iSYSTEM

BLUEPRINT

Enabling Safer Embedded Systems

Issue #04, September 2016

Upcoming Events:

Seminar: Development ends only when the testing is complete

6th/7th October 2016
Gothenburg, Sweden

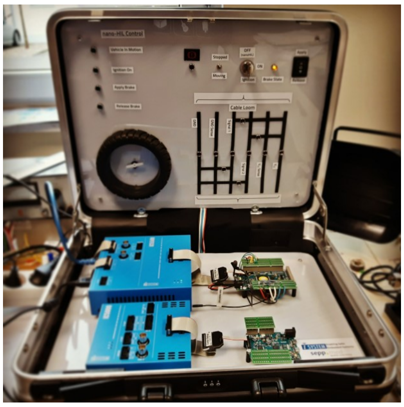
12th October 2016
Eindhoven, Netherlands

15th/17th November 2016
Paris/Toulouse, France

For more info, please call us on:

+49 (8138) 6971-0

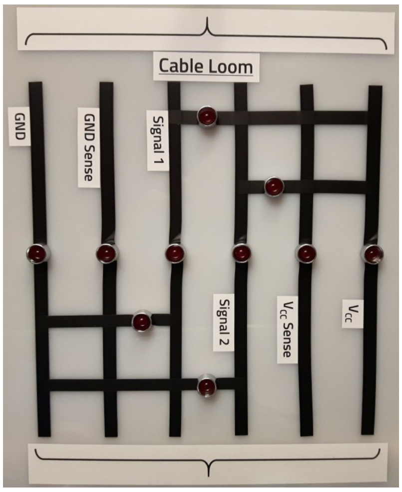
Over the last few months we were fortunate enough to have Markus Huber, a student from the Munich University of Applied Sciences, undertaking his Diploma Thesis with us. After the success of our "iCandGreet" visitor greeter, which we had enhanced to include a Hardware-in-the-Loop (HIL) function, we were keen to look at developing something closer to a realistic application that can be encountered daily.



Many of our customers are currently looking for ways to improve their code quality because a) they now need to fulfil safety standards that did not apply to them previously, and b) they want to achieve a higher level of reliability in the embedded systems they develop. Topics that often arise include the need to automate testing, generate code coverage for certification or documentation purposes and even integrating their development process into a continuous integration (CI) system such as Jenkins.

To this end we gave Markus the challenge of implementing an Electric Parking Brake (EPB) that fulfils the standards defined in ISO 26262. In fact, due to the complexity of such an application, we simply focused on the input rocker-switch used to apply and release the brake. An additional goal was to make use of industry standard tools for Original Binary Code (OBC) unit testing. The design also had to include a method whereby a simple HIL system could be developed to enable the application of a Model-Based Test approach for the HIL.

The result is to be seen here (left), a compact case containing the complete embedded solution along with debugging tools, HIL implementation and a simple drum brake implementation to add to the demonstrator. Having come to the conclusion that the application would be classified as an ASIL D solution, he set about implementing a dual-MCU solution based upon Arduino DUE and M0-Pro development boards. The DUE serves as the master of the system, evaluating the state of the Apply/Release rocker switch through a circuit consisting of resistor ladders. The M0-Pro serves as a slave to the DUE, delivering its opinion as to the state of the switch. If the two MCUs fail to agree on the switch state, the system enters a safe mode.



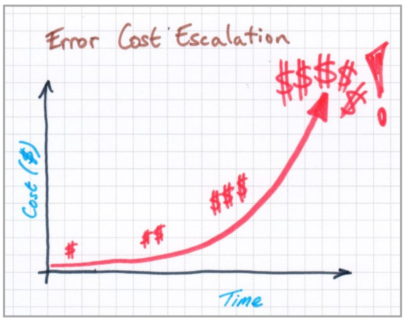
This compact "nano-HIL" uses relays to inject open– and short-circuit failures into the cable loom between the ECU and the rocker switch.

Design, documentation and code for the project can be found in our repository hosted at:

http://bit.ly/epb-demonstrator

How much do software errors in a project really cost?

If you've been in the business a while, you have no doubt seen a graph similar to the one below during a supplier presentation. There is no arguing the truth in



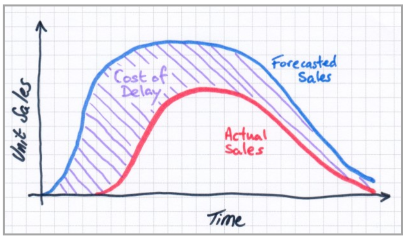
this graph. It seems rather too obvious to state that fixing a problem later in a project rather than earlier is likely to cost more the later it is found. Additionally this reality is backed up by the work of Frederick Brooks (The Mythical Man Month) as well as Barry Boehm (Software Engineering Economics). Even NASA has undertaken research in the form of an

> *"…fixing an issue found during final test costs 13 - 75 more to fix than finding it during design…"*

autopsy of several development projects. The analysis of real projects and models, drawing on real project data, showed that fixing a problem found during final testing cost between 13 and 75 times more than if it had been discovered during the design phase.

Another interesting approach to considering the cost impact of errors delaying the launch of a product comes from Eric Graves. He highlights that, in a commercial environment, a delayed product release reduces

sales revenue over the lifetime of the product. One first cause of the sales loss is due to customers deciding for alternative solutions for long-term projects, thus shutting the supplier out for an entire project cycle. This reduces the peak sales forecast for the product during the product definition phase. The second cause relates to the lifetime of the market need. With all likelihood, the market window for the window will still close in the timeframe forecast.



Increasingly development teams are turning to continuous integration tools, such as Jenkins, to help check code functionality on a daily basis during design to raise quality and deliver on time, submitting unit tests along with source code into the project's repository. For more follow our link: bit.ly/bh-cost-of-errors

Getting students up and running for short-term projects

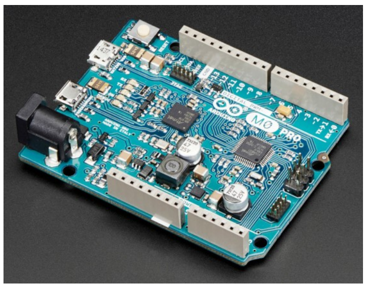
Students are a great addition to any team, bringing new ideas and curiosity which challenge the status-quo. However, with their sometimes limited practical experience, trying to give them an MCU, compiler, APIs, debug tool and IDE all in one hit can be quite overwhelming.

Tenured engineers often underestimate how much tacit knowledge they have acquired that helps them to quickly understand datasheet peripheral descriptions and the expected dependencies of software and hardware upon one another.

Internally we have been impressed with how the 32-bit based Arduino M0-Pro helps to ease the path to success in such situations. Firstly, the Arduino C++ programming environment helps to ensure that ideas can be quickly implemented without first tackling oscillator configurations, flash memory access or GPIO multiplexers.

Additionally, with its integrated CMSIS-DAP debugger, students can use industry standard development tools, like winIDEA Open, to debug their sketches and even develop unit tests. Practical examples are available via our BugHunter blog too!

Links: bit.ly/winideaopen and bit.ly/bughunterblog



iSYSTEM AG für Informatiksysteme
Carl-Zeiss-Str. 1
85247 Schwabhausen
Germany

Phone: +49(8138)6971-0
Fax: +49(8138)6971-46

E-mail: info@isystem.com
Web: www.isystem.com

Commercial register of the local courts (Amtsgericht) München HRB 148751
VAT identification number: DE128231221

Board of Management: Erol Simsek, Werner Fischer, Martin Gröstenberger

Responsible for the content § 10 Absatz 3 MDStV: Erol Simsek