

Communication under Control

RTI CAN MultiMessage Blockset tests large CAN setups



In our last issue, we looked at multi-tool workflows for the CAN bus in the dSPACE tool chain. This second article describes in detail how the RTI CAN MultiMessage Blockset supports the testing of large-scale CAN setups. With its easy-to-use graphical interface and a wide range of configuration options, the blockset proves its practical value time and time again.

Typical use cases for the RTI CAN MultiMessage Blockset range from setting up simple restbus simulations to defining extensive tests on CAN network communication. Using a single Simulink block in the real-time model, developers can control, configure, and edit large numbers of CAN messages quickly and individually. Once these configurations have been made, they can be reused in downstream process steps, for example with experiment and automation tools.

Starting with a Communication Matrix

A communication matrix (usually a DBC file) is the basis for configuring communication tests with the RTI CAN MultiMessage Blockset. The information in the matrix – cycle times, assignments between messages and ECUs, signal thresholds, defaults, etc. – can be used directly for configuring a restbus simulation. Whenever the DBC file is updated during the course of a project, only a “delta” configuration is necessary. This is especially useful if the databases or integration stages change frequently during vehicle development.

Another useful feature is the ability to use different restbus configurations for one and the same CAN bus. The blockset can be used to assign up to 20 versions of a test configuration to a single CAN controller. The different database or configuration versions can be changed during run time by an assigned model variable or an automatable trace variable.

Simple Restbus Simulations

When the data has been imported, a basic restbus simulation can be created by simply choosing the Rx (receive) and Tx (transmit) messages for specific selected ECUs. This static simulation does not yet

allow exchange data with the plant model, but works with defaults instead. Even so, such simulations, where fixed values are sent, are often sufficient for running tests when ECUs are first put into operation.

If more comprehensive simulations are required, there is a wide range of test functions available. With versatile settings, these support not only the error-free simulation of CAN communication (good tests), but also the insertion of specific errors into the communication process (bad tests). The blockset's functions are subdivided into message-level and signal-level functions for easy use.

Tests at Message Level

The message-level test functions control aspects such as message transmission and the failure of specific messages or even whole ECUs. One example of message transmission manipulation is the function for changing cycle times.

The “Message cycle time defaults” functionality is used to set cycle times and delay times for the messages to be transmitted. Users can either take the information contained in the database or define it themselves in the graphical view. If they choose to define data, they can specify the cycle time for the messages and also a delay time that has to expire before a specific

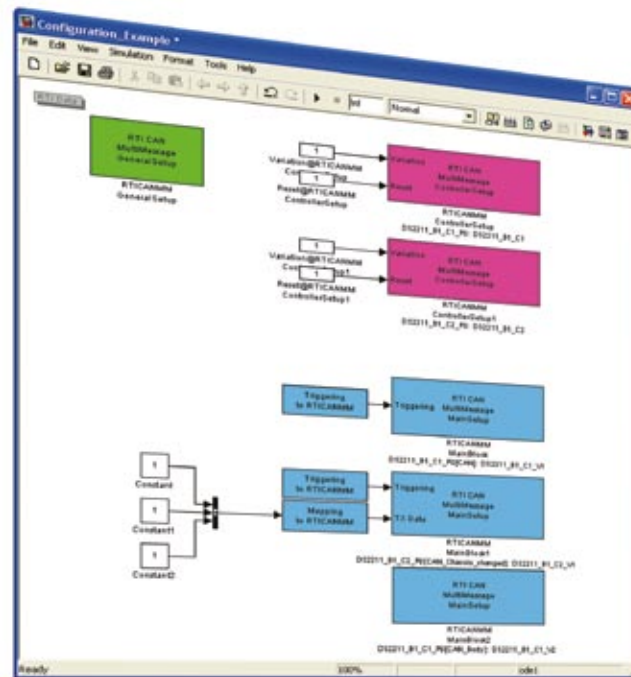


Fig. 1: Bus configurations performed with the RTI CAN MM Blockset.

