MCAL CODE GENERATION, DEVELOPMENT AND TESTING
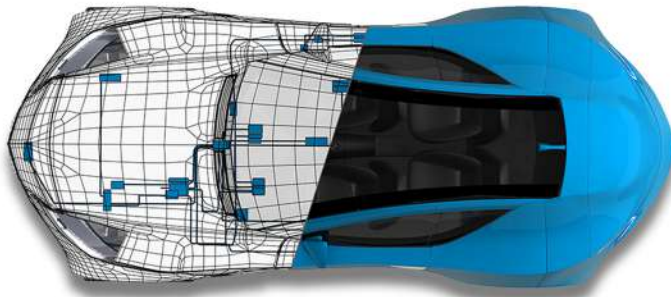
Comasso | DaVinci Configuration & Generator Tool | EB Tresos Studio

A Handbook on AUTOSAR

An Introduction: What is MCAL in AUTOSAR?

MCAL Stands for Microcontroller Abstraction Layer. MCAL is one of the core components of AUTOSAR compliant software architecture.

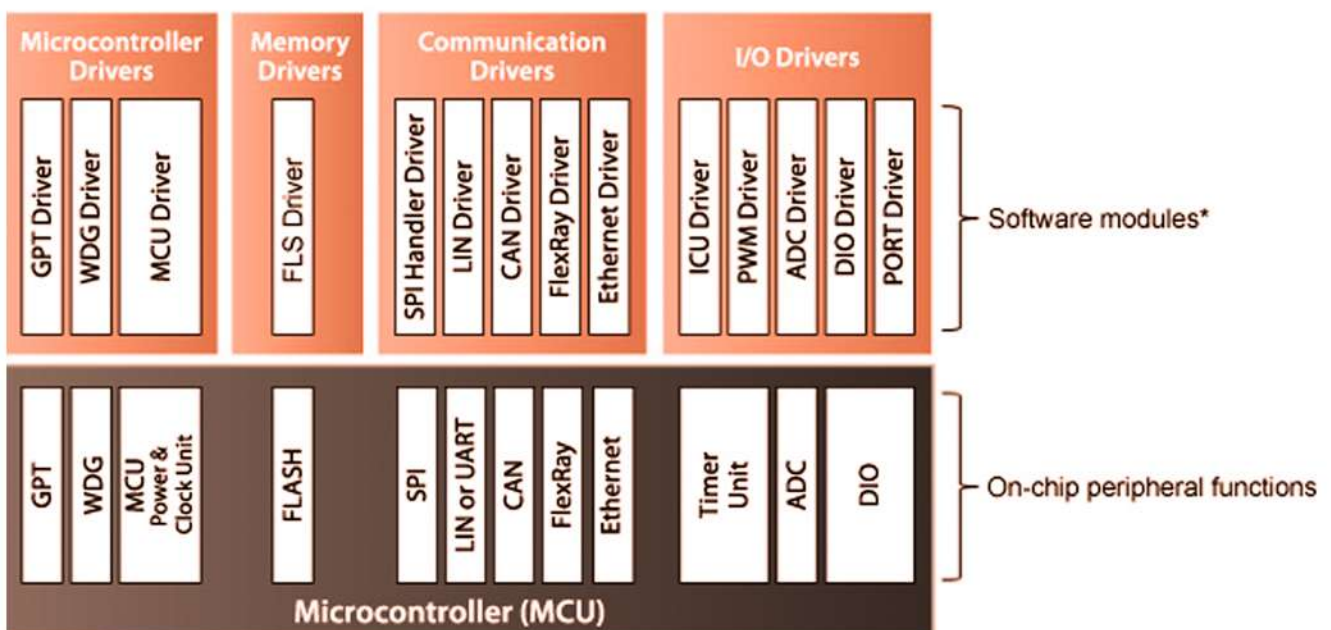MCAL, as a software component, plays a very pivotal role in implementing the objective of the AUTOSAR Consortium.

This objective is achieving the standardization in the design of the embedded software, for automotive products developed by global automotive OEMs' and Suppliers.

