



AUTOSAR

Classic Platform Demonstrator

...how to bring AUTOSAR technology into University Classes,
creating a community, supported by a working demonstrator

Presenter:
Dr. Urbina Fuentes

JULY 9-13, 2023

**MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA**



HUAWEI | PISA RESEARCH
CENTER



Motivation

The AUTOSAR standard is nowadays the predominant architecture for the development of automotive industrial software. In particular, AUTOSAR Classic is widely used among most of the world major car-makers.

Nevertheless, the side of software development workload for automotive projects has been growing exponentially in the last years so OEMs and suppliers are significantly struggling to find the proper developers for specific AUTOSAR-related implementations (e.g., application, BSW, integration, etc). On another hand, the study of AUTOSAR as part of the academy studying plan became extremely difficult due to the highly expensive dependencies to commercial AUTOSAR tools and lack of open source-based working environments for allowing university students to familiarize with the AUTOSAR world.

In this work we introduce the concept of the AUTOSAR classic platform demonstrator as part of the AUTOSAR University Package. This classic platform demonstrator describes an AUTOSAR classic development environment that will be developed using open-source tools and AUTOSAR modules and may be easily adopted by the universities with minimum financial investments. The Classic Platform Demonstrator (CPD) addresses the following problems:

- Dependency on embedded hardware boards
- Dependency on proprietary AUTOSAR software stacks
- Complexity of the creation of a complete example
- Size of the standard (>20000 pages!), and not availability of a “stable” training material

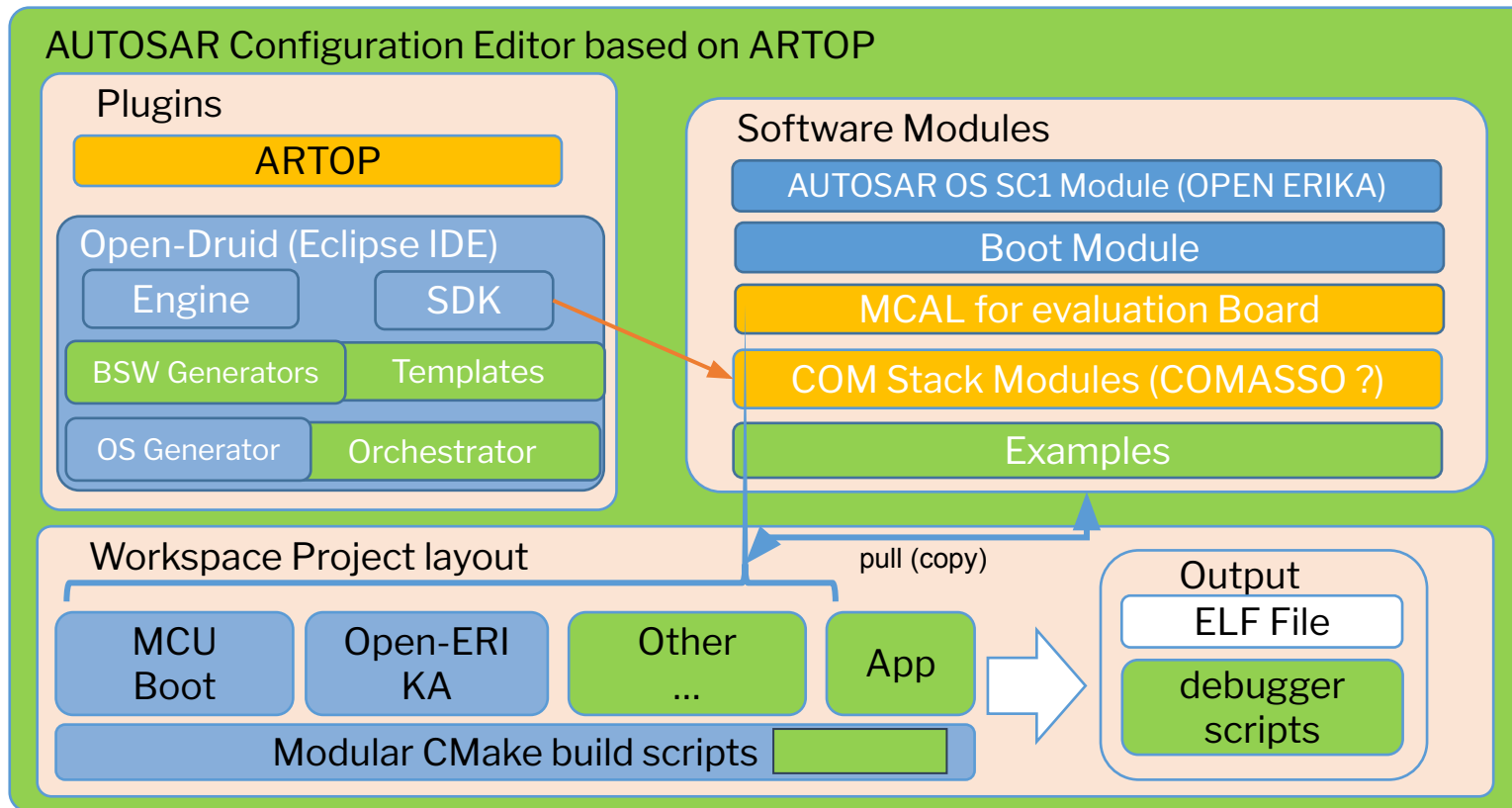
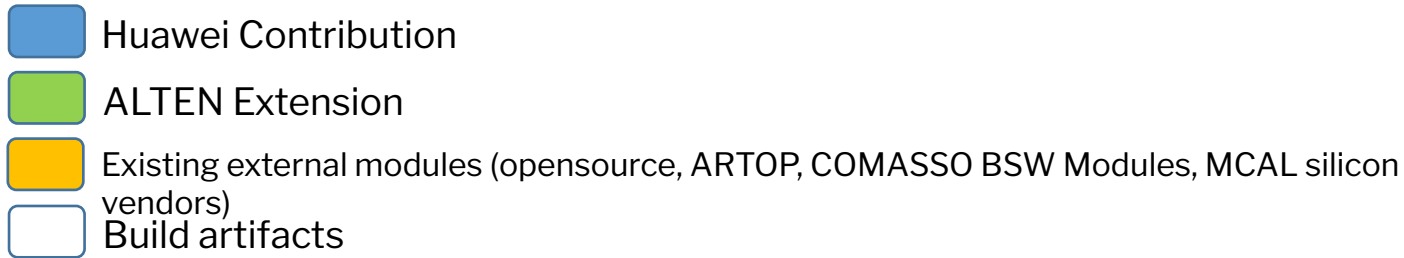


