Analysis and residual bus simulation of FlexRay networks

This article presents a system for analysis and residual bus simulation of FlexRay networks in conjunction with the fieldbus exchange format FIBEX as standardized in ASAM.

Time triggered systems such as FlexRay require prior planning of all communication parameters, where it is defined which messages are to be sent and received by the nodes at which times. For analysis and test systems, this means that depending on the task in the network of the analysis nodes, different configurations are necessary. Time-controlled communication systems require planning of the communication between the communication nodes. As in time-triggered systems a collision-free bus access must be achieved, it is necessary to assign unique time parameters, "time triggers" to the message, transmission source and message sink. With FlexRay, this time definition of a message is described via the slot ID, the cycle number or cycle repeat and cycle offset. All messages in a FlexRay system can be shown as a matrix via slot ID and cycle number, the so-called FlexRay schedule. **Figure 1**

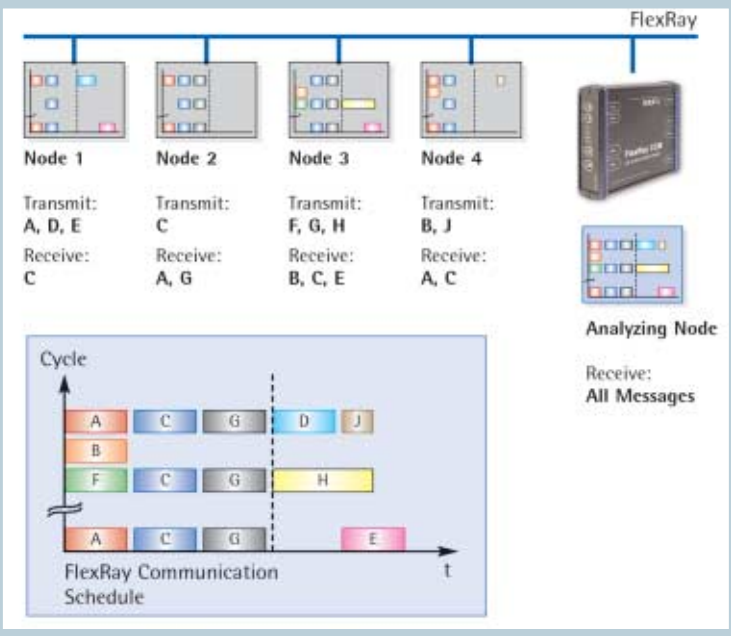

Figure 1: Configuration of an Analyzing node to receive all FlexRay messages.

© automotive

illustrates this with the aid of an example. Each individual node must know a part of the FlexRay schedule, depending on the assignment of the transmit and receive messages. Basic time parameters of the schedule are known to all nodes of the systems.

Configuration of analysis and test node

The FlexRay analysis node shown in *Figure 1* should be able to receive and analyze all messages of the connected bus system. For this it is sufficient to know the basic time parameters of the FlexRay schedule. The receive buffers can be set in an analysis node as a FIFO to receive all messages. The analysis hardware does not have to have the complete information on the times and numbers of messages for configuration. It is sufficient to store the schedule in the PC analysis system. Here the messages can be checked for errors or absents.

If functions are to be triggered in FlexRay networks by analysis or test devices, i.e. test or diagnostics devices transmit FlexRay messages, this must already be included in the original planning of the FlexRay networks. Transmit and receive messages and their parameters are to be reserved for the required functions.

With residual bus simulation, a FlexRay interface should simulate parts of the communication system, so that a functional test of individual or several electronic control units (ECU) is possible. For the FlaxRay analysis node, this means that now parts of the schedule have to be used for configuration, which correspond to the nodes to be replaced.

Whereas the messages are described in the schedule, but with residual bus simulation the application parameters, i.e. the signals are of interest. A comprehensive description of the FlexRay network with communication parameters, signal interpretation and functional description is necessary. In the past this was implemented by proprietary solutions.

Combined tool for analysis and residual bus simulation

Based on the powerful PC Interfaces for FlexRay and CAN, the FlexRay CCM, IXXAT offers a tool for the analysis and residual bus simulation with full FIBEX support. The modular architecture of the IXXAT Multibus-Analyser for FlexRay is designed as a client server model. The server is based on the open programming interface of the hardware platform and provides the PC analysis system with the information and services of the connected bus systems FlexRay and CAN. Analysis functions are provided by modules, which call the server functions e.g. various analysis functions such as recording of the connected bus systems, monitoring and signal analysis. With the aid of the complete network description from the FIBEX file, the clusters with ECUs and corresponding messages and signals can now be shown. This enables fast assignment of signal sequences and messages to devices, both for FlexRay and for CAN. Thus, FlexRay-CAN gateway applications can also be successfully analyzed.

