AUTOSAR Training

AUTOSAR is open system architecture to support automotive OEMs and Tier-1 suppliers to improve ECU software quality, reduce development costs, and avoid re-development of similar ECU software components repeatedly for the same vehicular applications. It is a growing and evolving standard defining a layered architecture for software.

Automotive trends like autonomous driving, v2x connectivity, OTA updates, predictive maintenance, and many other innovative features are based on in-vehicle software functions. For all these functions to work seamlessly and to cater to real-time in-vehicle functionalities, each ECU must work efficiently. Modern-day high-end vehicles have up to 100 ECUs, which communicate with each other via CAN bus, CAN FD, or Ethernet network to support complex vehicular functions.

Earlier, ECU software used by OEMs were on different platforms. There was no standard software architecture that was being used by tier 1 suppliers and their vendors to design the ECU software for OEMs. So, whenever any OEM wishes to switch to a new tier 1 supplier or vice-versa, the transition was very difficult. The new supplier used to face enormous challenges in understanding the existing software architecture, hardware platforms, and standards used in ECU software development. Thus, it was nearly impossible for a new supplier to drive an on-going project from the midst of its production life cycle.

To streamline the coordination between OEMs and tier 1 suppliers, to improve ECU software quality and reduce development time and costs, tier 1 automotive suppliers, semiconductor manufacturers, software suppliers, tool suppliers, and others came forward in 2003 and created a consortium called **AUT**omotive **O**pen **S**ystem **AR**chitecture (**AUTOSAR**).

What is AUTOSAR?

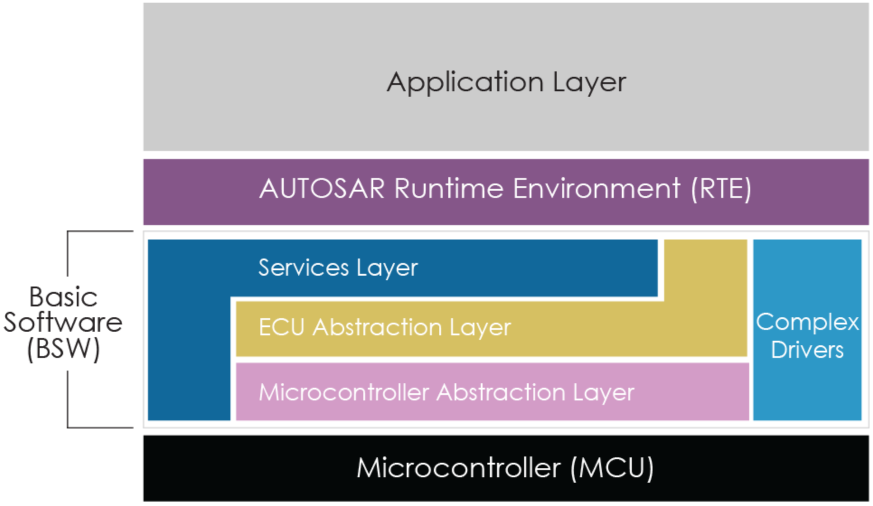
Automotive Open System Architecture (AUTOSAR) is an open and standardized automotive software architecture, which supports standardization in interfaces between application software and basic vehicular functions and it helps in establishing common ECU software architecture for all the AUTOSAR members.

AUTOSAR is intended to provide inherent benefits to the members to manage increasingly complex E/E in-vehicle environments like easy integration and exchange of functions in complex ECU network and control over the entire product lifecycle.

AUTOSAR Architecture

AUTOSAR is an open system architecture for automotive software development and provides standards for developing common automotive software applications. It is a growing and evolving standard that defines a layered architecture for the software. The classic AUTOSAR platform runs on a microcontroller and is divided into 3 main layers; let us discuss them in detail:

- Basic Software Architecture- It is common to any AUTOSAR ECU.
- AUTOSAR Runtime Environment
- Application Layer



AUTOSAR Architecture

Basic Software Architecture (BSW)

AUTOSAR Basic Software Architecture consists of hundreds of software modules structured in different layers and is common to any AUTOSAR ECU. This means the supplier who has designed BSW can share it with other suppliers that are working on ECUs of engine, gearbox, etc.

Basic software architecture in AUTOSAR consists of three layers:

- **Microcontroller Abstraction Layer (MCAL):** MCAL is also known as a hardware abstraction layer and implements interface for the specific microcontroller. MCAL has layers of software, which are integrated with the microcontroller through registers, and offers drivers like system drivers, diagnostics drivers, memory drivers, communication drivers (CAN, LIN, Ethernet, etc.), I/O drivers and more.
- **ECU Abstraction Layer:** The prime task of the ECU abstraction layer is to deliver higher software layers ECU specific services. This

layer and its drivers are independent of the microcontroller and dependent on the ECU hardware and provide access to all the peripherals and devices of ECU, which supports functionalities like communication, memory, I/O, etc.

