	This subsystem contains all the PDUs for sending/receiving static signals to or from an application task with the name <Application_Task_Name>. This subsystem exists for each application task which sends/receives PDUs.
UPDATE	This block can be used to update the model. See FLEXRAYCONFIG UPDATE on page 16.
Application Tasks	This subsystem contains all the created application tasks.
<Application_Task_Name>	Timetable task for triggering the application subsystem. <Application_Task_Name> is the name of a created application task. See Hardware-Triggered Runnable Function Block (Model Interface Package for Simulink Reference 📖).
<Application_Task_Name>_Subsystem	Function-call-triggered subsystem for the blocks of the application task.
Monitor PDU Blocks	This subsystem contains all the PDUs which can be monitored.
<ECU_Name>	Short name of the ECU which sends/receives PDUs. This subsystem exists for each ECU which sends/receives signals.
<Application_Task_Name> (MON_<ECU_Name>)	This subsystem contains all the PDU blocks for monitoring PDUs by an application task with the name

Level	Description / Related FlexRay Configuration Block
	<Application_Task_Name>. This subsystem exists for each application task which monitors PDUs.
<PDU-Name>_X_Y_Z_PDU_RX_Ch	PDU block for receiving dynamic signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU RX on page 38.
Monitor_RX (MON_<ECU_Name>)	This subsystem contains all the PDUs which can be monitored by the ECU.
<PDU-Name>_X_Y_Z_PDU_RX_Ch	PDU block for receiving dynamic signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle) ▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See FLEXRAYCONFIG PDU RX on page 38.

Related topics

References

Overview of an Automatically Generated FlexRay Model.. 11

FlexRay Configuration Blockset

Where to go from here

Information in this section

FLEXRAYCONFIG UPDATE

Purpose

To update the FlexRay blocks in the Simulink model.

Where to go from here

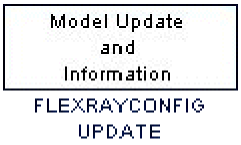
Information in this section

Information in other sections

How to Update the FlexRay Blocks in Simulink Models (Model Interface Package for Simulink - Modeling Guide 📖)
If you used the blocks of the automatically generated FlexRay model in your Simulink model and afterwards changed the FlexRay configuration, you must update your Simulink model.

Block Description (FLEXRAYCONFIG UPDATE)

Block



Purpose

To update a Simulink model based on a generated FlexRay configuration and to provide information on the configuration to ConfigurationDesk.

The block is added to the automatically generated FlexRay model during the generation process. Some block parameters are read-only.

Description

The FLEXRAYCONFIG UPDATE block updates the Simulink model based on the generated FlexRay configuration and provides the configuration information of the model to ConfigurationDesk.

When the Simulink model is updated, a reduced automatically generated FlexRay model is created. Its name is the original file name extended by "_diff". It is located in the current working folder. It contains only the Simulink blocks of the FlexRay configuration which are not used in your FlexRay model. For details, refer to How to Update the FlexRay Blocks in Simulink Models (Model Interface Package for Simulink - Modeling Guide 📖).

> **Note**
>
> The mapping subsystems which belong to the FLEXRAYCONFIG PDU RX and FLEXRAYCONFIG PDU TX blocks are not updated. You must update them manually.

The update process is logged in a log file located in the current working folder. When the update process is completed, a confirmation dialog opens with a link to the `<ModelName>_UpdateSummary.log` file. The log file lists the unused blocks from the library and old blocks in the model, and indicates the number of updated blocks. The link to the `<ModelName>_UpdateSummary.log` file is also contained in the MATLAB workspace.

> **Note**
>
> There must be exactly one FLEXRAYCONFIG UPDATE block for each FlexRay configuration in the Simulink model. Because a Simulink model can contain up to four FlexRay configurations, there can be up to four FLEXRAYCONFIG UPDATE blocks in a Simulink model.

Dialog settings

The dialog settings can be specified on the following pages:

- Model Update Page (FLEXRAYCONFIG UPDATE) on page 18
- Configuration Info Page (FLEXRAYCONFIG UPDATE) on page 19
- CRC Settings Page (FLEXRAYCONFIG UPDATE) on page 20
- XCP Frames (FLEXRAYCONFIG UPDATE) on page 21

Related topics

References

Overview of an Automatically Generated FlexRay Model.. 11
Structure of the Automatically Generated FlexRay Model.. 12

Model Update Page (FLEXRAYCONFIG UPDATE)

Purpose

To update the FlexRay blocks in the Simulink model.

Parameters

Basis for FlexRay configuration Lets you select the automatically generated FlexRay model which is the base for the update procedure.

Use path relative to the configuration folder Indicates whether the path to the automatically generated FlexRay model is relative to the model path or absolute.

Delete obsolete block(s) Indicates whether obsolete blocks are deleted during the model update. Obsolete blocks are blocks which are used in the model but do not exist in the automatically generated FlexRay model.

Keep CRC path settings (on tab page CRC Settings) Indicates whether the CRC path settings are kept during model update. The CRC path settings specified in the FlexRay Configuration Tool can differ from the settings on the CRC Settings page. If you select this checkbox, the settings specified on the CRC Settings page are kept.

Check if FlexRay Configuration blocks are derived from custom libraries Lets you enable or disable the check for FlexRay Configuration blocks from custom libraries during the model update process.

- If the checkbox is selected, the entire Simulink model to be updated is scanned for FlexRay Configuration blocks that are derived from custom libraries. If one or more blocks are found, the update process aborts, and a dialog displays the referenced custom libraries.

> **Note**
>
> Activating the check can increase memory consumption by MATLAB, because the check is performed recursively through the custom libraries used in your model. In extreme cases, for example, with very large models, MATLAB might crash.

- If the checkbox is cleared, no search for FlexRay Configuration blocks from custom libraries is executed. The model update ignores any FlexRay Configuration blocks derived from custom libraries that are in your Simulink model. If your model contains such FlexRay Configuration blocks, an error message indicating the invalid blocks will appear during the build process.

Enabling the check is useful if you want to find out whether the FlexRay Configuration blocks in your Simulink model to be updated come from custom libraries.

Update Lets you start the update of the model.

Note

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

Related topics

References

Block Description (FLEXRAYCONFIG UPDATE).. 16
CRC Page (FlexRay Configuration Tool Reference 🕮)
CRC Settings Page (FLEXRAYCONFIG UPDATE).. 20

Configuration Info Page (FLEXRAYCONFIG UPDATE)

Purpose

To get information on the generated FlexRay configuration.

Note

The block parameters are read-only.

Parameters

FlexRay Configuration frame Displays information on the FlexRay configuration:

Parameter	Description
Project name	Name of the project file used for configuration
Generated by	Tool used for configuration and platform for which the blocks are configured
Version	Version number of the FlexRay configuration tool used
Configuration ID	Configuration ID which was set in the FlexRay Configuration Tool if the multiple bus option is enabled

Configuration data file frame Displays information on the configuration data file:

Parameter	Description
File name	Name of the configuration data file
Date	Date and time of the configuration data file
Generated at	Date and time when the configuration data file was generated

FIBEX data base frame Displays information on the database:

Parameter	Description
File name	Name of the database (FIBEX or AUTOSAR system description file) the model was generated from
Date	Date and time when the database was generated
Version	Version of the database
Short name	Short name of the database
Long name	Long name of the database
Revision label	Revision label of the database
Number of channels	Number of channels used for communication which is specified in the database
Used channels	Channel type which is used for communication (A, B, or AB)

Related topics

References

Block Description (FLEXRAYCONFIG UPDATE).. 16

CRC Settings Page (FLEXRAYCONFIG UPDATE)

Purpose

To select another CRC C file for checksum calculation.

