FlexRay Technology at dSPACE
Contents

FlexRay Technology 3

dSPACE Products for FlexRay Applications 4
  dSPACE FlexRay Configuration Package 5
  RTI Bypass Blockset 7
  NEW: ControlDesk® Next Generation 8
DS4505 FlexRay Interface Board 10
DS4340 FlexRay Interface Module 10
DS1450 Bus FIU Board 11
  NEW: MicroAutoBox II 11

Application Areas 12
Use Cases 12
  Rapid Control Prototyping within a FlexRay Network 12
  Task-Synchronous Bypassing via XCP on FlexRay 14
  Simulating Nodes of a FlexRay Network 16
  ECU Calibration and Measurement via XCP on FlexRay 18

Application Examples 19
FlexRay Technology

About FlexRay

**The FlexRay Standard**

FlexRay has established itself as a de-facto standard for in-vehicle, time-triggered communication systems. The basic principle of time-triggered systems is that tasks are executed and messages are sent according to a pre-defined schedule. In FlexRay, activities are aligned to a global time base, whereas in nondeterministic communication protocols such as CAN, they are mainly event-triggered at run time. FlexRay is targeted to support data rates of up to 10 Mbit/s with increased flexibility for easy system extension and the dynamic use of bandwidth.

For more information on the FlexRay standard, see [www.flexray.com](http://www.flexray.com)

**The FlexRay Roadmap at dSPACE**

Our product range has been designed in close cooperation with advanced users of the FlexRay protocol. It provides comprehensive FlexRay support, ranging from hardware such as prototyping systems and I/O boards with FlexRay interfaces to software for the real-time simulation of models in FlexRay networks. Development tasks like function prototyping and testing FlexRay ECUs are supported. We will continue our FlexRay activities, creating products that are tailor-made for our systems. To help protect our customers’ investments, we are committed to standards that are future-proof.
dSPACE Products for FlexRay Applications
Tools for controller development

- Powerful products for designing and testing FlexRay applications
- Proven automotive development solutions
- Scalable from small to large development systems
- Single-source product offering
- Simple configuration of FlexRay systems
- Commitment to standards
- Designed in close cooperation with advanced users of the FlexRay protocol

**Configuration of FlexRay Systems**
When you use dSPACE tools for FlexRay, you have proven automotive development solutions combined with an emerging communication system for the next generation of cars. You can completely configure a dSPACE platform which is connected to the FlexRay bus. You can also integrate further tools from dSPACE for experimentation, visualization, or tests. Known and proven features of dSPACE systems can be used with the FlexRay application.

dSPACE is committed to standards and uses ASAM-MCD2 FBX (FIBEX) and AUTOSAR System Templates for the description of the network topology, signals, frames or PDUs, with timing information.

**Support of FlexRay Communication Protocol (Version 2.1)**
- Static and dynamic messages
- Read and write messages
- Generating communication drivers and controller configuration
- Network management messages according to the AUTOSAR standard

**The FIBEX Standard**
dSPACE uses, amongst others, the FIBEX data exchange format (FIBEX = Field Bus EXchange, ASAM-MCD2 FBX) for describing FlexRay communication schedules. It is an XML-based, common file format for all automotive frame-based bus communication.
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 MathWorks® Simulink® model via the RTI FlexRay Configuration Blockset. The resulting FlexRay application can be executed on a dSPACE system.

**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 MathWorks® 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 MicroAutoBox or modular systems. The package makes FlexRay configurations easy and hides much of FlexRay’s complexity. Configuration settings can be stored as project files. The number of FlexRay controllers needed can be optimized. The dSPACE FlexRay Configuration Package has been designed in close cooperation with advanced users of the FlexRay protocol to ensure it meets your requirements. The package supports two FlexRay channels.

**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 System Template (*.arxml) representation. Numerous consistency checks are performed when the FIBEX or AUTOSAR System Template file 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.

**Selecting Signals, PDUs and Frames**
The dSPACE FlexRay Configuration Tool is the bridge between the network or system level view and the node- or ECU-specific view. After a FIBEX or AUTOSAR System Template file is imported, the FlexRay network description and scheduling data are displayed in a clear hierarchical view. This is combined with additional display and sorting options, and you can easily select all the
signals, frames for FIBEX 2.x and PDUs for FIBEX 3.0 / FIBEX 3.1, and AUTOSAR System Templates\(^1\) you want to use in your simulation. If you want to set up a restbus simulation for a single ECU, just select the ECU and let the tool look up all the frames sent to it.

