ANCIT- SDV Training Proposal

Company Confidential **ANCIT**

ANCIT- SDV Training Proposal

Delivery Format	:	Offline in-Campus Delivery / Online
Duration	:	10 Days
Target Group	:	Embedded Engineers in AUTOMOTIVE, ECU Developers, Project Leaders
Prerequisites	:	Knowledge about basic Linux OS, Traditional Vehicle Architecture, C/C++/Python for application development, knowledge on Automotive Protocols
Outcome	:	Participants will gain hands-on experience in initializing and configuring the HPC: S32G274A platform, enabling multicore communication, implementation of Hypervisor and Containerization. They will also implement Adaptive AUTOSAR, SOME/IP communication, and OTA updates for real-world ADAS use cases.

S32G274A Platform Initialization

Day 1: Introduction to SDV & Linux OS

1. Introduction to SDV
2. SDV Architecture - Vehicle SOA
3. Introduction about S32G274a HPC Gold box
4. Architecture of S32G274a HPC Gold box
5. Overview of Embedded Linux
6. Linux based ES component stack
7. Bootloader Operations
8. Kernel Architecture
9. Anatomy of a Linux based system.
10. Configuration & Build process of an Embedded Linux system
11. Communication between user and kernel space

Day 2: Getting Exposed to S32DS for M7 application development

1. Getting started with S32Design studio
2. Modules that needs to be configured to get the board up & running
3. Developing a simple use case for M7 Core
4. Getting the binary executables ready for implementation via BareMetal approach in S32DS.

Day 3 & 4: Initial Setting up process of A53 core on S32G274A

1. Yocto build project- a basic Introduction
2. Configuring & building a custom Yocto project for A53 Core
3. Flashing of Custom Yocto Image onto the SD Card
4. Booting process of Linux System on A53
5. Running a simple LED Blink Test example to Validate GPIO Access with A53 core
6. Test Ethernet Connectivity with A53 (via a simple Hello World Example)
7. Hands-on practice /Board Bring-up

ANCIT- SDV Training Proposal

Delivery Format	:	Offline in-Campus Delivery / Online
Duration	:	10 Days
Target Group	:	Embedded Engineers in AUTOMOTIVE, ECU Developers, Project Leaders
Prerequisites	:	Knowledge about basic Linux OS, Traditional Vehicle Architecture, C/C++/Python for application development, Knowledge on Automotive Protocols
Outcome	:	Participants will gain hands-on experience in initializing and configuring the HPC: S32G274A platform, enabling multicore communication, implementation of Hypervisor and Containerization. They will also implement Adaptive AUTOSAR, SOME/IP communication, and OTA updates for real-world ADAS use cases.

S32G274A Platform Initialization

Day 5: S32G274A Use case Implementations: Multicore Communication (IPCF)

1. Establish Multicore Communication Between A53 and M7
   - Custom Yocto Build for IPCF
   - Compile Kernel Object (KO) Files for IPCF
   - Mount IPCF KO Files
   - Flash IPCF Binary onto M7 Core
   - Validate Communication via Shared Memory

Day 6: S32G274A Use case Implementations: CAN communication via LLPC, Suspend-to-RAM, Containerization

1. Implement Suspend-to-RAM (STR) on A53
   - Introduction about STR
   - Develop C Program for Counter Logic
   - Execute STR via RTC Wake-Up Source in A53

2. Read CAN Signal on A53 Core via LLPC

3. Containerizing a sensor application
   - Why we need Dockers
   - Docker Installation
   - Basic commands used for Containerization process
   - Capture camera data using a Raspberry Pi node to send vision data to an ADAS ECU.
   - Containerize the application and demonstrating container execution on the Raspberry Pi.

Day 7: S32G274A Use case Implementations: Implementation of XEN Hypervisor

1. Implement Xen Hypervisor on S32G274A platform and run two guest OS on the Hypervisor
   - Introduction about Hypervisors & Types of Hypervisors
   - Custom yocto build for xen by modifying the configuration files [Local.config]
   - Flashing & Verifying the Privileged and Unprivileged Operating systems working on Xen

ANCIT- SDV Training Proposal

Delivery Format	:	Offline in-Campus Delivery / Online
Duration	:	10 Days
Target Group	:	Embedded Engineers in AUTOMOTIVE, ECU Developers, Project Leaders
Prerequisites	:	Knowledge about basic Linux OS, Traditional Vehicle Architecture, C/C++/Python for application development, Automotive Protocols
Outcome	:	Participants will gain hands-on experience in initializing and configuring the HPC: S32G274A platform, enabling multicore communication, implementation of Hypervisor and Containerization. They will also implement Adaptive AUTOSAR, SOME/IP communication, and OTA updates for real-world ADAS use cases.

S32G274A Platform Initialization, Adaptive AUTOSAR Framework Implementation, OTA Updates

Day 8: S32G274A Use case Implementations: ADAS use case over XEN Hypervisor

1. ADAS Use case Explanation
2. Introduction to Some-Ip concepts and different Methods of implementation
3. Implementation of SOME IP communication on A53 over one of the Guest OS with Pi as the server for fetching vision data for ADAS applications [Parking Assist System].

Day 9: Adaptive AUTOSAR Concepts and Applications

1. Introduction to Adaptive AUTOSAR Methodology
2. Need for Adaptive AUTOSAR
3. Overview of Service-Oriented Architecture (SOA) and its Necessity
4. Understanding POSIX and Why POSIX Compliance is Required
5. Architectural Differences Between Adaptive and Classic AUTOSAR
6. Design and Implementation of Adaptive AUTOSAR Application
   - Creation of arxml using ARTOP tool
   - Application design
   - Execution Manifest
   - Service interface manifest
   - Process design
   - And integrate with existing machine and network manifest
7. Service-Interface design for Communication within the ECU
8. Detailed explanation of ara::com
9. Hands-on Implementation of Adaptive AUTOSAR Communication via Some-Ip communication – [Specifically focuses on Method call technique].

ANCIT- SDV Training Proposal

Delivery Format	:	Offline in-Campus Delivery / Online
Duration	:	10 Days
Target Group	:	Embedded Engineers in AUTOMOTIVE, ECU Developers, Project Leaders
Prerequisites	:	Knowledge about basic Linux OS, Traditional Vehicle Architecture, C/C++/Python for application development, Automotive Protocols
Outcome	:	Participants will gain hands-on experience in initializing and configuring the HPC: S32G274A platform, enabling multicore communication, implementation of Hypervisor and Containerization. They will also implement Adaptive AUTOSAR, SOME/IP communication, and OTA updates for real-world ADAS use cases.

S32G274A Platform Initialization, Adaptive AUTOSAR Framework Implementation, OTA Updates

Day 10: OTA Updates and Cloud Connectivity via S32G274A as Gateway

1. Updating the application on Pi/G2 via Some-Ip

Approach1 – without AWS

- Setting up Local Server (Node.js/Django) for OTA
- Execute Script to Trigger Image Update and Reset the System
- Validate OTA Update Process

Approach2 – via AWS

- Deploy Local server on AWS
- Execute Script to Trigger Image Update and Reset the System
- Validate OTA Update Process

2. Updating the driver on G2/Pi via Some-Ip - [FOTA] / Kernel Update via another HPC [G2]

Note: (* under testing)

- Configure UBoot for Loading A53 Image from SD Card
- Write Script to Download New A53 Image and Store It on SD Card
- Configure UBoot to Load the New Image.

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

+91-9840378602/ 9483541953

info@ancitconsulting.com

www.ancitconsulting.com
www.ancitedutech.com