Description

The dialog elements are enabled only if checksum calculation is specified in the FlexRay Configuration Tool (see Basics on Implementing Checksum Algorithms (FlexRay Configuration Tool Guide 🕮)).

If you want to keep the settings of this page during model update, you must select a checkbox on the **Model Update** page.

Parameters

CRC source file Lets you select the CRC C file which contains the CRC algorithms.

Use relative path Lets you specify whether the path to the CRC file is relative to the model path or absolute.

Enable include path specification Lets you specify whether another include path is to be used. If the checkbox is selected, you can select a path for including additional header files.

Use relative path Lets you specify whether the path to the additional header file is relative to the model path or absolute.

Related topics

HowTos

How to Handle Checksum Calculation for a PDU (Model Interface Package for Simulink - Modeling Guide 📖)

References

Block Description (FLEXRAYCONFIG UPDATE).. 16
Model Update Page (FLEXRAYCONFIG UPDATE).. 18

XCP Frames (FLEXRAYCONFIG UPDATE)

Purpose

To display the XCP frame information.

Parameters

ECUName Displays the name of the ECU which has XCP frames defined by the FIBEX file or AUTOSAR system description file.

Master Displays the master/slave configuration of the ECU which has XCP frames:

- Master: ECU is configured as XCP master
- Slave: ECU is configured as XCP slave

Number of Frames Displays the number of XCP frames defined in the FIBEX file or AUTOSAR system description file.

Configured Displays the number of XCP frames configured in the FlexRay Configuration Tool.

Related topics

References

Block Description (FLEXRAYCONFIG UPDATE).. 16

FLEXRAYCONFIG PDU TX

Purpose

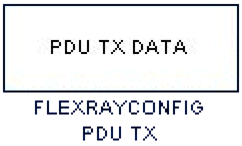
To send a PDU via a FlexRay network.

Where to go from here

Information in this section

Block Description (FLEXRAYCONFIG PDU TX)

Block

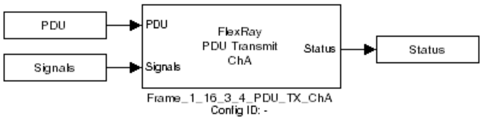


Purpose

To send a PDU via a FlexRay network.

Description

The FLEXRAYCONFIG PDU TX block integrates all PDU-specific control and status information and all PDU signals in one block.



The **FLEXRAYCONFIG PDU TX** block can be connected to **PDU**, **Signal** and/or **Status** mapping subsystems. Each of these mapping subsystems contains an interface for connecting the real-time model with the PDU data. The mapping subsystems can be replaced by user-defined buses.

Note

The mapping subsystems and the included blocks are added to the model during the generation process. The subsystems are not included in the blockset library and are not updated when you update the Simulink model. The block parameters are read-only.

Note

If you use user-defined buses instead of the generated mapping subsystems, all the signal inputs defined for the PDU block must also be specified in the user-defined bus.

I/O characteristics

The following table shows the mapping subsystems that connect signals to the inports:

Mapping Subsystem	Meaning
PDU	Sets and manipulates the PDU-specific data and options. For details, refer to PDU mapping subsystem on page 24.
Signals	Sets and manipulates the signals of the PDU. For details, refer to Signals mapping subsystem on page 28.

The following table shows the mapping subsystem that connects signals from the outport:

Mapping Subsystem	Meaning
Status	Provides the PDU-specific status information. For details, refer to Status mapping subsystem on page 30.

Dialog settings

The dialog settings can be specified on the following pages:

- PDU Page (FLEXRAYCONFIG PDU TX) on page 33
- Signal Page (FLEXRAYCONFIG PDU TX) on page 35
- PDU Option Page (FLEXRAYCONFIG PDU TX) on page 36
- Signal Option Page (FLEXRAYCONFIG PDU TX) on page 37

Related topics

References

FLEXRAYCONFIG PDU TX Mapping Subsystems

Purpose

To map PDU-specific control signals to the ports of a FLEXRAYCONFIG PDU TX block.

Description

The subsystems are created automatically. They can be used to map the signals of your real-time model to the FLEXRAYCONFIG PDU TX block. The subsystems contain BusCreator and BusSelector Simulink blocks which are used for mapping the signals.

PDU mapping subsystem

The following table shows all the control elements that can be part of the PDU bus. The control elements / options that are actually part of the PDU bus depend on the configuration of the PDU and the PDU type.

Simulink Inport	Range	Simulink Data Type	Meaning
TxTrigger	0, 1, 2	UInt32	Triggers the sending of an event PDU (static and dynamic) ▪ 0: Sending is not triggered. ▪ 1: Sending is triggered. ▪ 2: Sending is triggered and additionally the sending buffer is checked for access violation.
TxEnable	0 / 1	Boolean	Enables the cyclic sending of a dynamic cyclic PDU ▪ 0: PDU is not sent cyclically. ▪ 1: PDU is sent cyclically.
TransmissionModeSelector[1)]	0, 1, 98, 99	UInt32	Specifies the transmission mode that is used with the PDU: ▪ 0: Transmission mode `False` is used. ▪ 1: Transmission mode `True` is used. ▪ 98: Transmission mode `LPDU timing triggered` is used. ▪ 99: Transmission mode `User-Defined` is used. For information on PDU transmission modes, refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 📖).
MDTEnable	0 / 1	Boolean	Enables or disables the specified minimum delay time for the PDU. ▪ 0: Minimum delay time support is disabled. ▪ 1: Minimum delay time support is enabled.

Simulink Inport	Range	Simulink Data Type	Meaning
			The minimum delay time of a PDU, which is defined in the AUTOSAR system description file, specifies the minimum delay time between successive transmissions of the new data of the PDU.
TriggerMultiplexerSwitchCode	0 - 2^{32}	UInt32	Specifies the switch code for a triggered sub-PDU.
EnableMultiplexerSwitchCode	0 - 2^{32}	UInt32	Specifies the switch code for a cyclic sub-PDU.
HWEnable	0 / 1	Boolean	Enables or disables, via hardware, the sending of a static TX PDU, 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. ▪ 0: The sending of a static TX PDU is disabled. ▪ 1: The sending of a static TX PDU is enabled. **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	0 / 1	Boolean	Enables or disables, via software, the sending of a static TX PDU/frame. ▪ 0: The sending of the static TX PDU/frame is disabled. However, there are cases where the

Simulink Inport	Range	Simulink Data Type	Meaning
			static TX PDUs might nevertheless be sent (see below). ▪ 1: The sending of the static TX PDU/frame is enabled. 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 the following settings in the FlexRay Configuration Tool: ▪ SWEnable Configuration (see General Page (FlexRay Configuration Tool Reference 🕮)) ▪ Static TX buffer transmission mode (see Generators Page (FlexRay Configuration Tool Reference 🕮)) The following describes which data is sent in each case. *Static PDU is disabled via software (SWEnable = 0)*: ▪ FIBEX version ≤ 2.0: The SW Enable Configuration property in the General Properties dialog is always set to Control of L-PDU commit to FlexRay buffer. You cannot change this setting. ▪ Null frames are sent if the Static TX buffer transmission mode property of the CHI Code Generator is set to Event (null frame used). ▪ Old data is sent if the Static TX buffer transmission mode property is set to State (old value used). ▪ FIBEX+, FIBEX 3.x, FIBEX 4.1.x, or AUTOSAR System Template: If the SW Enable Configuration property in the General Properties dialog is set to Control of L-PDU commit to FlexRay buffer: ▪ Null frames are sent if the Static TX buffer transmission mode property of the CHI Code Generator is set to Event (null frame used). ▪ Old data is sent if the Static TX buffer transmission mode property is set to State (old value used). The update bit has the value that was set before SWEnable was set to 0. 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.