MCAL module is the baseline for migration of a software to AUTOSAR Architecture. This enables a software from being ECU specific to being more application oriented.
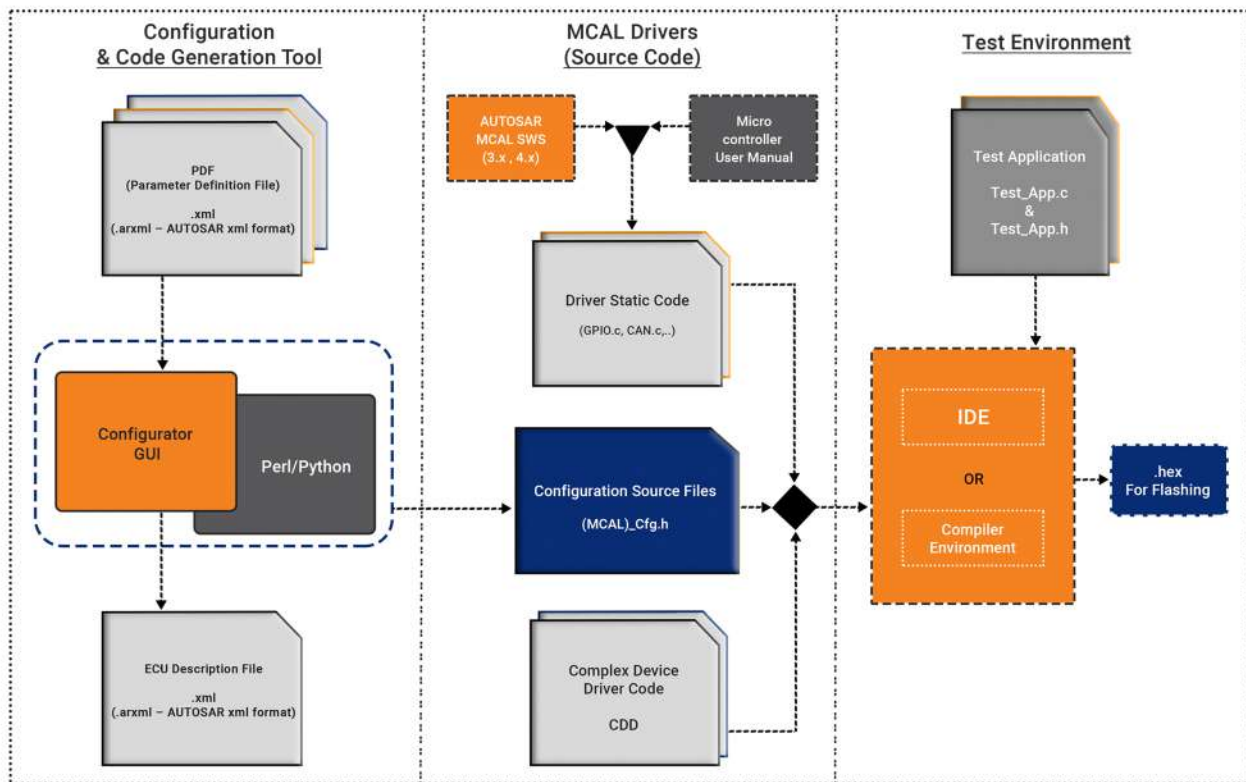
Hence MCAL essentially makes the upper software layer completely independent of the hardware platform (MCU).

A microcontroller vendor provides MCAL software along hardware platform. However, the MCAL drivers need to be configured based on the project's requirements. In order to carry out this configuration, MCAL configuration and code generation tools are deployed.

A Snapshot of the AUTOSAR Architecture of the MCAL Layer

Understanding the MCAL Driver Development Workflow

Configuration & Code Generation Tool

PDF (Parameter Definition File)
.xml (.arxml – AUTOSAR xml format)

Configurator GUI

Perl/Python

ECU Description File
.xml (.arxml – AUTOSAR xml format)

MCAL Drivers (Source Code)

AUTOSAR MCAL SWS (3.x , 4.x)

Micro controller User Manual

Driver Static Code (GPIO.c, CAN.c,..)

