RTI FlexRay Configuration Blockset

Reference

For RTI 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 RTI Reference provides a full description of the RTI FlexRay Configuration Blockset. The blockset is used to connect a dSPACE real-time system to a FlexRay bus. It has two parts:

- The RTI FlexRay Configuration Blockset contains RTI blocks to model a FlexRay node under Simulink. The root level of the library contains the blocks for analyzing and handling the FlexRay network.
- The Blocks for Configuration Generation library contains RTI FlexRay blocks which are used in the generation process. They are the basis for the automatically generated FlexRay model and cannot be used directly.

> **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.

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.

Symbol	Description
?	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.

Examples:

- Where you find terms such as `rti<XXXX>` replace them by the RTI platform support you are using, for example, `rti1007`.
- Where you find terms such as `<model>` or `<submodel>` in this document, replace them by the actual name of your model or submodel. For example, if the name of your Simulink model is `smd_1007_sl.slx` and you are asked to edit the `<model>_usr.c` file, you actually have to edit the `smd_1007_sl_usr.c` file.

RTI block name conventions All I/O blocks have default names based on dSPACE's board naming conventions:

- Most RTI block names start with the board name.
- A short description of functionality is added.
- Most RTI block names also have a suffix.

Suffix	Meaning
B	Board number (for PHS-bus-based systems)
M	Module number (for MicroAutoBox II)
C	Channel number
G	Group number
CON	Converter number
BL	Block number
P	Port number
I	Interrupt number

A suffix is followed by the appropriate number. For example, DS2201IN_B2_C14 represents a digital input block located on a DS2201 board. The suffix indicates board number 2 and channel number 14 of the block. For more general block naming, the numbers are replaced by variables (for example, DS2201IN_Bx_Cy).

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 RTI FlexRay Configuration Blockset

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

Library layout / model

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

The blockset consists of different parts.

RTI FlexRay Configuration Blockset The blockset contains RTI blocks to model a FlexRay node under Simulink. The root level of the library contains the blocks for analyzing and handling the FlexRay network. Refer to Overview of the RTI FlexRay Configuration Blockset on page 13.

Blocks for Configuration Generation The blockset contains RTI FlexRay blocks which are used in the generation process. They are the basis for the automatically generated FlexRay model and cannot be used directly. Refer to Overview of the Blocks for Configuration Generation on page 15.

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 16.

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 several dSPACE platforms and FlexRay modules. For a list of the supported hardware, refer to Connecting Real-Time Systems to the FlexRay Bus (FlexRay Configuration Features 🕮).

Demo model

For Simulink models that show how to use the FlexRay RTI blocks, refer to the RTI demo library of the RTI FlexRay Configuration Blockset.

Related topics

Basics

Setting up a FlexRay Network (FlexRay Configuration Features 🕮)

References

Overview of the Blocks for Configuration Generation.. 15
Overview of the RTI FlexRay Configuration Blockset.. 13

Overview of the RTI FlexRay Configuration Blockset

Introduction

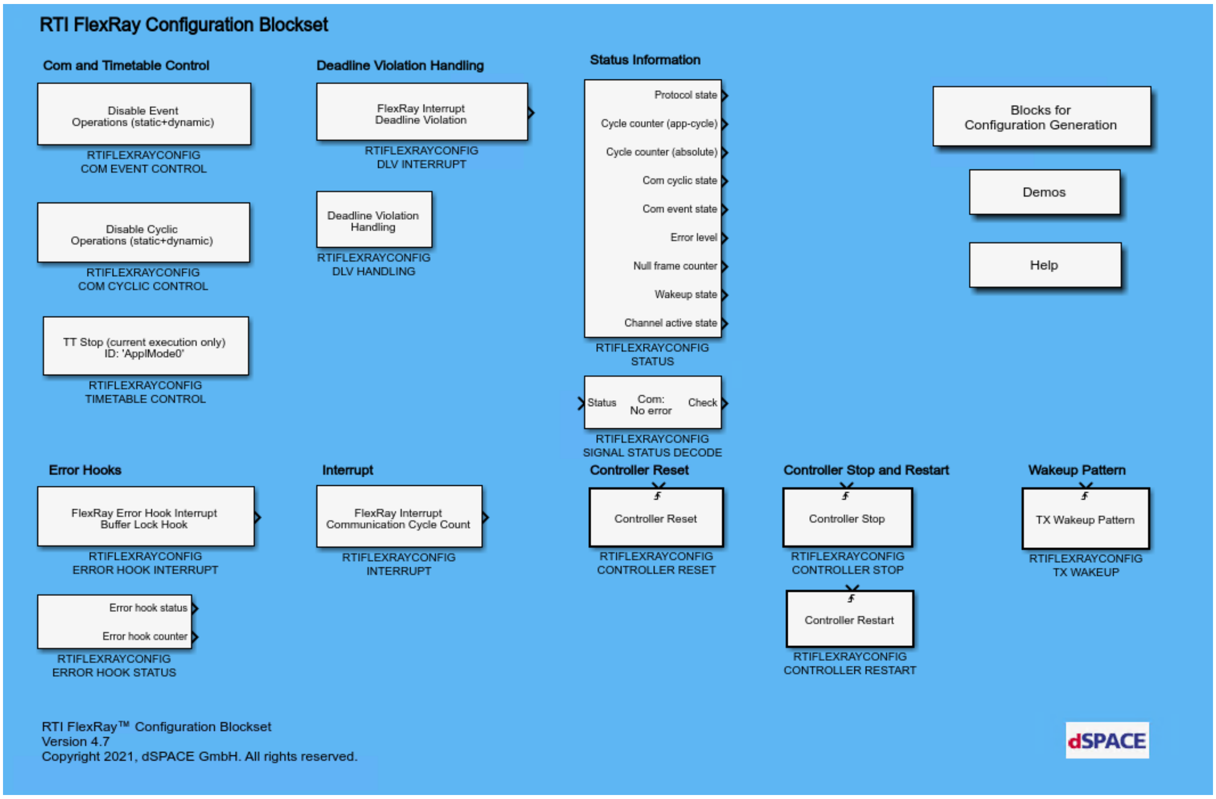
This topic gives you a short description of the RTI FlexRay Configuration Blockset and its block library.

RTI FlexRay Configuration Blockset

The RTI FlexRay Configuration Blockset is a Simulink blockset for simulating a FlexRay node with a real-time system.

Library access

In the MATLAB Command Window, enter `rtiflexrayconfig` or `rtifrconf`. The block library of the RTI FlexRay Configuration Blockset is displayed. You can also open the blockset via the DS1006, DS1007, or MicroAutoBox Blockset by clicking: Blockset - RTI FR CONF Blockset.


RTI FlexRay Configuration Blockset
Com and Timetable Control
Disable Event Operations (static+dynamic)
RTIFLEXRAYCONFIG COM EVENT CONTROL
Disable Cyclic Operations (static+dynamic)
RTIFLEXRAYCONFIG COM CYCLIC CONTROL
TT Stop (current execution only) ID: 'ApplMode0'
RTIFLEXRAYCONFIG TIMETABLE CONTROL
Deadline Violation Handling
FlexRay Interrupt Deadline Violation
RTIFLEXRAYCONFIG DLV INTERRUPT
Deadline Violation Handling
RTIFLEXRAYCONFIG DLV HANDLING
Status Information
Protocol state
Cycle counter (app-cycle)
Cycle counter (absolute)
Com cyclic state
Com event state
Error level
Null frame counter
Wakeup state
Channel active state
RTIFLEXRAYCONFIG STATUS
Status
Com: No error
Check
RTIFLEXRAYCONFIG SIGNAL STATUS DECODE
Blocks for Configuration Generation
Demos
Help
Error Hooks
FlexRay Error Hook Interrupt Buffer Lock Hook
RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT
Error hook status
Error hook counter
RTIFLEXRAYCONFIG ERROR HOOK STATUS
Interrupt
FlexRay Interrupt Communication Cycle Count
RTIFLEXRAYCONFIG INTERRUPT
Controller Reset
Controller Reset
RTIFLEXRAYCONFIG CONTROLLER RESET
Controller Stop and Restart
Controller Stop
RTIFLEXRAYCONFIG CONTROLLER STOP
Controller Restart
RTIFLEXRAYCONFIG CONTROLLER RESTART
Wakeup Pattern
TX Wakeup Pattern
RTIFLEXRAYCONFIG TX WAKEUP
RTI FlexRay™ Configuration Blockset
Version 4.7
Copyright 2021, dSPACE GmbH. All rights reserved.
dSPACE

RTI blocks For details on the RTI blocks, refer to RTI FlexRay Configuration Blockset on page 21.

Blocks for Configuration Generation Opens the sublibrary with the blocks for configuration generation, refer to Overview of the Blocks for Configuration Generation on page 15.

Demos Opens a sublibrary with the available example models.

Help Displays this reference information.

Related topics

References

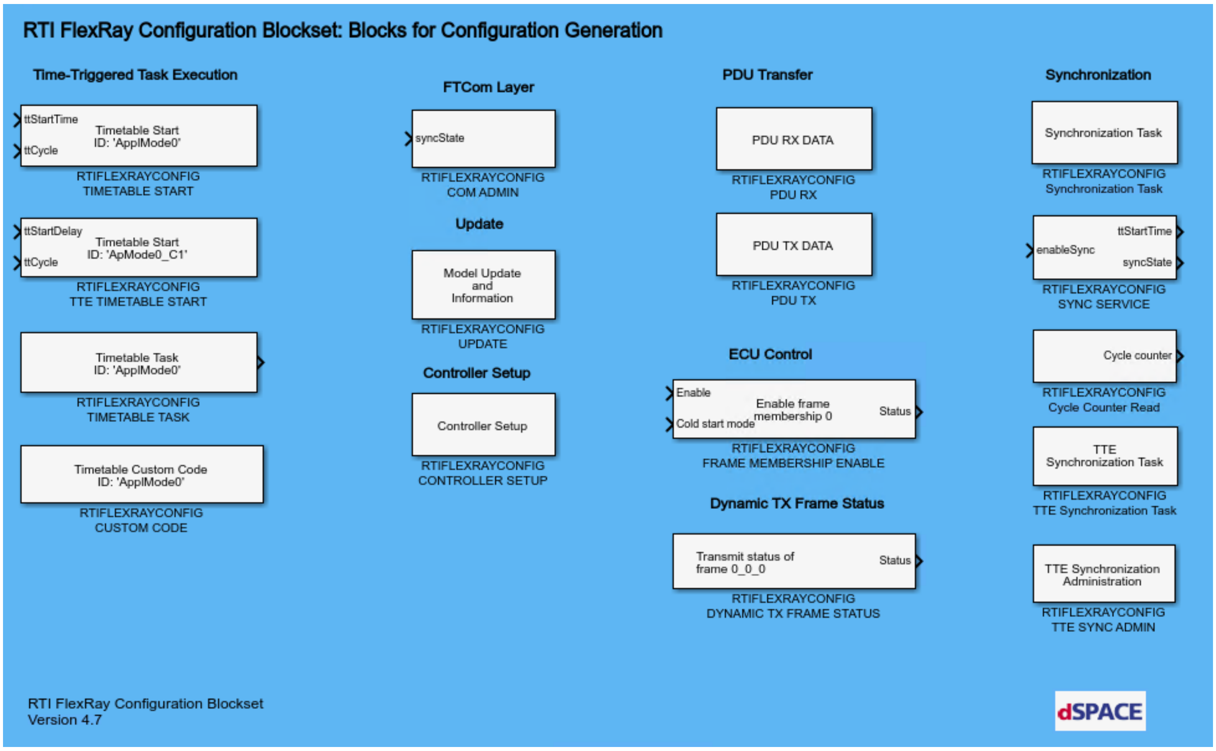
Com and Timetable Control Blocks..22
Error Hook Blocks and Deadline Violation Handling........................29
Interrupt and Controller Reset Blocks....................................48
Status Information and Wakeup Pattern Blocks.............................38

Overview of the Blocks for Configuration Generation

Blocks for configuration generation

The blocks of this library cannot be used directly. They are used by the generation process. The FlexRay Configuration Tool generates code for your FlexRay configuration (Simulink configuration data). The `rtiflexrayconfig_modelgenerate` command uses the Simulink configuration data to generate a Simulink model containing the configured blocks (see rtiflexrayconfig_modelgenerate on page 131). You can use the blocks of the generated model in your Simulink model to connect your real-time system to the FlexRay bus. For details, refer to Introduction to the FlexRay Configuration Tool (FlexRay Configuration Tool Guide 🕮).

The following illustration shows the blocks for configuration generation.