Simulink Inport	Range	Simulink Data Type	Meaning
			If the Static TX buffer transmission mode property is set to Event (null frame used), note the following information: **Note** An update bit value of 0 resulting from the corresponding SWEnable setting can be overruled, depending on whether the update bit of a PDU is manipulable or not. If the UpdateBitEnable inport is 1, the automatic calculation of the update bit of a PDU is disabled. The update bit of the PDU is set to the value specified by the UpdateBitValue inport. If SWEnable is 0 and the update bit value fed into the UpdateBitValue inport is 1, the PDU still sends old data. If UpdateBitEnable is 0/not used, the sending of static TX PDUs is enabled and disabled according to the SWEnable settings. *Static PDU is enabled via software (SWEnable = 1)*: ▪ If the SW Enable Configuration property in the General Properties dialog is set to Control of L-PDU commit to FlexRay buffer and the Static TX buffer transmission mode property of the CHI Code Generator is set to Event (null frame used): ▪ A null frame is sent if the LPDU to be sent contains exactly one PDU, and this PDU has not been updated and does not have a PDU update bit. ▪ Payload data is sent for all other LPDUs. ▪ For all other combinations of the SW Enable Configuration and Static TX buffer transmission mode properties settings: ▪ Payload data is sent.
RawDataEnable	0 / 1	Boolean	Enables or disables access to raw data of the PDU ▪ 0: Raw data access for the PDU is not active. ▪ 1: Raw data access for the PDU is active. If RawDataEnable is set to 1, the values of any parallel signals on the Signals bus are overwritten by raw data. You can enable or disable raw data overwriting by setting this parameter.
RawDataTxBytes	0 ... 254	UInt8	Provides a vector of bytes which contains the data to be written to the TX PDU.
RawDataStartPosition	0 ... PayloadLengthValue · 8 - 1	UInt32	Specifies the start position within the TX PDU in bits. The value must be specified in bits and is therefore not limited to bytes. RawDataStartPosition = 0 starts with the first bit position. The maximum value

Simulink Inport	Range	Simulink Data Type	Meaning
			depends on the payload length of the TX PDU in bytes.
RawDataLength	0 ... PayloadLengthValue · 8	UInt32	Specifies the number of bits which are accessed. If it is 0, no data is written. The maximum value depends on the payload length of the TX PDU in bytes.
PayloadLengthEnable[2]	0 / 1	Boolean	Enables/disables the manipulation of the payload length ▪ 0: Payload length is not manipulated. The PDU is sent with the maximum payload length. ▪ 1: Payload length is set to the value specified with the **PayloadLengthValue** port.
PayloadLengthValue[3]	0 ... max payload length of PDU	UInt32	Specifies the payload length value in bytes of the according PDU.
CRCEnable	0 / 1	Boolean	Enables or disables the checksum calculation for the selected TX PDU. The PDU must be configured for checksum calculation using the FlexRay Configuration Tool. For details, refer to Basics on Implementing Checksum Algorithms (FlexRay Configuration Tool Guide 🕮) and How to Handle Checksum Calculation for a PDU (Model Interface Package for Simulink - Modeling Guide 🕮). ▪ 0: The checksum is not calculated. The CRC function is not called by the Com code. ▪ 1: The checksum is calculated. The CRC function is called by the Com code.
CRCType	0 ... *n*	UInt32	Specifies which CRC algorithm of the CRC C file is used. *n* is the highest ID used in the CRC C file for selecting the CRC algorithm.
UpdateBitEnable	0 / 1	Boolean	Selects how the update bit is set: ▪ 0: The update bit is set automatically in the generated Com code. ▪ 1: The update bit is set by the block using the UpdateBitValue inport.
UpdateBitValue	0 / 1	Boolean	Specifies the value of the update bit when the UpdateBitEnable inport is 1.

[1] The port is available only if it is activated in the FlexRay Configuration Tool. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮).

[2] The port is available only if generation of a payload length manipulation port is activated in the FlexRay Configuration Tool. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮).

[3] The port is available only if generation of a payload length port is activated in the FlexRay Configuration Tool. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮).

Signals mapping subsystem

The following table shows all the signals that can be part of the **Signals** bus. The signals that are actually part of the Signals bus depend on the configuration of the PDU and the PDU type. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮). <Signal name> is a placeholder for the signal name.

Simulink Inport	Range	Simulink Data Type	Meaning
<Signalname>_Value	-	Depends on the signal configuration in the FIBEX file or AUTOSAR system description file. The data types Int64, UInt64, ASCII, and OTHER are interpreted as 8-byte arrays. The UInt8[n] data type (used for data with opaque byte format, which can be defined in AUTOSAR system description files) is realized by a dynamic n x UInt8 array (i.e.: by n x 8 bytes of the UInt8 data type), where n depends on the signal length.	Writes a signal of the Simulink model to the FlexRay bus. This inport is available only for signals with 'Coded' or 'Physical' port data type.
<Signalname>_Value_coded	-	Depends on the signal configuration in the FIBEX file or AUTOSAR system description file. The data types Int64, UInt64, ASCII, and OTHER are interpreted as 8-byte arrays.	Writes a coded signal value to the FlexRay bus. This inport is available only for signals with the SCALE_LINEAR_TEXTTABLE computation method for which the 'Coded and physical' port data type is selected. It can be used for transmitting text table values, for example. Values of this inport are transferred as they are, i.e., without conversion.
<Signalname>_Value_phy	-	Depends on the signal configuration in the FIBEX file or AUTOSAR system description file. The data types Int64, UInt64, ASCII, and OTHER are interpreted as 8-byte arrays.	Specifies a physical signal value to be sent. This inport is available only for signals with the SCALE_LINEAR_TEXTTABLE computation method for which the 'Coded and physical' port data type is selected. Values of this inport are converted to coded values by the defined linear computation scale before they are packed into the PDU.
<Signalname>_Coded_phy_switch	0 / 1	UInt32	Specifies whether the value from the coded value inport or the physical value inport is used when the PDU is packed. The following values are possible: ▪ 0: The value from the <Signal name>_Value_coded inport is used. ▪ 1: The value from the <Signal name>_Value_phy inport is used. (This is the default setting.) This inport is available only for signals with the SCALE_LINEAR_TEXTTABLE computation method for which the 'Coded and physical' port data type is selected, and if the Physical data type conversion layer option is set to COMMUNICATION on the General page of the General Properties dialog.
<Signalname>_Validity[1)]	0, 1, 2, 4, 8, 16	UInt32	Specifies the validity of the transmitted signal. The following validity values are possible: ▪ 0: NOT VALID ▪ 1: VALID ▪ 2: ERROR ▪ 4: NOT AVAILABLE

Simulink Inport	Range	Simulink Data Type	Meaning
			▪ 8: NOT DEFINED ▪ 16: OTHER

[1] The port is available only if it is activated in the FlexRay Configuration Tool. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮).