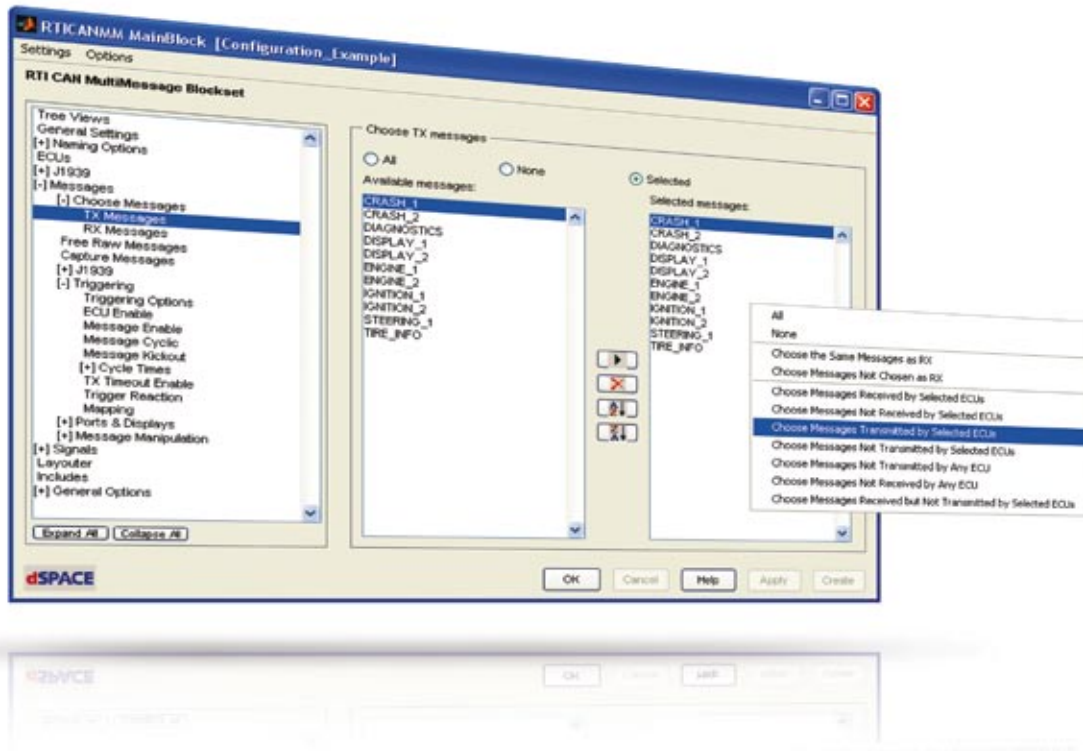


Fig. 2: Selecting Rx (receive) and Tx (transmit) messages for a basic restbus simulation.

message is transmitted during the test. What is particularly useful is that this value can be changed during test execution, either by a model variable or by a trace variable.

Not only the transmission itself is modifiable. Users can also manipulate specific aspects of the messages such as their IDs, lengths, or contents. One frequently used feature is the option to define or modify the checksums for messages and the signals they contain in order to check data integrity. This is done in two steps. Step One: Define the message classes or cases that require different checksum algorithms. For example, one class can be defined to comprise the en-

gine ECU's messages to the transmission ECU, while a second class contains messages to be received by the ESP ECU. The blockset now generates header files for these classes as frameworks for users to integrate their own checksum algorithms – simple look-up-tables – to be used for the tests. Step Two: Assign messages or even individual signals to the defined classes. The easiest way of doing this is to use settable filters, which concludes the configuration of the checksum calculation.

Tests at Signal Level

The blockset and its powerful test function also support testing at signal level. There is the usual op-

tion for transmitting static values, plus an option for transmitting signals from the real-time model. Another very useful feature is the possibility to assign signal names from the database and signal names from the real-time model to each other.

All the typical functions for defining contents and manipulating signals are available: counters, error values, parity signals, etc. One such function is "dynamic signal values", which fixes the value to be transmitted for a predefined number of transmissions. When these transmissions have been completed, the previous signal source or manipulation becomes effective again.