Overview of provided features:

- AUTOSAR Editor for AUTOSAR BSW configuration of AUTOSAR OS and part of the BSW (COM Stack Modules and MCAL for a specific Evaluation board). Generation of ARXML files based on Editor configurations.
- Generation of Code from ARXML configuration files integrated with third party Basic Software Modules (such as Silicon Vendor MCAL Libraries, COMASSO, and others).
- IDE for handling project C and header files. Smooth creation, editing and compilation of a project by employment of a Template/Example generator (to make simple the instantiation of events), a Build Orchestrator (able to trigger the build of various subsystems handled by third party configurators), and possibly a graphical configurator.
- Integrate with third party Tier2 tools (in order to add ARTI tracing support, debugging, static code checks, and others features).
- Default USE Cases based on OS and COM Stack implementations (CAN, Ethernet)



Target Classic Platform Demonstrator Architecture



Starting point: Open-ERIKA

The starting point of the AUTOSAR Classic Demonstrator is the result of the ERIKA3 Project:



<http://www.erika-enterprise.com>

The original software package of the ERIKA3 Project is extended to provide an initial platform concept consisting of:

- Already existing open-source packages: Eclipse, EMF, XText, CDT, Jenkins, Vagrant, Ubuntu.
- AUTOSAR Tool Platform (ARTOP), which is an EMF-based implementation of common base functionality for AUTOSAR development tools.
- Extended Open-ERIKA codebase: Full support for AUTOSAR OS Scalability Class 4 (including Memory Protection, Stack Monitoring, Timing Protection, ARTI); Full support for GCC/HighTec compilers; Build system based on CMAKE; Support for various architectures and microcontrollers (see Table I for the complete list); Support for minimal boot and MCU support for the selected boards; Support for a number of built-in examples useful to explain the main features of the kernel; The code of Open-ERIKA will be packaged as an Eclipse Plugin to ease its distribution inside a standard Eclipse environment.
- Extended Open-Druid code generator: Code generator based on AUTOSAR ARTOP EMF Models; Code generation based on XText/XTend; Integration in Eclipse through a dedicated plugin; Support for the most common Automotive and educational boards (Cortex-M, AURIX, RISC-V, etc), focused initially on CortexM boards and on Tricore AURIX Family).