RTI FlexRay Configuration Blockset: Blocks for Configuration Generation
Time-Triggered Task Execution
ttStartTime
ttCycle
Timetable Start ID: 'ApplMode0'
RTIFLEXRAYCONFIG TIMETABLE START
ttStartDelay
ttCycle
Timetable Start ID: 'ApMode0_C1'
RTIFLEXRAYCONFIG TTE TIMETABLE START
Timetable Task ID: 'ApplMode0'
RTIFLEXRAYCONFIG TIMETABLE TASK
Timetable Custom Code ID: 'ApplMode0'
RTIFLEXRAYCONFIG CUSTOM CODE
FTCom Layer
syncState
RTIFLEXRAYCONFIG COM ADMIN
Update
Model Update and Information
RTIFLEXRAYCONFIG UPDATE
Controller Setup
Controller Setup
RTIFLEXRAYCONFIG CONTROLLER SETUP
PDU Transfer
PDU RX DATA
RTIFLEXRAYCONFIG PDU RX
PDU TX DATA
RTIFLEXRAYCONFIG PDU TX
ECU Control
Enable
Cold start mode
Enable frame membership 0
Status
RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE
Dynamic TX Frame Status
Transmit status of frame 0_0_0
Status
RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS
Synchronization
Synchronization Task
RTIFLEXRAYCONFIG Synchronization Task
enableSync
ttStartTime
syncState
RTIFLEXRAYCONFIG SYNC SERVICE
Cycle counter
RTIFLEXRAYCONFIG Cycle Counter Read
TTE Synchronization Task
RTIFLEXRAYCONFIG TTE Synchronization Task
TTE Synchronization Administration
RTIFLEXRAYCONFIG TTE SYNC ADMIN
RTI FlexRay Configuration Blockset Version 4.7
dSPACE

Related topics

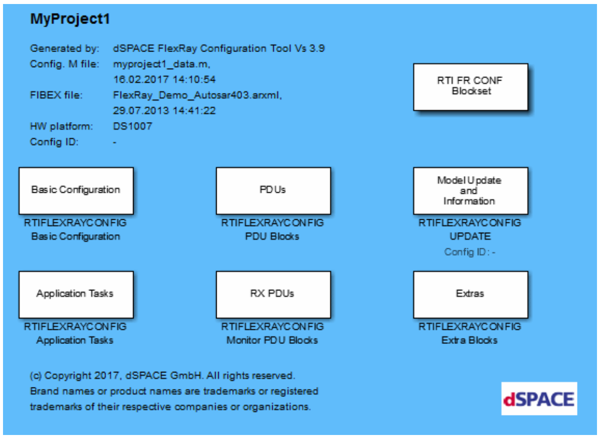
References

Blocks for Configuration Generation.. 59

Overview of an Automatically Generated FlexRay Model

Automatically generated FlexRay model

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



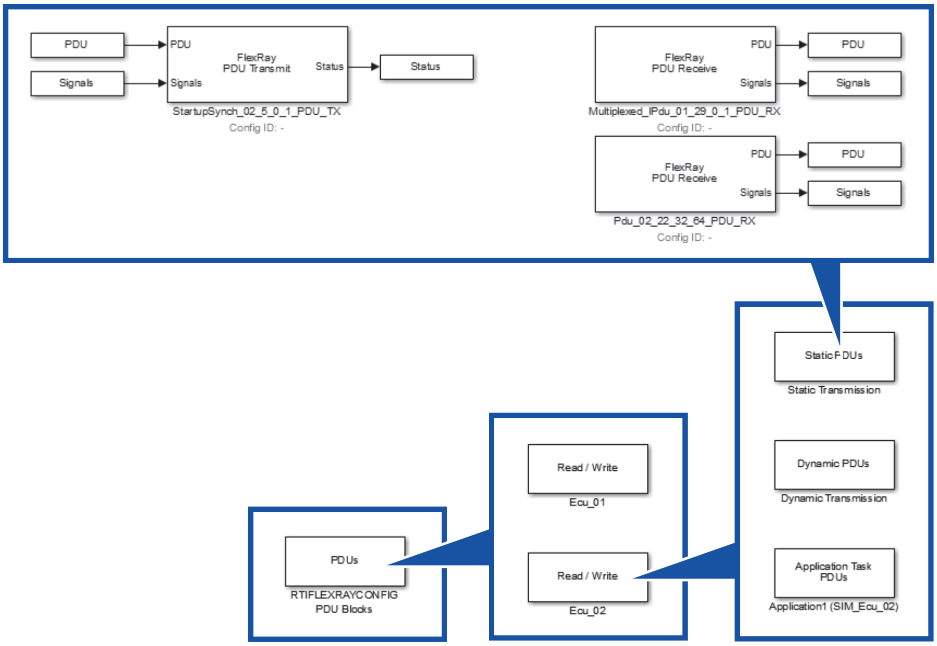
MyProject1

Generated by: dSPACE FlexRay Configuration Tool Vs 3.9
Config. M file: myproject1_data.m,
16.02.2017 14:10:54
FIBEX file: FlexRay_Demo_Autosar403.arxml,
29.07.2013 14:41:22
HW platform: DS1007
Config ID: -

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

RTI FR CONF Blockset
Basic Configuration
RTIFLEXRAYCONFIG Basic Configuration
PDUs
RTIFLEXRAYCONFIG PDU Blocks
Model Update and Information
RTIFLEXRAYCONFIG UPDATE
Config ID: -
Application Tasks
RTIFLEXRAYCONFIG Application Tasks
RX PDUs
RTIFLEXRAYCONFIG Monitor PDU Blocks
Extras
RTIFLEXRAYCONFIG Extra Blocks
dSPACE

The model contains subsystems with the configured RTI FlexRay blocks. All the generated FlexRay blocks are structured in such subsystems. The following illustration shows an example of the hierarchy of generated subsystems.



Related topics

HowTos

How to Generate RTI Blocks for Designing a FlexRay Node (FlexRay Configuration Features 🕮)

References

rtiflexrayconfig_modelgenerate..131
Structure of the Automatically Generated FlexRay Model..17

Structure of the Automatically Generated FlexRay Model

Structure

The following table shows the structure of a FlexRay model, with the names of the blocks and subsystems (the RTIFLEXRAYCONFIG 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 RTI Block**
1st	**2nd**	**3rd**	**4th**	
Basic Configuration				This subsystem contains all the blocks for basic configuration.
	Controller			This subsystem contains all the blocks required to set up the FlexRay controller.
		CONTROLLER SETUP Controller name: [Name]		See RTIFLEXRAYCONFIG CONTROLLER SETUP on page 76.
	Com Tasks			This subsystem contains all the communication tasks. See also Automatic Task Creation (FlexRay Configuration Tool Reference 🕮).
		ComTaskStaticFetch		See RTIFLEXRAYCONFIG CUSTOM CODE on page 66.
		ComTaskStaticDeliver		See RTIFLEXRAYCONFIG CUSTOM CODE on page 66.
		ComTaskDynamicFetch		See RTIFLEXRAYCONFIG CUSTOM CODE on page 66.
		ComTaskDynamicDeliver		See RTIFLEXRAYCONFIG CUSTOM CODE on page 66.
	Synchronization Task			This subsystem contains the blocks and subsystem for synchronization.
		SYNCHRONIZATION TASK		See RTIFLEXRAYCONFIG SYNC SERVICE on page 115.
		Synchronization Subsystem		See Synchronization Task Subsystem on page 120. If multiple buses are enabled and the configuration ID is 1, 2 or 3, see TTE Synchronization Task Subsystem on page 127.

Level				**Description / Related RTI Block**
1st	**2nd**	**3rd**	**4th**	
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 RTI blocks for sending/receiving static signals in a PDU.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU TX on page 93.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU RX on page 79.
		Dynamic Transmission		This subsystem contains all the RTI blocks for sending/receiving dynamic signals in a PDU.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU TX on page 93.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU RX on page 79.
		Network management		This subsystem contains all the PDUs for sending/receiving signals for network management.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU TX on page 93.
			<PDU-/Frame-Name>_X_Y_Z_PDU_RX_Ch	PDU block for receiving static signals. ▪ X = ID (SlotID) ▪ Y = BC (BaseCycle)

Level				Description / Related RTI Block
1st	2nd	3rd	4th	
				▪ Z = CR (CycleRepetition) ▪ Ch = Channel (A, B, AB) (only for dual-channel FIBEX or AUTOSAR system description files) See RTIFLEXRAYCONFIG PDU RX on page 79.
		<Application_Task_Name> (SIM_<ECU_Name>)		This subsystem contains all the PDUs for sending static signals to or from an application task with the name <Application_Task_Name>. This subsystem exists for each application task which sends signals.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU TX on page 93.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU RX on page 79.
UPDATE				This block can be used to update the model. See RTIFLEXRAYCONFIG UPDATE on page 70.
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.
	<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 signals by an application task with the name <Application_Task_Name>. This subsystem exists for each application task which monitors signals.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU RX on page 79.

Level				**Description / Related RTI Block**
1st	**2nd**	**3rd**	**4th**	
		Monitor_RX (MON_<ECU_Name>)		This subsystem contains all the PDUs which can be monitored by the ECU.
			<PDU-/Frame-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 RTIFLEXRAYCONFIG PDU RX on page 79.
Extra Blocks				This subsystem contains some extra blocks.
	DYNAMIC TX FRAME STATUS			See RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS on page 113.
	FRAME MEMBERSHIP ENABLE			See RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE on page 109.

Related topics

References

Overview of an Automatically Generated FlexRay Model.. 16

RTI FlexRay Configuration Blockset

Blocks of the RTI FlexRay Configuration Blockset

You can use the blocks of the RTI FlexRay Configuration Blockset to model a control for your FlexRay network.

Where to go from here

Information in this section

Information in other sections

Com and Timetable Control Blocks

Introduction

The RTI FlexRay Configuration Blockset contains blocks for controlling the execution of the Com (communication) layer and timetable. Event-based (dynamic) and cyclic frames are controlled by separate blocks. The blocks can be combined to control all the Com layers.

Where to go from here

Information in this section

Information in other sections

How to Control the Communication Layer (FlexRay Configuration Features 📖)
You can control the execution of event-based and cyclic frames of the communication layer.

RTIFLEXRAYCONFIG COM EVENT CONTROL

Purpose

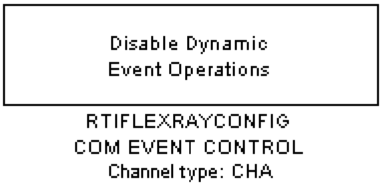
To control the execution of event-based (dynamic) frames of the communication layer.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG COM EVENT CONTROL)

Block



Purpose

To control the execution of event-based frames of the Com layer.

Description

Use this block to control the event-based (dynamic) frames of the communication layer. The block defines the operation mode for the RTIFLEXRAYCONFIG COM ADMIN block. You can configure cyclic and event-based frames separately using the RTIFLEXRAYCONFIG COM EVENT CONTROL and the RTIFLEXRAYCONFIG COM CYCLIC CONTROL block.

You can specify the operation mode for channel A, channel B, and both channels.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG COM EVENT CONTROL) on page 23

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13
RTIFLEXRAYCONFIG COM ADMIN.. 68
RTIFLEXRAYCONFIG COM CYCLIC CONTROL.. 24

Unit Page (RTIFLEXRAYCONFIG COM EVENT CONTROL)

Purpose

To specify the operation mode and select the channel type.

Dialog settings

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority

of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 🕮).

Operation mode Lets you select one of the following operation modes:

- Disable dynamic event operations
- Enable dynamic event operations
- Disable dynamic event reception
- Enable dynamic event reception
- Disable dynamic event transmission
- Enable dynamic event transmission

Channel type Lets you choose the channel type for which the operation mode is selected (channel A, channel B, or both channels).

Related topics

References

Block Description (RTIFLEXRAYCONFIG COM EVENT CONTROL).. 23

RTIFLEXRAYCONFIG COM CYCLIC CONTROL

Purpose

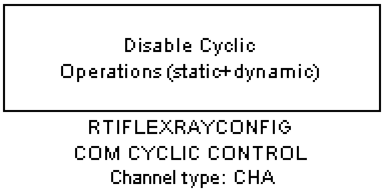
To control the execution of cyclic (static and dynamic) frames of the communication layer.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG COM CYCLIC CONTROL)

Block



Disable Cyclic
Operations (static+dynamic)

RTIFLEXRAYCONFIG
COM CYCLIC CONTROL
Channel type: CHA

Purpose

To control the execution of cyclic frames of the Com layer.

Description

Use this block to control the cyclic frames of the communication layer. The block defines the operation mode for the RTIFLEXRAYCONFIG COM ADMIN block. You can configure cyclic and event-based (dynamic) frames separately using the RTIFLEXRAYCONFIG COM EVENT CONTROL and the RTIFLEXRAYCONFIG COM CYCLIC CONTROL block.

You can specify the operation mode for channel A, channel B, and both channels.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG COM CYCLIC CONTROL) on page 25

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13
RTIFLEXRAYCONFIG COM ADMIN.. 68
RTIFLEXRAYCONFIG COM EVENT CONTROL.. 22

Unit Page (RTIFLEXRAYCONFIG COM CYCLIC CONTROL)

Purpose

To specify the operation mode and select the channel type.

Dialog settings

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority

of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖).

Operation mode Lets you select one of the following operation modes:

- Disable cyclic operations (static + dynamic)
- Enable cyclic operations (static + dynamic)
- Disable cyclic reception (static + dynamic)
- Enable cyclic reception (static + dynamic)
- Disable cyclic transmission (static + dynamic)
- Enable cyclic transmission (static + dynamic)

Channel type Lets you choose the channel type for which the operation mode is selected (channel A, channel B, or both channels).

> **Note**
>
> When using only one buffer to send TX PDUs for both channels in a dual-channel FlexRay system, you have to note the following rules if the operation mode is configured for a single channel:
>
> - If a disable operation mode is specified for one channel, the same operation mode is used for both channels together (AB), regardless of the operation mode specified for the other channel.
> - If an enable operation mode is specified for one channel, the same operation mode is used for both channels together (AB), if this operation mode is also specified or already active for the other channel.

Related topics

References

Block Description (RTIFLEXRAYCONFIG COM CYCLIC CONTROL)..25

RTIFLEXRAYCONFIG TIMETABLE CONTROL

Purpose

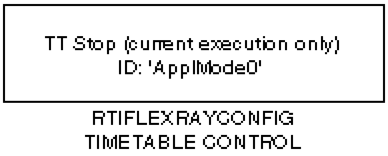
To control the execution of a timetable task.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG TIMETABLE CONTROL)

Block



Purpose

To control the execution of a timetable task.

> **Note**
>
> For each RTIFLEXRAYCONFIG TIMETABLE CONTROL block, there must be at least one corresponding RTIFLEXRAYCONFIG TIMETABLE START block with the same timetable ID.

Description

The referenced timetable is started by an RTIFLEXRAYCONFIG TIMETABLE START block during run time. Depending on the operation mode set, the RTIFLEXRAYCONFIG TIMETABLE CONTROL block does the following:

- Stops the execution of a specific timetable. All tasks of the timetable which are not in the running or suspended state (preempted tasks) are discarded. The timetable is automatically started again the next time the RTIFLEXRAYCONFIG TIMETABLE START block is executed for it.
- Stops the execution of a specific timetable permanently. This option has the same functionality as the option described above, except that all subsequent starts for the timetable are disabled. That is, even if the corresponding RTIFLEXRAYCONFIG TIMETABLE START block is executed, the timetable is not started again.
- Enables the execution of a specific timetable which has been stopped permanently. After the timetable has been enabled, executing an RTIFLEXRAYCONFIG TIMETABLE START block starts it again.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG TIMETABLE CONTROL) on page 28

Related topics

References

Unit Page (RTIFLEXRAYCONFIG TIMETABLE CONTROL)

Purpose

To set the parameters for the RTIFLEXRAYCONFIG TIMETABLE CONTROL block.

Dialog settings

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 🕮).

