PiEST Systems
PIEST SYSTEMS (OPC) PRIVATE LIMITED

Automotive Embedded Software Developer Syllabus

Automotive Embedded Software Development – 6 Month Industry-Ready Course

Duration: 24 Weeks (6 Months)
Mode: Online / Offline / Hybrid
Focus: Full Automotive Stack (Protocols, RTOS, Diagnostics, Safety Standards, AI Integration, 2 Projects)

Month 1 – Foundations

Week 1–2: Embedded System & MCU Basics

- Basics of Electronics, Sensors, Actuators, ECUs
- STM32 Architecture, Memory Map, Boot Sequence

Lab Practical

- **Basic Electronics Component Identification**
  - Identify resistors, capacitors, diodes, sensors, actuators
  - Skill: Hardware familiarity
- **Multimeter Measurements**
  - Measure voltage, current, resistance, continuity
  - Skill: Measurement confidence
- **STM32 Board Familiarization**
  - Identify pins, power, clock, reset, debug interface
  - Skill: MCU hardware understanding
- **Boot Sequence Observation**
  - Observe reset, vector table, Flash execution
  - Skill: Startup flow clarity

Week 3–4: Embedded C & Peripheral Programming

- Register-Level Coding, GPIO, Timers, UART, ADC
- Makefile, Compiler, Linker Script

Lab Practical

- LED blink using GPIO
- Button input & debounce observation
- UART logging for debugging

- Timer interrupt generation
- Makefile build analysis

Month 2 – Real-Time Concepts & RTOS

Week 5–6: Operating Systems Fundamentals

- OS vs RTOS
- Processes, Threads, Scheduling

Lab Practical

- Bare-metal vs RTOS timing comparison
- Task scheduling visualization

Week 7–8: FreeRTOS

- Task Creation, Priorities, Queues, Semaphores
- Inter-Task Communication, Memory Management

Lab Practical

- Multi-task LED control
- Queue-based UART logger
- Semaphore protection demo

Week 9–10: Zephyr RTOS

- Zephyr Architecture, Build System (CMake/Kconfig)
- Threads, Device Tree, Drivers

Lab Practical

- Zephyr build & flash using west
- CAN communication using Zephyr

Month 3 – Automotive Communication & Diagnostics

Week 11–12: CAN, CAN-FD, CAN-XL

- Frame Structure, Bit Timing, Arbitration
- STM32 CAN Peripheral Tx/Rx

Lab Practical

- CAN Tx/Rx using STM32
- Arbitration observation
- Error counter monitoring

Week 13–14: CAN-TP & UDS

- Multi-Frame Communication, Flow Control
- UDS Services (0x10–0x3E, DTC Management)

Lab Practical

- Multi-frame CAN-TP transfer
- UDS 0x22 ReadDataByIdentifier
- UDS 0x2E WriteDataByIdentifier

Week 15: LIN, FlexRay, Ethernet

- LIN Scheduling
- FlexRay Overview
- TCP/UDP Basics

Lab Practical

- LIN frame observation
- Ethernet packet capture (Wireshark)

Month 4 – Debugging, Toolchain & Development Workflow

Week 16–17: Software Development Workflow

- Git, GitHub, Jira
- Make, CMake, West
- Startup File, Linker File, Cross-Compilation

Week 18: Debugging & Bug Fixing

- Breakpoints, Trace Debugging
- Memory Faults, Stack Overflow, Race Conditions
- Adding New Features to Existing Project

Lab Practical

- Git version control workflow
- Debugging hard faults & stack overflow
- Adding new feature to existing code

Month 5 – AI Tools & Automation

Week 19–20: Advanced AI Integration

- Using ChatGPT & Copilot for requirement understanding
- AI-assisted code generation & documentation
- Automated Unit Test & Review Scripts
- Embedded ML (TinyML, Edge Impulse)

Lab Practical

- ChatGPT for code explanation
- AI-assisted documentation generation
- Basic automated test idea

