Testing of CAN, LIN, and FlexRay ECU characteristics to assure reliability

Manfred Schneider, Jens Muenzberg, - August 22, 2013

Electronic control units (ECUs) in modern vehicles are all linked together via various communication interfaces. The information, which each ECU needs to function correctly, is received from other system components. For example, an instrument cluster can read the current speed or engine RPM from ESP and engine ECUs, respectively. In addition, the ECU puts data about its own status on the network for reception and evaluation by other control units.

The CAN bus is the most common data network in modern vehicles. However, as the trend continuously moves towards an increasing number of ECUs per additional communication networks have found their ways into new vehicle designs. The LIN bus, for example, is a subsystem of the CAN network and is used to relieve the bus load of the supervising CAN network and to reduce overall costs.

On the other hand, FlexRay communication between ECUs has become popular for safety critical applications, such as drive-by-wire. All the different networks within vehicles are connected via gateway ECUs that require special test cases like routing and transmission time.
Modern vehicle ECU’s also support overall vehicle safety features on a system level. It is therefore of paramount importance that the ECU works faultlessly under all operating conditions. It is furthermore required that an ECU will not affect the overall communication in the vehicle environment or goes out of control in the event of a communications breakdown if the unit is connected to the vehicle bus.

For the purposes of test the ECU’s system functions can be divided into physical, i.e. usually electrical, and communication characteristics. The test system measures electrical characteristics without a network connection. In order to measure the communication characteristics it is necessary to simulate the vehicle environment via one or more bus nodes.

Furthermore, failure simulations are yet other test tasks not to be neglected. The scope of failure simulation includes hardware faults as well as failures of the communication protocol written into the software.

**System architecture requirements**
The development of the test system started already 15 years ago, originally developed and implemented around the specifications for high speed and low speed CAN buses, CAN transport protocols and CAN diagnosis (KWP 2000) requirements. Further developments have added the ability to test K-line functions and LIN bus systems. Meanwhile, the test system is already in its fourth generation. It has been continuously enhanced, containing all current requirements in networking vehicle ECUs up to applications, such as network management, subnetwork operations or gateway test.

Basic test needs were drawn from car manufacturers design specifications which define the interface conditions of the communication networks in the vehicle. These specifications determine the purpose and content of individual test steps and define the stimuli and measurements which must be made as well as the precision required. Next, a test requirements matrix was constructed to ensure that the tester configuration encompassed all of these highly complex test needs. Detailed analysis resulted in a number of mandatory core test components and so the base configuration of
the tester for checking network characteristics was derived:

- Oscilloscope and multimeter functions for electrical measurements
- CAN interfaces - the controllers’ onboard software must support measuring functions and also be able to generate fault protocols
- Integration of rest bus simulation for each UUT
- Programmable and expandable unit under test (UUT) power supplies
- The ability to simulate line faults
- Diagnostic interface to access the UUT’s internal fault memory

The LIN bus and FlexRay bus communications have become integral parts in modern vehicles. The expansion to cover LIN and FlexRay testing demands additional intelligent interface instruments, each dedicated to their specific type of bus. The LIN bus controller, for example, has three active LIN interfaces, and is therefore able to simulate all possible network configurations defined by the LIN protocol. The FlexRay controller provides two independent FlexRay interfaces, each of them with A and B channel.

Normally, the majority of an ECU’s functions can be simulated via diagnostic services, which is already a part of the test system’s core configuration. However, for test purposes and also depending on the ECU’s function it may be necessary to stimulate and monitor the ECU’s regular inputs and outputs directly in order to invoke appropriate bus messages from the ECU or check the ECU’s response to bus messages.

**Hardware configuration**

The system architecture (image 2) is centred around a PXI stimulus and measurement module which accommodates all required instruments except the UUT power supply and stress assemblies for fault simulation. The system was based on the PXI bus because of its compact structure and the fact that a wide range of test instruments are already available from GOEPEL electronic’s existing Automotive Test product lines. These include communication interface modules, signal generators and I/O assemblies which are especially designed to simulate typical signals found in ECUs used for chassis, drive and equipment controllers.
An arbitrary signal generator, multimeter and oscilloscope are provided for stimulus and measurement in the basic configuration of the test system. The different networks must be available simultaneously (to test control units with gateway functions), hence it is necessary to switch the measuring instruments onto various test points using relays.