Timetable ID Displays the name of the timetable currently assigned to the task. The string length can be in the range 1 ... 10. The valid characters are "a ... z", "A ... Z", "0 ... 9" and "_". The Timetable ID parameter is read-only.

Operation mode Lets you choose the operation mode of this block:

Operation Mode	Description
Timetable stop (stop current execution only)	Stops the execution of a specific timetable.
Timetable stop and timetable start disable	Stops the execution of a specific timetable permanently.
Timetable start enable	Enables the execution of a specific timetable which has been stopped permanently.

For detailed description of the operation modes, refer to Block Description (RTIFLEXRAYCONFIG TIMETABLE CONTROL) on page 27.

Related topics

References

Error Hook Blocks and Deadline Violation Handling

Introduction

You can use the following blocks to implement error handling and for deadline violation handling.

Where to go from here

Information in this section

Information in other sections

How to Implement Error Handling (FlexRay Configuration Features 📖)
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.

RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT

Purpose

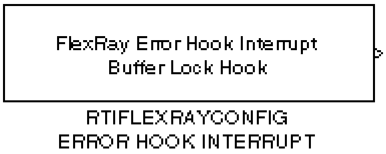
To make FlexRay error hooks available as trigger sources.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT)

Block



Purpose

To make FlexRay error hooks available as trigger sources.

Description

During execution of communication code, several error conditions may occur. The **RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT** block allows you to use the error hooks to call a subsystem for error handling. The error type which triggers the interrupt can be selected on the **Unit** page. You can select whether the interrupt is triggered buffer-specifically or not on the **Options** page. You can use the **RTIFLEXRAYCONFIG ERROR HOOK STATUS** block to get information on the number and type of errors that occurred.

Dialog settings

The dialog settings can be specified on the following pages:

- Options Page (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT) on page 31
- Unit Page (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT) on page 31

Related topics

References

Unit Page (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT)

Purpose

To specify the error hook type and the controller name.

Dialog settings

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖).

Error hook type Lets you choose the error hook type:

- Buffer lock hook
- Buffer full hook
- Local buffer corrupted hook
- Invalid frame hook
- Trigger interrupt on any hook

Controller name Lets you choose the name of the controller to set the error handling for.

Related topics

References

Block Description (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT).. 30

Options Page (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT)

Purpose

To specify buffer-specific error handling

Dialog settings

Enable FlexRay buffer specific error hook handling Indicates whether a buffer-specific error hook is enabled or not. If it is enabled, you must specify the buffer ID.

Buffer id Lets you select the buffer ID. It is available only if the error handling is buffer-specific. The range depends on the type of the controller type. The table shows the standard values of the ranges:

Controller Type	Range of Buffer ID
MFR4200	0 ... 58
MFR4300 (used on DS4340)	0 ... 127

The dialog cannot check whether the buffer ID specified for a MFR4200 is out of range. In the build process, it is checked whether the buffer ID is in the actual range of buffer IDs of the selected controller type.

Related topics

References

Block Description (RTIFLEXRAYCONFIG ERROR HOOK INTERRUPT)... 30

RTIFLEXRAYCONFIG ERROR HOOK STATUS

Purpose

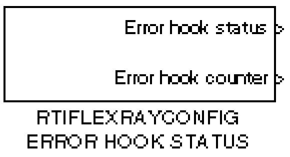
To get information on the number and type of FlexRay errors that occurred.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG ERROR HOOK STATUS)

Block



Purpose

To get information on the number and type of FlexRay errors that occurred.

Description

This block gives you status information on the FlexRay errors delivered by the Com code. It outputs the FlexRay errors of all FlexRay buses specified in the real-time model. A block output indicates the type of the last FlexRay error which occurred since the previous block execution. Another block output shows the number of FlexRay errors that occurred since the start of the real-time application.

I/O characteristics

The table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Error hook status	0, 1, 2, 3 , 4	UInt16	Type of the last error: ▪ 0: No error hook detected ▪ 1: Buffer locked error ▪ 2: Buffer full error ▪ 3: Local buffer corrupted ▪ 4: Invalid frame error
Error hook counter	0, 1, ... 2^{32}-1	UInt32	Number of errors detected

Dialog settings

The dialog settings can be specified on the following pages:

- Options Page (RTIFLEXRAYCONFIG ERROR HOOK STATUS) on page 34
- Unit Page (RTIFLEXRAYCONFIG ERROR HOOK STATUS) on page 33

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13

Unit Page (RTIFLEXRAYCONFIG ERROR HOOK STATUS)

Purpose

To get information on FlexRay-specific errors.

Dialog settings

No parameters

Related topics

References

Block Description (RTIFLEXRAYCONFIG ERROR HOOK STATUS).. 32

Options Page (RTIFLEXRAYCONFIG ERROR HOOK STATUS)

Purpose

To set the sample time for the RTIFLEXRAYCONFIG ERROR HOOK STATUS block.

Dialog settings

Sample time Lets you specify the sample time for this Simulink block in seconds. The sample time defines the time interval for the Simulink block to be executed.

Sample Time	Meaning
–1	Inherited sample time
= 0	Discrete sample time of the subsystem
> 0	Discrete sample time as specified

Related topics

References

Block Description (RTIFLEXRAYCONFIG ERROR HOOK STATUS).. 32

RTIFLEXRAYCONFIG DLV INTERRUPT

Purpose

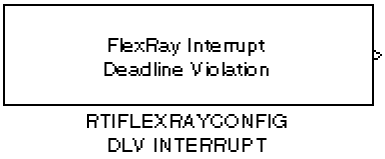
To generate a trigger signal when a deadline violation (DLV) occurred in the application task or communication task.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG DLV Interrupt)

Block



Purpose

To handle a deadline violation that occurred in the application task or communication task.

Description

The block can be used to specify an error handler for a deadline violation (DLV) that occurred in the application task or communication task. When a deadline violation occurs, the block generates an interrupt which you can use to call a subsystem.

If you use an RTIFLEXRAYCONFIG DLV INTERRUPT block, an RTIFLEXRAYCONFIG DLV HANDLING block is not necessary. The behavior is automatically set to warning.

Dialog settings

This block has no dialog settings.

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13
RTIFLEXRAYCONFIG DLV HANDLING.. 36

Unit Page (RTIFLEXRAYCONFIG DLV Interrupt)

Purpose

No purpose

Dialog settings

No parameters

Related topics

References

Block Description (RTIFLEXRAYCONFIG DLV Interrupt).. 35

RTIFLEXRAYCONFIG DLV HANDLING

Purpose

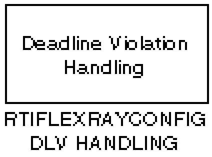
To specify the behavior of a real-time application when a deadline violation occurs.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG DLV HANDLING)

Block



Purpose

To specify the behavior of a real-time application when a deadline violation occurs.

Description

By default, a real-time application is terminated when a deadline violation (DLV) occurs. You can use this block to avoid the termination. The real-time application is then terminated or not according to your settings. A warning is displayed in the log viewer of ControlDesk. It distinguishes between a deadline violation that occurred in the synchronization task and a deadline violation that occurred in the application task or communication task.

You can specify an error handler for a deadline violation that occurred in the application task or communication task using the RTIFLEXRAYCONFIG DLV INTERRUPT block.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG DLV HANDLING) on page 37

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13
RTIFLEXRAYCONFIG DLV INTERRUPT.. 34

Unit Page (RTIFLEXRAYCONFIG DLV HANDLING)

Purpose

To specify the behavior of the RTIFLEXRAYCONFIG DLV HANDLING block.

Dialog settings

DLV of Synchronization Task Specifies how the real-time application reacts when a deadline violation (DLV) occurs in the synchronization task.

Setting	Description
terminate	The real-time application is terminated.
warn	The real-time application keeps running but a warning is issued. The warning is displayed in the log viewer.

DLV of application or communication tasks Lets you specify how the real-time application reacts when a deadline violation occurs in the application task or communication task.

Setting	Description
terminate	The real-time application is terminated.
warn	The real-time application keeps running but a warning is issued. The warning is displayed in the log viewer. This behavior is set automatically if an RTIFLEXRAYCONFIG DLV INTERRUPT block is used.

Related topics

References

Block Description (RTIFLEXRAYCONFIG DLV HANDLING).. 36
RTIFLEXRAYCONFIG DLV INTERRUPT.. 34
Synchronization Task Subsystem.. 120

Status Information and Wakeup Pattern Blocks

Introduction

You can use an RTI block to get status information on the FlexRay controller and to decode FlexRay signals. The wakeup pattern block lets you specify the basic settings and the trigger conditions for transmitting wakeup patterns on the FlexRay bus.

Where to go from here

Information in this section

Information in other sections

How to Monitor the Status of FlexRay Controllers (FlexRay Configuration Features 📖)
You can monitor the status of a FlexRay controller. The status values are available as signals in your Simulink model and can be visualized in ControlDesk.

RTIFLEXRAYCONFIG STATUS

Purpose

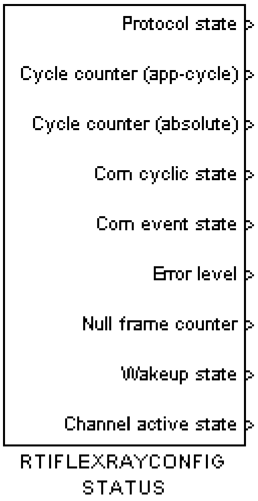
To monitor FlexRay controller specific information.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG STATUS)

Block



Purpose

To monitor FlexRay controller-specific information.

Description

The **RTIFLEXRAYCONFIG STATUS** block allows you to display information on a FlexRay controller (see I/O characteristics below). The protocol state output lets you monitor the states of a FlexRay controller. The FlexRay controller and the channel type are specified on the **Unit** page. You can enable or disable the output ports on the **Options** page.

The block outputs common controller states and channel-related states. If you want to read the channel-related states of both channels, you must use two blocks in your Simulink model.

I/O characteristics

The table shows the Simulink block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Protocol state	0 ... 52	UInt32	Displays the protocol state assigned to the following values: ▪ 0: Configuration ▪ 1: Initialize schedule ▪ 2: Normal active operation

Simulink Output	Range	Simulink Data Type	Meaning
			▪ 3: Normal passive operation ▪ 4: Integration consistency check ▪ 5: Integration listen ▪ 11: Wake up ▪ 21: Coldstart listen ▪ 22: Integration coldstart check ▪ 23: Join coldstart ▪ 24: Coldstart collision resolution ▪ 25: Coldstart consistency check ▪ 26: Coldstart gap ▪ 50: Default configuration ▪ 51: Ready ▪ 52: Halt
Cycle counter value (app-cycle)	0 ... max (see right)	UInt32	Displays the value of the FlexRay communication counter value corresponding to the application cycle. The communication counter starts with 0. The maximum value is `APPLICATION_CYCLE_DURATION/COMMUNICATION_CYCLE_DURATION - 1`.
Cycle counter value (absolute)	0 ... 63	UInt32	Displays the value of the FlexRay communication counter value corresponding to the value of the CCCVR (current cycle counter value register) of the FlexRay controller. The communication counter starts with 0. It runs independently of the application cycle.
Com cyclic state	0 ... 3	UInt32	Displays the state of the cyclic part of the communication layer. It is "on" if it is being executed or "off" if it is not currently being executed. The state is set to "off" if the FlexRay RCP node is not synchronized or if an error hook function of the layer was called. The following values describe the state of the cyclic part of the communication layer: ▪ 0: It is offline ▪ 1: Receiving is online, sending is offline ▪ 2: Receiving is offline, sending is online ▪ 3: It is online
Com event state	0 ... 3	UInt32	Displays the state of the event part of the communication layer. It is "on" if it is being executed or "off" if it is not currently being executed. The state is set to "off" if the FlexRay RCP node is not synchronized or if an error hook function of the layer was called. The following values describes the state of the event part of the communication layer: ▪ 0: It is offline ▪ 1: Receiving is online, sending is offline ▪ 2: Receiving is offline, sending is online ▪ 3: It is online
Error level	0 ... 3	UInt32	Displays the error level of the FlexRay controller depending on the FlexRay module used. Error level used by MFR4200/V9: ▪ 0: Everything is OK (green) ▪ 1: Some minor problems have occurred (yellow) ▪ 2: There is a serious problem (red) Error level used by MFR4300/V11: ▪ 0: ACTIVE (green) ▪ 1: PASSIVE (yellow) ▪ 2: COMM_HALT (red)

Simulink Output	Range	Simulink Data Type	Meaning
			Error level used by E-Ray: ▪ 0: ACTIVE (green) ▪ 1: PASSIVE (yellow) ▪ 2: COMM_HALT (red) ▪ 3: Reserved
Null frame counter	0 ... 2^{32}-1	UInt32	Displays the number of received null frames that occurred during communication. Received null frames are counted only if their ID is valid for the simulated FlexRay node. The value is incremented until the counter overflows. If the value overflows, the counter starts with 0.
Wakeup state	0 ... 6	UInt32	The following values describe the wakeup state of the controller. The wakeup state relates to wakeup patterns that are transmitted via the RTIFLEXRAYCONFIG TX WAKEUP block. ▪ 0: DSFLEXRAY_WUP_STAT_UNDEFINED The wakeup state is undefined. ▪ 1: DSFLEXRAY_WUP_STAT_REC_HEADER The communication controller has received a frame header without coding violation during the initial listen phase. ▪ 2: DSFLEXRAY_WUP_STAT_REC_WUP The communication controller has received a valid wakeup pattern during the initial listen phase. ▪ 3: DSFLEXRAY_WUP_STAT_COLL_HEADER The communication controller has detected a collision during wakeup pattern transmission by receiving a valid header during the ensuing detection phase. ▪ 4: DSFLEXRAY_WUP_STAT_COLL_WUP The communication controller has detected a collision during wakeup pattern transmission by receiving a valid wakeup pattern during the ensuing detection phase. ▪ 5: DSFLEXRAY_WUP_STAT_COLL_UNKNOWN The communication controller has detected a collision but did not detect a subsequent reception event that would allow the collision to be categorized as either DSFLEXRAY_WUP_STAT_COLL_HEADER or DSFLEXRAY_WUP_STAT_COLL_WUP. ▪ 6: DSFLEXRAY_WUP_STAT_TRANSMITTED The wakeup pattern was completely transmitted.
Channel active state	0, 1	UInt32	Displays the state of the channel: ▪ 0: Channel is inactive, i.e., no valid frame/PDU was received during the current and the previous execution of the status block. ▪ 1: Channel is active. Note that the block observes only the time between the current and the previous call. If no valid frame/PDU was received in this time, the block returns 0 even if the channel is active. It is therefore recommended to call the block only once per cycle or less.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG STATUS) on page 42
- Options Page (RTIFLEXRAYCONFIG STATUS) on page 42

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13

Unit Page (RTIFLEXRAYCONFIG STATUS)

Purpose

To specify the controller name and select the channel type.