Status mapping subsystem

The following table shows all the signals that can be part of the Status bus. The signals that are actually part of the Status bus depend on the configuration of the PDU and the PDU type.

Simulink Outport	Range	Simulink Data Type	Meaning
TriggerStatus	0, 256, 257, 258	UInt32	Displays the triggering status. ▪ 0: PDU was not triggered yet ▪ 256: Triggering was successful ▪ 257: Send buffer is full (available only when TriggerEnable = 2) ▪ 258: The TriggerMultiplexerSwitchCode inport has an invalid switch code.
TransmitStatus	69, 70	UInt32	Displays the buffer status. ▪ 69: Buffer is empty. ▪ 70: Buffer is full. The status is updated 2 times per cycle (At the beginning of the static and at the beginning of the dynamic part). It is recommended to place a PDU block with transmit status in an application task instead of a periodic task.
EnableStatus	256, 258	UInt32	Displays the enable status of a dynamic cyclic TX PDU. ▪ 256: The PDU was successfully enabled. ▪ 258: The EnableMultiplexerSwitchCode inport has an invalid switch code.
HWEnableStatus	0 / 1	UInt32	Displays the HW enable status of a static TX PDU. ▪ 0: Enabling or disabling was successful. ▪ 1: Enabling or disabling was not successful.
SWEnableStatus	0 / 1	UInt32	Displays the SW enable status of a static TX PDU. ▪ 0: Enabling or disabling was successful. ▪ 1: Enabling or disabling was not successful.
RawDataAccessStatus	0, 1, 4	UInt32	Displays status information. The statuses have different meanings with: ▪ 0: Raw data successfully written ▪ 1: Access to raw data failed ▪ 4: Invalid values at RawDataStartPosition and/or RawDataLength inport
PayloadLengthStatus	0, 1, 18	UInt32	Displays status information with payload length manipulation. ▪ 0: The payload length of the frames was changed successfully ▪ 1: Access to payload length of the frame failed ▪ 18: PayloadLength is greater than the maximum payload length which is specified in the FIBEX file or AUTOSAR system description file for the frame

Simulink Outport	Range	Simulink Data Type	Meaning
EnableSwitchCodeStatus	256, 258	UInt32	Displays the enable multiplexer switch code status of a static TX PDU with sub-PDUs. ▪ 256: The multiplexer switch code was successfully set. ▪ 258: The EnableMultiplexerSwitchCode inport has an invalid switch code.
ContainedPDUSendStatus	0, 1, 2, 4	UInt32	Displays the send status of the contained IPDU and the associated container: ▪ 0: The contained IPDU was not triggered. ▪ 1: The contained IPDU was triggered but not added to the container IPDU yet. ▪ 2: The contained IPDU was triggered and packed into the container IPDU. The container IPDU has not been sent yet. ▪ 4: The container IPDU (and thus the contained IPDU) was sent.
ContainerPDUTriggerStatus	0, 1, 2, 4, 8, 16	UInt32	Displays the trigger status of the container IPDU: ▪ 0: The container IPDU was not triggered. ▪ 1: The container IPDU was triggered because the threshold was exceeded. ▪ 2: The container IPDU was triggered because it was full. ▪ 4: The container IPDU was triggered by the first added contained IPDU (`FirstContainedTrigger`). ▪ 8: The container IPDU was triggered because this contained IPDU was added to the container IPDU (`TRIGGER_ALWAYS`) ▪ 16: The container IPDU was triggered by the container IPDU's timeout. 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`).
MDTStatus	0, 1, 2, 3, 4, 8, 12	UInt32	Displays information on the last captured trigger and on whether and how it was considered for triggering (if minimum delay time support is enabled for the PDU). ▪ 0: No recent trigger. ▪ 1: Accepted event trigger. The event-based triggering occurred outside the minimum delay time. ▪ 2: Accepted cyclic trigger. The cyclic triggering occurred outside the minimum delay time. ▪ 3: Accepted event trigger, discarded cyclic trigger. The simultaneous event-based and cyclic triggering occurred outside the 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.
MDTTime	–	Double	Displays the remaining minimum delay time.
GTS_CRC	0 ... 254	UInt8	Displays the CRC value.

Simulink Outport	Range	Simulink Data Type	Meaning
GTS_TimeDomainID	0 ... 15	UInt8	Displays the identifier of the global time domain that is transferred in the time synchronization message.
GTS_E2ESequenceCounter	0 ... 254	UInt8	Displays the end-to-end protection sequence counter (SC) that is transferred in the time synchronization message.
GTS_FlexRayCycleCounter	0 ... 254	UInt8	Displays the value of the FlexRay cycle counter (FCNT) at the time the global time synchronization PDU is built.
GTS_TimeGatewaySynchronizationStatus	0 ... 254	UInt8	Displays the value of the SYNC_TO_GATEWAY (SGW) bit from the Time Base status of the time base manager instance.
GTS_UserByte0	0 ... 254	UInt8	Displays the UserByte 0 (US0) that is transferred in the time synchronization message.
GTS_UserByte1	0 ... 254	UInt8	Displays the UserByte 1 (US1) that is transferred in the time synchronization message.
GTS_Seconds	–	Double	Displays the seconds that are transferred via the FlexRay bus in the 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 GTS_TotalTime.
GTS_Nanoseconds	0 ... 254	UInt32	Displays the nanoseconds that are transferred via the FlexRay bus in the 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 GTS_TotalTime.
GTS_TotalTime	–	Double	Displays the time that is transferred via the FlexRay bus. This is the time that was read when the global time synchronization message was coded.

Related topics

Basics

Basics of Code Generation (FlexRay Configuration Tool Guide 🕮)
Sending Dynamic PDUs or Sub-PDUs (Model Interface Package for Simulink - Modeling Guide 🕮)
Sending Static PDUs and Sub-PDUs (Model Interface Package for Simulink - Modeling Guide 🕮)

HowTos

How to Handle Checksum Calculation for a PDU (Model Interface Package for Simulink - Modeling Guide 🕮)
How to Send or Receive Signals of PDUs (Model Interface Package for Simulink - Modeling Guide 🕮)

References

Block Description (FLEXRAYCONFIG PDU TX).. 22

PDU Page (FLEXRAYCONFIG PDU TX)

Purpose

To display frame or PDU information.

Dialog settings

Information type Displays whether the data is interpreted as a PDU or a frame. If the information type is PDU, the name is the PDU name. If the information type is frame, the name is the frame name. In FIBEX+, FIBEX 3.x, FIBEX 4.1.x, and AUTOSAR System Templates, the information is interpreted as PDUs. In FIBEX 2.0 and lower versions, the information is interpreted as frames.

Name Displays the name of the PDU derived from the short name of the related FIBEX file or AUTOSAR system description file.

Send ECU name Displays the short name of the ECU which sends the PDU. The short name is derived from the FIBEX file or AUTOSAR system description file.

Parent ECU name Displays the short name of the ECU to which the currently displayed PDU is assigned.

FlexRay channel Displays the channel on which the PDU is sent. The following values are possible:

- A
- B
- A & B

Search identical blocks Searches for an identical block in the model. For example, if the PDU is configured for channel A, you can search the same PDU configured for channel B. If identical blocks are found, the MATLAB Command Window contains hyperlinks to them.

Length Displays the PDU length in bytes, which is specified in the FIBEX file or AUTOSAR system description file.