The UUT power supply comprises of two isolated and separately controlled power supplies to achieve a ground offset. The power supplies are controlled by means of Ethernet.

The UUT is connected using a universal connection panel equipped with a plug that mates directly with UUT. There are additional sockets wired in parallel to allow further control units to be connected. Additional monitor sockets for the communication interfaces and test sockets for external measuring devices or load switching are also provided.

**System software**

The overall software architecture is designed to accommodate the user’s desire to execute individual tests in an interactive manner as well as to automatically run complex test program sequences. In order to optimise the efforts in ECU test program generation, a special automation method was developed, which independently generates all necessary test parameters from the data bases of vehicle manufacturers (in the formats dbc, ldf or Fibex).

![System software architecture](image)

The GUI and test run controls are based on test sequencing software which are part of an existing product based on LabVIEW. It has been used successfully for several years in test systems for both R&D and production environments.

Image 4 shows the system software structure. The test sequencing software allows the user to create test sequences by calling up individual test steps (macros) from a macro library. Each macro has a specific pre-defined function with a number of parameters, which are set by the user via an appropriate graphical user interface. The overall test sequence is made up of a combination of macros, i.e. test steps, with user defined parameters. Examples of these test steps are shown in the
following images with the macro interfaces for some of the CAN functions. A significant advantage of this approach is to focus the user on the physical aspects of the test rather than the details of the algorithms behind it.

Image 5: Main menu of the test program editor.

The macro library can modularly be upgraded according to user test tasks. Examples are the availabilities of test step libraries for CAN, LIN and FlexRay diagnosis and network management.
The test sequencer software features a debugger to allow test programs to be run step by step. There is also a run module for running test sequences automatically. Each test step is executed by the run module and returns pass/fail information based on the measurements taken within the macros compared to the user defined parameters. The results are stored in a test protocol file which covers the entire test sequence run on the UUT. The test protocol can be recorded in HTML and XML formats and, for instance, allows links to measurement tables and screenshots of waveforms captured by the oscilloscope. It’s also possible to export measurement values to NI DIADEM for evaluation purposes.

Since the automated test benches’ introduction the users have noted the following improvements regarding network characteristic testing:

- Reducing the test time to less than one third of that required for laboratory testing before.
- The test system enables the user to record and evaluate time-synchronous runs in a timescale of under 1 ms.
- By multiple vehicle bus and diagnostic functions within a single test system special measurements can be carried out which were not possible before.

As requirements change, the modular architecture of the network tester opens up the possibility for system upgrades covering a variety of communication interfaces. In addition to the obvious benefit of cost saving, the main thrust of this approach is in testing gateway functions between different communication networks. The ability of the PXI bus to trigger and synchronize its instruments is invaluable in establishing a common timing basis for these tests.
And last but not least, the most important conclusion drawn from using the network tester should be grounds for some serious thought among the designers and manufacturers of vehicles and control units:

**Nearly 100% of all tested ECUs show defects in the implementation of bus specifications!**

This alarming finding indicates an urgent need to establish the network test system as a key part of the quality management system in addition to its existing presence in R&D departments. Communication faults in the comfort area such as radio or GPS systems may only lead to annoying malfunctions. However, considering more critical applications such as the advent of drive-by-wire systems the introduction of effective test strategies becomes absolutely mandatory.

*This article originally appeared on EE|Times Europe*

**Also see:**
- Handheld device simulates ECU messages
- Understand CAN, LIN, and FlexRay buses for automotive apps
- Debugging automotive serial buses: CAN, LIN and FlexRay exposed
- R&S scope analyzes CAN/LIN at the protocol layer
- dSpace software monitors ECU tests
- PXI functional test system increases ECU throughput