Parameters

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖).

Controller name Lets you choose the specified FlexRay controller. See RTIFLEXRAYCONFIG CONTROLLER SETUP on page 76.

Channel type Lets you choose the channel type whose status information should be read (channel A (CHA) or channel B (CHB)).

Related topics

References

Block Description (RTIFLEXRAYCONFIG STATUS).. 39

Options Page (RTIFLEXRAYCONFIG STATUS)

Purpose

To select the status outport to be enabled and to specify the sample time.

Description

You can enable or disable the output ports of the RTIFLEXRAYCONFIG STATUS block on this page. For details on the status information, refer to Block Description (RTIFLEXRAYCONFIG STATUS) on page 39.

Parameters

Protocol State Enables the Protocol state outport, which indicates the states of the FlexRay network.

Cycle counter value (app-cycle) Enables the Cycle counter (app-cycle) outport.

Cycle counter value (absolute) Enables the Cycle counter (absolute) outport.

Error level Enables the Error level outport.

Wakeup state Enables the Wakeup state outport.

Com cyclic state Enables the Com cyclic state outport.

Com event state Enables the Com event state outport.

Null frame counter Enables the Null frame counter outport.

Channel active state Enables the Channel active state outport.

Sample Time Lets you enter the sample time (the intervals the data should be read at).

Sample Time	Meaning
–1	Inherited sample time
= 0	Discrete sample time of the subsystem
> 0	Discrete sample time as specified

Related topics

References

Block Description (RTIFLEXRAYCONFIG STATUS)..39

RTIFLEXRAYCONFIG SIGNAL STATUS DECODE

Purpose

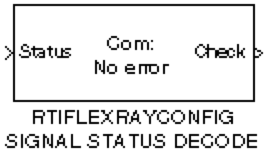
To decode the status of received FlexRay signals.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG SIGNAL STATUS DECODE)

Block



Purpose

To check the status of a received signal.

Description

This block checks the status of a received signal. You can specify a status type on the **Parameters** page. If the specified status type at the received signal occurs, the block outport is set to 1.

This block does not check whether the received signal is valid or not. The validity status of the signal is checked by the **RTIFLEXRAYCONFIG PDU RX** block.

I/O characteristics

The following table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Status	0, 1, 2, 4, 4096	UInt32	Status of the read signal which should be checked. This port must be connected to the Status outport of the **RTIFLEXRAYCONFIG PDU RX** block.

The following table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Check	0 / 1	Boolean	Displays whether the specified status type occurs ▪ 0: Status type is not occurred ▪ 1: Status type is occurred

Dialog settings

The dialog settings can be specified on the following pages:

- Parameters Page (RTIFLEXRAYCONFIG SIGNAL STATUS DECODE) on page 45

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.......... 13
RTIFLEXRAYCONFIG PDU RX.......... 79

Parameters Page (RTIFLEXRAYCONFIG SIGNAL STATUS DECODE)

Purpose

To specify the status type.

Dialog settings

Status type Specifies the status type which should be checked.

- Com: No error
- Com: Message not received
- Com: Access error
- Signal: Last correct data not read

Related topics

References

RTIFLEXRAYCONFIG TX WAKEUP

Purpose

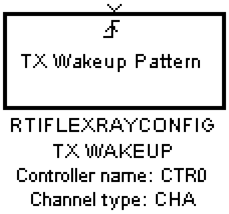
To specify the basic settings and trigger conditions for transmitting wakeup patterns on the FlexRay bus.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG TX WAKEUP)

Block



RTIFLEXRAYCONFIG
TX WAKEUP
Controller name: CTR0
Channel type: CHA

Purpose

To specify the basic settings and trigger conditions for transmitting wakeup patterns on the FlexRay bus.

Description

The **RTIFLEXRAYCONFIG TX WAKEUP** block allows you to specify the FlexRay controller which transmits the wakeup pattern on channel A or channel B and to define the trigger condition for transmitting the wakeup pattern. The FlexRay controller and the channel type are specified on the **Unit** page.

The block can send the wakeup pattern on channel A or channel B. If you want to send the wakeup pattern on both channels you must use two blocks.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG TX WAKEUP) on page 46
- Options Page (RTIFLEXRAYCONFIG TX WAKEUP) on page 47

Related topics

HowTos

How to Send Wakeup Patterns (FlexRay Configuration Features 🕮)

Unit Page (RTIFLEXRAYCONFIG TX WAKEUP)

Purpose

To select the controller which transmits the wakeup pattern and to specify the trigger type.

Parameters

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority

of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖).

Controller name Lets you select the name of the controller which transmits the wakeup pattern.

Channel type Lets you select the channel type to send the wakeup pattern on (channel A or channel B).

Trigger type Lets you select the trigger type of wakeup pattern transmission. The following settings are available:

Trigger Type	Meaning
Rising	Triggers the wakeup pattern transmission when the control signal rises from a negative or zero value to a positive value (or to zero if the initial value is negative).
Falling	Triggers the wakeup pattern transmission when the control signal falls from a positive or a zero value to a negative value (or to zero if the initial value is positive).
Either	Triggers the wakeup pattern transmission when the signal is either rising or falling.
Function call	The wakeup pattern is triggered by a function call that decides when to send the wakeup pattern.

Related topics

References

Block Description (RTIFLEXRAYCONFIG TX WAKEUP).. 46

Options Page (RTIFLEXRAYCONFIG TX WAKEUP)

Purpose

To specify the wakeup symbols of a wakeup pattern.

Parameters

Number of wakeup symbols Lets you specify the number of wakeup symbols.

Wakeup symbol low-time Lets you specify the low time of the wakeup symbol.

Wakeup symbol idle-time Lets you specify the idle time of the wakeup symbol.

Related topics

References

Block Description (RTIFLEXRAYCONFIG TX WAKEUP).. 46

Interrupt and Controller Reset Blocks

Introduction

You can use several events to trigger an interrupt and you can reset the controller.

Where to go from here

Information in this section

RTIFLEXRAYCONFIG INTERRUPT

Purpose

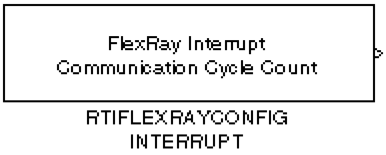
To make a FlexRay interrupt available as a trigger source.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG INTERRUPT)

Block



Purpose

To specify interrupts for a FlexRay bus from a FlexRay bus controller.

Description

The controller name must have been specified by an RTIFLEXRAYCONFIG CONTROLLER SETUP block.

For each event, only one interrupt is allowed for the selected controller. For the Communication Cycle Count, several interrupts are allowed, with the following limitations:

- There can be only one interrupt that is generated at start of every communication cycle.
- There can be only one interrupt for each specific cycle count value.

The RTIFLEXRAYCONFIG INTERRUPT block provides several event sources that can be used to trigger interrupts.

Event	Meaning
Missing Rate Correction Error	Triggers an interrupt when a controller loses synchronization and enters the passive state because the maximum odd cycles without rate correction value has been exceeded.
Communication Cycle Count	Triggers an interrupt when a defined number of communication cycles is reached. You can specify the number on the Unit page.
Receiving wakeup pattern (CHA)	Triggers an interrupt when the wakeup pattern is received on channel A.
Receiving wakeup pattern (CHB)	Triggers an interrupt when the wakeup pattern is received on channel B.

If you want to trigger an interrupt when the wakeup pattern is received on channel A or channel B, you must use two blocks.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG INTERRUPT) on page 50

Related topics

References

Unit Page (RTIFLEXRAYCONFIG INTERRUPT)

Purpose

To specify a FlexRay interrupt as a trigger source.

Dialog settings

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 🕮).

Controller name Lets you select the name of the controller the interrupt is specified for.

Interrupt type Lets you select the trigger type that is used to trigger the interrupt. The possible settings are:

- Missing Rate Correction Error
- Communication Cycle Count
- Receiving Wakeup Pattern (CHA)
- Receiving Wakeup Pattern (CHB)

For detailed information on the interrupt types, see Block Description (RTIFLEXRAYCONFIG INTERRUPT) on page 49.

If you select the Communication Cycle Count interrupt type, you must select whether the interrupt is generated at start of every communication cycle or at a specific communication cycle:

Generate interrupt at start of every communication cycle Lets you trigger an interrupt when a new communication cycle is started.

Generate interrupt for specific communication cycle Lets you specify the number of a specific communication cycle at the beginning of which an interrupt has to be generated. At the moment, the valid value is only 0. If the value is 0, interrupts are triggered at the beginning of each communication cycle.

Related topics

References

Block Description (RTIFLEXRAYCONFIG INTERRUPT).. 49

RTIFLEXRAYCONFIG CONTROLLER RESET

Purpose

To specify the basic settings for resetting the FlexRay communication controller.

Where to go from here

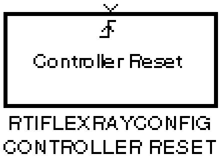
Information in this section

Information in other sections

How to Reset FlexRay Controllers (FlexRay Configuration Features 📖)
You can soft reset a FlexRay controller and specify whether it then reintegrates itself into the FlexRay cluster or may initiate a new start-up of the FlexRay communication.

Block Description (RTIFLEXRAYCONFIG CONTROLLER RESET)

Block



Purpose

To specify the basic settings for resetting the FlexRay communication controller.

Note

It is strongly recommended that the RTIFLEXRAYCONFIG CONTROLLER RESET block is not triggered while the FlexRay controller is in NORMAL state. Protocol errors can be generated if the FlexRay controller enters the RESET (soft-reset) state while the FlexRay cluster is running. For information on the protocol state, see Block Description (RTIFLEXRAYCONFIG STATUS) on page 39.

Description

The RTIFLEXRAYCONFIG CONTROLLER RESET block sets the FlexRay controller to the configuration mode and restarts it afterwards. It allows you to restart or reintegrate a FlexRay controller which already entered configuration mode or to restart/reintegrate a running FlexRay controller. The controller reset can be triggered by different events. For more information, see the Unit page. You can set the cold start inhibit flag of the FlexRay controller (see below).

I/O characteristics

The table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Cold start mode	0 / 1	UInt32	To 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 the cluster communication. ▪ If the flag is not set, the FlexRay controller can act as a leading coldstart node. This can impair the communication of a running FlexRay cluster. You can disable this block inport on the Options page.

Dialog settings

The dialog settings can be specified on the following pages:

- Options Page (RTIFLEXRAYCONFIG CONTROLLER RESET) on page 53
- Unit Page (RTIFLEXRAYCONFIG CONTROLLER RESET) on page 52

Related topics

References

Overview of the RTI FlexRay Configuration Blockset.. 13

Unit Page (RTIFLEXRAYCONFIG CONTROLLER RESET)

Purpose

To specify the controller name and the trigger type for resetting the FlexRay controller.

Parameters

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority

of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖).

Controller name Lets you select the name of the controller to be reset.

Trigger type Lets you select the trigger type of the controller reset. The following settings are available:

Trigger Type	Meaning
Rising	Triggers the execution of the subsystem when the control signal rises from a negative or zero value to a positive value (or to zero if the initial value is negative).
Falling	Triggers the execution of the subsystem when the control signal falls from a positive or a zero value to a negative value (or to zero if the initial value is positive).
Either	Triggers the execution of the subsystem when the signal is either rising or falling.
Function call	The reset command is triggered by a function call that decides when to send the reset signal.

Related topics

References

Block Description (RTIFLEXRAYCONFIG CONTROLLER RESET).. 51

Options Page (RTIFLEXRAYCONFIG CONTROLLER RESET)

Purpose

To specify the options for a controller reset.

Parameters

Enable cold start mode inport Lets you enable or disable the cold start mode block inport. For details on the cold start mode, refer to Block Description (RTIFLEXRAYCONFIG CONTROLLER RESET) on page 51.

Related topics

References

Block Description (RTIFLEXRAYCONFIG CONTROLLER RESET).. 51

RTIFLEXRAYCONFIG CONTROLLER STOP

Purpose

To specify the basic settings for stopping a FlexRay communication controller.

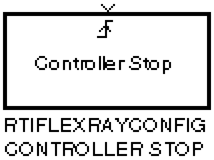
Where to go from here

Information in this section

Information in other sections

Block Description (RTIFLEXRAYCONFIG CONTROLLER STOP)

Block



Purpose

To specify the basic settings for resetting the FlexRay communication controller.

Note

It is strongly recommended that the RTIFLEXRAYCONFIG CONTROLLER STOP block is not triggered while the FlexRay controller is in NORMAL state. Protocol errors can be generated if the FlexRay controller enters the configuration mode while the FlexRay cluster is running. For information on the protocol state, see Block Description (RTIFLEXRAYCONFIG STATUS) on page 39.

Description

The RTIFLEXRAYCONFIG CONTROLLER STOP block sets the FlexRay controller to the configuration mode. To restart the FlexRay controller, you can use the RTIFLEXRAYCONFIG CONTROLLER RESTART block.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG CONTROLLER STOP) on page 55

Related topics

References

Unit Page (RTIFLEXRAYCONFIG CONTROLLER STOP)

Purpose

To specify the controller name and the trigger type for stopping the FlexRay controller.

Parameters

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 🕮).

Controller name Lets you select the name of the controller to be stopped.

Trigger type Lets you select the trigger type which stops the controller. The following settings are available:

Trigger Type	Meaning
Rising	Triggers the execution of the subsystem when the control signal rises from a negative or zero value to a positive value (or to zero if the initial value is negative).
Falling	Triggers the execution of the subsystem when the control signal falls from a positive or a zero value to a negative value (or to zero if the initial value is positive).
Either	Triggers the execution of the subsystem when the signal is either rising or falling.
Function call	The reset command is triggered by a function call that decides when to send the reset signal.