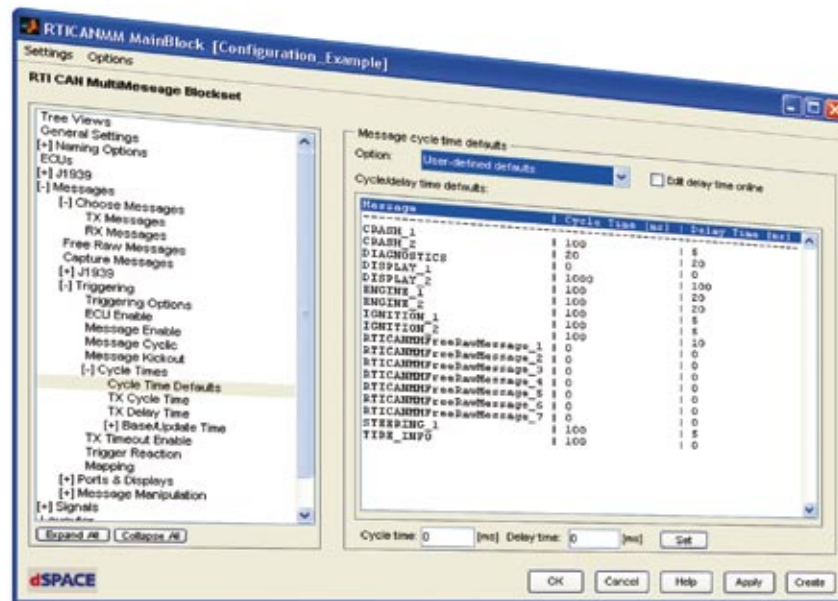


Fig. 3: Defining message cycle times and transmission delays.

Glossary

DBC file – Database container; file format to describe a CAN communication network.

Restbus simulation – Reproduces messages from ECUs that are not present as real devices in the network.

Communication matrix – Describes all the nodes (ECUs) involved in communication and the data exchange between them.

Trace variable – Reference to a model variable that can be used for experimentation and automation.

Extra Options

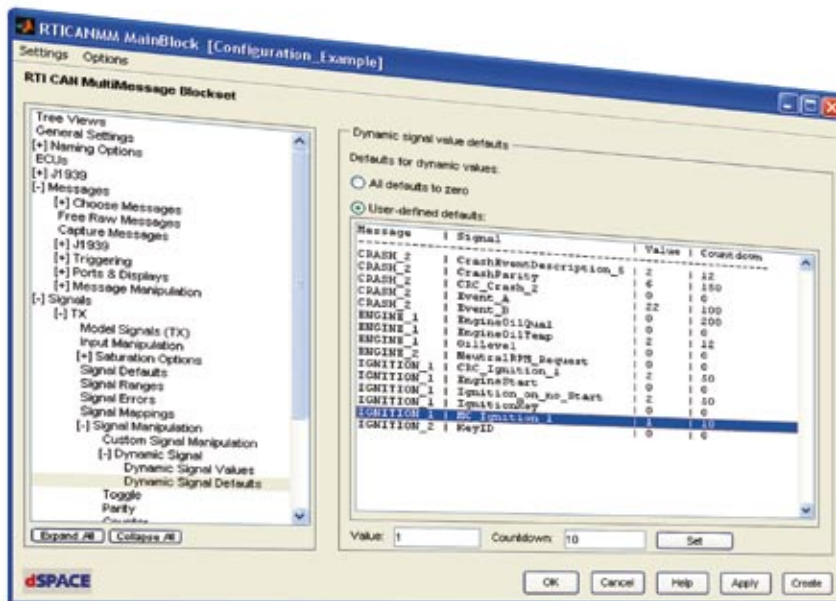
The messages and signals that are configured are not only ones defined in the database. The RTI CAN MultiMessage Blockset also provides an option for creating messages outside the database. These special messages can be changed during run time, so they are useful for sending experimental messages during diagnostic tests, ECU commissioning, etc.

Multiple Bus Support

With its versatile functionality and multi-tool handling (see dSPACE Magazin 1/2008), the RTI CAN MultiMessage Blockset makes it easy to handle and test even complex CAN setups. Once a system has been configured, it is available for comprehensive experiments in

ControlDesk or AutomationDesk and can easily be adapted to new data versions. ■

For details on using the RTI CAN MultiMessage Blockset in a development project, turn to pages 26 – 31.



The transmission controls and signal manipulations can be changed during test execution.



Fig. 4: Graphical page for specifying dynamic transmit signals.

The RTI CAN MultiMessage Blockset:

The RTI CAN MultiMessage Blockset is used for combining dSPACE systems with CAN communication networks and configuring the networks. It is an extension to dSPACE's implementation software Real-Time Interface (RTI). The blockset can be used in rapid control prototyping applications and also in hardware-in-the-loop applications. Users have an intuitive user interface with extensive configuration options for CAN communication. Even complex CAN setups can be handled easily with the RTI CAN MultiMessage Blockset, since a very large number of CAN messages can be controlled via a single

Simulink block. This has a positive effect on the model size and shortens code generation and build

times. The blockset also provides import options for DBC, MAT and FIBEX files.

