

## Complex Test System for the Automated Test of Standard Software Modules

*Peter Opgen-Rhein, Manfred Melzig, Caroline Seubert; Hella KGaA Hueck & Co.*

*Sven Petersen, Gerd Schmitz; S.E.A. Datentechnik GmbH*

### Abstract

This paper introduces the test system FIT (Firmware Integration Testsuite) developed by Hella.

The test system is capable of carrying out manual and automated tests of standard software modules. As these are standardized software elements that are parameterized in specific applications at a later stage, the test environment has to fulfil high demands on flexibility.

The paper describes how the hardware concept of the FIT test system fulfils these high demands. Stimuli and system reactions are controlled by a PXI system. Access to the software under test is implemented with an emulator from iSYSTEM. Using the `isystem.connect` interface, all emulator control and debug functions can be used from within National Instruments LabVIEW™.

The test device is controlled with the TestMaster® software of S.E.A. It integrates the different test bench components and manages the test specifications and test reports. Apart from the presentation of the hardware and software concept, the paper provides an outlook on planned activities concerning extensions of the test case description form currently implemented in TestMaster to a universal XML-based test case description form for Hella-wide use.

### Introduction

Increasing competitive pressure and the growing complexity of automotive products require new methodologies and development processes. Aim of the CASA@Hella (Common Automotive Software Architecture) project is the development and introduction of a standardized basic software system. Designed like a modular construction system, it provides an interface abstracted by the microcontroller on the controller periphery [1].

The software modules developed within the scope of the CASA@Hella project, for example, transmit or read-in PWM signals or convert analog voltage signals to digital ones. Prerequisite for using the basic software in a tangible customer project is the combination and parameterization of the modules for manager, handler and driver level.

Before the standard software modules can be used in the concrete project, the correct specification of the modules needs to be verified. Especially the cross-project use of modules requires extensive software testing prior to delivery.

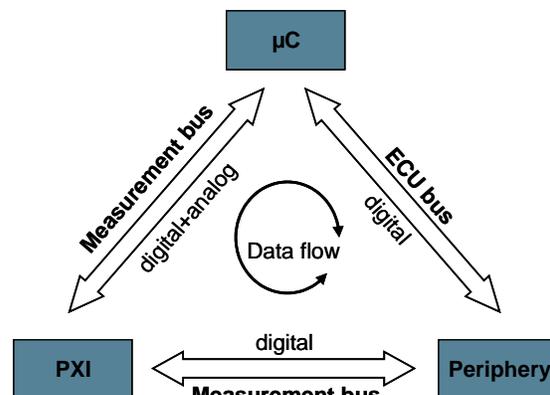


Figure 1: Communication within the test system

As module parameterization takes place at a later stage in the product line, however, the test environment has to fulfill even higher flexibility requirements.

Currently used microcontrollers already have more than 120 freely configurable I/O pins. The peripheral components required for testing, such as shift registers or motor bridges, add to the complexity of the software to be tested. A largely automated test process and a test environment as realistic as possible are additional requirements. Reusability of test cases is a key issue for standard software modules. The diversity of constellations primarily applies to parameterizations, such as channels and controller pins, so that a high degree of test automation is the stated goal.

### Hardware Concept of the FIT Test System

A highly flexible hardware connection is one of the key challenges when implementing a test system. The developed test bench masters complexity with the use of a slide-in system which integrates various circuit boards for the controller, peripheral components and variable loads.

An adapter board connects a National Instruments PXI test system with the slide-in system. Connection of the digital and analog input and output signals between the PXI system and controller as well as the peripheral components is implemented using these boards. Modularity based on diverse circuit boards facilitates the emulation of the various hardware configurations. Using a backplane, circuit boards are interconnected on the rear panel of the slide-in system. Two types of buses are classified for the backplane, the measurement bus and the ECU bus (figure 1). The measurement bus

connects the signal paths of the PXI system with the microcontroller and the periphery. The selected signals are a representative cross-section of the spectrum of possible controller signal types. These include analog signal paths as well as digital inputs and outputs representing special functionalities like SSI, PWM, PWD or time measurement signal paths. The signals between periphery and PXI system are usually digital signals which do not meet high demands to time or clock rate.