**Creating a Task Schedule**

You can create a task schedule by selecting signals, frames for FIBEX 2.x and PDUs for FIBEX 3.0 / FIBEX 3.1 or AUTOSAR System Templates\(^1\). 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 Freescale and Bosch E-Ray controller implementations for FlexRay\(^2\).

**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 network 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 frame 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.

For more information on the dSPACE FlexRay Configuration Package, please see


---

\(^1\) For information on supported versions, please contact dSPACE.

\(^2\) For more information on supported controllers, please contact dSPACE.
RTI Bypass Blockset

- Dialog-based configuration of bypass applications and real-time ECU access
- Integrated A2L file browser for selecting ECU variables
- Failure checking and data consistency mechanisms for bypass communication
- Powerful API for configuring ECU interface and bypass blocks via scripts
- Wide portfolio of bypass interfaces including XCP on FlexRay

**Application Areas**

For developing ECU functions by means of the external bypass approach, it is usually necessary to configure the bypass interface in the modeling environment and to change the input and output signals of the bypass model without ECU code modifications. The RTI Bypass Blockset provides a generic user interface for this purpose, with the same look and feel no matter what ECU interface is actually used for bypassing.

**Key Benefits**

The RTI Bypass Blockset lets you concentrate on designing bypass functions for ECUs and makes it easy for you to configure the bypass interface to the ECU via a dialog-based user interface in Simulink. A broad selection of ECU interfaces is supported, for example, XCP on CAN, XCP on Ethernet and XCP on FlexRay. The same blockset can also be used for ECU test purposes, allowing dSPACE hardware-in-the-loop simulators to read and write ECU variables in real-time.

**Real-time ECU Access via XCP on FlexRay**

The XCP on FlexRay option of the RTI Bypass Blockset is based on the dSPACE FlexRay Configuration Package. The configuration tool in the package lets you assign the XCP master node and select and configure the XCP slots of the FlexRay¹ cycle which are to be used for bypassing, using the information in the ASAM MCD2 FBX (FIBEX) file. A Simulink library that matches the FlexRay configuration is generated, after which the bypass model can be implemented. For this, dedicated blocks of the RTI Bypass Blockset allow you to select a matching XCP slot and to associate this slot with variables to be read from and written to the ECU. The set of variables available for selection is in the ECU description (A2L) file. You can also create new variables based on any ECU address with the bypass blockset. Synchronization between the ECU and the rapid prototyping system can be implemented using either the FlexRay message schedule or interrupts which trigger the bypass model calculation as soon as all the model inputs are available.
NEW: ControlDesk® Next Generation

- Universal, modular experiment and instrumentation software for ECU development
- Integrated ECU calibration, measurement and diagnostics access (CCP, XCP, ODX)
- Synchronized data capture across ECUs, RCP and HIL platforms, and bus systems
- Powerful layouting, instrumentation, measurement and post-processing

Application Areas
ControlDesk Next Generation is the dSPACE experiment software for seamless ECU development. It performs all the necessary tasks and gives you a single working environment, from the start of experimentation right to the end. These are some of the tasks it can be used for:
- Rapid control prototyping (fullpass, bypass)
- Hardware-in-the-loop simulation
- ECU measurement, calibration, and diagnostics
- Access to vehicle bus systems (CAN, LIN, FlexRay)
- Access to PC offline simulation

Key Benefits
ControlDesk Next Generation unites functionalities that until today required several specialized tools. It provides access to simulation platforms as well as to connected bus systems and can perform measurement, calibration and diagnostics on ECUs, e.g., via standardized ASAM interfaces. Its flexible modular structure provides high scalability to meet the requirements of specific application cases. This gives you clear advantages in terms of handling, the amount of training needed, the required computing power, and costs.

FlexRay Communication with ControlDesk Next Generation
- ControlDesk Next Generation has an optional module, the Bus Navigator, which lets you handle several types of items for all the platforms in a project, for example, settings in the dSPACE FlexRay Configuration Package (used for configuring dSPACE systems in FlexRay networks). With the help of the Bus Navigator, you can manipulate signals, frames, and PDUs before transmission, exclude them from being transmitted, etc.
- To configure an XCP on FlexRay device, you can select the FlexRay interface and configure XCP and FlexRay features. The configuration parameters are taken from the variable description and the FIBEX file which is referenced in the variable description file. The settings can be changed if necessary.
- ControlDesk Next Generation supports quick start measurements on ECUs with XCP on FlexRay.