In addition to the analysis function, a series of different simulation modules are also available for FlexRay and CAN. The interface of the server of the Multibus-Analyser is also

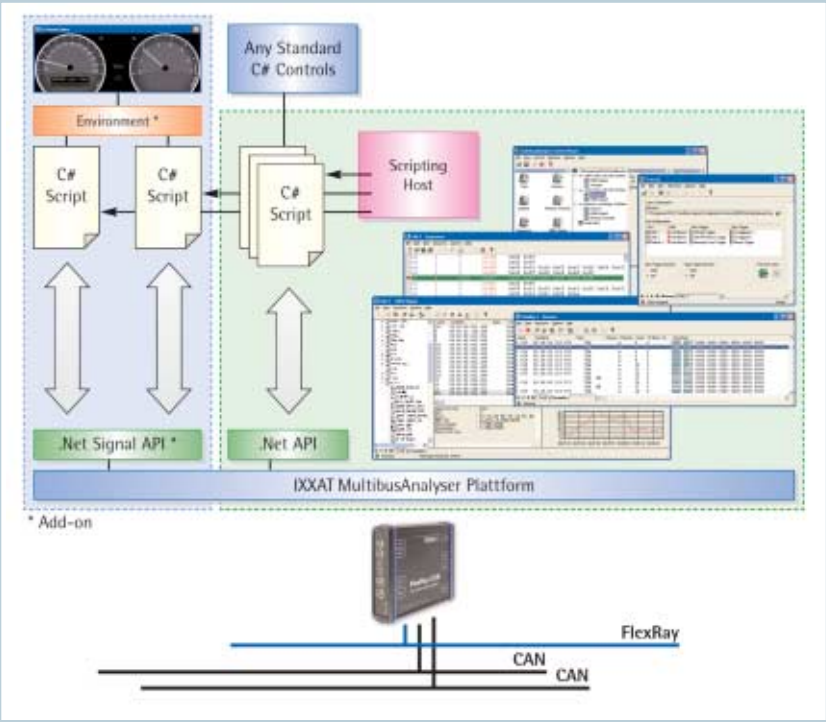

Figure 2: PC based residual bus simulation and analyzing of FlexRay and CAN networks

designed as a .Net interface and open, so that customer-specific modules can be implemented. Together with the scripting host, C# scripts can be started in the Multibus-Analyser context (figure 2). Based on C# scripts, any analysis and simulation functions can therefore be implemented. The particular advantage of the environment implemented in .Net technology is the use of standardized languages and therefore wide-ranging support. In this way, a large number of different modules are available on the Internet to users of this technology [2]. Therefore, even very specific solutions can be implemented very quickly. Anther advantage is the no-cost availability of C# development systems [3].

A residual bus simulation could easily be implemented on the basis of the C# scripting environment, but it is desirable that for the description of the device models it is possible to work directly with signals and not at FlexRay message level.

The information from the FIBEX file is provided to the scripts as global variables via an extension package for the Multibus-Analyser, so that signal interpretation does not have to take place via function calls or via code. An extended environment interface with notification interface is also available, so that all scripts and modules can communicate with each other and event-oriented calling of functions is possible. With the aid of the Multibus-Analyser for FlexRay and a residual bus extension package, PC-supported residual bus simulation can be implemented at low cost for FlexRay and CAN based on standardized technologies.

On-board acceleration of the residual bus simulation

If increased requirements conflict with a pure PC-supported residual bus simulation, for example if it must be possible to reply to FlexRay messages within a cycle, this can only be done with immediate processing on the FlexRay hardware platform. For such requirements, the FlexRay CCM can be extended by on-board functions. An environment adapted for the residual bus simulation is available, which supplies the function modules with the messages. At the same time, a generic PC API for the on-board function interface is implemented, so that data and parameters can be transferred to the hardware and the on-board function. The shortest possible reaction time for devices on FlexRay systems is given by the time of reception of the message to be processed and the processing time until the message to be transmitted is supplied to the FlexRay communication controller. The message to be transmitted must be entered in the transmit buffer before the time of transmission on the bus, so that the new data are transferred in the current cycle. If reactions are to take place within a cycle, the times are to be set sufficiently far apart. For example, a reaction of two messages with consecutive slot IDs within a cycle would be unrealistic. If the times of the receive and reaction message are too close together, the new data are only transmitted in the next cycle. Reactions in the next cycle are normally achievable without restrictions. With cycle times of 5ms, for example, the PC system can already react with a delay of one cycle, but this is not guaranteed due to the Windows technology.

An ideal solution for the residual bus simulation is produced by a combination of on-board processing for the real-time-critical messages and a PC-based processing of all other messages and functions. This fully exploits the flexibility of the PC-base solution and the real-time capabilities of the hardware.

Roman Hofmann works as product manager at IXXAT Automation GmbH. He is responsible for the hardware products and tools.

Literature and further information:

[1] MCD-2 [FBX] Version 1.1 FIBEX – Field Bus Exchange Format www.ASAM.net
[2] www.IT-Visions.de; http://sharptoolbox.com
[3] The Open Source Development Environment for .NET www.icsharpcode.net
[4] Multibus-Analyser for FlexRay www.ixxat.de
[5] FlexRay CCM PC-Interface for FlexRay and CAN www.ixxat.de
[6] www.ETAS.de
[7] www.softing.com
[8] www.FlexRay.com