Developed Demo Sessions

- The pre-liminary CPD provides a set of additional content, including:
- AUTOSAR Training material about real-time operating systems and AUTOSAR OS, with the following structure:
 - Details on the design of small operating systems for microcontrollers
 - Details on fixed priority real-time scheduling, and stack sharing techniques
 - The AUTOSAR OS Standard, including: basic tasks, preemption and non preemption, resources, application modes, events, alarms, and hooks;
- In addition to slides, the system comes with examples that can be used to understand realtime AUTOSAR OS features such as preemptions and periodic activations, as well as AUTOSAR specific features such as stack sharing, multiple activations, events, and ORTI/ARTI debugging. The examples are made available on a set of embedded boards (typically cheap maker embedded boards) which are typically used during University training courses.



Used Toolchains

- MCAL files from the silicon vendors
 - MCAL files can be taken from the silicon vendors such as Infineon, NXP, etc.
- MCAL configuration by using EB Tresos
 - EB Tresos is one the most common MCAL configuration tool.
 - Most of the case silicon vendors are giving Tresos tool with temporary license to use.
- BSW configuration and generation in BSWDT by using COMASSO stack
 - COMASSO Package contains mainly Communication Stack, Diagnosis stack, System Services stack, Memory stack, Service Library stack and ARCEL stack (AUTOSAR Central Elements) that contain the modules in the table.
 - Each module contains all the relevant source file, templates, scripts and Module documents.

Communication Stack	Diagnosis Stack	SysServices Stack	Memory Stack	Service Library Stack	ARCEL Stack
• CanIf	• CanTp	• DEM	• NvM	• Bfx	• CompilerAbstraction
• CanSM	• FrTp	• FIM	• MemIf		• ComStackTypes
• CanNm	• J1939Tp	• DET	• Fee		• PlatformTypes
• Com	• DCM	• DLT	• Ea		• StandardTypes
• IpduM		• ComM			
• PduR		• EcuM			
• Nm		• StbM			
• FrNm					
• FrSM					
• FrIf					
• LinIf					
• LinSM					



Development process of the example project

- MCAL configuration,
 - After receiving the MCAL files from the silicon vendors, MCAL configuration can be done by using EB Tresos
- ARXML and code generation by using EB Tresos
 - Code generation with respect to our configuration
 - ARXML file is needed for BSW configuration
- BSWDT
 - Importing the ARXML file to BSWDT, so we can refer MCAL configurations that we will need for BSW configuration.
 - Importing meta-model files from MCAL
 - Basic Software modules configuration in BSWDT
 - Code generation
- ERIKAOS
 - OS configuration and code generation
- RTDruid
 - Code integration and build

