Release Presentation

ASAM AE HIL API 1.0.2 (Maintenance)

Stuttgart, 2012-02-29 (ASAM TSC Meeting)





Agenda

- Introduction: Hardware-in-the-Loop Simulation (HIL)
- Architecture of a HIL Test Bench
- Motivation / Current Situation
- Project's Major Goals
- Standardisation Context and Technological Approach
- Deliverables of HIL API 1.0.2
- What's new in HIL API 1.0.2
- Compatibility



What is Hardware-in-the-Loop Simulation ?



SAM 3

Architecture of a HIL Test Bench





Motivation / Current Situation



- Test cases directly depend on the used Test Hardware
- Test Automation Software and Test Hardware are rigidly coupled
- Not always the ,best' test software can be combined with the ,best' testing hardware
- Know-how cannot be transferred from one test bench to the other (additional training costs for employees)
- Switch to the newest testing technology is difficult because of tool specific formats and test hardware compatibility
- Test cases cannot be ported from one test system to another (e.g. test case exchange between OEM and supplier)



- Reuse of test cases within the same test automation software on different test hardwares.
- Reduction of effort for test hardware integration into test automation software
- Long-term protection of software investments and test case development efforts





Standardisation Context







Advertising and Marketing

- Today, test cases often directly depend on the used test hardware
- ASAM HIL API 1.0.2 Decouples test cases from test systems using ports
- Easy test case exchange between different HIL test benches
- Better know-how transfer from one test bench to the other
- Reduced training costs for employees
- Crosstests among different vendors are planned by June of 2012



Deliverables of HIL API 1.0.2

The deliverable of ASAM AE HIL Version 1.0.2 includes:

Directory ProgrammersGuide

- ASAM_AE_HIL_BS-1-3_API-for-ECU-Testing-via-HIL-Simulation_V1-0-2.pdf

Directory Generic_UML_Model

- ASAM_AE_HIL_BS_V1-0-2.EAP

and ASAM_AE_HIL_BS_V1-0-2_HTMLExport.zip

- ASAM_AE_HIL_BS_ErrorOverview_V1-0-2.xls

Directory Templates

different subdirectories with XML schema files for EES Configuration and Stimulus Signal Description (*.xsd, *.xml, *.sti)

Directory Technology_Reference_Interfaces

Sub Directory Python contains

- Mapping_Rules (ASAM_AE_HIL_BS-3-3_Python-API-Technology-Reference-V1-0-2.pdf)

and specific for each Port and Common

- Interfaces (py files)
- Sample Code
- Dummy Implementation

Sub Directory C# contains

- Mapping_Rules (ASAM_AE_HIL_BS-2-3_Python-API-Technology-Reference-V1-0-2.pdf) and specific for each Port and Common

- Interfaces (cs files)

- Sample Code

- Dummy Implementation

- setup binaries (in subfolder Installation)

Release Presentation



Introduction to HIL API 1.0.2

For the ASAM AE HIL API Version 1.0.1 a corresponding maintenance project was started.

The projected lasted from February 2011 until February 2012.

A lot of feedback coming from first implementations of the different HIL API ports (MAPort, EESPort) was evaluated regarding errors, unclear documentation or missing parameterisations.

Maintenance was done based on current ASAM TSC Guidelines.

All received change requests were collected, developed, assessed and after decision integrated. All items were managed in a corresponding Bugzilla configuration. All proposals are stored in the project archive.



What's new in HIL API 1.0.2

Removed inconsistencies:

- •Differences between C# Interfaces und UML Model
- •State machine of EESPort
- •XML Scheme EESConfiguration
- •Literal Errors in Documentation removed
- •Precise definition of measurement units
- (e.g. duration of Loose Contact Errors)

Distribution of binaries:

•ASAM Interface- and Implementation-Assemblies provided with Deliverable Motivation: these vendor independent assemblies are supposed to exist only once on a target pc. Windows provides the Global Assembly Cache (GAC) to ensure this.

•A setup installs the ASAM-Binaries (ASAM.HILAPI.Implementation.dll, ASAM.HILAPI.Interfaces.dll) into the GAC and is able to do de-installation.

•This setup (used by all vendors) reduces the risk of installation/de-installation errors, if setups of different vendors manage those ASAM binaries on a pc.



Compatibility

Following the TSC guidelines a maintenance release have to be backward compatible to its base release, here to Version 1.0.0 of ASAM HIL API. Backward compatible in this sense means, that a script, written for a V1.0.0 or V1.0.1 implementation have to run also with a V1.0.2 implementation.

Java language is not supported.

Supported languages are C# and Python.