Cyclic timing Displays the list of cyclic timings of the PDU. If sub-PDUs are defined in the FIBEX file or AUTOSAR system description file, all the subitems are listed here as well.

- Multiplexer switch

  Displays the switch codes of a sub-PDU or a subframe.
- Cyclic timing

  Displays the cyclic timing in seconds. The cyclic timing for event-controlled PDUs or frames is 0. If no timing is defined for a subframe, the timing value of its parent frame is displayed. If no timing is defined for a sub-PDU, this sub-PDU is interpreted as an event-controlled PDU and has the cyclic timing value 0.

- Timing type

  Displays the timing type of the PDU or frame. The following values are possible:

Timing Type	Description
SE	Static event
SC	Static cyclic
DE	Dynamic event
DC	Dynamic cyclic
SEC	Static event cyclic
DEC	Dynamic event cyclic
SSE	Subframe static event
SSC	Subframe static cyclic
SDE	Subframe dynamic event
SDC	Subframe dynamic cyclic
SSEC	Subframe static event cyclic
SDEC	Subframe dynamic event cyclic
Event	User-defined event-controlled timing

- Transmission mode

  Displays the transmission mode that the timing is assigned to. The following values are possible:

  - 0: Transmission mode 'False'
  - 1: Transmission mode 'True'
  - 98: Transmission mode 'LPDU timing triggered'
  - 99: Transmission mode 'User-defined'

- Minimum delay time

  Displays the start time of the minimum delay time of the PDU, if the **Enable Minimum Delay Time** property is set to True for the PDU in the FlexRay Configuration Tool.

  In the following cases the minimum delay time feature is not supported and '–' is displayed instead:

  - The PDU has the timing type SC, SSC, DC, or SDC, and the **ECU MDT for cyclic transmission** property is False in the FlexRay Configuration Tool.
  - The transmission mode 'User-defined' or 'LPDU timing triggered' is assigned to the PDU.
  - No timing is specified for the sub-PDU in the communication cluster file.

Absolutely scheduled timing Displays the list of absolutely scheduled timings of the PDU. The list shows the timing, FlexRay channel and controller name for each PDU.

Related topics

References

Block Description (FLEXRAYCONFIG PDU TX)..22

Signal Page (FLEXRAYCONFIG PDU TX)

Purpose

To display the settings of the signals of the PDU.

Dialog settings

Signal selection Lets you select a signal to be displayed.

Name Displays the short name of the selected signal.

Coded bit length Displays the signal length in bits.

Physical data type Displays the physical data type of the selected signal, which is defined in the FIBEX file or AUTOSAR system description file or configured in the FlexRay Configuration Tool. The radio button indicates whether the data type is selected. You can configure the data type mode of a signal (physical or coded) via the FlexRay Configuration Tool.

Coded data type Displays the coded data type of the signal which is defined in the FIBEX file or AUTOSAR system description file. The radio button indicates whether the data type is selected. You can configure the data type of a signal via the FlexRay Configuration Tool.

Constraint Displays the valid ranges of a signal:

- Index
- Validity
- Coded [Min...Max]
- Physical [Min...Max]

The Validity short name and the Min/Max values are derived from the FIBEX file or AUTOSAR system description file. If no Min/Max value pair is defined in the FIBEX file or AUTOSAR system description file, the field remains empty.

Computation method Displays the computation methods of a signal:

- Conv. Type
- Conv. Offset
- Conv. Factor
- Unit
- Text

Conv. Factor and Conv. Offset are defined in the following form:

```
y = Conv.Factor · X + Conv.Offset
```

Related topics

References

Block Description (FLEXRAYCONFIG PDU TX)..22

PDU Option Page (FLEXRAYCONFIG PDU TX)

Purpose

To display information about the PDU block interface and configuration.

Dialog settings

Input port configuration Displays the input ports of the PDU-specific signal. The input ports are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names.

Output port configuration Displays the output ports of the PDU-specific signal. The output ports are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names.

UpdateBit is ignored Displays whether the update bit for the PDU is ignored. You can configure this value via the FlexRay Configuration Tool.

HW Enable static frame(s) after modeling start Displays whether a frame is activated in hardware between the simulation start and the first calculation. You can configure this value via the FlexRay Configuration Tool.

SW Enable static frame after modeling start Displays whether a frame is activated in software between the simulation start and the first calculation. You can configure this value via the FlexRay Configuration Tool.

Max. number of raw data bytes Displays the maximum number of raw data bytes to be sent. This value must be smaller than or equal to the payload length of the PDU. You can configure this value via the FlexRay Configuration Tool to enhance processor performance and simplify modeling.

HW disabled affected PDUs Displays the PDUs which could be affected when this PDU is disabled by hardware because they share the selected slot(s).

Related topics

References

Block Description (FLEXRAYCONFIG PDU TX)..22

Signal Option Page (FLEXRAYCONFIG PDU TX)

Purpose

To display the signal interfaces and configuration.

Dialog settings

Input port configuration Displays the signal inputs that are available for the PDU block. The signal inputs are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names. The inputs are displayed in a tree view. If a signal is included in subframes or sub-PDUs, its associated multiplexed switch codes are also displayed after the signal name.

Output port configuration Displays the signal outputs that are available for the PDU block. The signal outputs are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names. The outputs are displayed in a tree view. If a signal is included in subframes or sub-PDUs, its associated multiplexed switch codes are also displayed after the signal name.

Related topics

References

Block Description (FLEXRAYCONFIG PDU TX).. 22

FLEXRAYCONFIG PDU RX

Purpose

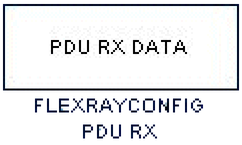
To read a PDU from the FlexRay bus.

Where to go from here

Information in this section

Block Description (FLEXRAYCONFIG PDU RX)

Block

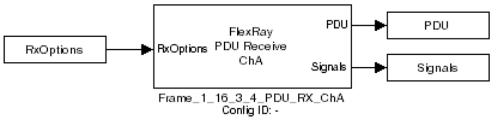


Purpose

To receive a PDU via a FlexRay network.

Description

The **FLEXRAYCONFIG PDU RX** block integrates all PDU-specific receiving information and all PDU receiving signals in one block.



The **FLEXRAYCONFIG PDU RX** block can be connected to **PDU**, **Signal** and/or **RxOptions** mapping subsystems. Each of these mapping subsystems contains an interface for connecting the real-time model with the PDU data. The mapping subsystems can be replaced by user-defined buses.

Note

The mapping subsystems and the included blocks are added to the model during the generation process. The subsystems are not included in the blockset library and are not updated when you update the Simulink model. The block parameters are read-only.

Note

If you employ user-defined buses instead of the generated mapping subsystems, all the signal inputs defined for the PDU block must be also specified in the user-defined bus.

I/O characteristics

The following table shows the mapping subsystem that connects signals to the inports:

Mapping Subsystem	Meaning
RxOptions	Sets the receiving options such as enabling the CRC analysis. For details, refer to RxOptions mapping subsystem on page 40.

The following table shows the mapping subsystems that connect signals from the outport:

Mapping Subsystem	Meaning
PDU	Provides PDU-specific data and status. For details, refer to PDU mapping subsystem on page 41.
Signals	Provides all the signal values of a PDU and their states. For details, refer to Signals mapping subsystem on page 46.