Configuration Source Files (MCAL)_Cfg.h

Complex Device Driver Code CDD

Test Environment

Test Application Test_App.c & Test_App.h

IDE

OR

Compiler Environment

.hex For Flashing

The MCAL Workflow (Configuration-Development-Testing)

As shown in the diagram, the following are the broad-level steps involved in design and development of MCAL :
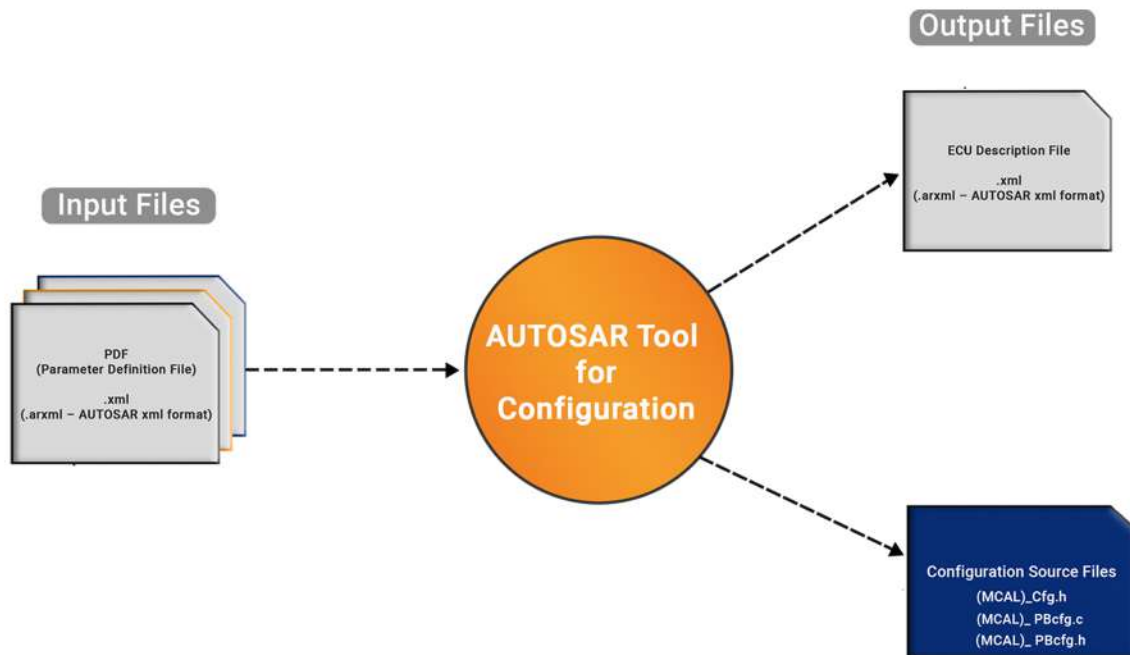
- Configuration and Code Generation
- MCAL Driver Development
- Testing of MCAL Drivers

1. Configuration and Code Generation (understanding the input and output files) :

The Configuration and Code Generation for MCAL drivers, requires an in-depth expertise in AUTOSAR Tools.

An AUTOSAR tool for Configuration is a GUI that facilitates the configuration of MCAL Drivers. In the background of this tool, runs a script that enables the Code Generation.

The Parameter Definition File is the input for the configuration tool GUI. Based on these parameters, Code Generator Tool (usually a pearl or a python script) generates configuration source file (MCAL)_Cfg.h, (MCAL)PBcfg.c/.h)

Input Files

PDF
(Parameter Definition File)
.xml
(.arxml – AUTOSAR xml format)

AUTOSAR Tool for Configuration

Output Files

ECU Description File
.xml
(.arxml – AUTOSAR xml format)

Configuration Source Files
(MCAL)_Cfg.h
(MCAL)_ PBcfg.c
(MCAL)_ PBcfg.h

Parameter definition file is in .xml / .arxml (arxml is an AUTOSAR xml file format with some schema for representing Parameters). In simple terms it consits of parameters that can be configured for a particular MCAL driver

---

Interestingly, the Code Generation activity (Perl/Python Script) is not dependent on the availability of the GUI of any Configurator Tool.

