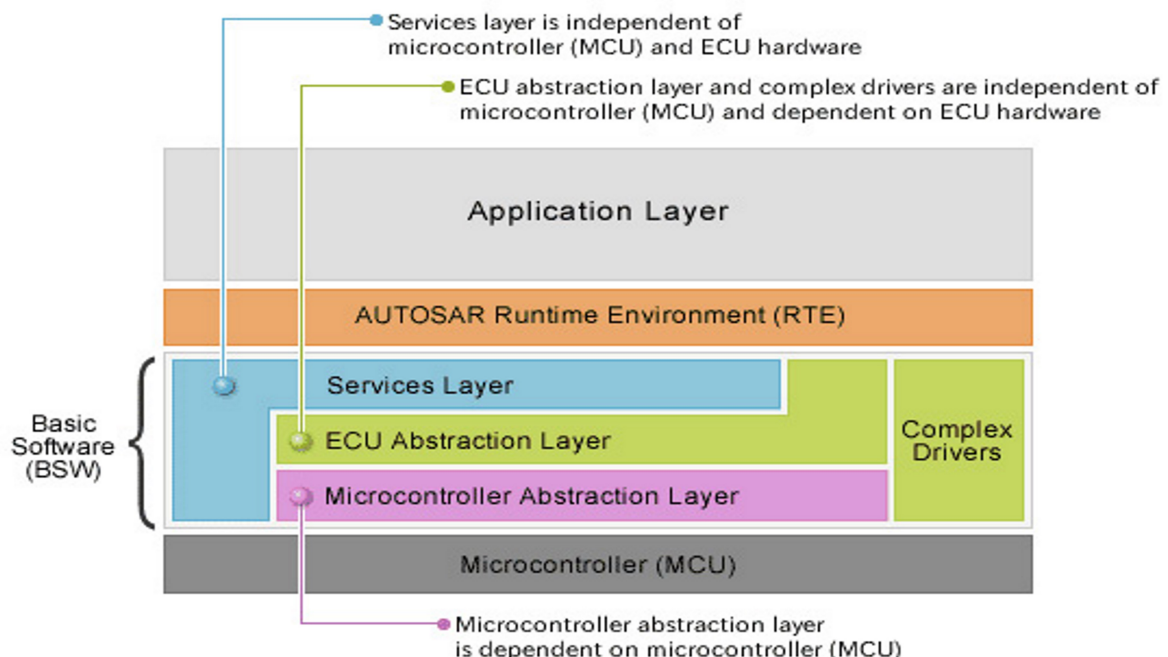


AUTOSAR

12 February 2021 12:34 PM

1) AUTOSAR

- **AUTOSAR (Automotive Open System Architecture)** can be defined as a common platform for the whole automotive industry that is designed to enhance the scope of application for vehicle functionality without affecting the current operating model. AUTOSAR is basically an open and standard software architecture which was **jointly developed by automobile manufacturers, suppliers and tool developers**.
- The main motto of AUTOSAR is **“Cooperate on standards, compete on implementation”**. This unique architecture was developed in order to establish and maintain a common standard among the manufacturers, software suppliers, and tool developers so that the outcome of the process can be delivered without the need of any alterations.
- Different layer in AUTOSAR Architecture



- **Application Layer**
 - The AUTOSAR application layer consists of various applications and specific software components that are designed to perform a specific task as per the given instructions.
 - The application layer is the topmost layer of AUTOSAR's software Architecture that's why it's critical for all the vehicle applications.
- **AUTOSAR RTE(Runtime Environment Layer)**
 - Middleware runtime environment layer providing communication services for AUTOSAR software components (SWC) and applications containing AUTOSAR sensor/actuator parts.
 - RTE has the responsibility of managing the life cycle of the SWC, It should start-up and shutdown the functions based on the needs.
 - It also acts as a separation layer between the Application Software (ASW) and the Basic Software layer(BSW) where the Base software had the permission to call any API function or other modules directly, but the Application software can only communicate through ports.
- **BSW(Basic Software Layer)**
 - The Basic Software layer can be defined as the standardized software that can provide

services to the AUTOSAR software components and it is also used to run the functional part of the software. The Basic software includes the standardized and ECU specified components.

- The Basic Software layer is further divided into 4 Major parts namely Services Layer, ECU Abstraction Layer, Microcontroller Abstraction Layer and Complex Drivers.

- **Service Layer**

- It is the topmost layer of the basic software layer, It provides the basic software modules to the application software and it is independent of the micro-controller and ECU hardware.
- The service layer provides functions such as
 - Memory Services (NVRAM Management)
 - Diagnostic services (Including UDS <protocol> communication and error memory)
 - Vehicle network communications and management
 - ECU state management
 - Operating System (OS)
 - This layer's mounting is specialized for micro-controller (MCU), Parts of the ECU hardware and their applications.

- **ECU Abstraction Layer**

- This layer acts as an interface of the micro-controller abstraction layer which also contains some drivers of external devices. It has access to the peripherals and the devices no matter where they are located either inside or the outside of the micro-controller. It also offers the API to interface with the micro-controller.

- **Microcontroller Abstraction Layer(MCAL)**

- Microcontroller layer is the access route to communicate with the hardware. This layer was framed in order to avoid direct access to micro-controller registers. The micro-controller Abstraction Layer(MCAL) is a hardware layer designed to ensure the standard interface to the components of basic software. It provides micro-controller independent values for the components of the basic software and also manages the micro-controller peripherals.
- The MCAL is provided with a notification mechanism so that it can support the distribution of commands, responses, and information to different process. Apart from this the MCAL can include some of the functions and devices such as Digital I/O (DIO), Analog/Digital Converter (ADC), Pulse Width (De) Modulator (PWM, PWD) , EEPROM (EEP), Flash (FLS) , Capture Compare Uni(CCU), Watchdog Timer (WDT), Serial Peripheral Interface (SPI), I2C Bus.

- **Complex Device Driver (CDD)**

- This layer has special timing and functional requirement for dealing with complex sensors and actuators. The CDD is used for handling complex functions, it can't be found in any other layers and it has the ability to access the microcontroller directly.
- The complex functions include injection control, Control of electrical values, Position increase detection, etc