Dialog settings

The dialog settings can be specified on the following pages:

- PDU Page (FLEXRAYCONFIG PDU RX) on page 48
- Signal Page (FLEXRAYCONFIG PDU RX) on page 50
- PDU Option Page (FLEXRAYCONFIG PDU RX) on page 51
- Signal Option Page (FLEXRAYCONFIG PDU RX) on page 52

Related topics

References

Overview of an Automatically Generated FlexRay Model.. 11
Structure of the Automatically Generated FlexRay Model.. 12

FLEXRAYCONFIG PDU RX Mapping Subsystems

Purpose

To map PDU-specific control signals to the ports of a FLEXRAYCONFIG PDU RX block.

Description

The subsystems are automatically created. They can be used to map the signals of your real-time model to the FLEXRAYCONFIG PDU RX block. The subsystems contain BusCreator and BusSelector Simulink blocks which are used for mapping the signals.

RxOptions mapping subsystem

The following table shows all the signals that can be part of the RxOptions bus. The signals that are actually part of the RxOptions bus depend on the configuration of the PDU.

Simulink Inport	Range	Simulink Data Type	Meaning
RawDataStartPos	0 ... *PayloadLength* · 8 - 1	UInt32	Specifies the start position within the RX frame in bits. The value must be specified in bits and is therefore not limited to bytes. The value = 0 means starting with the first bit position. The maximum value depends on the payload length of the RX frame in bytes.
RawDataLength	0 ... *PayloadLength* · 8	UInt32	Specifies the number of bits which are accessed. If the value = 0, no data is read and the RxBytes outport contains the previously read data. The maximum value depends on the payload length of the RX PDU in bytes.
CRCEnable	0 / 1	Boolean	Enables or disables the checksum calculation for the selected RX frame ▪ 0: The checksum is not calculated. The CRC function is not called by the Com code. ▪ 1: The checksum is calculated. The CRC function is called by the Com code.

Simulink Inport	Range	Simulink Data Type	Meaning
			Note If checksum calculation is enabled and a wrong checksum is detected, only the raw data of the RX PDU is updated. The RX signal values of the RX PDU remain unchanged.
CRCType	0 ... *n*	UInt32	Specifies which CRC algorithm of the CRC C file is used. *n* is the highest ID used in the CRC C file for selecting the CRC algorithm.

PDU mapping subsystem

The following table shows all the control elements that can be part of the PDU bus. The control elements/options that are actually part of the PDU bus depend on the configuration of the PDU and the PDU type.

Simulink Outport	Range	Simulink Data Type	Meaning
RawDataRxBytes	0 ... 254	UInt8	Provides a vector of bytes which contains the raw data to be read.
RawDataAccessStatus	0 ... 254	UInt8	Displays status information: ▪ 0: Raw data successfully read ▪ 1: Access to raw data failed ▪ 2: No data received ▪ 4: Invalid values at RawDataStartPos and/or RawDataLength inport ▪ 8: Checksum of frame not correct
PayloadLengthValue[1)]	0 ... max payload length of frame	UInt32	Provides the payload length value in bytes of the frame. It is always an even number.
UpdateBitValue	0 / 1	Boolean	Indicates whether the Update bit of the received frame was set or not ▪ 0: Update bit was not set. ▪ 1: Update bit was set. The port is only generated if the specified PDU has defined an Update bit.
DataReceived	0 / 1	Boolean	Indicates whether a frame was received ▪ 0: The frame was not received in the current sampling step. ▪ 1: The frame was received in the current sampling step.
Nullframe	0 / 1	Boolean	Indicates whether the received frame was a null frame ▪ 0: The last received frame was not a null frame ▪ 1: The last received frame was a null frame.
Error	0 - 65535	UInt32	Indicates various statuses. Each status has its own bit, so parallel statuses are possible. In that case the error values are added. For details on the error statuses, see below.

Simulink Outport	Range	Simulink Data Type	Meaning
UpdateContainedPDU	0 / 1	Boolean	Indicates whether the contained IPDU was in the container IPDU ▪ 0: The contained IPDU was not contained in the container IPDU. ▪ 1: The contained IPDU was contained in the container IPDU.
RXTimestamp	–	Double	Provides the time at which the corresponding LPDU was received on the FlexRay bus.
GTS_CRC	0 ... 254	UInt8	Provides the CRC value that was transferred with the global time synchronization message via the FlexRay bus.
GTS_E2EProtectionStatus	0 ... 65535	UInt32	Displays the 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 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

Simulink Outport	Range	Simulink Data Type	Meaning
			correct/initial reception, but this is within the configured tolerance range. ▪ 0x040: 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.
GTS_TimeDomain	0 ... 15	UInt8	Displays the identifier of the global time domain that is transferred in the global time synchronization message.
GTS_E2ESequenceCounter	0 ... 254	UInt8	Displays the end-to-end protection sequence counter that is transferred in the global time synchronization message.
GTS_FlexRayCycleCounter	0 ... 254	UInt8	Displays the FlexRay cycle counter (FCNT) that was current at the time the global time synchronization PDU was built.
GTS_TimeGatewaySynchronizationStatus	0 / 1	UInt8	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.
GTS_UserByte0	0 ... 254	UInt8	Displays the user byte 0 (US0) that is transferred in the global time synchronization message.
GTS_UserByte1	0 ... 254	UInt8	Displays the user byte 1 (US1) that is transferred in the global time synchronization message.
GTS_Seconds	–	Double	Displays the seconds that are 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 **GTS_TotalTime**.
GTS_Nanoseconds	0 ... 65535	UInt32	Displays the nanoseconds that are 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 **GTS_TotalTime**.
GTS_TotalTime	–	Double	Displays the time that is transferred via the FlexRay bus. This is the time that was read when the global time synchronization message was coded.
GTS_TimeBaseStatus	0 ... 65535	UInt32	Displays status information on the time base manager instance (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

Simulink Outport	Range	Simulink Data Type	Meaning
			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
GTS_Status	0 ... 65535	UInt32	Displays status information on the time synchronization message (name, bit position, description): ▪ MSG_TYPE_UNSUPPORTED, 0, The first eight bits of the messages do not have the value 0x20, which is required for GTS messages. ▪ Reserved, 1 ▪ Reserved, 2 ▪ MSG_TD_WRONG, 3, The bits 3-0 of byte 2 contain an unexpected Time Domain ID. ▪ MSG_NANOSECONDS_INVALID, 4, The value for nanoseconds in the bytes 12-15 is larger than 1000000000. ▪ Reserved, 5 ▪ E2E_PROTECTION_API_ERROR, 6, The value for E2E_Protection does not match E2E_P02STATUS_OK. ▪ Reserved, 7 ▪ Reserved, 8 ▪ TBM_UPDATED_BUT_SC_NOT_OK, 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

Simulink Outport	Range	Simulink Data Type	Meaning
			E2E_P02STATUS_OKSOMELOST or E2E_P02STATUS_INITIAL, for example. ▪ TBM_UPDATED_BUT_CRC_NOT_OK, 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. ▪ TBM_UPDATED_ALL_OK, 11, The time base was synchronized and both the CRC and the sequence counter check were completed successfully. If none of the bits is set, the status value 0 is displayed. In this case, valid status information is available at the GTS_E2EProtectionStatus port, which you should display.

[1] The port is available only if generation of a payload length port is activated in the FlexRay Configuration Tool. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮).

Error status The following table shows the different errors in different bit position.