XCP on FlexRay
The Universal Measurement and Calibration Protocol (XCP) is the successor to the well established CAN Calibration Protocol (CCP). XCP offers strict separation of the protocol and transport layers and allows implementation on both present and future in-vehicle communication standards. XCP on FlexRay is part of the XCP family and standardized by the Association for Standardisation of Automation and Measuring Systems (ASAM). XCP is based on a master/slave concept with the ECU as the XCP slave. The FlexRay bus and XCP on FlexRay are gaining in importance in an increasing number of ECU projects.

© ControlDesk Next Generation can access (via XCP on Ethernet) virtual ECUs generated with SystemDesk® and Simulink® plant models that are simulated by PC offline simulation. For more details on PC offline simulation with ControlDesk Next Generation, please contact dSPACE.
Synchronous Data Acquisition

Multiple data acquisitions with ControlDesk Next Generation can be very useful in FlexRay development. For example, a FlexRay setup for brake-by-wire development could include four MicroAutoBoxes (for controlling dedicated brake actuators) and one AutoBox (for executing the control algorithm). With ControlDesk Next Generation, data can be captured and recorded from these different platforms. The data can be synchronized to a global time, so that detailed timing analysis can be performed on it with the ControlDesk Next Generation plotter instrument. These recordings can also be saved by ControlDesk Next Generation for further postprocessing tasks.

For more information on ControlDesk Next Generation, please see www.dspace.com/goto?ControlDesk
DS4505 FlexRay Interface Board

**Application Areas**
The DS4505 FlexRay Interface Board allows a dSPACE system with modular hardware to be connected to a FlexRay communication system. Specially developed for carrying the DS4340 FlexRay Interface Module, it increases the performance of the communication system. It is ideally suited to rapid control prototyping and testing FlexRay applications with a dSPACE prototyping system or Simulator. The board can carry up to four FlexRay Interface modules and can be used with the DS1005 PPC Board and DS1006 Processor Board.

**Key Benefits**
The DS4505 FlexRay Interface Board equipped with a FlexRay Interface Module enables you to test FlexRay applications with dSPACE modular hardware in a FlexRay network. Setting up and testing FlexRay communications is easy with the available toolset, such as the dSPACE FlexRay Configuration Blockset (p. 5) or ControlDesk Next Generation (p. 8). All this ensures full real-time support during controller development in FlexRay networks. For restbus simulation of FlexRay applications, the board supports flexible controller settings to ensure the bus behaves correctly on startup.

For more information on the DS4505, please see [www.dspace.com/goto?DS4505](http://www.dspace.com/goto?DS4505)

**FlexRay Interface Solution**
The FlexRay Interface solution allows a dSPACE system with modular hardware to be connected to a FlexRay communication system. It can be used for rapid control prototyping and testing FlexRay applications with a dSPACE prototyping system or Simulator. The solution is based on the DS4501 carrier board and can hold up to four FlexRay controller modules with IndustryPack (IP) format.

**DS4340 FlexRay Interface Module**
The FlexRay Interface Module can be used with the DS4505 FlexRay Interface Board, DS4501 IP carrier board, or MicroAutoBox II (1401/1511/1512 and 1401/1505/1507). The module supports FlexRay protocol specification 2.1. It provides a hardware-configurable, switchable termination circuit.
DS1450 Bus FIU Board

The DS1450 Bus FIU (Failure Insertion Unit) Board has been especially designed for modular dSPACE HIL systems. The unit offers a broad range of electrical failure insertion options for the physical layers, for example, of FlexRay. These options include circuit to ground/to UBat, open wire, and varying termination resistance. One DS1450 can support 4 FlexRay channels in parallel. The unit can be triggered from dSPACE’s experiment software ControlDesk Next Generation (additional Failure Simulation Module required). dSPACE offers specialized ControlDesk Next Generation failure patterns for the DS1450.

NEW: MicroAutoBox II

- 2nd generation of dSPACE’s robust and compact stand-alone prototyping unit
- High performance with IBM PowerPC running at 900 MHz
- Improved I/O performance
- Comprehensive I/O incl. CAN, LIN, K/L line, FlexRay, Ethernet, and LVDS/bypass interfaces

**Application Areas**

MicroAutoBox is a real-time system for performing fast function prototyping from scratch. It operates without user intervention, just like an ECU. MicroAutoBox can be used for many different rapid control prototyping (RCP) applications, for example:

- Chassis control
- Powertrain
- Body control
- Drives control
- Electric drives control
- X-by-wire applications
- Advanced driver assistance systems (ADAS)

**Key Benefits**

The special strength of the MicroAutoBox hardware is its unique combination of high-performance, comprehensive automotive I/O, and an extremely compact and robust design. In addition to the standard I/O, MicroAutoBox offers variants with FPGA functionality as well as variants with interfaces for all major automotive bus systems: CAN, LIN, K/L line, FlexRay, and Ethernet.

