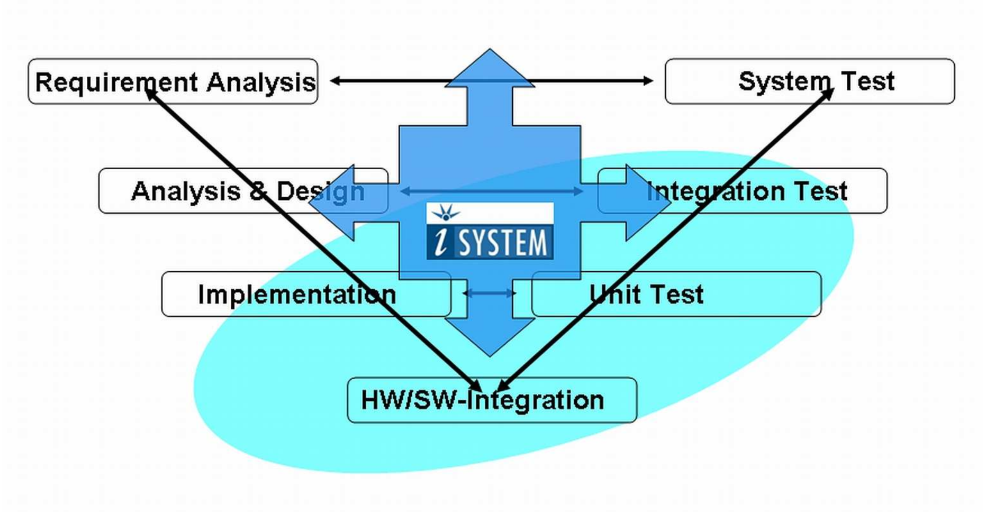
Solutions for Embedded Systems Development

Future trends in medical software development and test

Intense testing for software bugs has taken up more and more space in the overall development process. Different standards for various markets are reissued or adjusted over and over to enhance system and software safety: IEC61508 (functional safety of electrical/electronic/programmable electronic safety-related systems), ISO CD 26262 (functional safety of road vehicles), DO-178B/C (required by FAA and EASA for the certification of software for use in aviation), ISO/TS 16949 (combines existing general quality management system requirements), DIN EN 62304 (software lifecycle processes of medical device software), IEC 60601-1 (3rd edition, safety requirements for medical electrical equipment) etc. These standards and many more, have considerable impact on the actual software development process, particularly the specified test scope (Pic. 1).

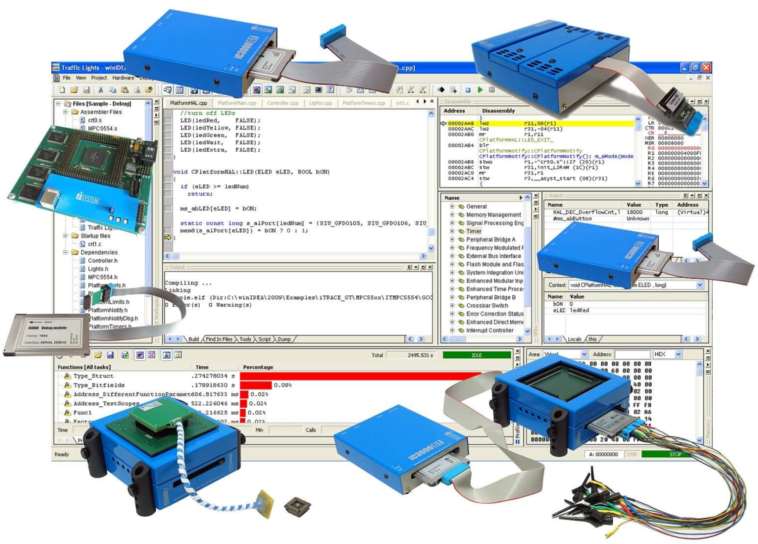
Hence, modern development systems, in the conventional sense, have to be more flexible than ever. In-circuit emulators (ICE) and on-chip debuggers (OCD) as manufactured by iSYSTEM are good examples *(Pic. 2)*. In the past, only microcontroller experts used them as hardware-near development tools, but today they have found their way into many various stages of software development. Here, the ICE and OCD still act as a link to the actual target hardware, via standard debug interfaces, so as to simulate, implement and test embedded software as closely as possible to the actual hardware. Besides basic interface functionality to the target hardware, the ICE and OCD have offered functionality for professional bug finding and thus software testing for a long time, enabling engineers to trace the execution of running software. In addition, the program state can be inspected and program execution can be halted under certain conditions, without or only slightly impacting the software under test. With professional debug solutions as offered by iSYSTEM, customers can moreover record software execution at real-time (tracing), protocol execution times in the clock cycle range and evaluate the processed software sequences relevant to testing (e.g., code coverage).



Pic. 1: Today's challenges are automated development and test processes with smooth tool transitions.

Connectivity (open and public APIs)

Tool vendors like iSYSTEM have to provide generic interfaces to integrate these tools in their customers' development and test processes so that this functionality can be used in full. The interfaces have to support a wide range of tasks (development, test, verification and validation of software and hardware). The trend is towards supporting many different programming languages (C, C++, C#, Java, ...) and script languages (Python, Perl, Tcl, ...) to "remotely control" the development tool from within another application (also customer-specific). In principle, specific process sequences can thus be automated both during development and test.