Error Bit Position	Value	Description
0	1	Communication controller (CC) synchronization error ▪ 0: CC is synchronized with the FlexRay bus ▪ 1: CC is not synchronized with the FlexRay bus
1	2	Buffer access error ▪ 0: No buffer access error ▪ 1: Error while accessing buffer
2	4	Boundary violation ▪ 0: No boundary violation in the FlexRay slot detected ▪ 1: Boundary violation in the FlexRay slot detected
3	8	Syntax error ▪ 0: No syntax error in the FlexRay slot detected ▪ 1: Syntax error in the FlexRay slot detected
4	16	Content error ▪ 0: No content error in the FlexRay slot detected ▪ 1: Content error in the FlexRay slot detected
5	32	Empty slot (FlexRay static frames only) ▪ 0: Data received ▪ 1: No data received in the FlexRay static slot
6[1]	64	Software CRC calculation error ▪ 0: Software CRC from the last received frame is correct. This is also the default value if no CRC calculation is activated. ▪ 1: Software CRC from the last received frame is not correct.
7 to 15		None

[1] This error bit is also used for global time synchronization PDUs.

Generally, the **Error** output is 0 if the value of **Data_received** is 1. The value of the **Data_received** output is 0 if the **Error** output is greater than 0. However, there are some exceptions:

- Boundary Violation

  It is possible that a frame is received while a boundary violation occurs. In this case, the **Data_received** and **Error** outputs are not 0.

- SW_CRC_ERROR

  The software CRC can only be calculated if a frame is received successfully, so the value of **Data_received** is 1 when a software CRC error occurs.

  The software CRC of a frame with update bit is only calculated when the update bit is set, so the values of **Data_received** and the **UpdateBit** are 1 when a software CRC error occurs.

Signals mapping subsystem

The following table shows all the signals that can be part of the **Signals** bus. The signals that are actually part of the **Signals** bus depend on the configuration of the PDU and the PDU type. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 📖). <Signalname> is a placeholder for the signal name.

Simulink Outport	Range	Simulink Data Type	Meaning
<Signalname>_Value	-	Depends on the signal configuration in the FIBEX file or AUTOSAR system description file. The data types Int64, UInt64, ASCII, and OTHER are interpreted as 8-byte arrays. The UInt8[n] data type (used for data with opaque byte format, which can be defined in AUTOSAR system description files) is realized by a dynamic n x UInt8 array (i.e.: by n x 8 bytes of the UInt8 data type), where n depends on the signal length.	Reads a signal of the Simulink model from the FlexRay bus. This outport is available only for signals with 'Coded' or 'Physical' port data type.
<Signalname>_Value_coded	-	Depends on the signal configuration in the FIBEX file or AUTOSAR system description file. The data types Int64, UInt64, ASCII, and OTHER are interpreted as 8-byte arrays.	Reads a coded signal value from the FlexRay bus. This outport is available only for signals with the SCALE_LINEAR_TEXTTABLE computation method for which the 'Coded and physical' port data type is selected. It can be used for transmitting text table values, for example. Values of this outport are used as they are, i.e., without conversion.
<Signalname>_Value_phy	-	Depends on the signal configuration in the FIBEX file or AUTOSAR	Reads a physical signal value from the FlexRay bus. This outport is available only for signals with the SCALE_LINEAR_TEXTTABLE computation method for

Simulink Outport	Range	Simulink Data Type	Meaning
		system description file. The data types Int64, UInt64, ASCII, and OTHER are interpreted as 8-byte arrays.	which the 'Coded and physical' port data type is selected. Values of this outport were converted from the coded values that were transferred via the FlexRay bus using the defined linear computation scale.
<Signalname>_Texttable_status	0 / 1	UInt32	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 from the <Signalname>_Value_phy port. ▪ 1: The received value is a text table value. The received value can be read from the <Signalname>_Value_coded port. Regardless of the status value, both the physical signal value and the coded signal value are updated. This outport is available only for signals with the SCALE_LINEAR_TEXTTABLE computation method for which the 'Coded and physical' port data type is selected.
<Signalname>_Validity[1]	0, 1, 2, 4, 8, 16	UInt32	Returns the constraint of the signal: ▪ 0: NOT VALID ▪ 1: VALID ▪ 2: ERROR ▪ 4: NOT AVAILABLE ▪ 8: NOT DEFINED ▪ 16: OTHER
<Signalname>_Status[1]	0, 1, 2, 4, 8, 4096	UInt32	Receive status of the read signal. The following values are returned: ▪ 0: No error; data is valid, but previously read ▪ 1: Access error ▪ 2: Signal was not received ▪ 4: Signal is not valid (signal validity status ≠ 'VALID') ▪ 8: CRC not ok ▪ 4096: Data of the signal is new or not read
<Signalname>_UpdateBit	0 / 1	Boolean	Provides the read signal update bit value: ▪ 0: The PDU was not received ▪ 1: The PDU was received

[1] The port is available only if it is activated in the FlexRay Configuration Tool. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 📖).

Related topics

Basics

Sending and Receiving Signal- and Signal-Group-Specific Update Bits (Model Interface Package for Simulink - Modeling Guide 🕮)

HowTos

How to Send or Receive Signals of PDUs (Model Interface Package for Simulink - Modeling Guide 🕮)

References

Block Description (FLEXRAYCONFIG PDU RX)... 38

PDU Page (FLEXRAYCONFIG PDU RX)

Purpose

To display frame or PDU information.

Dialog settings

Information type Displays whether the data is interpreted as a PDU or a frame. If the information type is PDU, the name is the PDU name. If the information type is frame, the name is the frame name. In FIBEX+, FIBEX 3.x, FIBEX 4.1.x, or AUTOSAR System Templates, the information is interpreted as PDUs. In FIBEX 2.0 and lower versions, the information is interpreted as frames.

Name Displays the name of the PDU derived from the short name of the related FIBEX file or AUTOSAR system description file.

Send ECU name Displays the short name of the ECU which sends the PDU. The short name is derived from the FIBEX file or AUTOSAR system description file.

Parent ECU name Displays the short name of the ECU to which the currently displayed PDU is assigned.

FlexRay channel Displays the channel on which the PDU is sent. The following values are possible:

- A
- B
- A & B

Search identical blocks Searches for an identical block in the model. For example, if the PDU is configured for channel A, you can search the same PDU configured for channel B. If identical blocks are found, the MATLAB Command Window contains hyperlinks to them.

Length Displays the PDU length in bytes, which is specified in the FIBEX file or AUTOSAR system description file.

Cyclic timing Displays the list of cyclic timings of the PDU. If sub-PDUs are defined in the FIBEX file or AUTOSAR system description file, all the subitems are listed here as well.

- Multiplexer switch

  Displays the switch codes of a sub-PDU or a subframe.
- Cyclic timing

  Displays the cyclic timing in seconds. The cyclic timing for event-controlled PDUs or frames is 0. If no timing is defined for a sub-frame, the timing value of its parent frame is displayed. If no timing is defined for a sub-PDU, this sub-PDU is interpreted as an event-controlled PDU and has the cyclic timing value 0.
- Timing type

  Displays the timing type of the PDU or frame. The following values are possible:

Timing Type	Description
SC	Static cyclic
DE	Dynamic event
DC	Dynamic cyclic
DEC	Dynamic event cyclic
SDE	Subframe dynamic event
SDC	Subframe dynamic cyclic
SDEC	Subframe dynamic event cyclic
SSC	Subframe static cyclic

Absolutely scheduled timing Displays the list of absolutely scheduled timings of the PDU. The list shows the timing, FlexRay channel and controller name for each PDU.