- **Service Layer:** The service layer is the topmost layer of AUTOSAR Basic Software Architecture. The service layer constitutes an operating system, which runs from the application layer to the microcontroller at the bottom. The OS has an interface between the microcontroller and the application layer and can schedule application tasks. The service layer in BSW is responsible for services like network services, memory services, diagnostics service, communication service, ECU state management, and more.

AUTOSAR Runtime Environment (RTE Layer)

AUTOSAR Run-time Environment is a middleware layer of the AUTOSAR software architecture between the BSW and the application layer and provides communication services for the application software.

Application Layer

The application layer is the first layer of the AUTOSAR software architecture and supports custom functionalities implementation. This layer consists of the specific software components and many applications which perform specific tasks as per instructions.

The AUTOSAR application layer consists of three components which are: *application software components, ports of software components, and port interfaces.*

AUTOSAR ensures standardized interfaces for software components in the application layer and application software components help in generating simple applications to support the vehicle functions.

The communication between software components is enabled via specific ports using a virtual Function Bus. These ports also facilitate communication between software components and AUTOSAR Basic Software (BSW).

The above-explained architecture of AUTOSAR is its classic platform, which supports real-time requirements and safety constraints. Based on the microcontroller, the classic platform is capable of supporting applications in the field of networking and security by allowing ECUs to access vehicle sensors and actuators.

Below are the modules on which professional training is available.

1. Overview of Automotive SW Process, SWDL, Tools.
2. Introduction and deep into AUTOSAR architecture.
3. Application Layer of AUTOSAR architecture.
4. VFB design and RTE configuration.
5. AUTOSAR Communication Stack for CAN protocol.
6. AUTOSAR Diagnostic Stack.

(Note: Detail syllabus given on next page.)

Who can learn?

Anyone who wants to enhance their skills in field of automotive software development

Prerequisite:

Embedded system developers who has hands on experience on embedded C, communication protocols and embedded software architecture.

AUTOSAR Syllabus: Duration of Course: 76 hrs.

Overview of Automotive SW Process, SWDL, Tools: Duration: 4hrs

1. Overview of product development life cycle used in Automotive
    A. V model
    B. Agile Methodology
2. Overview of ASPICE process.
3. Functional safety standards: ISO26262
4. Overview of Static Analysis tools
    A. QAC
    B. Polyspace
    C. LDRA

Introduction and deep into AUTOSAR architecture: Duration: 6 hrs

1. AUTOSAR Layered Architecture and Methodology and Tools
2. Building AUTOSAR Application Software Component development (M2 level)
3. Building RTE and Configuring with OS and integrating with ASWC
4. Understanding the modules of the BSW stack
5. COM Stack
6. Mode Management
7. Memory Stack
8. IO Stack
9. Diagnostic Stack
10. CDD
11. Crypto
12. Other Modules in AUTOSAR-WDG, Microcontroller services

Application Layer of AUTOSAR architecture:
Duration: 6 hrs.

1. What is Software component and COE design?
2. Types of Software components.
3. Runnable design.
4. Communication Ports.
5. Types of Ports and Its utilization in SWC.
6. OS Task creation in AUTOSAR configuration tool.
7. Runnable mapping to OS tasks.
8. Components validation and code generation procedure.

AUTOSAR Communication Stack for CAN protocol:
Duration: 30 hrs.

1. Introduction to AUTOSAR communication stack.
2. Communication stack service layer modules:
    A. COM
    B. PDUR
    C. CANTP
3. Communication hardware abstraction layer modules:
    A. CANIF
    B. CAN TrCv
4. Communication Driver (MCAL).
5. Introduction to CAN DBC.
6. CAN n/w Designing using DBC editor.
7. CAN DBC configuration in AUTOSAR configurator tool.
8. AUTOAR communications stack configuration steps for configurator tool.
9. Communication Modules validation and code generation demo.

AUTOSAR Diagnostic Stack:
Duration: 30 hrs.

1. What is Vehicle Diagnostic?
2. Diagnostic N/W view
3. Detail concepts of diagnostic service.
   A. Diagnostic Services
   B. Diagnostic Service: Positive response
   C. Diagnostic Service: Negative response
   D. NRCs of Diagnostic services.
   E. Different response timeouts: P2 , P2*, S3 timer.
4. UDS Protocol
5. UDS services
6. Introduction to AUTOSAR diagnostic stack.
7. Diagnostic Communication manager (DCM).
8. Introduction to Cdd file used in Diag. Stack.
9. Cdd file creation.
10. Cdd file configuration in diagnostic stack using configurator tool.
11. Diag. stack validation and Code generation demo.