VECTOR >

A Look Behind the Scenes

ECU testing with XCP support

Blackbox tests are typically conducted in the framework of ECU development or in analyzing faulty ECU behavior. This involves connecting an ECU's inputs and outputs to a test system for stimulation and measurement. Although this method lets the test engineer perform extensive analysis, certain tests require looking directly into the ECU. This is the only way to obtain meaningful test results or reduce testing effort.

In most cases, it is actually sufficient to look at the ECU's inputs and outputs to functionally test a component **(Figure 1)**. However, this becomes difficult when state machines are used in the ECU. Their current states can only be derived indirectly by their effects at the ECU's outputs. In the case of sensors whose values are not transmitted over the network system, it is also very difficult for the test engineer to localize errors to the software interface. From outside the ECU, it is not clear exactly where the sensor value was incorrectly processed.

Different methods that offer access to internal ECU data are used, depending on the phase of ECU development. In early phases, for example, internal ECU values are often output in so-called "reserved development messages" **(Figure 1)**. For the functional developer at a supplier, this is an effective and quick method that precisely targets a specific objective. However, these supplemental messages must be removed for later development phases, especially for system integration and series production. They induce additional bus load, and in the worst case they might even collide with messages of other system components.

Another way to access internal values is through diagnostics **(Figure 1)**. Some information is available directly via
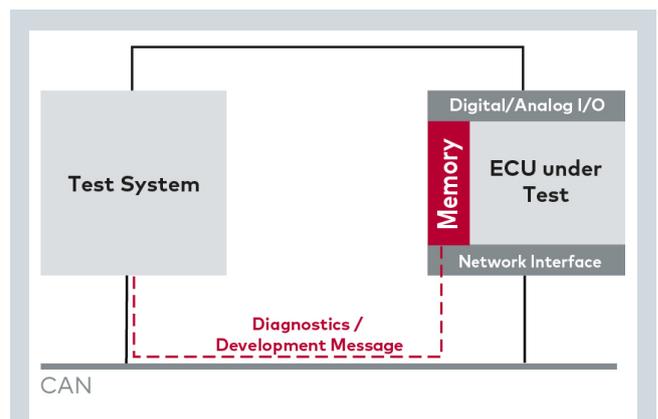
Figure 1: Conventional test system for ECU testing with limited access to an ECU's internal values via diagnostic functionality or special messages created by the developer

diagnostics, e.g. diagnostics offers access to fault memory. Special diagnostic services are also provided to read the required values from memory. The advantage here is that a standardized access method is used. The only precondition is full integration of the diagnostic driver; this is generally provided in today's ECUs. The disadvantage of this method is that a lot of unnecessary diagnostic protocol information

is transmitted along with the actual measured values, and this adds load to the network system interface. A data flow analysis of many values is not possible, especially since the measured values do not contain time stamp information.

XCP for Test Access

If network interface load needs to be kept low, an alternative is to use a calibration protocol. Originally, such protocols were developed for the ECU calibrator. They let calibrators modify parameters or characteristic maps in the ECU to optimize their algorithms. With the XCP protocol standardized by ASAM, the user can read individual values directly from the ECU as needed. The protocol can also periodically supply a defined set of measured values from the ECU via so-called Data Acquisition (DAQ) lists. The XCP protocol was defined for efficient provision of data over the network medium. As an example, after configuration the DAQ lists can be transmitted in response to a single identifier from the test system. In addition, measurement times of the DAQ lists can be synchronized to internal ECU processes. Automated test systems place similar requirements on the system. Use of the XCP protocol makes it possible to integrate internal values in test sequences without excessive loading of the ECU or the network system used.

Another reason that a widely used standard like XCP is ideal is that it is very easy to configure in the tool chain. All necessary information is already in the A2L file such as internal program memory locations with their names and communication parameters. Depending on the development environment, the A2L file is either automatically generated, or it may need to be generated in a separate step from the linker-map information. In the test tool, the user only has to configure this file once for each ECU used in the test. In a second step, the user selects the symbols needed for the test sequences from the A2L file.

CANoe Option .AMD/XCP

Option .AMD/XCP supplements the CANoe test tool from Vector with the convenient option of reading and writing internal ECU values. Besides supporting the XCP standard, it also supports the previous protocol CCP. Once the A2L file has been configured and the necessary values selected, CANoe automatically acquires them and maps them as system variables. The user can then use these variables in any of the testing tasks. Besides offering access to ECU inputs and outputs, they also provide an in-depth look into the ECU's memory **(Figure 2)**.

In simple analysis tasks, users can display the data in the Trace or Graphic Window and use panels to evaluate the results. For more complex test sequences, CANoe's Test Feature Set offers extensive options for creating test cases and automatically evaluating them. For example, this enables checking of the Network Management state machine

Figure 2: CANoe test system with direct access to ECU-internal values via the Option .AMD/XCP

for correct functionality. The necessary stimulation is performed in the CANoe rest-of-bus simulation, and the ECU's reaction is not just measurable on the network; it is directly measurable in the ECU over XCP.

The effort required to execute test cases is also significantly reduced, e.g. for test cases that require sensors. The test system writes the sensor values directly to memory cells in the ECU over XCP. This eliminates the need to connect and control original sensors at the ECU inputs – a demanding task. The ECU is notified that the sensor and associated hardware driver have measured the values correctly. The same approach can be used in the other direction. Here it is assumed that the output stage and actuator have been tested and accepted. In this case, the test system measures the value that the application prescribes to the driver stage over XCP.

Access with Large Quantities of Data

If large quantities of data need to be exchanged between the test system and the ECU in a test case, or if especially quick processes need to be monitored, an XCP connection over a CAN network is no longer effective. In such cases, direct access to the ECU's debug interfaces is recommended. This could be implemented via a NEXUS or JTAG interface, for example. These protocols directly access the ECU memory – partly without load on the microcontroller. Taking this approach, the user can quickly read out very large quantities of data from the system without loading the network and the ECU.

Vector VX hardware, for example, offers direct access to an ECU's NEXUS or JTAG interface **(Figure 2)**. Since this hardware communicates with the test system via XCP-on-Ethernet, integration in CANoe is as easy as integration for XCP access over CAN. Combining VX hardware with the CANoe test system further improves test system performance, without any negative effects on the communication medium.

Oliver Falkner
studied electrical engineering at the University of Stuttgart, Germany. After his studies he joined Vector in Stuttgart in 1999. He is group leader in product management of the Networks and Distributed Systems product line.

Image rights:
Vector Informatik GmbH