Related topics

References

RTIFLEXRAYCONFIG CONTROLLER RESTART

Purpose

To specify the basic settings for restarting the FlexRay communication controller.

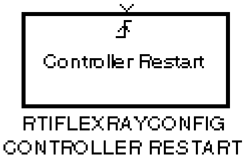
Where to go from here

Information in this section

Information in other sections

Block Description (RTIFLEXRAYCONFIG CONTROLLER RESTART)

Block



Purpose

To specify the basic settings for restarting a FlexRay communication controller.

Description

The **RTIFLEXRAYCONFIG CONTROLLER RESTART** block allows you to restart or reintegrate a FlexRay controller which already entered the configuration mode by using the **RTIFLEXRAYCONFIG CONTROLLER STOP** block before. The controller restart can be triggered by different events. For more information, see the **Unit** page. You can set the cold start inhibit flag of the FlexRay controller (see below).

To stop a FlexRay controller, you can use the **RTIFLEXRAYCONFIG CONTROLLER STOP** block.

I/O characteristics

The table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Cold start mode	0 / 1	UInt32	To 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 the cluster communication. ▪ If the flag is not set, the FlexRay controller can act as a leading coldstart node. This can impair the communication of a running FlexRay cluster. You can disable this block inport on the **Options** page.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG CONTROLLER RESTART) on page 57
- Options Page (RTIFLEXRAYCONFIG CONTROLLER RESTART) on page 58

Related topics

References

Overview of the RTI FlexRay Configuration Blockset........ 13
RTIFLEXRAYCONFIG CONTROLLER STOP........ 53

Unit Page (RTIFLEXRAYCONFIG CONTROLLER RESTART)

Purpose

To specify the controller name and the trigger type for restarting a FlexRay controller.

Parameters

Configuration ID Lets you choose a configuration ID, which is required for modeling several FlexRay buses in one real-time model. The configuration ID identifies a configuration and therefore a FlexRay bus, and specifies the priority

of the FlexRay bus. For details, refer to Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 📖).

Controller name Lets you select the name of the controller to be restarted.

Trigger type Lets you select the trigger type of the controller restart. The following settings are available:

Trigger Type	Meaning
Rising	Triggers the execution of the subsystem when the control signal rises from a negative or zero value to a positive value (or to zero if the initial value is negative).
Falling	Triggers the execution of the subsystem when the control signal falls from a positive or a zero value to a negative value (or to zero if the initial value is positive).
Either	Triggers the execution of the subsystem when the signal is either rising or falling.
Function call	The reset command is triggered by a function call that decides when to send the reset signal.

Related topics

References

Block Description (RTIFLEXRAYCONFIG CONTROLLER RESTART).. 56

Options Page (RTIFLEXRAYCONFIG CONTROLLER RESTART)

Purpose

To specify the options for a controller restart.

Parameters

Enable cold start mode inport Lets you enable or disable the cold start mode block inport. For details on the cold start mode, refer to Block Description (RTIFLEXRAYCONFIG CONTROLLER RESTART) on page 56.

Related topics

References

Block Description (RTIFLEXRAYCONFIG CONTROLLER RESTART).. 56

Blocks for Configuration Generation

Blocks for configuration generation

The blocks for configuration generation cannot be used directly. They are used by the **`rtiflexrayconfig_modelgenerate`** command to build the automatically generated FlexRay model.

Where to go from here

Information in this section

Information in other sections

Blocks for PDU-Based Bus Simulation

Where to go from here

Information in this section

RTIFLEXRAYCONFIG TIMETABLE START

Purpose

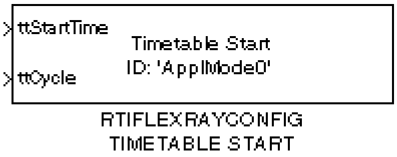
To start assigned timetable tasks.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG TIMETABLE START)

Block



Purpose

To start assigned timetable tasks.

Note

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

Description

The RTIFLEXRAYCONFIG TIMETABLE START block is part of the synchronization task (refer to Synchronization Task Subsystem on page 120). The timetable which corresponds to an RTIFLEXRAYCONFIG TIMETABLE START block is started each time the block is evaluated. One RTIFLEXRAYCONFIG TIMETABLE START block can start several timetable tasks with the same timetable ID. An absolute time on the dSPACE system or a delay time that is set relative to the point in time of block execution can be specified by using the optional inport. The tasks are executed using the RTIFLEXRAYCONFIG TIMETABLE TASK or RTIFLEXRAYCONFIG CUSTOM CODE blocks.

I/O characteristics

The table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
ttStartTime	–	Double or Vector (2 UInt32)	The time to start task execution. The **Start Time inport data type** option on the **Parameters** page specifies whether it is an absolute start time (microticks and macroticks of the dSPACE timer device) or a delay time in seconds.
ttCycle	0 ... 63	UInt32	The number of timetable cycles

Dialog settings

The dialog settings can be specified on the following pages:

- Parameters Page (RTIFLEXRAYCONFIG TIMETABLE START) on page 62

Related topics

References

Parameters Page (RTIFLEXRAYCONFIG TIMETABLE START)

Purpose

To display the parameters set for starting the related timetable task.

Note

The block parameters are read-only. They are specified during the generation process.

Dialog settings

Timetable ID Displays the name of the timetable currently assigned to the timetable tasks. The string length must be in the range 1 ... 10. The valid characters are “a ... z”, “A ... Z”, “0 ... 9” and “_”.

Number of timetable cycles Displays the number of timetable cycles. This edit field is disabled if the block is used for a FlexRay model.

Sample time Lets you enter the sample time (the intervals the block is executed at).

Sample Time	Meaning
–1	Inherited sample time
= 0	Discrete sample time of the subsystem
> 0	Discrete sample time as specified

Start timetable once at simulation start Shows whether to automatically start the timetable once at the beginning of the simulation.

Enable timetable start time inport Shows whether the timetable start-time inport is enabled or disabled. The inport allows you to specify a delay time for the timetable start.

Start-time inport data type Shows the selected data type of the start-time inport signal:

Variable	Data Type	Meaning
delay time (Double)	Double	Supports the input of a delay time.
absolute time (time stamp)	Vector (2 UInt32)	Supports the input of an absolute time (time stamp), for example, by connecting an RTIFLEXRAYCONFIG SYNC SERVICE block (ttStartTime outport).

Related topics

References

Block Description (RTIFLEXRAYCONFIG TIMETABLE START)........61
RTIFLEXRAYCONFIG SYNC SERVICE........115

RTIFLEXRAYCONFIG TIMETABLE TASK

Purpose

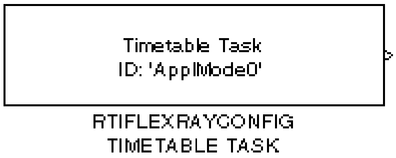
To assign a task to a timetable.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG TIMETABLE TASK)

Block



Purpose

To assign a task to a timetable.

> **Note**
>
> The block is added to the model during the generation process. The block parameters are read-only.

Description

The referenced timetable is started by an RTIFLEXRAYCONFIG TIMETABLE START block during run time.

For each RTIFLEXRAYCONFIG TIMETABLE TASK block, there must be at least one corresponding RTIFLEXRAYCONFIG TIMETABLE START block with the same timetable ID.

For detailed information, refer to RTI Task Configuration Dialog - Task Groups (RTI and RTI-MP Implementation Reference 📖).

I/O characteristics

The table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Function Call	–	Scalar	The outport can be used to trigger a function-called subsystem.

Dialog settings

The dialog settings can be specified on the following pages:

- Parameters Page (RTIFLEXRAYCONFIG TIMETABLE TASK) on page 65

Related topics

References

Overview of the Blocks for Configuration Generation.......... 15
RTIFLEXRAYCONFIG TIMETABLE START.......... 61

Parameters Page (RTIFLEXRAYCONFIG TIMETABLE TASK)

Purpose

To display the parameters set for the RTIFLEXRAYCONFIG TIMETABLE TASK block.

> **Note**
>
> The block parameters are read-only. They are specified during the generation process.

Dialog settings

Timetable ID Displays the name of the timetable the task is currently assigned to. The string length must be in the range 1 ... 10. The valid characters are “a ... z”, “A ... Z”, “0 ... 9” and “_”.

Base cycle Displays the timetable cycle in which the task should be executed for the first time.

Cycle repetition Displays the repetition in which the task should be executed.

End cycle Displays the timetable cycle in which the task should be executed for the last time. The task is executed in cycles complying the following condition (with `k ≥ 0`):

```
CycleIndex = BaseCycle + k · CycleRepetition ≤ EndCycle
```

Task Trigger Delay Displays the task start time(s) in relation to the timetable start event. Several delay times can be specified by using a vector, for example, [0.0001 0.0003]. The valid values must be ≥ 0. Workspace variables are allowed.

Deadlines Displays the task execution finish. If several start delay times are specified (`Task Trigger Delay` parameter), the vector of the `Deadlines` parameter must be the same length, for example, [0.0002 0.0004]. A deadline of “0.0” (zero) indicates that no task deadline check is required. The valid values must be ≥ 0. Workspace variables are allowed.

Task entry code Displays C code that was added for execution before the task is started. This can be any C code that can be used to execute external communication functions, for example, to read data from the controller before the task is started. The default value is: `/* Task entry code string */`.

Task exit code Displays C code that was added for execution after the task has finished. This can be any C code that can be used to execute external communication functions, for example, to write data to the controller after the task was executed. The default value is: `/* Task exit code string */`.

Related topics

References

Block Description (RTIFLEXRAYCONFIG TIMETABLE TASK).. 64

RTIFLEXRAYCONFIG CUSTOM CODE

Purpose

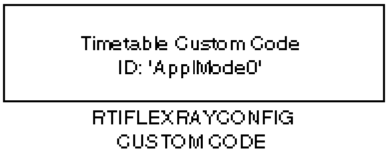
To assign a task with custom code to a timetable.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG CUSTOM CODE)

Block



Purpose

To assign a task with your own C code to a timetable.

Note

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

Description

The **RTIFLEXRAYCONFIG CUSTOM CODE** block is used to include communication code which is executed using a separate task. This is the case if the communication code is not executed directly before or after an application task.

For each **RTIFLEXRAYCONFIG CUSTOM CODE** block, there must be at least one corresponding **RTIFLEXRAYCONFIG TIMETABLE START** block with the same timetable ID.

For detailed information, refer to RTI Task Configuration Dialog - Task Groups (RTI and RTI-MP Implementation Reference 🕮).

Dialog settings

The dialog settings can be specified on the following pages:

- Parameters Page (RTIFLEXRAYCONFIG CUSTOM CODE) on page 67

Related topics

References

Overview of the Blocks for Configuration Generation.......... 15
RTIFLEXRAYCONFIG TIMETABLE START.......... 61

Parameters Page (RTIFLEXRAYCONFIG CUSTOM CODE)

Purpose

To display the parameters set for the RTIFLEXRAYCONFIG CUSTOM CODE block.

Note

The block parameters are read-only. They are specified during the generation process.

Dialog settings

Timetable ID Displays the name of the timetable currently assigned to the task. The string length must be in the range 1 ... 10. The valid characters are “a ... z”, “A ... Z”, “0 ... 9” and “_”.

Base cycle Displays the timetable cycle in which the task should be executed for the first time.

Cycle repetition Displays the repetition in which your C code should be executed.

End cycle Displays the timetable cycle in which the task should be executed for the last time. The task is executed in cycles complying the following condition (`k ≥ 0`):

```
CycleIndex = BaseCycle + k · CycleRepetition ≤ EndCycle
```

Custom code trigger delay Displays the task start time(s) in relation to the timetable start event. Several delay times can be specified by using a vector, for example, [0.0001 0.0003]. The valid values must be ≥ 0. Workspace variables are allowed.

Deadlines Displays the task execution finish. If several start delay times are specified (`Custom code trigger delay` parameter), the vector of the `Deadlines` parameter must be the same length, for example, [0.0002 0.0004].

A deadline of “0.0” (zero) indicates that no task deadline check is required. The valid values must be ≥ 0. Workspace variables are allowed.

Custom Code Displays C code to be executed. The default value is: `/* custom code string */`.

Related topics

References

RTIFLEXRAYCONFIG COM ADMIN

Purpose

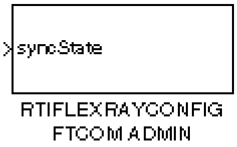
To enable and disable the communication layer according to the synchronization state and settings made using RTIFLEXRAYCONFIG Com Control blocks.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG COM ADMIN)

Block



Purpose

To enable and disable the communication (Com) layer according to the synchronization state and settings made using RTIFLEXRAYCONFIG Com Control blocks.

Note

The block is added to the model during the generation process. Use it only in the synchronization task, see Synchronization Task Subsystem on page 120.

Description

The RTIFLEXRAYCONFIG COM ADMIN block switches the Com layer offline when the host is not synchronized. When the host is synchronized, the RTIFLEXRAYCONFIG COM ADMIN block switches the Com layer according to the settings made with the RTIFLEXRAYCONFIG COM CYCLIC CONTROL and RTIFLEXRAYCONFIG COM EVENT CONTROL blocks. You can switch the whole Com layer, or the sending or receiving task only.

Tip

The RTIFLEXRAYCONFIG COM ADMIN, RTIFLEXRAYCONFIG COM CYCLIC CONTROL, and RTIFLEXRAYCONFIG COM EVENT CONTROL blocks are useful in error handling tasks. If an error has occurred, you can switch the Com layer offline to prevent the call of Com-specific error hooks.

I/O characteristics

The table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
syncState	0 / 1	Double	Indicates the synchronization status: ▪ Value = 1: FlexRay host is synchronized ▪ Value = 0: FlexRay host is not synchronized

Dialog settings

The block has no dialog settings.

Related topics

References

RTIFLEXRAYCONFIG COM CYCLIC CONTROL..24
RTIFLEXRAYCONFIG COM EVENT CONTROL..22

Parameters Page (RTIFLEXRAYCONFIG COM ADMIN)

Purpose

To display the parameters set for the RTIFLEXRAYCONFIG COM ADMIN block.

Dialog settings

No parameters.

Related topics

References

RTIFLEXRAYCONFIG UPDATE

Purpose

To update the FlexRay blocks in the Simulink model.

