



Low cost FlexRay cluster design in three steps

The FIBEX-Editor simplifies creation and description of FlexRay networks. Three major steps of the development process are described as well as the validation check of the results.

In the past, the most car manufacturers preserved their own car data base format. To come to a more suitable solution some of the main car manufacturers and suppliers defined a more „exchangeable“ data base standard for the physical description of the bus system, the frames and the signals, the FIBEX (Field Bus Exchange Format) Standard was born.

The FIBEX standard, proposed and defined by the ASAM e.V., is a standard XML-Format, which describes complex message based communication systems, e.g. vehicle communication databases. Based on this XML data format the description of FlexRay, CAN, LIN and MOST networks and the talking nodes as well as the data exchange of the design, configuration, simulation and monitoring tools can be defined in a straight forward manner. A little bit more complicated is the design and the maintenance of such a FIBEX XML database. The following article describes in three steps, how the FIBEX-Editor® can be used to setup a FIBEX XML database e.g. for a simple FlexRay network consisting of

CLUSTER PARAMETERS

Cluster-Parameter	Description	Value/Unit
Makrotick (MT)	gdMacroTtick	1 µs
Cycle Duration	gdCycle	10.000 MT
Static Slot Duration	gdStaticSlot	35 MT
Number of Static Slots	gNumberOfStaticSlots	90
Number of Minislots	gNumberOfMiniSlots	250
Symbol Window	gdSymbolWindow	0 MT
Network Idle Time	gdNIT	400 MT

two Electronic Control Units (ECU 1 and ECU 2) exchanging some frames containing a set of signals.

Step 1: Definition of Cluster Parameters

In the first step a new project has to be opened and the physical cluster parameters of the FlexRay network have to be designed and entered into the FIBEX-Editor. The user input data are validated against the FIBEX schemata and with additional strictly defined rules. Time overlappings, wrong parameters and other inconsistencies are signaled to the user immediately.

Figure 1 shows the data representation inside the FIBEX-Editors cluster dialog.

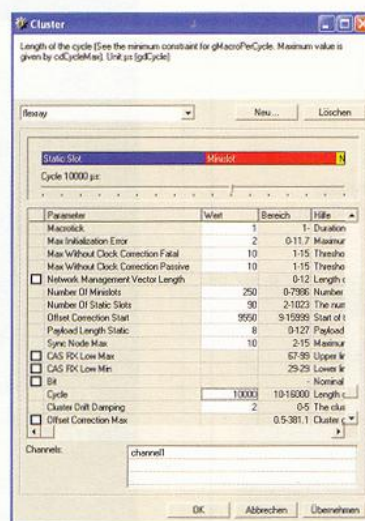


Figure 1: Defining FlexRay cluster parameters with the FIBEX-Editor

Step 2: Definition of Frames and Signals

The second step compounds the design of frames and signals. The different frame parameters, e.g. frame size, frame type as well as the signal and signal coding parameters have to be defined first. For a frame, the frame name, the frame size and the frame type, e.g. "Application", have to be defined. For a signal the signal name, coding name, bit position and byte order (little/big endian) are mandatory. For a signal coding the coding name, the amount of bits, the physical scale and offset as well as the limit and error values must exist. Invalid entered user data, e.g. overlapping signals inside a frame, are signaled immediately in red color inside the logical frame/signal window. The whole frame/signal/coding creation process has to be repeated four times for the four frames (Figure 2).

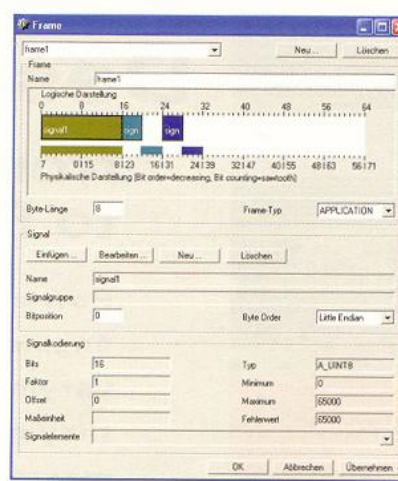


Figure 2: Defining FlexRay Frame, Signal and Signal Coding values with the FIBEX-Editor

© automotive

Step 3: Definition of ECUs and Channels

In the third step the two ECUs (ECU1, ECU2) have to be created and named, thereafter each ECU has to be assigned at least one appropriate channel (Channel 1), the channel itself has to be bound to an ECU connector.

The relation between the frame sending ECUs and a specific Slot-ID used for sending is made by selecting the appropriate ID cell inside the ID grid and inserting a (previously created frame) with the "Insert Frame" button respectively frame dialog (Figure 3). The receiving ECU for the particular frame can be chosen in the frame dialog from a list containing all available ECUs in the network.

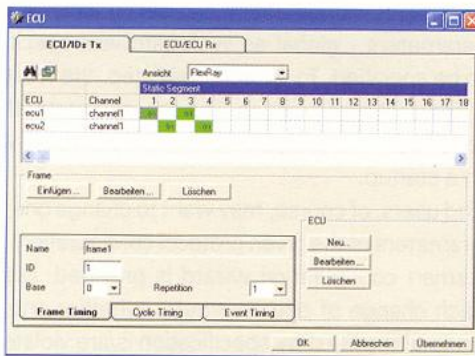


Figure 3: The ECU dialog to link the frames between sending and receiving ECUs

© automotive

Step 4: Validation and plausibility check (optional)

In step four a validation of the entered data can be done, but this is only an option. The validation can be made against the FIBEX schemata and a set of rules, which can be selected from a list of reasonable checks. The check list includes the following rules:

The results of the validation, errors (red) and warnings (yellow), are shown in a clearly arranged error list (Figure 4).

Conclusion

The FIBEX-Editor is a versatile tool, not only to edit, change or view FIBEX files in graphic form, but also to create a completely validated FIBEX XML description for automotive network, containing FlexRay, CAN and LIN ECUs.

A set of user selectable validation rules allows to check the FIBEX design for data integrity.

The FIBEX XML description file created can be used e.g. as data base for a lot of software tools, which need a car data base to map the FlexRay, CAN or LIN raw data stream into easy readable and interpretable physical values.

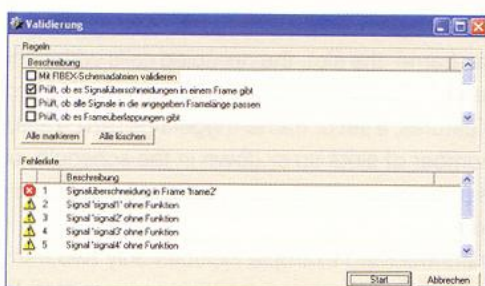


Figure 4: Validation of the FIBEX design

© automotive



Dipl.-Ing. B. Sc. (FH/TU)
Thomas Bachmann is Software Developer and responsible for the product lines FlexRay and FIBEX at CRST, Gauting, Germany.



B. Sc. (TU) Pawel Brzakala is Software Developer and responsible for the product lines FlexRay and FIBEX at CRST, Gauting, Germany.