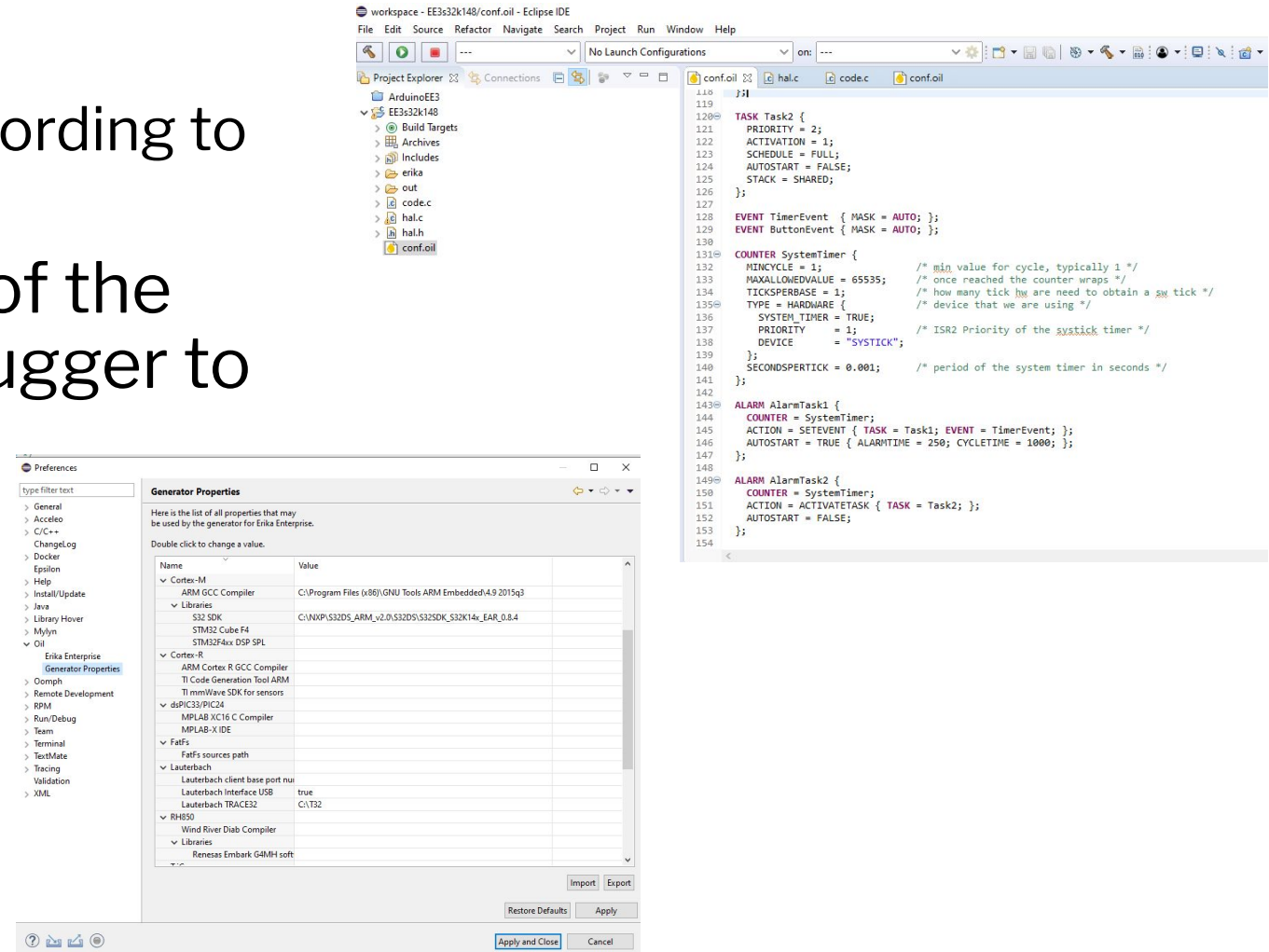


- According to our use case, we used Can, Dio, Gpt, Icu, Port, etc. modules to configure in Tresos.
- Basic software modules configuration in BSWDT by importing Arxml file and meta-model files from MCAL.
- While the code generation in BSWDT we have faced several issues regarding to scripts to validate the configuration and to generate the code. Some of the configuration controls written for specific microcontroller family and this was causing issues for our microcontroller. To solve these issues, we had to adjust script files.



Configuration and Generation of Erika OS by Using Druid

- OIL file configuration
 - OIL files can be updated according to use case
- We can import directories of the libraries, compiler and debugger to use built-in make files.



Summary and Future Steps

The AUTOSAR University Package for classic platform is an ambitious project that spans from training material for Universities and Research Centers, to a working AUTOSAR classic demonstrator based on the new version Open-ERIKA of the pre-existing project ERIKA Enterprise. In this work we presented the first phase of an AUTOSAR Classic Platform Demonstrator that provides:

- A modular AUTOSAR classic platform based on production code derived from the ERIKA3 project
- Full source code of an AUTOSAR OS together with specific demo sessions

Future Steps:

- Proper licensing scheme (currently under discussion in AUTOSAR)
- A complete integration and build of the existing pieces
- Integration of COMASSO
- ARXML Editor
- RTE Generator
- ... but it's worth it. Would you like to help and join us?

