dSPACE FlexRay Configuration Package
Configuring dSPACE systems in FlexRay networks

Highlights
- Efficient configuration of FlexRay simulations
- For rapid control prototyping and hardware-in-the-loop simulation, including restbus simulation
- Support of end-to-end protection, secure onboard-communication and global time synchronization

Application Areas
The dSPACE FlexRay Configuration Package is used to integrate dSPACE hardware as simulation or monitoring nodes in a FlexRay network. Nodes are configured with the dSPACE FlexRay Configuration Tool according to a communication matrix containing scheduling information for signals and frames transmitted via the FlexRay bus. The communication information is linked to a MATLAB®/Simulink® model via the RTI FlexRay Configuration Blockset. The resulting FlexRay application can be executed on a dSPACE system.

Key Benefits
The dSPACE FlexRay Configuration Package is an extensive solution for using FlexRay in dSPACE’s real time systems and consists of two parts: dSPACE FlexRay Configuration Tool and RTI FlexRay Configuration Blockset. The package makes FlexRay configurations easy and hides much of FlexRay’s complexity. Configuration settings can be stored as and exchanged with project files. The dSPACE FlexRay Configuration Package has been designed in close cooperation with advanced users of the FlexRay protocol to ensure it meets your requirements.

dSPACE FlexRay Configuration Tool
The dSPACE FlexRay Configuration Tool lets you configure a dSPACE system as a simulation node in a FlexRay network. It relies on the network and scheduling data available in a FIBEX or AUTOSAR XML representation. Numerous consistency checks are performed when the communication description is imported. Various views help in managing the FlexRay configuration. The tool generates the communication code and controller configuration.

RTI FlexRay Configuration Blockset
Application-specific Simulink models can be created using the RTI FlexRay Configuration Blockset as a basis. The block attributes are filled with data generated by the dSPACE FlexRay Configuration Tool. The blockset contains additional blocks that can be used for task execution control, interrupt and error handling, status information, and controller reset. The RTI FlexRay Configuration Blockset supports the sending and receiving of protocol data units (PDUs), which are also used in AUTOSAR. Such units comprise several signals, which can be handled in the model using only one Simulink block per PDU.
# Functionality Overview

**dSPACE FlexRay Configuration Tool**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
</table>
| Importing communication descriptions<sup>1)</sup> | - Reads communication descriptions for FlexRay and validates via plausibility checks  
- Ignores irrelevant data  
- Support of various FIBEX versions<sup>2)</sup>  
- Support of various AUTOSAR System Template versions<sup>2)</sup> |
| Handling | - Hierarchical view of the communication description  
- Visualizes parameters  
- Filter and view options, as well as search masks for handling larger descriptions  
- Generates a report for the present configuration |
| Signal and frame selection | - Select signals, PDUs and frames for creating blocks with the RTI FlexRay Configuration Blockset  
(for use in simulation)  
- All the frames sent to a real ECU are automatically identified and configured for simulation (restbus simulation) |
| PDU handling | - PDU update bit manipulation  
- Alive counter for PDUs  
- CRC check for PDUs by user CRC C-code file  
- Raw data access for PDUs |
| Frame handling | - Raw data access for frames with or without signals  
- CRC check for frames by user CRC C-code file  
- Enable/disable static frames via software: A null frame or old value (CHI Code option) is sent  
- Enable/disable buffers of static slots via hardware: There is no bus activity in the slots  
- NEW: RX time stamp support |
| Task configuration | - Generate and configure communication tasks automatically or manually and according to the selected signals, PDUs and frames  
- Consistency checks for eliminating invalid configurations in tasks  
- Configure dSPACE hardware, including the connected FlexRay communication controllers |
| Support of AUTOSAR functions | - Container IPDU  
- End-to-end protection  
- Secure onboard communication  
- Global time synchronization |

<sup>1</sup> dSPACE tools for FlexRay rely on the existence of a communication description containing information for time-triggered message scheduling. The information is used to configure a dSPACE hardware platform to communicate messages, using a MATLAB/Simulink model for handling the messages.

<sup>2</sup> Please contact dSPACE for appropriate version.