Where to go from here

Information in this section

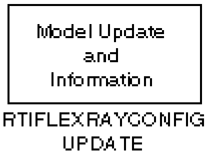
Information in other sections

How to Update the FlexRay Blocks in Simulink Models (FlexRay Configuration Features 🕮)

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 a special RTI block to update your Simulink model automatically. This is useful when you have a lot of changes in a large model.

Block Description (RTIFLEXRAYCONFIG UPDATE)

Block



Purpose

To update a Simulink model based on a generated FlexRay configuration.

> **Note**
>
> The block is added to the model during the generation process. Some block parameters are read-only.

Description

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 (FlexRay Configuration Features 🕮).

> **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.

The update process is logged in two log files located in the current working folder. When the update process is completed, a confirmation dialog opens with links to these two log files (these links can also be found in the MATLAB workspace):

- `*_UpdateBlocks.log` lists all the updated blocks.
- `*_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.

There must not be more than one RTIFLEXRAYCONFIG UPDATE block in a Simulink model.

Dialog settings

The dialog settings can be specified on the following pages:

- Model Update Page (RTIFLEXRAYCONFIG UPDATE) on page 72
- Configuration Info Page (RTIFLEXRAYCONFIG UPDATE) on page 73
- CRC Settings Page (RTIFLEXRAYCONFIG UPDATE) on page 74
- XCP Frames (RTIFLEXRAYCONFIG UPDATE) on page 75

Related topics

References

Model Update Page (RTIFLEXRAYCONFIG UPDATE)

Purpose

To update the FlexRay blocks in the Simulink model.

Parameters

Basis RTI 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 or absolute.

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

Exchange the Basic Configuration subsystem Indicates whether the RTIFLEXRAYCONFIG Basic Configuration subsystem is replaced during model update. If you select this checkbox, the new subsystem replaces the old subsystem in the model. The old subsystem is deleted.

Keep CRC path 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 RTI FlexRay Configuration blocks are derived from custom libraries Lets you enable or disable the check for RTI 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 RTI 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 RTI FlexRay Configuration blocks from custom libraries is executed. The model update ignores any RTI FlexRay Configuration blocks derived from custom libraries that are in your Simulink model. If your model contains such RTI 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 RTI 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 RTIFLEXRAYCONFIG PDU RX and RTIFLEXRAYCONFIG PDU TX blocks are not updated. You must update the mapping subsystems manually.

Related topics

References

Block Description (RTIFLEXRAYCONFIG UPDATE)....................71
CRC Page (FlexRay Configuration Tool Reference 📖)
CRC Settings Page (RTIFLEXRAYCONFIG UPDATE)....................74

Configuration Info Page (RTIFLEXRAYCONFIG UPDATE)

Purpose

To get information on the generated RTI 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 of the configuration data file
Generated at	Date when the configuration data file is generated

FIBEX data base frame Displays information on the data base:

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

Related topics

References

Block Description (RTIFLEXRAYCONFIG UPDATE).. 71

CRC Settings Page (RTIFLEXRAYCONFIG 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 or absolute.

Enable include path specification Lets you select a path for including additional header files.

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

Related topics

References

Block Description (RTIFLEXRAYCONFIG UPDATE)...71
Model Update Page (RTIFLEXRAYCONFIG UPDATE)...72

XCP Frames (RTIFLEXRAYCONFIG 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 defined XCP frames in the FIBEX file or AUTOSAR system description file.

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

Related topics

References

Block Description (RTIFLEXRAYCONFIG UPDATE)...71

RTIFLEXRAYCONFIG CONTROLLER SETUP

Purpose

To specify the basic settings of the FlexRay controller.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG CONTROLLER SETUP)

Block

Controller Setup
RTIFLEXRAYCONFIG
CONTROLLER SETUP

Purpose

To specify the basic settings of the FlexRay controller.

Note

The block is added to the model during the generation process.

Description

The block specifies the board type, the board number and the controller number of a FlexRay controller.

For information on how to connect your dSPACE platform to the FlexRay bus, refer to

- DS1006/DS1007 with DS4501: DS4501 IP Carrier Board (PHS Bus System Hardware Reference 📖)
- DS1006/DS1007 with DS4505: DS4505 Interface Board (PHS Bus System Hardware Reference 📖)

- MicroAutoBox II: Connecting to a FlexRay Bus (MicroAutoBox II Hardware Installation and Configuration Guide 📖)

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG CONTROLLER SETUP) on page 77
- Options Page (RTIFLEXRAYCONFIG CONTROLLER SETUP) on page 78

Related topics

Basics

Connecting Real-Time Systems to the FlexRay Bus (FlexRay Configuration Features 📖)

References

Overview of the Blocks for Configuration Generation.. 15

Unit Page (RTIFLEXRAYCONFIG CONTROLLER SETUP)

Purpose

To specify the basic settings of the FlexRay controller.

Dialog settings

Board/Module type Specifies the board type or module type.

Board/Module number Specifies the board number (DS4501 or DS4505) or module number (MicroAutoBox II). The board number is in the range 1 ... 16. The module number is 1 or 2.

Controller number Specifies the controller number. A DS4501 or DS4505 can carry up to 4 FlexRay controllers. On the DS4501 and DS4505, controller 1 corresponds to the FlexRay module inserted in slot 1.

Controller name Specifies the controller name which is set in the generation process.

Controller module Specifies the controller module which is set in the generation process.

Buffer count Specifies the buffer count of the controller module which is set in the generation process.

Related topics

Basics

Connecting Real-Time Systems to the FlexRay Bus (FlexRay Configuration Features 📖)

HowTos

How to Configure Hardware (FlexRay Configuration Tool Guide 📖)

References

Block Description (RTIFLEXRAYCONFIG CONTROLLER SETUP)... 76
Signal Mapping to I/O Pins (PHS Bus System Hardware Reference 📖)
Signal Mapping to I/O Pins (DS4505/DS4340) (PHS Bus System Hardware Reference 📖)

Options Page (RTIFLEXRAYCONFIG CONTROLLER SETUP)

Purpose

To specify the termination modes and cold start mode.

Description

The following options are enabled only if your controller module is a DS4340 FlexRay Interface Module. You can specify the termination for the bus lines and cold start mode.

For more information on the bus termination, refer to

- DS1006/DS1007 with DS4501 or DS4505: DS4340 Connections in Different Topologies (PHS Bus System Hardware Reference 📖)
- MicroAutoBox II: DS4340 Connections in Different Topologies (MicroAutoBox II Hardware Installation and Configuration Guide 📖)

Dialog settings

bus lines are terminated (CHA) Specifies the termination mode of channel A. If you choose this option, the bus lines of channel A are terminated.

bus lines are feed through (CHA) Specifies the termination mode of channel A. If you choose this option, the bus lines of channel A are fed through.

bus lines are terminated (CHB) Specifies the termination mode of channel B. If you choose this option, the bus lines of channel B are terminated.

bus lines are feed through (CHB) Specifies the termination mode of channel B. If you choose this option, the bus lines of channel B are fed through.

Inhibit cold start Lets you select whether the FlexRay controller should act as a following coldstart node only. With this checkbox selected, 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 the cluster communication.

Related topics

References

RTIFLEXRAYCONFIG PDU RX

Purpose

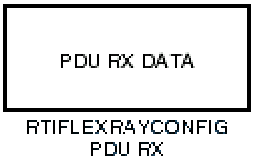
To read a PDU from the FlexRay bus.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG PDU RX)

Block



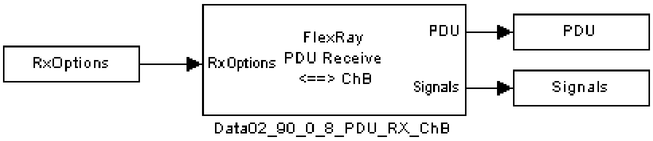
Purpose

To receive a PDU via a FlexRay network.

> **Note**
>
> The information type of the **PDU** page indicates whether the frame-specific data of the FIBEX file is interpreted as PDUs or as frames. For FIBEX 2.0 and lower versions, the data is interpreted as frames. In all the FIBEX file versions greater than FIBEX 2.0, the data is interpreted as PDUs. Therefore, the PDUs can also be understood as frames according to the FIBEX version used.
> For AUTOSAR System Templates, the data is always interpreted as PDUs.

Description

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



The **RTIFLEXRAYCONFIG 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 use 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 81.

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 82.
Signals	Provides all the signal values of a PDU and their states. For details, refer to Signals mapping subsystem on page 87.

Dialog settings

The dialog settings can be specified on the following pages:

- PDU Page (RTIFLEXRAYCONFIG PDU RX) on page 89
- Signal Page (RTIFLEXRAYCONFIG PDU RX) on page 91
- PDU Option Page (RTIFLEXRAYCONFIG PDU RX) on page 92
- Signal Option Page (RTIFLEXRAYCONFIG PDU RX) on page 93

Related topics

References

RTIFLEXRAYCONFIG PDU RX Mapping Subsystems

Purpose

To map PDU-specific control signals to the ports of an RTIFLEXRAYCONFIG PDU RX block.

Description

The subsystems are automatically created. They can be used to map the signals of your real-time model to the RTIFLEXRAYCONFIG 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 and the PDU type.

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, which is given 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, which is given 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. **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.

Simulink Outport	Range	Simulink Data Type	Meaning
			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.
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.
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 value of user byte 0 that is transferred in the global time synchronization message.
GTS_UserByte1	0 ... 254	UInt8	Displays the value of user byte 1 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 / 1	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 an 8-byte array. 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 an 8-byte array.	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 an 8-byte array.	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 (FlexRay Configuration Features 🕮)
Sending Dynamic PDUs or Sub-PDUs (FlexRay Configuration Features 🕮)
Sending Static PDUs and Sub-PDUs (FlexRay Configuration Features 🕮)

HowTos

How to Handle Checksum Calculation for a PDU (FlexRay Configuration Features 🕮)
How to Manipulate the Payload Length of a PDU (FlexRay Configuration Features 🕮)
How to Manipulate the Update Bit of a PDU (FlexRay Configuration Features 🕮)
How to Receive PDUs in Raw Format (FlexRay Configuration Features 🕮)
How to Send or Receive Signals of PDUs (FlexRay Configuration Features 🕮)
How to Send PDUs in Raw Format (FlexRay Configuration Features 🕮)

References

Block Description (RTIFLEXRAYCONFIG PDU RX).. 79

PDU Page (RTIFLEXRAYCONFIG 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 3.0, FIBEX 3.1, 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 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	Sub-frame dynamic event
SDC	Sub-frame dynamic cyclic
SDEC	Sub-frame dynamic event cyclic
SSC	Sub-frame 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 (RTIFLEXRAYCONFIG PDU RX).. 79

Signal Page (RTIFLEXRAYCONFIG 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 is configured in the FlexRay Configuration Tool. The radio button indicates whether the data type is selected. You can configure the data type of a signal 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 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 (RTIFLEXRAYCONFIG PDU RX).. 79

PDU Option Page (RTIFLEXRAYCONFIG 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 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 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 (RTIFLEXRAYCONFIG PDU RX).. 79

Signal Option Page (RTIFLEXRAYCONFIG PDU RX)

Purpose

To display the signal interfaces and configuration.

Dialog settings

Input configuration Displays which signal inputs 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 configuration Displays which signal outputs 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 (RTIFLEXRAYCONFIG PDU RX).. 79

RTIFLEXRAYCONFIG PDU TX

Purpose

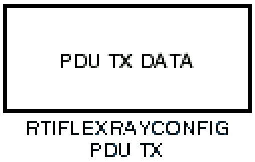
To send a PDU via a FlexRay network.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG PDU TX)

Block



Purpose

To send a PDU via a FlexRay network.

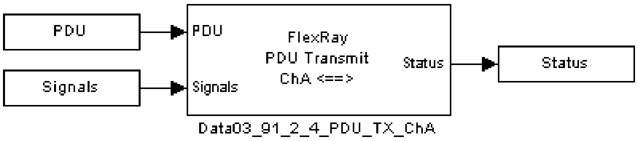
Note

The information type of the **PDU** page indicates whether the frame-specific data of the FIBEX file is interpreted as PDUs or as frames. For FIBEX 2.0 and lower versions, the data is interpreted as frames. In all the FIBEX file versions greater than FIBEX 2.0, the data is interpreted as PDUs. Therefore, the PDUs can also be understood as frames according to the FIBEX version used.

For AUTOSAR System Templates, the data is always interpreted as PDUs.

Description

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



The **RTIFLEXRAYCONFIG PDU TX** block can be connected to **PDU**, **Signals** 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 96.
Signals	Sets and manipulates the signals of the PDU. For details, refer to Signals mapping subsystem on page 100.

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 102.

Dialog settings

The dialog settings can be specified on the following pages:

- PDU Page (RTIFLEXRAYCONFIG PDU TX) on page 105
- Signal Page (RTIFLEXRAYCONFIG PDU TX) on page 107
- PDU Option Page (RTIFLEXRAYCONFIG PDU TX) on page 108
- Signal Option Page (RTIFLEXRAYCONFIG PDU TX) on page 109

Related topics

References

Overview of an Automatically Generated FlexRay Model... 16
Overview of the Blocks for Configuration Generation... 15
Structure of the Automatically Generated FlexRay Model... 17

RTIFLEXRAYCONFIG PDU TX Mapping Subsystems

Purpose

To map PDU-specific control signals to the ports of an RTIFLEXRAYCONFIG PDU TX block.

Description

The subsystems are automatically created. They can be used to map the signals of your real-time model to the RTIFLEXRAYCONFIG 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.
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. 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

Simulink Inport	Range	Simulink Data Type	Meaning
			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 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

Simulink Inport	Range	Simulink Data Type	Meaning
			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, which is given 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, which is given 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 (FlexRay Configuration Features 📖). ▪ 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 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
<Signal name>_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.
<Signal name>_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.
<Signal name>_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.
<Signal name>_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.
<Signal name>_Validity[1)]	0, 1, 2, 4, 8, 16	UInt32	Specifies the validity status 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 status of triggering ▪ 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.
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, 4	UInt32	Displays status information. The statuses have different meanings with: ▪ 0: Raw data successfully written. ▪ 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.
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.

Simulink Outport	Range	Simulink Data Type	Meaning
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.
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.

