AUTOSAR Interface Activities

Project Training –Automotive Overview

26/07/2024
Areeb Hammad N
Anuratha N
Atraindra Gupta
Anji Babu

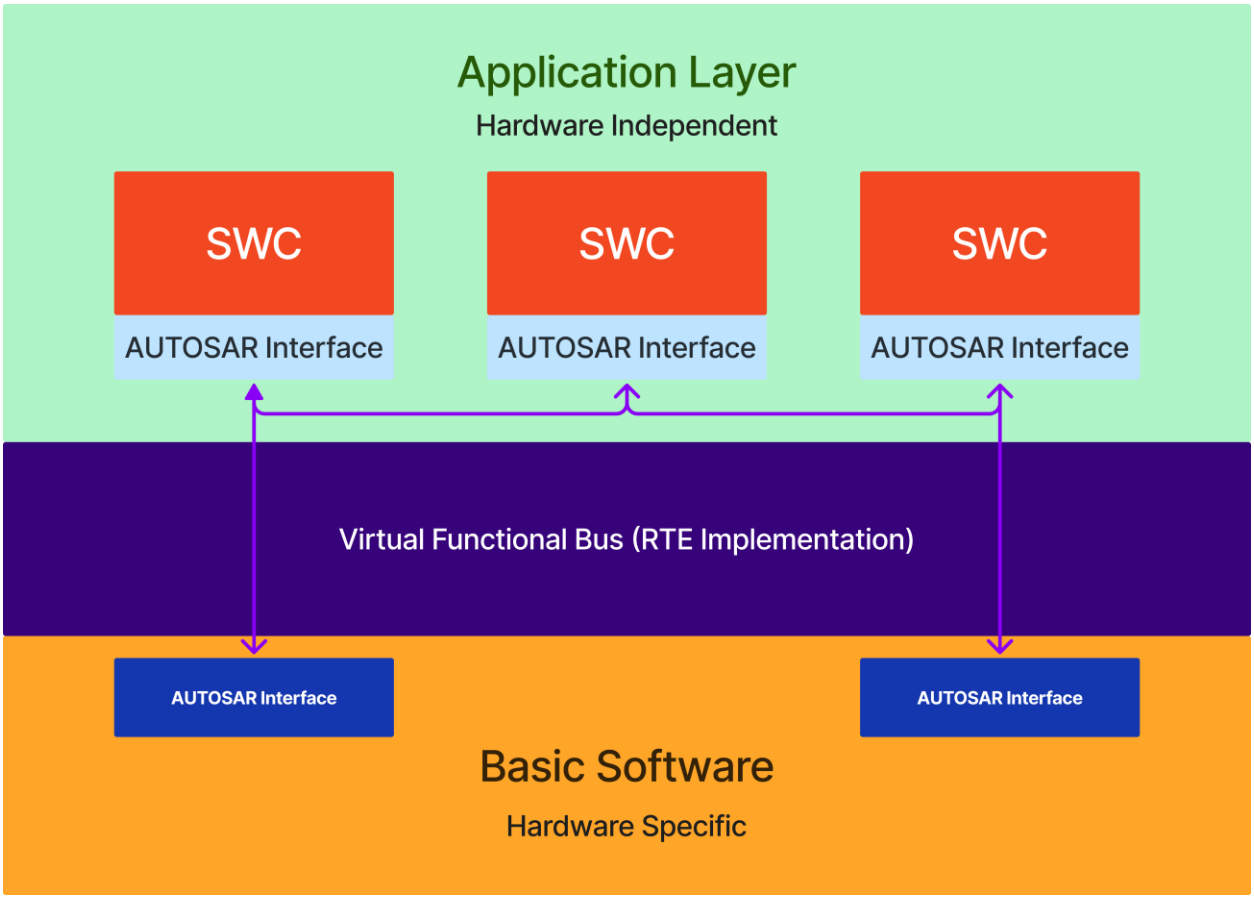
Version: 1.0

Created: 26/07/2024

Last Updated: 26/07/2024

Status: DRAFT (The status would change to finalized post the BA, PM and dev team review and sign off)

Task 1: Taking one example of each interface and design the AUTOSAR activities



AUTOSAR Interface Documentation

Introduction

AUTOSAR (Automotive Open System Architecture) is a standardized automotive software architecture. It aims to standardize software architecture, methodology, and application interfaces, thereby facilitating the

reuse of software across different platforms and suppliers. In this document, we will elaborate on the different types of interfaces within AUTOSAR, providing a detailed example for each interface type to illustrate their functionality and implementation.