From a test environment point of view, the ECU bus is the internal connection between microcontroller and periphery. Voltage supply of the carrier board is effected through the ECU bus; the signals form a digital connection between the microcontroller and the peripheral components. So-called CPLDs manage the routing of the digital signals at the interfaces between the buses and circuit boards. The advantages of these logic devices are defined cycle times and a non-volatile memory due to which the program remains in the CPLD after configuration and does not have to be reloaded after each system start. Each CPLD in the test bench provides bidirectional connections of up to 160 digital signals.

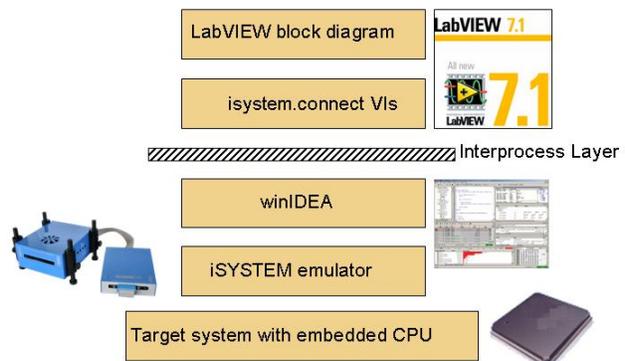


Figure 2: isystem.connect as link between LabVIEW and emulator



The hardware structure of the various circuit boards is mapped as an XML data structure. Within a configuration, this data structure is the basis for the programming of defined signal paths, for instance, between the controller and peripheral components, resulting in a VHDL code for the specific constellation. The CPLDs can be programmed via a JTAG interface using the generated code.

As described before, the standard software modules under test provide a layer between the hardware and the application. Hardware access is effected via the PXI system, the software is accessed through an emulator interface. The software to be tested is executed on an iSYSTEM emulator; in addition, the `isystem.connect` interface facilitates the addressing of all emulator control and debug functions from within National Instruments LabVIEW™ (figure 2). This is achieved with the help of VIs programmed in LabVIEW which address the winIDEA functionalities [2].

### Software Concept of the FIT Test System

The modular hardware requires highly flexible software that can be extended with specific modules at any time and facilitates test generation and parameterization.

The TestMaster® software [3,4] by company S.E.A. Datentechnik GmbH is used to this end. It integrates the various test bench components and manages test specifications and test reports.

The complete PXI system hardware in the test bench is controlled and coordinated by the hardware abstraction layer contained in TestMaster (figure 3). This means that all signals are assigned a symbolic name, thus

providing the user with a transparent mapping of the used hardware components. Consequently, hardware can be exchanged with other hardware or, if required, with a simulation simply by changing the configuration.

With the generated hardware drivers, it is possible to map the signals of the iSYSTEM CPU emulator and the National Instruments FPGA hardware used to simulate specific communication protocols by means of simple signals. On the National Instruments FPGA board, the interface with TestMaster is realized through the host VI of the implemented test driver. This can be easily created and adjusted to current requirements by the user of the test system in the National Instruments FPGA development environment.

When the VI is loaded in TestMaster, the FPGA interfaces are created in TestMaster in the form of logic signals and are available as logic systems without any further configuration.

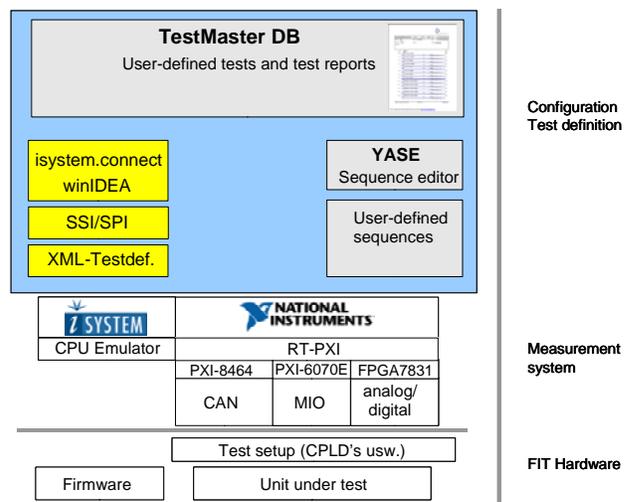


Figure 3: Modular structure of TestMaster in the test bench environment



The implementation of the emulator is based on the software interface `isystem.connect` provided by iSYSTEM.

