

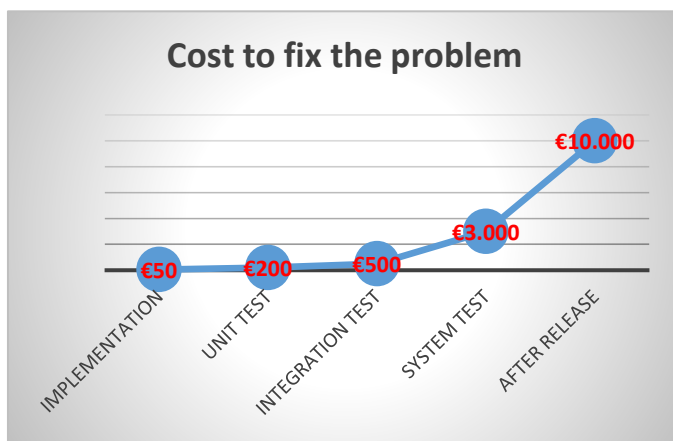
Hardware-in-the-Loop- and Software-in-the-Loop-Tests For Everybody

Test First für Embedded Systeme

Thomas Schütz, PROTOS Software GmbH, Toolchains für Embedded Software

Die meisten Embedded Systeme werden zu spät oder nur unzureichend getestet. Fakt ist: Je später Fehler entdeckt werden, desto teurer ist die Behebung - die Kosten steigen exponentiell mit Fortschreiten des Projekts an. Im Extremfall einer Rückrufaktion für bereits gelieferte Produkte kann das zu Kosten in Millionenhöhe führen.

Es muss also Ziel sein, deutlich früher zu testen – im Idealfall bereits während der Implementierung.



Warum ist es schwierig Embedded Systeme zu testen?

Es gibt vielfältige Gründe warum Embedded Systeme so schwer zu testen sind: Im Gegensatz zu „normaler“ Desktop Software haben Embedded Systeme Schnittstellen zu physikalischen Systemen und nicht nur zu anderen Softwaresystemen. Die Software, welche diese Systeme kontrolliert, ist hochgradig nebenläufig und zustandsbehaftet. Mit den üblichen Testverfahren wie z.B. sequenziellen Unit Tests kann man solche Systeme meist nur schlecht testen.

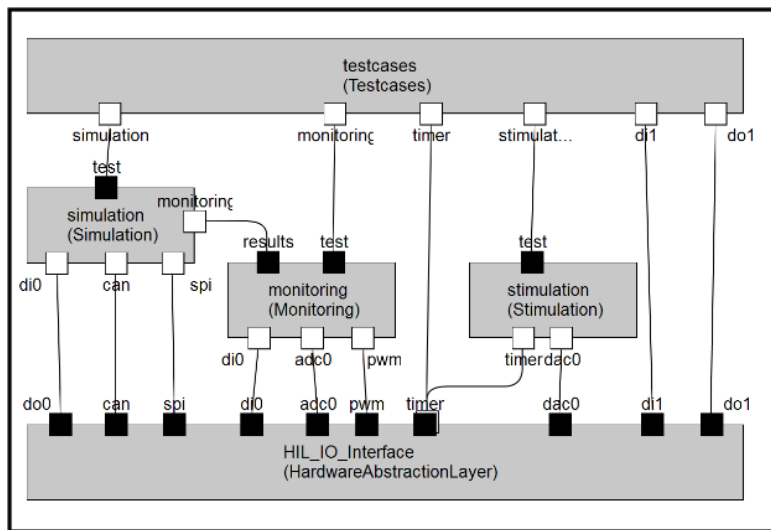
Simulationsbasierte Software in the Loop- (SIL) oder Hardware in the Loop-Tests (HIL) sind grundsätzlich gut geeignet für solche Tests. Die gängigen Hardwareplattformen und Softwaretools sind aber häufig zu teuer, um sie bereits während der Entwicklung an jedem Arbeitsplatz einzusetzen. Außerdem ist für die Entwicklung der Tests anderes Methoden- und Toolwissen nötig als für die Entwicklung der Applikation. In den meisten Fällen werden diese Methoden daher erst sehr viel später in der Testabteilung angewendet.

Der Entwickler kann so kaum strukturiert testen sodass die entstehenden Embedded Systeme während der Entwicklung häufig nicht getestet sondern eher „ausprobiert“ werden.

Wie kann man dennoch entwicklungsbegleitend testen?

Um dennoch entwicklungsbegleitend testen zu können sollten folgende Voraussetzungen erfüllt sein:

- Der Entwickler sollte die Methoden und Tool bereits kennen, oder sich schnell einarbeiten können
- Hardwareplattformen und Softwaretools müssen günstig sein wenn sie an jedem Arbeitsplatz eingesetzt werden sollen
- Methoden und Tools sollten die Entwicklung von Tests für Embedded Systeme auf unterschiedlichen Ebenen unterstützen (zumindest Komponenten- und Integrationstests)



Das Open Source-Modellierungswerkzeug *Eclipse eTrice* ermöglicht den Aufbau einer kostengünstigen und dennoch mächtigen Testplattform: Im Modell können Komponenten für Stimulation, Monitoring und Simulation entwickelt und generiert werden. Mit dem *eTrice* Add-on *CaGe* können Komponenten mit Testcases für das System beschrieben und generiert werden. Die kombinatorische Testcasegenerierung ermöglicht eine sehr schnelle Entwicklung von Testcases mit hoher Abdeckung.

Alle Komponenten zusammen bilden einen portablen, echtzeitfähigen „Test Harness“ für die Applikation. Diesen kann man als SIL-Test (z.B. auf dem Entwicklungsrechner) oder als HIL-Test auf kostengünstiger Hardware (z.B. Standard Evaluation Boards von Infineon oder ST-Microelectronics) ausführen.

Die vollautomatische Durchführung der Tests wird durch einen Jenkins Continuous Integration Server (ebenfalls Open Source) übernommen.

Test First für Embedded Systeme!

Durch die gewählte Kombination aus Standard-Hardware und größtenteils frei verfügbaren Open Source Tools kann man eine modellgetriebene SIL- oder HIL-Testlösung aufbauen. Diese ermöglicht es bereits während der Applikationsentwicklung Embedded Systeme mit hoher Abdeckung automatisiert zu testen.

<http://www.eclipse.org/etrice/> - <http://www.protos.de>