A Perl/Python script can be designed and run to generate a standalone executable package (.exe file).

Such executable files can be easily integrated, in the build process to generate a configuration source files from the ECU Description File.

What is our expertise in Configuration and Code Generation for MCAL?

- Development of Eclipse based Configuration tool and Perl/Python based Code Generation Tool

- In-depth expertise in customization of the configuration tool, as per your requirements

- Our team can also deliver a standalone executable package (code generation tool).

- This .exe file can be easily integrated and configured with any Configuration Tool, as per the project requirement.

2. ISO 26262 Complaint MCAL Drivers Development:

The MCAL Layer comprises of several device drivers, that act as an interface between the hardware peripherals and the application layer. These drivers are designed specifically for the underlying microcontroller platform (the source code of the drivers need to be changed if one migrates to a different hardware platform).

AUTOSAR MCAL drivers are based on requirements specified in AUTOSAR MCAL Driver Software Specification document.

These specifications are defined for each MCAL module like communication drivers, I/O drivers, memory drivers, microcontroller drivers and others.

Software Specification Document of the AUTOSAR MCAL Drivers, consists of the following:

- Software requirements that need to be fulfilled by the MCAL Drivers.

- API details and parameters that should be passed to the API, as well as their return value

- Data types to be used and configurable parmeters to be provided

- API prototypes, their functionality and implementation specifications

What is our expertise in development of ISO 26262 complaint MCAL Drivers?

- Support for identifying the ASIL Value (ASIL B/ASIL C/ASIL D) and full-stack development of the MCAL Drivers as per the ISO 26262 Standard

- Development of Complex Device Drivers (CDD access MCAL drivers/APIs)

- Integration of CDD with existing configuration

3. Testing of MCAL Drivers:

The testing or validation of the MCAL drivers needs to be executed in a specifically created testing environment.

To create this testing environment, an IDE or Compiler and the source code of the MCAL drivers are required.

Details of the components required for the Validation of MCAL drivers:

- Test Application files written specifically for every MCAL component viz. Communication, Memory, vehicle communication protocols (CAN, FlexRay, Ethernet, SPI, UART)

- Static code of the MCAL Drivers (GPIO.c, CAN.c, etc)

- Configuration Source Code of the MCAL Drivers ((MCAL)_PBcfg.c/.h & (MCAL)_Cfg.h)

- Stub files to replicate BSW when BSW* (Base Software Module) is not available

  > * BSW is required by MCAL driver to make API calls to the upper layer of AUTOSAR.

- A scheduler, IDE and/or compiler

What is our expertise in MCAL Validation and Test Environment Creation?

- Creation of test plan for each MCAL Driver component

- Creation of Test Applications for each driver component

- Creation of test environment with simple scheduler and stub files.

- Creation of Make and Batch files for Test environment Build

- Integration of Code Generation Executable/tool along with Test Environment build process.

Partner with our Functional Safety and AUTOSAR Consultants for migration to AUTOSAR 4.3:

Our team of Functional Safety (FuSa) consultants work in collaboration with our AUTOSAR domain experts to deliver solutions that are compliant with ISO 26262 standard and AUTOSAR Architecture.

Our cross-functional teams of Functional Safety and AUTOSAR, have partnered with OEMs' and Suppliers from US, Germany, Europe, India, China, Taiwan and South Korea.

The following is the snapshot of our expertise in delivering ISO 26262 Compliant MCAL Drivers:

- Support for HARA and FMEA in order to derive Safety goals, Safety Mechanisms and ASIL Value

- HLD, LLD, Software Development and Testing services, as per the assigned ASIL Value

- Unit Testing, QAC and Polyspace Analysis of the developed MCAL Driver

Hope you enjoyed reading this handbook.
For more queries and/or demos, please contact us at
sales@embitel.com