**FlexRay Applications with MicroAutoBox**

FlexRay is an automotive standard. The MicroAutoBox lets you test your FlexRay applications directly in the vehicle. The MicroAutoBox (1401/1511/1512, 1401/1505/1507, and 1401/1507) can hold up to 2 FlexRay Interface Modules and can be configured using the dSPACE FlexRay Configuration Package (p. 5).

For more information on MicroAutoBox II, please see www.dspace.com/goto?MicroAutoBox
Application Areas
For rapid control prototyping, hardware-in-the-loop simulation, and ECU calibration

- Developing and optimizing your control designs without manual programming
- Systematic ECU tests
- Restbus simulation
- ECU calibration and measurement

Use Case:
Rapid Control Prototyping within a FlexRay Network

Description
Rapid Control Prototyping is a method of developing and ensuring controller function models. The function models are simulated on the prototyping systems, which take the place of the electronic control units (ECUs) that are being developed or modified. This tests the functional behavior of the controllers and their interaction with the bus communication.

One or more nodes in a FlexRay network can be realized by dSPACE prototyping systems with FlexRay controller modules. Real-time simulation of the controller model, including startup behavior and error handling, and time-synchronous tracing of signals, can be performed on the prototyping system.

In this configuration example, MicroAutoBox is used for realizing the nodes. It is also possible to use dSPACE’s modular hardware, that is, the DS1005 PPC Board in combination with the DS4505 FlexRay Interface Board, mounted in an AutoBox, instead.

For time-synchronous tracing of signals dSPACE’s experiment software ControlDesk Next Generation (p. 8) can be used.

Working with dSPACE Tools
dSPACE hardware systems – from modular hardware (p. 10) to MicroAutoBox II (p. 11) – can be used for various aspects of FlexRay applications. The systems are equipped with slots for IP modules containing a FlexRay communication controller, such as the DS4340 FlexRay Interface Module (p. 10). The application models are created in MATLAB®/Simulink® in combination with the dSPACE FlexRay Configuration Package (p. 5). The models are executed on the hardware in real time. Model execution and bus access are synchronized. The results can be visualized in an experiment environment such as ControlDesk Next Generation (p. 8).
### Third-Party Components

**PC**

- **Modeling software**: MathWorks® MATLAB®/Simulink®/Stateflow®
- **Real-time code generation**: MathWorks® Real-Time Workshop® and Stateflow Coder®

### Software Components

**Implementation software**

- Real-Time Interface
- dSPACE FlexRay Configuration Package
- PowerPC Compiler

**Test and experiment software**

- ControlDesk Next Generation - Basic Version
- Standard Platforms Module
- MLIB/MTRACE

### Hardware Components

**IP modules with FlexRay communication controller, such as the DS4340 FlexRay Interface Module**

- MicroAutoBox hardware: MicroAutoBox II or
- Modular hardware: DS1005 PPC Board, DS4505 FlexRay Interface Board or

---

1) For available products and latest version information, please contact us.
Use Case:
Task-Synchronous Bypassing via XCP on FlexRay

Description
Due to the complexity of modern electronic control units (ECU) and the limited time available for the development of new ECU generations, the entire ECU software is only rarely developed from scratch. Typically, only the existing ECU code is adapted or extended. In this context the external bypass method is an efficient approach, allowing new algorithms to be developed on a rapid prototyping system while the original ECU executes all the functions that remain unchanged. The input and output variables of the bypass model are exchanged, and task execution on the ECU and the prototyping system is synchronized, via existing ECU interfaces such as XCP on FlexRay. The external bypass approach gives you great flexibility during the design phase, since you have almost no resource constraints such as RAM, ROM, processor performance, or I/O channels. Real-time behavior is guaranteed even with complex bypass functions. In addition, the autoboot options of the prototyping systems allow you to validate the behavior of the new functions in realistic scenarios, for example, during test drives. In this configuration example, the dSPACE MicroAutoBox is used as a real-time prototyping system. The bypass function is calculated synchronously to an ECU task. As soon as all the input data of the bypass model is available, an interrupt is triggered and the bypass task on the MicroAutoBox is executed. Alternatively, you can also synchronize task execution on the two systems by means of the FlexRay message schedule. The bypass task is then time-triggered and independent of the input data available.
### Third-Party Components