Types of AUTOSAR Interfaces

1. **Sender/Receiver Interface**
2. **Client/Server Interface**
3. **Mode Switch Interface**
4. **NV Data Interface**
5. **Parameter Interface**
6. **Trigger Interface**

Example Scenario

Consider a vehicle's Climate Control System (CCS), which consists of several software components (SWCs) to manage temperature, air quality, and fan speed. This example will demonstrate how different AUTOSAR interfaces can be used within this system.

Components in the Climate Control System (CCS)

Temperature Sensor SWC
Fan Control SWC
Air Quality Sensor SWC
User Interface SWC
Memory Management SWC

1. Sender/Receiver Interface

The Sender/Receiver interface facilitates data communication between SWCs, where one component sends data and another receives it.

Example:

- **Sender SWC:** Temperature Sensor
- **Receiver SWC:** Fan Control

Activities:

Define Data Element:

– Create a data element `CurrentTemperature`.

Configure Sender SWC:

– Define `CurrentTemperature` as an output port in the Temperature Sensor SWC.

– Configure the data type and sampling rate.

Configure Receiver SWC:

– Define `CurrentTemperature` as an input port in the Fan Control SWC.

– Configure the reception method.

Mapping in RTE:

– Map the sender and receiver ports to the RTE to ensure proper data exchange.

2. Client/Server Interface

The Client/Server interface allows communication where one SWC requests services and another provides these services.

Example:

- **Client SWC:** User Interface
- **Server SWC:** Memory Management

Activities:

Define Service:

- Define a service `SaveUserSettings` provided by the Memory Management SWC.

Configure Server SWC:

- Implement the `SaveUserSettings` service in the Memory Management SWC.
- Specify input parameters (e.g., settings data) and output (e.g., success status).

Configure Client SWC:

- Define a client port in the User Interface SWC to request the `SaveUserSettings` service.
- Implement the request handling in the User Interface.

Mapping in RTE:

- Map the client and server ports to the RTE to enable service requests and responses.

3. Mode Switch Interface

The Mode Switch interface is used to manage different operational modes of an SWC.

Example:

- **SWC:** Fan Control

Activities:

Define Modes:

- Define modes such as `Off`, `Low`, `Medium`, and `High` for the fan.

Configure Mode Management:

- Implement mode switching logic in the Fan Control SWC.
- Define conditions and triggers for switching between modes.

Mapping in RTE:

- Map the mode switch interface to the RTE for real-time mode management.

4. NvData Interface

The NvData interface accesses non-volatile memory data, ensuring data persistence across power cycles.

Example:

- **SWC:** User Interface

Activities:

Define NV Data:

- Define non-volatile data elements, such as `PreferredTemperature`.

Configure SWC:

- Implement read and write operations for the NvData elements in the User Interface SWC.

Mapping in RTE:

- Map the NvData interface to the RTE to ensure data persistence.

5. Parameter Interface

The Parameter interface accesses configuration parameters that can be dynamically adjusted.

Example:

- **SWC:** Air Quality Sensor

Activities:

Define Parameters:

- Define parameters such as `SensitivityLevel` for the air quality sensor.

Configure SWC:

- Implement parameter access logic in the Air Quality Sensor SWC.
- Ensure parameters can be updated and retrieved.

Mapping in RTE:

- Map the parameter interface to the RTE for consistent parameter management.

6. Trigger Interface

The Trigger interface initiates actions based on specific events or conditions.

Example:

- **SWC:** Air Quality Sensor

Activities:

Define Triggers:

- Define triggers such as `PoorAirQualityDetected`.

Configure SWC:

- Implement trigger handling logic in the Air Quality Sensor SWC.
- Specify actions to be taken upon trigger events, such as activating the air purifier.

Mapping in RTE:

- Map the trigger interface to the RTE to ensure timely responses to events.