Pic. 2: Typical tool sets for embedded software developers and testers

Example Connectivity for software test: ZIM project eMOTE funded by the BMWi (Federal Ministry of Economics and Technology)

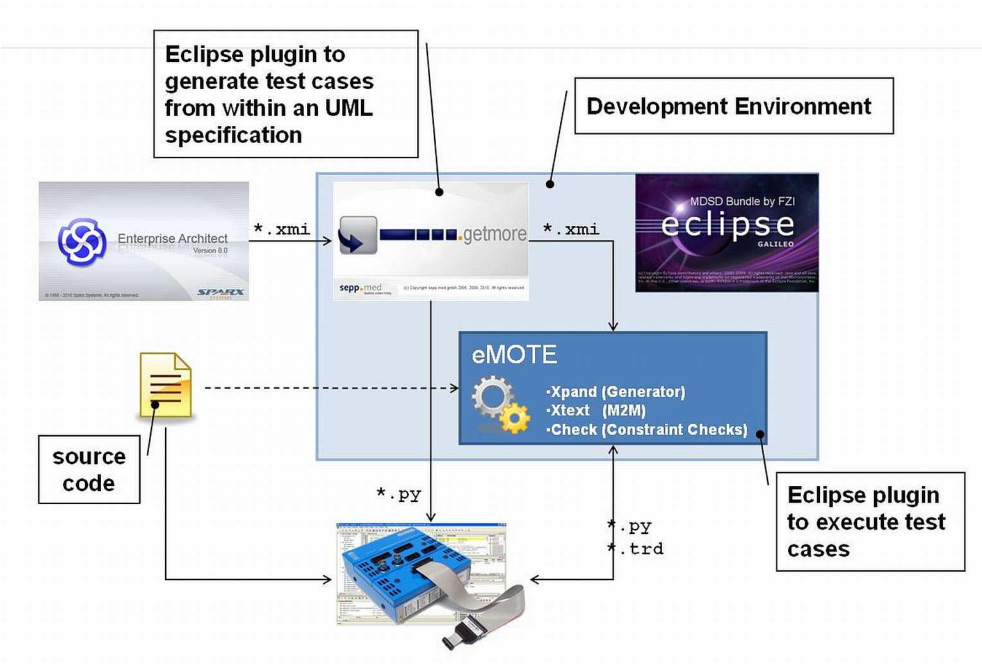
A test process serves to detect anomalies in software under test (SUT) and can be roughly divided into test creation (design) and test execution (deployment). Today, modeling languages like UML are increasingly used for the creation of tests. They facilitate the direct graphical description of test cases or the creation of a test model. Here, the trend is to extend the conventional development system ICE or OCD to a test execution engine that takes over test vectors, partly created automatically from such a test model, executes them and returns the result. Here, the tests are executed without program code instrumentation (as otherwise typical of test tools).

sepp.med GmbH, iSYSTEM AG and the FZI (Research Center for Information Technology in Karlsruhe) cooperate to define and enhance methods and tools for testing embedded software. The project titled eMOTE ("*E*mbedded *Mo*del-based *Te*sting") is funded by the BMWi (Federal Ministry of Economics and Technology) and will last for two years.

Primary objective of the project is to define and enhance test methods for embedded systems software regarding their quality and efficiancy. One important step is to increase the systematic aspect of test design. By the use of test models, and the raise of the automation level through automatic test generation as well as the combination of black box and with box tests with code coverage analysis while running functional tests, an increase in test coverage should be achieved without increasing test efforts. This is realized through the combination of model based approaches with powerful development systems.

Main goal is to adopt sepp.med's model based test procedures and their test generation methods to the specifics and constraints of real time embedded software. iSYSTEM's debug solutions are extended and customized to optimally support test execution in real time. In particular following aspects should be covered and/or solutions developed:

- Coverage of existing and currently evolving standards for embedded software development and test in different industry sectors
- Non-intrusive testing, in particular without code instrumentation
- Test of real time conditions and their description in the model
- Combination of test methods with code coverage analysis
- Inclusion of data sources (for external stimulation of an embedded system) in the test pattern and the test itself
- Integration in a model based development process



Pic. 3 Tool architecture for model based software development and test

Further Trends

Microcontrollers per se feature ever increasing integration and they get faster and more complex. Various types of on-chip debug interfaces (JTAG, BDM, SWD, SDI, OnCE, NEXUS, ETM, ...) as central point of access for development tools are available as standard. However, the quality of information obtained through such an interface may vary enormously – a real challenge for customers and tool vendors. At the early stage of hardware design, engineers already have to consider the positions at which certain debug interfaces shall be conveyed outside. If the demand for specific proof (and tests) to be furnished for product certification is known from the start, the functionality of such a debug interface has to be taken into consideration as well (trace, no trace, etc.). Multicore processors have found their way into a wide range of applications. Here as well, suitable development tools will offer solutions to support the development and test of dedicated multicore applications. Future development tools will also provide so-called "mini HIL" functionality (hardware-in-the-loop, measurement and stimulus modules for testing), to generate and record digital and analog signals and integrate them in testing at an early stage. This makes it possible to develop software very close to reality and to test as early as necessary. Important is that this can be done from wihtin the usual environment, and virtually on-the-fly.

Erol Simsek, iSYSTEM AG
Tel.: +49 (8138) 6971-56
erol.simsek@isystem.com
blog: www.embedded.typepad.com
twitter: www.twitter.com/isystemag