Simulink Outport	Range	Simulink Data Type	Meaning
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

Sending Dynamic PDUs or Sub-PDUs (FlexRay Configuration Features 📖)
Sending Static PDUs and Sub-PDUs (FlexRay Configuration Features 📖)

HowTos

How to Handle Checksum Calculation for a PDU (FlexRay Configuration Features 📖)
How to Manipulate the Payload Length of a PDU (FlexRay Configuration Features 📖)
How to Manipulate the Update Bit of a PDU (FlexRay Configuration Features 📖)
How to Receive PDUs in Raw Format (FlexRay Configuration Features 📖)
How to Send or Receive Signals of PDUs (FlexRay Configuration Features 📖)
How to Send PDUs in Raw Format (FlexRay Configuration Features 📖)

References

Block Description (RTIFLEXRAYCONFIG PDU TX).. 94

PDU Page (RTIFLEXRAYCONFIG 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 3.0, FIBEX 3.1, 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 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
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 (RTIFLEXRAYCONFIG PDU TX)..........94

Signal Page (RTIFLEXRAYCONFIG 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 is configured in the FlexRay Configuration Tool. The radio button indicates whether the data type is selected. You can configure the data type of a signal 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.

Constraints 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

The Conv. Factor and Conv. Offset are defined in the following form:
`y = Conv. Factor · X + Conv. Offset`.

Related topics

References

Block Description (RTIFLEXRAYCONFIG PDU TX)..94

PDU Option Page (RTIFLEXRAYCONFIG 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.

Update bit is ignored This setting is not relevant.

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 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

References

Block Description (RTIFLEXRAYCONFIG PDU TX)..94

Signal Option Page (RTIFLEXRAYCONFIG PDU TX)

Purpose

To display the signal interfaces and configuration.

Dialog settings

Input port configuration Displays which signal inputs 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 which signal outputs 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 (RTIFLEXRAYCONFIG PDU TX).. 94

RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE

Purpose

To enable or disable the communication of a specified group of ECUs (frame membership group).

Where to go from here

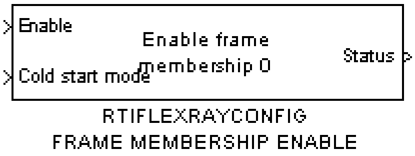
Information in this section

Information in other sections

How to Enable or Disable the Communication of an ECU (FlexRay Configuration Features 🕮)
The communication of simulated ECUs can be disabled to simulate an ECU malfunction or to replace them by the real ECUs.

Block Description (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE)

Block


Enable
Cold start mode
Enable frame membership 0
Status
RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE

Purpose

To enable or disable the communication of a specified group of ECUs (frame membership group).

Note

The block is added to the model during the generation process if a TX frame membership group ID was selected in the FlexRay Configuration Tool, indicating that enabling/disabling the frame membership group during run time is allowed. Refer to Building Frame Membership Groups (FlexRay Configuration Tool Guide 🕮).

Description

The block is used to enable or disable a frame membership group. A frame membership group contains all the frame messages sent by one or several ECUs. Enabling or disabling a frame membership therefore affects the communication of an entire ECU. Using this feature, you can simulate the malfunction on an ECU or replace a simulated ECU by a real one without changing the real-time model. The frame membership groups are specified in the FlexRay Configuration Tool. Refer to Building Frame Membership Groups (FlexRay Configuration Tool Guide 🕮).

Receive frames can be assigned to the same frame membership as the send frames or to the global RX pool. All receive frames assigned to the global RX pool cannot be disabled.

You must use one **RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE** block for each frame membership group which is configured in the FlexRay Configuration Tool.

If the RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block enables a frame membership group, the controllers of the ECUs are enabled. It is not possible to reset or restart the controllers using the RTIFLEXRAYCONFIG CONTROLLER RESET or RTIFLEXRAYCONFIG CONTROLLER RESTART blocks.

The RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block is locked against multiple calls, so frame memberships are enabled or disabled only once.

You can set the cold start inhibit flag of the FlexRay controller (see below).

I/O characteristics

The following table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Enable	0/1	UInt32	Activates the sending and receiving of a frame membership group ▪ 0: Stops sending/receiving of the frame membership group ▪ 1: Starts sending/receiving of the frame membership group
Cold start mode	0 / 1	UInt32	To 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 the cluster communication. ▪ If the flag is not set, the FlexRay controller can act as a leading coldstart node. This can impair the communication of a running FlexRay cluster. You can disable this block inport on the Unit page.

The following table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Status	0, 1, 2, 4	UInt32	Displays the membership status: ▪ 0: DSFTCOM_MS_DISABLED Disabling the frame membership group was successful. ▪ 1: DSFTCOM_MS_ENABLED Enabling the frame membership group was successful. ▪ 2: DSFTCOM_MS_DISABLE_NOK Disabling the frame membership group failed. ▪ 4: DSFTCOM_MS_ENABLE_NOK Enabling the frame membership group failed.

The Status port is available only if it is enabled on the Unit page.

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE) on page 112
- Initialization Page (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE) on page 112

Related topics

References

Overview of the Blocks for Configuration Generation.. 15
RTIFLEXRAYCONFIG CONTROLLER RESET... 50
RTIFLEXRAYCONFIG CONTROLLER RESTART.. 56

Unit Page (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE)

Purpose

To select the frame membership group for enabling or disabling the sending and possibly receiving of all frames of the members, to specify the cold start mode, and to set the membership status outport of the RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE block.

Dialog settings

Group selection Specifies the frame membership group to be enabled or disabled. The group selection contains all frame membership groups which are configured in the FlexRay Configuration Tool.

Enable cold start mode inport Lets you enable or disable the cold start mode block inport. For details on the cold start mode, refer to Block Description (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE) on page 110.

Enable membership status outport Lets you enable or disable the membership status outport.

Related topics

Basics

Building Frame Membership Groups (FlexRay Configuration Tool Guide 📖)

References

Block Description (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE)..................................... 110

Initialization Page (RTIFLEXRAYCONFIG FRAME MEMBERSHIP ENABLE)

Purpose

To set the initialization state of a frame membership group.

Dialog settings

Enable frame membership after model start Specifies the initialization state for the frame membership group.

Related topics

References

RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS

Purpose

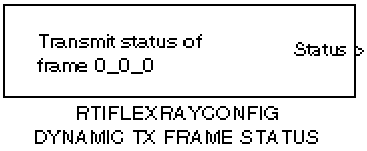
To monitor the transmit status of dynamic event-based FlexRay frames (DE or DEC).

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS)

Block



Purpose

To monitor the transmit status of event-based FlexRay frames.

Description

Using this block you can monitor the triggering status of dynamic event-based frames (DE or DEC). For information on the states and the events when the state changes, refer to Sending Dynamic PDUs or Sub-PDUs (FlexRay Configuration Features).

Monitoring the status of event-triggered subframes is not yet supported.

I/O characteristics

The following table shows the block outports:

Simulink Output	Range	Simulink Data Type	Meaning
Status	65, 66, 67, 68	UInt32	Displays the send status: ▪ 65: Frame is not triggered ▪ 66: Frame is triggered but frame is not in the send buffer ▪ 67: Frame is in the send buffer but it is not yet sent ▪ 68: Frame is sent

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS) on page 114

Related topics

References

Overview of the Blocks for Configuration Generation.. 15
RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS.. 113

Unit Page (RTIFLEXRAYCONFIG DYNAMIC TX FRAME STATUS)

Purpose

To show the transmitted frame status.

Dialog settings

Frame specification Displays the frames the signals are carried by. The frame is described in a list containing the following parameters:

Column	Description
ID	Identifier
BC	Base cycle
CR	Cycle repetition
CT	Cyclic timing
Type	Frame type
Ch	Channel the frame is sent on (A, B, A+B)
CTR	Name of the controller which sends the frame
SendECU	Name of the ECU which sends the frame
Name	Name of the frame

Parent ECU Displays the name of the ECU which reads the signal.

Related topics

References

RTIFLEXRAYCONFIG SYNC SERVICE

Purpose

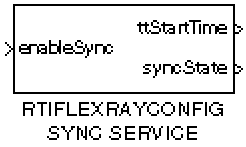
To specify the settings for synchronization.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG SYNC SERVICE)

Block



Purpose

To specify the settings for synchronization.

Note

The block is added to the model during the generation process. Some block parameters are read-only. You can only select the synchronization mode.

Description

The RTIFLEXRAYCONFIG SYNC SERVICE block is used to synchronize the clocks of the FlexRay cluster and the dSPACE FlexRay host system. It is automatically added to the model in conjunction with an RTIFLEXRAYCONFIG TIMETABLE START block (see RTIFLEXRAYCONFIG Synchronization Task on page 120). The RTIFLEXRAYCONFIG SYNC SERVICE block can be used to set the correct start time of a timetable and to switch the synchronization service on and off. The SyncState outport shows you the current state of synchronization (synchronized/not synchronized).

I/O characteristics

The table shows the block inport:

Simulink Input	**Range**	**Simulink Data Type**	**Meaning**
enableSync	0 / 1	Double	Enables or disables the synchronization service. If the value is > 0.5, the synchronization service is activated.

The table shows the block outport:

Simulink Output	**Range**	**Simulink Data Type**	**Meaning**
ttStartTime	–	Vector (2xUInt32)	Delivers the start time (microticks and macroticks of the dSPACE timer device) of the next timetable activation.
syncState	0 / 1	Double	Indicates the status of the synchronization: ▪ 0: FlexRay host is not synchronized ▪ 1: FlexRay host is synchronized

Dialog settings

The dialog settings can be specified on the following pages:

- Unit Page (RTIFLEXRAYCONFIG SYNC SERVICE) on page 117
- Options Page (RTIFLEXRAYCONFIG SYNC SERVICE) on page 117

Related topics

References

Overview of the Blocks for Configuration Generation.. 15
RTIFLEXRAYCONFIG TIMETABLE START... 61

Unit Page (RTIFLEXRAYCONFIG SYNC SERVICE)

Purpose

To specify the basic settings for timer synchronization.

Parameters

Start up synchronization mode Lets you select the synchronization mode:

- SMOOTH: The synchronization algorithm does not use more macroticks for correction than specified by the SyncLimit parameter.
- HARD: Synchronization is performed after the first communication cycle. The SyncLimit parameter is ignored.

Synchronization limit Displays the value of the SyncLimit parameter (read-only).

The SyncLimit parameter is determined using the following formulas:

- For a DS1006 with DS4501: `SyncLimit = 32 · MacrotickLength`
- For a DS1007 with DS4501: `SyncLimit = 21 · MacrotickLength`
- For a MicroAutoBox II: `SyncLimit = 36 · MacrotickLength`

Synchronization threshold Displays the synchronization threshold (read-only). If the difference between local and global time is less than the threshold value, the FlexRay host is set to the 'synchronized' state. The synchronization state is reset to 'unsynchronized' if the difference between local and global time is greater than the SyncLimit parameter.

The synchronization threshold is set using the following formulas:

- For a DS1006 with DS4501: `SyncThreshold = 4 · MacrotickLength`
- For a DS1007 with DS4501: `SyncThreshold = 3 · MacrotickLength`
- For a MicroAutoBox II: `SyncThreshold = 4 · MacrotickLength`

Related topics

References

Block Description (RTIFLEXRAYCONFIG SYNC SERVICE)..115

Options Page (RTIFLEXRAYCONFIG SYNC SERVICE)

Purpose

To enable or disable the block synchronization control inport and synchronization status outport.

> **Note**
>
> The block is added to the model during the generation process. The block parameters are read-only.

Parameters

Enable synchronization control inport Lets you enable the EnableSync inport that is used to control synchronization (enable/disable).

If synchronization is disabled, the outport still delivers a start time that must be fed into the RTIFLEXRAYCONFIG TIMETABLE START block of the synchronization task. This start time is calculated to match the planned activation time of the synchronization task. The calculated start time does not feature any correction terms as with enabled synchronization.

Enable synchronization status outport Lets you enable the syncState outport. The outport returns the following values:

Value	Meaning
0	FlexRay host is not synchronized with the FlexRay bus (cluster).
1	FlexRay host is synchronized with the FlexRay bus (cluster).

Related topics

References

Block Description (RTIFLEXRAYCONFIG SYNC SERVICE).. 115

RTIFLEXRAYCONFIG Cycle Counter Read

Purpose

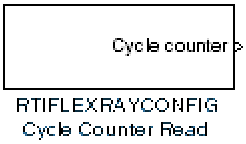
To read the cycle counter of the RTIFLEXRAYCONFIG configuration with configuration ID '-'.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG Cycle Counter Read)

Block



Purpose

To read the cycle counter of the RTIFLEXRAYCONFIG configuration with configuration ID '-'.

> **Note**
>
> The block is added to the model during the generation process.

Description

The **RTIFLEXRAYCONFIG Cycle Counter Read** block is part of the synchronization task (refer to Synchronization Task Subsystem on page 120) with configuration ID '-'.

In the FlexRay configurations with configuration ID 1, 2, or 3, the **RTIFLEXRAYCONFIG Status** block is used instead.

I/O characteristics

The table shows the block inport:

Simulink Output	Range	Simulink Data Type	Meaning
Cycle counter	0 ... 63	UInt32	Displays the value of the FlexRay communication counter.

Dialog settings

The block has no dialog settings.

Related topics

References

Overview of the Blocks for Configuration Generation... 15

Parameters Page (RTIFLEXRAYCONFIG Cycle Counter Read)

Purpose

To display the parameters set for the RTIFLEXRAYCONFIG Cycle Counter Read block.

Dialog settings

No parameters.

Related topics

References

Block Description (RTIFLEXRAYCONFIG Cycle Counter Read).. 119

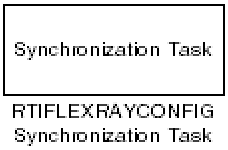
Synchronization Task Subsystem

Introduction

The following subsystem is used to synchronize the application tasks and the FlexRay bus.

RTIFLEXRAYCONFIG Synchronization Task

Subsystem



Purpose

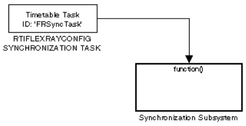
To synchronize the application tasks and the FlexRay bus.