**RTI FlexRay Configuration Blockset**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
</table>
| Model generation | - Generates MATLAB/Simulink blocks for FlexRay communication  
- Combine the model frames with application models, for example, for function prototyping or restbus simulation  
- Blocks for FlexRay controllers, tasks, and synchronization settings  
- Trigger blocks for dynamic frames  
- Blocks for sending and receiving protocol data units (PDUs) |
| Simulation stage | - Use a dSPACE platform as a monitoring or simulation node in a FlexRay communication system |
### Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>dSPACE FlexRay Configuration Package</td>
<td>FLEXRAY_BUS</td>
</tr>
</tbody>
</table>

### Relevant Software and Hardware

#### Software

<table>
<thead>
<tr>
<th>Required</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>For PHS-based systems, MicroAutoBox II, and MicroLabBox</td>
<td>Real-Time Interface&lt;sup&gt;1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>For SCALEXIO systems and MicroAutoBox III</td>
<td>ConfigurationDesk</td>
</tr>
<tr>
<td>For PHS-based systems</td>
<td>GNU C compiler for AMD Opteron&lt;sup&gt;2)&lt;/sup&gt; for DS1006</td>
</tr>
<tr>
<td>For MicroAutoBox II</td>
<td>Microtec PowerPC C Compiler</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optional</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>For all dSPACE systems</td>
<td>ControlDesk with Bus Navigator Module</td>
</tr>
</tbody>
</table>

#### Hardware

<table>
<thead>
<tr>
<th>Optional</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>For PHS-based systems</td>
<td>DS1006 Processor Board&lt;sup&gt;2)&lt;/sup&gt;</td>
</tr>
<tr>
<td>For MicroAutoBox</td>
<td>MicroAutoBox II</td>
</tr>
<tr>
<td>See relevant product information</td>
<td>MicroAutoBox III</td>
</tr>
<tr>
<td>See relevant product information</td>
<td>Interface modules with FlexRay communication controller, such as the DS4340 FlexRay Interface Module&lt;sup&gt;3)&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For SCALEXIO systems</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCALEXIO Processing Unit</td>
<td>DS2671</td>
</tr>
<tr>
<td>DS2680 I/O Unit (with bus support)</td>
<td>DS2680_2672</td>
</tr>
<tr>
<td>DS6311 FlexRay Board</td>
<td>DS6311</td>
</tr>
</tbody>
</table>

<sup>1)</sup> For information on standard hardware and software requirements for Real-Time Interface (RTI), please see the relevant product information.

<sup>2)</sup> With DS4505 Interface Board or the FlexRay Interface Board based on DS4501.

<sup>3)</sup> For further available products and the latest version information, please contact dSPACE.
Creating a Task Schedule

You can create a task schedule by selecting signals, frames for FIBEX 2.x and PDUs for FIBEX 3.x and AUTOSAR. The schedule includes communication routines for sending and receiving FlexRay frames, for both the static and the dynamic parts of the communication cycle. It also covers application tasks for your functional or restbus simulation models. The task schedule can be derived automatically according to a fixed scheme for positioning communication routines. Alternatively, you have full control to define it manually. When you do so, various checks are performed in the background to ensure the task schedule you define is consistent. The third and final category of tasks covers the synchronization task. This is executed at the end of each application cycle to align task execution to the global time on the FlexRay bus.
Code Generation
The configuration tool has been given all the information it needs to generate the actual communication code and the settings for initializing the FlexRay controller. The tool also calculates the required number of FlexRay controllers, taking into account the available communication buffers, and startup and synchronization behavior. The code generator is prepared to support NXP (Freescale) and Bosch E-Ray controller implementations for FlexRay.

Creating the FlexRay Model
You can now take the results of the configuration tool, which acts as a kind of preprocessor tool, and continue with the usual model-based design flow. When a FlexRay model is created for the first time, blocks from the RTI FlexRay Configuration Blockset library are copied to the model and their parameter values are set automatically according to the configuration data that was generated previously. The resulting model frame represents a complete interface to the FlexRay network and a local task schedule. It can be supplemented by the actual functional or restbus simulation models and further blocks from the library, for example, for receiving status information and handling error situations.

Modifying the Model
Later on, you will most likely face modifications to the communication description file, representing new integration stages in the vehicle project. To preserve the modeling results already obtained, the RTI FlexRay Configuration Blockset comes with an update mechanism for handling changes in the configuration data, for example, introducing new signal and PDU blocks and discarding old ones. The resulting FlexRay application model is compiled for execution on a dSPACE hardware system. The driver and initialization code of the configuration tool is integrated during this build process. The generated code is downloaded to the dSPACE hardware and acts as a full-fledged node in the FlexRay network, sending and receiving FlexRay frames in real time.

Selected Failure Simulation Methods
- Failure and restart of a FlexRay controller
- Enable/disable transmission of a static frame
- Enable/disable transmission of a cyclic dynamic FlexRay frame
- Enable/disable all event-based dynamic FlexRay frames
- Enable/disable all cyclic FlexRay frames (a null frame or old value is sent)
- Alive counter manipulation via TRC file
- Change CRC algorithms at run time
- Send and receive invalid signals
- Simulate synchronization service breakdown
- Simulate time-triggered task execution breakdown with optional restart at correct synchronization

For more information on supported controllers, please contact dSPACE.