**PC**

- **Modeling software**: MathWorks® MATLAB®, Simulink®, Stateflow®
- **Real-time code generation**: MathWorks® Real-Time Workshop® and Stateflow Coder®

### Software Components

**Implementation software**

- Real-Time Interface
- RTI Bypass Blockset
- dSPACE FlexRay Configuration Package
- PowerPC Compiler

**Test and experiment software**

- ControlDesk Next Generation – Basic Version
- Standard Platforms Module

### Hardware Components

- **IP modules with FlexRay communication controller, such as the DS4340 FlexRay Interface Module**

**Rapid prototyping hardware**

- MicroAutoBox II with FlexRay interface modules or
- DS1005 PPC Board with DS4505 FlexRay Interface Board or
- DS1006 Processor Board with DS4505 FlexRay Interface Board

---

1) For available products and latest version information, please contact us.
Use Case:
Simulating Nodes of a FlexRay Network

Description
Restbus simulation is a method of testing distributed controller systems which are only partly available. A dSPACE Simulator emulates the bus nodes that are missing from the network. Controller models are substituted and the simulator sends the results to the nodes. Restbus simulation requires a communication schedule.

In this example, a dSPACE Simulator Mid-Size with the DS1005 PPC Board is used for simulating nodes. The DS1006 Processor board can also be mounted in the simulator instead.
## Third-Party Components

### PC
- Modeling software
  - MathWorks® MATLAB®, Simulink®, Stateflow®
- Real-time code generation
  - MathWorks® Real-Time Workshop® and Stateflow Coder®

## Software Components

### Implementation software
- Real-Time Interface
- dSPACE FlexRay Configuration Package
- PowerPC Compiler

### Test and experiment software
- ControlDesk Next Generation – Basic Version
- Standard Platforms Module
- MLIB/MTRACE

## Hardware Components

### IP modules with FlexRay communication controller, such as the DS4340 FlexRay Interface Module\(^1\)

### Modular hardware
- DS1005 PPC Board or DS1006 Processor Board
- DS4505 FlexRay Interface Board or
- FlexRay Interface (based on DS4501)

### Simulator hardware
- Simulator-specific hardware such as dSPACE Simulator Mid-Size

---

\(^1\) For available products and latest version information, please contact us.
Use Case:
ECU Calibration and Measurement via XCP on FlexRay

Description
ControlDesk Next Generation is a universal, modular experiment and instrumentation software for ECU development that supports all major ECU interfaces and protocols associated with these use cases. Due to the increasing relevance of FlexRay in the automotive industry, ControlDesk Next Generation also provides an XCP on FlexRay interface for ECU calibration and measurement. To interface to the ECU, users just have to import the respective ECU variable description file (A2L file) in ControlDesk Next Generation. The A2L file typically contains a reference to the associated ASAM MCD 2 FBX (FIBEX) database describing the FlexRay communication of all the network nodes available in the vehicle, including the XCP master. Information that is specific to XCP on FlexRay and details on XCP slots and the associated XCP packet types are also available in the A2L file.

Third-Party Components

<table>
<thead>
<tr>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlexRay hardware</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Software Components

<table>
<thead>
<tr>
<th>Measurement and calibration software</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ ControlDesk Next Generation - Basic Version</td>
</tr>
<tr>
<td>■ ECU Interface Module</td>
</tr>
</tbody>
</table>
Application Examples

**Damper Control**
Application:
- New damper control for dynamic stabilization, based on a distributed controller with FlexRay communication
Details:
- Production code for function model generated with TargetLink® in a largely automated development process
- Further tools equipped with FlexRay used
- ECUs tested with dSPACE Simulator

**Intelligent Wedge Brake**
Application:
- Developing and testing an intelligent wedge brake in a real vehicle
Details:
- In-vehicle tests with dSPACE prototyping system
- Use of FlexRay tools from dSPACE
- Excellent braking performance achieved

**Developing a Fully X-by-Wire Vehicle**
Application:
- Full x-by-wire fuel cell application featuring FlexRay and CAN
Details:
- Several dSPACE MicroAutoBoxes with FlexRay and CAN interfaces used as the vehicle’s distributed control system