Note

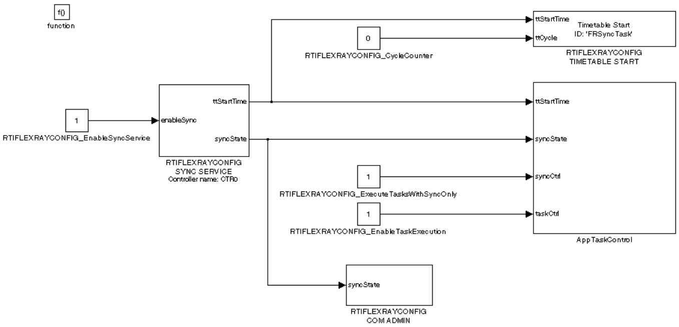
- A ground state task is added to the model during the generation process.
- You cannot change the ground state parameters within this block. The parameters are read-only. You can change only the Start up synchronization mode parameter in the RTIFLEXRAYCONFIG SYNC SERVICE block.

Description

The synchronization task synchronizes the application tasks and the FlexRay bus. The RTIFLEXRAYCONFIG Synchronization Task subsystem consists of several subsystems that contain two RTIFLEXRAYCONFIG TIMETABLE START blocks to start the FlexRay application tasks and the synchronization task itself. The synchronization task also contains an RTIFLEXRAYCONFIG TIMETABLE TASK block which drives the current synchronization subsystem:



The subsystem contains an RTIFLEXRAYCONFIG SYNC SERVICE block and an RTIFLEXRAYCONFIG TIMETABLE START block used to synchronize the FlexRay cluster with the dSPACE system and to start the application timetable tasks and the synchronization task again. Additionally, the subsystem contains an RTIFLEXRAYCONFIG COM ADMIN block to switch the communication layer depending on the synchronization state to on or off. The subsystem has the following structure:



The displayed AppTaskControl block is a subsystem which can be used to control the application tasks. You can control the application with ControlDesk by means of variables:

Start and stop application timetable tasks Modify the `RTIFLEXRAYCONFIG_EnableTaskExecution` variable to start and stop tasks without stopping the simulation:

- 1: Starts FlexRay application tasks.
- 0: Stops FlexRay application tasks.

Enable and disable the synchronization service Modify the `RTIFLEXRAYCONFIG_EnableSyncService` variable to enable or disable the synchronization service:

- 1: Enables the FlexRay synchronization service.
- 0: Disables the FlexRay synchronization service.

Inform whether or not the host is synchronized Observe the syncState outport to monitor the synchronization state. For information on the status outport, see RTIFLEXRAYCONFIG SYNC SERVICE on page 115.

Execution of application tasks even if the host is not synchronized By default, application tasks are executed only if the host and the FlexRay cluster are synchronized (syncState = 1). It is also possible to execute application tasks independently of the synchronization state. Use ControlDesk to set the RTIFLEXRAY_ExecuteTasksWithSyncOnly Constant block to “0”. The execution of application tasks while the host and the FlexRay cluster are not synchronized is important, for example, if no connection to a FlexRay cluster exists and application tasks must be executed for the purpose of testing execution time.

Related topics

References

Overview of the Blocks for Configuration Generation...15
RTIFLEXRAYCONFIG COM ADMIN..68

Blocks for Multiple Bus Simulation

Introduction

The blocks are additionally used if you want to create a real-time model for more than one FlexRay bus (based on several FlexRay configurations).

Where to go from here

Information in this section

Information in other sections

Modeling Several FlexRay Buses on One dSPACE Real-Time System (FlexRay Configuration Features 🕮)
You can use RTIFLEXRAYCONFIG blocks of different FlexRay configurations in one real-time model, for example, to model a gateway.

How to Create Configurations for Multiple Buses (FlexRay Configuration Tool Guide 🕮)
One FlexRay configuration project can contain the configuration of only one FlexRay bus. If you want to use several buses in one Simulink model, for example, to model a gateway, you must create several FlexRay projects with several configurations.

RTIFLEXRAYCONFIG TTE TIMETABLE START

Purpose

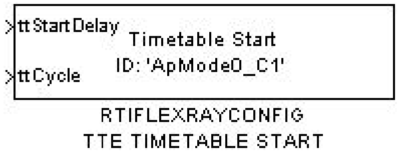
To start assigned timetable tasks for FlexRay buses with configuration IDs 1, 2, and 3.

Where to go from here

Information in this section

Block Description (RTIFLEXRAYCONFIG TTE TIMETABLE START)

Block



Purpose

To start assigned timetable tasks for FlexRay buses with configuration IDs 1 ,2, and 3.

Note

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

Description

The RTIFLEXRAYCONFIG TTE TIMETABLE START block is part of the synchronization task (refer to TTE Synchronization Task Subsystem on page 127) of FlexRay buses with configuration IDs 1, 2, and 3. The timetable which corresponds to an RTIFLEXRAYCONFIG TTE TIMETABLE START block is started each time the block is evaluated. One RTIFLEXRAYCONFIG TTE TIMETABLE START block can start several timetable tasks with the same timetable ID. An absolute time on the dSPACE system or a delay time that is set relative to the point in time of block execution can be specified by using the optional inport. The tasks are executed using the RTIFLEXRAYCONFIG TIMETABLE TASK or RTIFLEXRAYCONFIG CUSTOM CODE blocks.

I/O characteristics

The table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
ttStartDelay	–	Double or Vector (2 UInt32)	The time to start task execution. The **Start Time inport data type** option on the **Parameters** page specifies whether it is an absolute start time (microticks and macroticks of the dSPACE timer device) or a delay time in seconds.
ttCycle	0 ... 63	UInt32	The number of timetable cycles

Dialog settings

The dialog settings can be specified on the following pages:

- Parameters Page (RTIFLEXRAYCONFIG TTE TIMETABLE START) on page 125

Related topics

References

Overview of the Blocks for Configuration Generation.. 15

Parameters Page (RTIFLEXRAYCONFIG TTE TIMETABLE START)

Purpose

To display the parameters set for starting the related timetable task.

> **Note**
>
> The block parameters are read-only. They are specified during the generation process.

Dialog settings

Timetable ID Displays the name of the timetable currently assigned to the timetable tasks. The string length must be in the range 1 ... 10. The valid characters are “a ... z”, “A ... Z”, “0 ... 9” and “_”.

Number of timetable cycles Displays the number of timetable cycles. This edit field is disabled if the block is used for a FlexRay model.

Sample time Lets you enter the sample time (the intervals the block is executed at).

Sample Time	Meaning
–1	Inherited sample time
= 0	Discrete sample time of the subsystem
> 0	Discrete sample time as specified

Start timetable once at simulation start Shows whether to automatically start the timetable once at the beginning of the simulation.

Enable timetable start time inport Shows whether the timetable start-time inport is enabled or disabled. The inport allows you to specify a delay time for the timetable start.

Start-time inport data type Shows the selected data type of the start-time inport signal:

Variable	Data Type	Meaning
delay time (Double)	Double	Supports the input of a delay time.
absolute time (time stamp)	Vector (2 UInt32)	Supports the input of an absolute time (time stamp), for example, by connecting an RTIFLEXRAYCONFIG SYNC SERVICE block (ttStartTime outport).

Related topics

References

Block Description (RTIFLEXRAYCONFIG TTE TIMETABLE START)... 124
RTIFLEXRAYCONFIG SYNC SERVICE.. 115

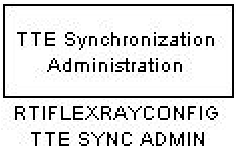
RTIFLEXRAYCONFIG TTE SYNC ADMIN

Purpose

To generate code for cluster and dispatcher cycle administration for RTIFLEXRAYCONFIG configurations with configuration ID 1, 2 or 3.

Block Description (RTIFLEXRAYCONFIG TTE SYNC ADMIN)

Block



Purpose

To generate code for cluster and dispatcher cycle administration of RTIFLEXRAYCONFIG configurations with configuration ID 1, 2 or 3.

Note

The block is added to the model during the generation process.

Description

This block is only used in TTE synchronization task subsystems. For details, refer to TTE Synchronization Task Subsystem on page 127.

Dialog settings

The block has no dialog settings.

Related topics

References

Overview of the Blocks for Configuration Generation... 15

TTE Synchronization Task Subsystem

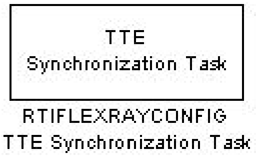
Introduction

The following subsystem is used to synchronize the application tasks and the FlexRay bus of FlexRay configuration with configuration IDs 1, 2, or 3.

For configuration with configuration ID '-', the Synchronization Task Subsystem is used. For details, refer to Synchronization Task Subsystem on page 120.

RTIFLEXRAYCONFIG TTE Synchronization Task

Subsystem



Purpose

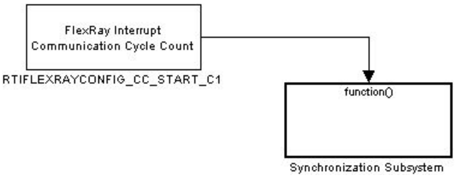
To synchronize the application tasks and the FlexRay bus with configuration IDs 1, 2, 3.

> **Note**
>
> - TTE Synchronization tasks are added to the model during the generation process for FlexRay configurations 1, 2, and 3.
> - For configuration with configuration ID '-', the Synchronization Task Subsystem is used. Refer to Synchronization Task Subsystem on page 120.

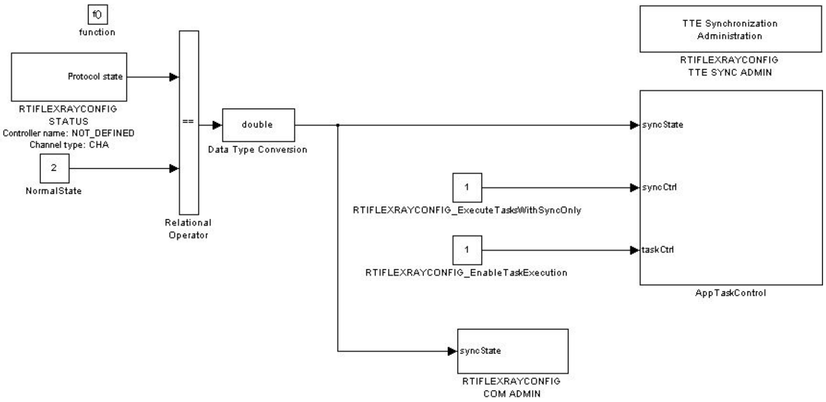
Description

The synchronization task synchronizes the application tasks and the FlexRay bus of configurations 1, 2, and 3. The subsystem is triggered by the Communication Cycle Count FlexRay interrupt of the FlexRay bus with the matching configuration ID. The RTIFLEXRAYCONFIG INTERRUPT block makes the interrupt available.

> **Note**
>
> The communication cycle count interrupt stops if the selected FlexRay controller is not in the NORMAL state. This can happen when the membership of the controller is disabled or the FlexRay controller itself changes its state caused by an external event. If the interrupt stops, FlexRay data is no longer received or transmitted.


FlexRay Interrupt
Communication Cycle Count
RTIFLEXRAYCONFIG_CC_START_C1
function()
Synchronization Subsystem

The subsystem contains an RTIFLEXRAYCONFIG STATUS block that provides the status of the FlexRay bus with the matching configuration ID. If its Protocol state output is 2 (normal active operation), the FlexRay cluster is synchronized with the controller selected with the RTIFLEXRAYCONFIG Status block, i.e., the application timetable tasks and the synchronization task are restarted. Additionally, the subsystem contains an RTIFLEXRAYCONFIG COM ADMIN block to switch the communication layer "on" and "off" according to the synchronization state. The subsystem has the following structure:



The AppTaskControl block is a subsystem which can be used to control the application tasks. You can control the application with ControlDesk by means of variables:

Start and stop application timetable tasks Modify the `RTIFLEXRAYCONFIG_EnableTaskExecution` variable to start and stop tasks without stopping the simulation:

- 1: Starts FlexRay application tasks.
- 0: Stops FlexRay application tasks.

Execute application tasks even if the host is not synchronized By default, application tasks are executed only if the host and the FlexRay cluster are synchronized (syncState = 1). It is also possible to execute application tasks independently of the synchronization state. Use ControlDesk to set the RTIFLEXRAY_ExecuteTasksWithSyncOnly Constant block to "0".

Related topics

References

Overview of the Blocks for Configuration Generation 15
RTIFLEXRAYCONFIG COM ADMIN 68
RTIFLEXRAYCONFIG INTERRUPT 48
RTIFLEXRAYCONFIG STATUS 38

RTI FlexRay Configuration Commands

rtiflexrayconfig_modelgenerate

Syntax

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

Purpose

To generate configured RTI FlexRay blocks from the Simulink configuration data.

Description

The FlexRay Configuration Tool generates Simulink configuration data. The `rtiflexrayconfig_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 16 and Structure of the Automatically Generated FlexRay Model on page 17.

Since dSPACE FlexRay Configuration Blockset 1.5, you can 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 RTIFLEXRAYCONFIG UPDATE block is time-consuming. For details, refer to How to Update the Automatically Generated FlexRay Model (FlexRay Configuration Features 🕮).

> **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 information required by the update process. Do not change or remove the MAT file 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 basis of the configuration data. 'Update': The command updates the automatically generated FlexRay model.
Blockset	▪ PDU	Specifies the usage of PDU-based modeling and generates the FlexRay model with PDU blocks.
SortOrder	▪ PDUName ▪ PDUTiming The default value is 'PDUName'.	Specifies the sort order of the generated PDU blocks.

Return value

errorcode 0 if no error occurred, 1 if an error occurred.

Examples

Generating an automatically generated PDU-based FlexRay model Enter

```
rtiflexrayconfig_modelgenerate('MyProject_data.m')
```

or

```
rtiflexrayconfig_modelgenerate('MyProject_data.m', 'BlockSet', 'PDU')
```

in the MATLAB Command Window to generate an automatically generated PDU-based FlexRay model on the basis of 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 To update an automatically generated PDU-based model, enter

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

or

```
rtiflexrayconfig_modelgenerate('MyProject_data.m', 'Blockset', 'PDU',
'GenerationMode', 'Update')
```

in the MATLAB Command Window.

`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 RTI Blocks for Designing a FlexRay Node (FlexRay Configuration Features 📖)
How to Update the Automatically Generated FlexRay Model (FlexRay Configuration Features 📖)
How to Update the FlexRay Blocks in Simulink Models (FlexRay Configuration Features 📖)

References

RTIFLEXRAYCONFIG UPDATE.. 70