AUTOSAR – Communication Stack (56 Hrs)

Company Confidential **ANCIT**

AUTOSAR – Communication Stack Agenda from ANCIT

Delivery Format	:	This Course is offered in Classroom or Online Format
Duration	:	58 Hours (7 days)
Target Group	:	Embedded Engineers working in AUTOMOTIVE
Prerequisites	:	Knowledge about software development for automotive systems
Outcome	:	AUTOSAR COM Stack Configuration, Implementation & Testing with more emphasis on the Communication Stack

Day 1. AUTOSAR(ASWC, RTE and OS)

- Overview and Introduction to Architecture
- Embedded vs Legacy Architecture
- AUTOSAR Architecture – Significance of each layer
- AUTOSAR Methodology
- Types of Interfaces Supported by AUTOSAR
- VFB Design
- Types of Application SWCs
- Application Port Interfaces → Client Server, Sender Receiver
- Application Ports
- Types of Connectors
- Implementation of VFB design in tool
- Internal Behavior - Runnables, Triggers and Access Points

Day 2. RTE & OS

RTE

- RTE Contract Phase
- RTE Generation Phase
- RTE Scheduling Process
- RTE Communication Process
- How RTE helps in ensuring Data Consistency
- Different types of Interfaces Supported by AUTOSAR
- RTE Code flow
- RTE Integration

Operating Systems

- AUTOSAR OS vs OSEK OS
- OS States
- OS handling Interrupts
- Nuances in configuring the OS task
- Connecting ASWC, RTE and OS
- Code generation and Testing

AUTOSAR – Communication Stack
Agenda ANCIT EDUTECH

Day 3. COMMUNICATION and CAN Stack with Implementation → Data Flow

- Communication Module
- PDUR
- CANIF
- CAN Driver
- CAN Transceiver
- CANTP – introduction only
- Implementation of Communication Stack
- Loading the dbc file and monitoring the code flow from Com Module to CAN using ECUC reference
- Compare the dbc file and the PDUs and make sure the data constraints are applied and the PDU config in stack and dbc are same.

Day 4. Configuring the Com Module, PDUR, CANIF and CAN controller → Data Flow

- Trace the Signal/PDU in the com stack
- Configure the Com Module for requirements
- Configure the other modules and fix all the dependency errors
- Do Data Mapping
- Generate the Code
- Differentiate Com Send and Com Receive signal Behavior in code
  a) Monitor Update Bit behavior in code
  b) Gateway Functionality – Signal Routing, Application Routing and PDU Routing
  c) Application to Check the status of the COM manager and implement the logic to transmit data only if COMM is in FULL COM
- Handling Callbacks and Callouts
- Integration and Testing

Configuring the ComM and CAN SM → State Management

- COMM and CAN SM states and how they are related the status of the CAN controller and Can transceivers
- Understanding the Port interfaces that are the COM manager
- Using the application to change the status of the CAN Controllers using the AUTOSAR defined Port Interfaces

Day 5. BSW Manager and ECU Manager→ Network Management

- Mode Management Introduction and Implementation
- Configuring the BSW Manager
  a. Communication Management
  b. BSW Manager Module Initialization
- ECU Manager
  a. ECU Manager States
  b. ECUM Callouts
  c. ECUM Initialization sequence
  d. Integrating Initialization of external peripherals

AUTOSAR – Communication Stack
Agenda ANCIT EDUTECH

Day 6. Network Management use case implementation

- Can Nm
- Network Manager
- Put the ECU to sleep if CAN NM is not received over a period of time
- Wake up the ECU when the CAN message is received

Day 7. Memory Stack

- EEPROMdriver
- FlashDriver
- Fee
- EA
- MemIf
- NVM
- Memory stack Implementation from Non volatile memorySWC
    a) Store the Faults detected in the CAN in NvM

ANCIT

SNO : 37 Gurusamy Nagar,
Codissia Road, Peelamedu,Coimbatore,
Tamil Nadu, India- 641004

+91-9840378602/ 9483541953

info@ancitconsulting.com

www.ancitconsulting.com
www.ancitedutech.com