Related topics

References

Block Description (FLEXRAYCONFIG PDU RX).. 38

Signal Page (FLEXRAYCONFIG PDU RX)

Purpose

To display the settings of the signals of the PDU.

Dialog settings

Signal selection Lets you select a signal to be displayed.

Name Displays the short name of the selected signal.

Coded bit length Displays the signal length in bits.

Physical data type Displays the physical data type of the selected signal, which is defined in the FIBEX file or AUTOSAR system description file or configured in the FlexRay Configuration Tool. The radio button indicates whether the data type is selected. You can configure the data type mode of a signal (physical or coded) via the FlexRay Configuration Tool.

Coded data type Displays the coded data type of the signal which is defined in the FIBEX file or AUTOSAR system description file. The radio button indicates whether the data type is selected. You can configure the data type of a signal via the FlexRay Configuration Tool.

Constraint Displays the valid ranges of a signal:

- Index
- Validity
- Coded [Min...Max]
- Physical [Min...Max]

The Validity short name and the Min/Max values are derived from the FIBEX file or AUTOSAR system description file. If no Min/Max value pair is defined in the FIBEX file or AUTOSAR system description file, the field remains empty.

Computation method Displays the computation methods of a signal:

- Conv. Type
- Conv. Offset
- Conv. Factor
- Unit
- Text

Conv. Factor and Conv. Offset are defined in the following form:

```
y = Conv.Factor · X + Conv.Offset
```

Related topics

References

Block Description (FLEXRAYCONFIG PDU RX)... 38

PDU Option Page (FLEXRAYCONFIG PDU RX)

Purpose

To display information about the PDU block interface and configuration.

Dialog settings

Input port configuration Displays the input ports of the PDU-specific signal. The input ports are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names.

Output port configuration Displays the output ports of the PDU-specific signal. The output ports are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names.

Update bit is ignored Displays whether the update bit for the PDU is ignored. You can configure this value via the FlexRay Configuration Tool.

HW Enable static frame(s) after modeling start (HW) Displays whether a frame is activated in hardware between the simulation start and the first calculation. You can configure this value via the FlexRay Configuration Tool.

SW Enable static frame after modeling start (SW) Displays whether a frame is activated in software between the simulation start and the first calculation. You can configure this value via the FlexRay Configuration Tool.

Max. number of raw data bytes Displays the maximum number of raw data bytes to be read. This value must be smaller than or equal to the payload length of the PDU. You can configure this value via the FlexRay Configuration Tool to enhance processor performance and simplify modeling.

HW disable affected PDUs Displays the PDUs which could be affected when this PDU is disabled by hardware because they share the selected slot(s).

Related topics

Basics

Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Guide 📖)

References

Block Description (FLEXRAYCONFIG PDU RX)... 38

Signal Option Page (FLEXRAYCONFIG PDU RX)

Purpose

To display the signal interfaces and configuration.

Dialog settings

Input port configuration Displays the signal inputs that are available for the PDU block. The signal inputs are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names. The inputs are displayed in a tree view. If a signal is included in subframes or sub-PDUs, its associated multiplexed switch codes are also displayed after the signal name.

Output port configuration Displays the signal outputs that are available for the PDU block. The signal outputs are the block interfaces of the PDU block and are displayed as MATLAB/Simulink port names. The outputs are displayed in a tree view. If a signal is included in subframes or sub-PDUs, its associated multiplexed switch codes are also displayed after the signal name.

Related topics

References

Block Description (FLEXRAYCONFIG PDU RX)... 38

FLEXRAYCONFIG TIMETABLE TASK

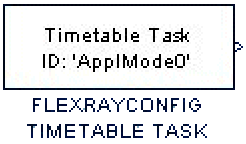
Purpose

To assign a task to a timetable.

> **Note**
>
> This block is obsolete. It is only provided for compatibility reasons.

Block Description (FLEXRAYCONFIG TIMETABLE TASK)

Block



Purpose

To assign a task to a timetable.

> **Note**
>
> This block is obsolete. As of FlexRay Configuration Blockset 2.0, the block is replaced by a Runnable Function block from the Model Interface Blockset.
>
> Runnable Function blocks are generated according to the Simulink configuration data from the FlexRay Configuration Tool and added to the automatically generated FlexRay model during the generation process. They are used to make the tasks configured in the FlexRay Configuration Tool available to the real-time simulation. The Runnable Function blocks are configured for your FlexRay network. You should not reconfigure these blocks manually. Particularly the preconfigured name must not be changed, since otherwise the mapping for the task handling in ConfigurationDesk will not work automatically.
>
> For further information on Runnable Function blocks, refer to Model Interface Blockset (Model Interface Package for Simulink Reference 📖).

Dialog settings

None

FlexRay Configuration Commands

dsfr_modelgenerate

Syntax

```
errorcode = dsfr_modelgenerate('FileName'[, 'Parameter', 'Value'[, ...]])
```

Purpose

To generate or update configured FlexRay blocks from the Simulink configuration data.

Generation

The FlexRay Configuration Tool generates Simulink configuration data. The `dsfr_modelgenerate` command uses the configuration data to generate a Simulink model, the automatically generated FlexRay model. The model contains all the FlexRay blocks configured for your FlexRay communication. For more information on the blockset, refer to Overview of an Automatically Generated FlexRay Model on page 11 and Structure of the Automatically Generated FlexRay Model on page 12.

Update

You can specify `Update` as a parameter of the `dsfr_modelgenerate` command to update an automatically generated FlexRay model. After using the update method, you can perform actions such as adding a FlexRay block to your Simulink model or exchanging FlexRay blocks manually. This is especially useful for minor changes because updating via the FLEXRAYCONFIG UPDATE block is time-consuming.

> **Note**
>
> During the generation process a MAT file is generated. It has the same name as the corresponding M file and is located in the same folder. The MAT file contains all the information required by the update process. Do not change or remove the MAT file, as otherwise the update process fails.

Parameters

FileName Name of the Simulink configuration data file. If the file is not stored in the MATLAB search path, you are prompted for the complete path.

Parameter, Value Optional parameter and its value, see the following table:

Parameter	Possible Values	Description
GenerationMode	▪ New ▪ Update The default value is 'New'.	'New': The command generates the automatically generated FlexRay model on the basis of the configuration data. 'Update': The command updates the automatically generated FlexRay model.
SortOrder	▪ PDUName ▪ PDUTiming The default value is 'PDUName'.	Specifies the sort order of the generated PDU blocks.

Examples

Generating an automatically generated FlexRay model Enter

```
dsfr_modelgenerate('MyProject_data.m')
```

or

```
dsfr_modelgenerate('MyProject_data.m', 'GenerationMode', 'New')
```

in the MATLAB Command Window to generate an automatically generated FlexRay model based on the configuration data created by the FlexRay Configuration Tool. `MyProject_data` is the file name of the Simulink configuration data file.

Updating the automatically generated FlexRay model Enter

```
dsfr_modelgenerate('MyProject_data.m', 'GenerationMode', 'Update')
```

in the MATLAB Command Window to update the automatically generated FlexRay model.

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

Related topics

HowTos

How to Generate Blocks for Modeling a FlexRay Communication (Model Interface Package for Simulink - Modeling Guide 📖)

References

FLEXRAYCONFIG UPDATE.. 16