All I/O signals can be manipulated and requested with the user-defined names in the interactive TestMaster user surface and the automated test sequences. Charts are available for the graphic visualization of signal curves. Combined in logic groups, the signals can be changed and visualized by means of simple entry windows. For special applications or process automation, user-specific windows can be configured in order to manipulate and visualize signals.

Test sequences are generated in the YASE editor in a simple sequence language which is TestMaster-specific and extendable with LabVIEW. A test sequence is generated by sequentially combining individual and preconfigured system-specific commands. Test sequences can be freely nested and called both in parallel and sequentially. A generated test sequence is the basis for an individual test step. The test steps are managed with the TestMaster database module in the FIT test system. Here, the test steps are parameterized and combined to form complete tests. All actions are managed according to version and dependencies.

The functional levels test, execution and interactive test system are clearly defined so that various tasks – test development, test execution and parameterization – can be distributed to persons with different qualifications. Thus, the software developer becomes capable of testing the newly developed modules even without having profound knowledge of the realization of the test processes in the test system.

A complete test concept was realized with TestMaster within only a few weeks time,

based on existing hardware combined with preliminary tests to validate simple functions. The developed system is maintained and upgraded by Hella and used for the final test of standard software modules before a new release is delivered to the development departments.

In the near future, an extension of the test process description form currently realized in TestMaster to a universal XML based test case description form is planned (Hella-wide use). Ideally, a cross-company standard should be employed here.

Interested in even more information about iSYSTEM winIDEA and NI LabVIEW?

The following links provide a direct path to relevant pages on the Internet:

- [winIDEA for LabVIEW – embedded test integration toolkit for use within the graphical programming environment LabVIEW from National Instruments](#)
- [winIDEA with isystem.connect – access winIDEA debug, IDE and profiling features from third party applications](#)
- [In-Circuit and On-Chip Emulators to accelerate NI LabVIEW embedded systems development](#)
- [www.isystem.com](http://www.isystem.com)

## Summary

This paper describes the FIT test system. The test system contains a specially designed hardware. The hardware generates signals that were deterministically generated in the

CPU emulator and, using CPLDs, provides for a scalable routing of these signals to the measurement hardware integrated in the PXI system. Conversely, signals generated by the PXI system are transmitted to the emulator. The emulator registers are read out in order to validate the function. Due to the use of FPGAs for the simulation of analog and digital signals, a fast realization as well as flexible adaptation to changing signal requirements and protocols is guaranteed. The modular setup of the hardware provides for a flexible extension of the test environment.

The test system is controlled with the TestMaster<sup>®</sup> software – a modular software tool that allows easy interactive access to measurement signals by means of a symbolic mapping of logic signals. The integrated sequencer facilitates the automation of processes and generation of logic test steps. These can be combined to form test runs, thus providing for an automated execution of software tests. The test bench meets the high requirements on software testing due to its modular hardware concept and high degree of automation and thus contributes to sustained quality assurance.

## Literature

- [1] Molt, T. Standardisierung und Modularität von Automobilsoftware. ATZ/MTZ System Partners – extra, p. 40-41, March 2003. (*Standardization and modularity of automotive software*)
- [2] Simsek, E. and J. Zäpf.  $\mu$ Controller Emulation meets LabVIEW – Virtuelle Instrumente für Embedded Systems Entwicklung und Test. Virtuelle Instrumente in der Praxis, Begleitband zum Kongress VIP 2005, Hüthig Verlag, Heidelberg, 2005. (*Virtual instruments for embedded system development and test.*)
- [3] Münstermann F. und Schmitz, G., Automatisierter Motorenprüfstand mit LabVIEW Real-Time und Engine Tester, Virtuelle Instrumente in der Praxis, Begleitband zum Kongress VIP 2004, Hüthig Verlag, Heidelberg, 2004. (*Automated engine test bench with LabVIEW real-time and engine tester.*)
- [4] U. Peiffer und G. Schmitz, Automatisierungslösungen mit LabVIEW und TestMaster, Virtuelle Instrumente in der Praxis, Begleitband zum Kongress VIP 2002, Messtechnik und Automatisierung, Hüthig Verlag, Heidelberg, 2002. (*Automation solutions with LabVIEW and TestMaster*)

