AUTOSAR – Basic 4 Weeks Training

Duration: **4 Weeks**

Delivery Format: **Classroom/Online**

Training Curriculum:

Week 1 – Automotive and AUTOSAR (ASWC, RTE, OS, IO and Cdd)

Day 1

AUTOSAR – Introduction & ASWC

- Overview and Introduction to Architecture
- Application Design in VFB Level
- Software Component
- Interfaces in AUTOSAR

Day 2

RTE & OS

- RTE Layer
- Implement RTE
- OS
- Implement the ASWC with OS and RTE

Day 3 and Day 4

RTE & OS Implementation

- Implementing the ASWC - Exploring all the possible RTE implementation.
- Integration of ASWC with other stacks
- Code navigation in Rte. Demonstrating how RTE Events trigger the runnables with the help of OS
- Code navigation of access points.
- Building wrappers for code migration

Day 5

Cdd and IO Stack

- IoHwAb Layer
- DIO Driver
- PORT Driver
- PWM Driver
- ICU/OCU Driver
- ADC Driver
- Implementation- Managing the IO Stack using IO Abstraction Layer
- Complex Device Drivers
- Implementation- Managing the IO Drivers using Complex Device Drivers- Seat Heater Application

Evaluation of Week 1

- Requirements to software requirements
- VFB design
- Accessing the hardware peripherals through CDD and IO
- Generating the code and testing the same

Week 2 – Communication Stack

Day 1

COMMUNICATION and CAN Stack with Implementation

- Communication Module
- PDUR
- CANIF
- CAN Driver
- CAN Transceiver
- IPDUM
- CAN TP
- 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 2

Configuring the Com Module, PDUR, CANIF and CAN controller

- 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 Behaviour in code
- Monitor Update Bit behaviour in code
- Gateway Functionality – Signal Routing , Application Routing and PDU Routing
- Application to Check the status of the COM manager and implement the logic to transmit data only if COMM is in FULL COM

Day 3

Handling Call Backs and Callouts

- Writing Call back functions for signal failure and other use cases
- Triggering IPDU callouts
- Handling Notifications
- Testing the Com stack using CANoe

Day 4

Mode Management and Implementation

- COM Manager
- CAN SM
- NMIF
- CAN NM
- BswM
- EcuM
- StdM
- SecOC
- Implementation of Mode Management – Configuring different wakeup sources
- Write application code to trigger the CAN bus and check the status of the Can bus before Can transmission.

Day 5

BSW Manager and ECU Manager

- Configuring the BSW Manager for Communication Stack and a mode Switch condition as given by ANCIT team
- ECU Manager understanding

Week 3 – Memory, Security and other managers

Day 1 and Day 2 Communication Stack Evaluation

Day 3

Memory Stack and Implementation

- EEPROM driver
- Flash Driver
- Fee
- EA
- MemIf
- NVM
- Implementation of Memory Stack from ASWC
- Boot Loader Introduction

Day 4

WatchDog Stack and Crypto

- WDG Driver
- WDG Manager
- WDG If
- Crypto Stack – CsM and CryIF
- Implementation

Day 5 and Day 6 – UDS Theory

Week 4 – Diagnostics Stack

Day 1, Day 2 and Day 3

UDS, Diagnostic Stack and Implementation

- DEM
- DCM
- FIM
- DET
- Implementation of Diagnostic Stack
    a. DTC Implementation
    b. NRC checks
    c. Service Id implementation

Day 4 and Day 5 Implementation of specific use case as evaluation