Month 6 – Projects & Automotive Standards

Week 21–22: Project 1 – CAN-Based ECU Communication

- STM32 + RTOS
- Sensor Data Transmission & Diagnostics

Week 23–24: Project 2 – Blind Spot Detection System (BSD)

- Radar/Ultrasonic Sensors
- CAN Bus Communication & DTC Implementation
- Final Documentation & Presentation

End Module – AUTOSAR & Functional Safety

- AUTOSAR Classic & Adaptive Architecture
- ISO 26262 Functional Safety, ASIL Levels
- MISRA-C Guidelines
- ASPICE, ISO/SAE 21434 Cybersecurity Basics
- ECU Integration Flow in OEMs

Outcome (6-Month Program)

- Master Automotive Protocols, RTOS & ECU Software Lifecycle
- Hands-on with FreeRTOS, Zephyr, UDS, AI Tools
- Industry-ready portfolio with **two real automotive projects**

Automotive Embedded Communication & Diagnostics – 3 Month Professional Course

Duration: 12 Weeks
Focus: CAN, CAN-TP, UDS + Project
(No RTOS included)

Month 1 – Communication Fundamentals & CAN

Week 1: Embedded Recap & ECUs

- STM32 GPIO, UART, Timers Recap
- Automotive ECUs Overview
- OSI Model Basics

Lab Practical

- UART and CAN signal comparison

Week 2–3: CAN (ISO 11898)

- Bus Topology, Arbitration, Frame Formats
- Standard vs Extended IDs
- STM32 CAN Tx/Rx
- Error Handling, Interrupts
- PCAN-View, BusMaster Testing

Lab Practical

- CAN frame transmission & reception
- Standard vs extended ID testing
- CAN filtering & masking
- Debugging using PCAN / BusMaster

Week 4: Advanced CAN

- ❖ Multi-Node Communication
- ❖ Filtering, Masking
- ❖ Debugging with Oscilloscope/Analyzer

Lab Practical

- ➢ Multi-node CAN communication
- ➢ Oscilloscope / logic analyzer observation

Month 2 – Transport & Diagnostics Protocols

Week 5–6: CAN-TP (ISO 15765-2)

- ❖ Segmentation & Reassembly
- ❖ Flow Control
- ❖ STM32 Implementation

Lab Practical

- ➢ Long data transmission using CAN-TP
- ➢ Flow control testing

Week 7–9: UDS (ISO 14229)

- ❖ Request/Response Model
- ❖ Key Services:
  - 0x10 – Diagnostic Session Control
  - 0x11 – ECU Reset
  - 0x22 – ReadDataByIdentifier
  - 0x2E – WriteDataByIdentifier
  - 0x19 – DTC Information
- ❖ UDS over CAN (DoCAN)
- ❖ Diagnostic Tester Simulation

Lab Practical

- ➢ Diagnostic session control (0x10)
- ➢ ECU reset (0x11)
- ➢ DID read/write (0x22, 0x2E)
- ➢ DTC storage and clearing

Month 3 – Project & AI Integration

Week 10–11: Project – CAN-UDS Based ECU

Example: Engine Data Diagnostic ECU

- Sensor Data via CAN
- UDS 0x22/0x2E Implementation
- Diagnostic Tester Verification
- Report & Documentation

Week 12: AI Tools for Embedded Developers

- ChatGPT, Copilot
- Code Optimization & Documentation
- Generating Test Cases
- Best Practices & Limitations

Outcome (3-Month Program)

- Full understanding of CAN, CAN-TP & UDS
- Able to design & test a diagnostic ECU
- Ready for entry-level automotive embedded roles

Automotive Embedded Software Development - 1 Month Fast-Track Course

Week 1 – Embedded Fundamentals

- Basic Electronics
- Sensors & Actuators
- Microprocessor vs Microcontroller
- Automotive ECUs Overview

Lab Practical

- Component identification
- Voltage, current, resistance measurement
- Ohm's law verification

Week 2 – ARM & STM32 Basics

- Cortex-M Architecture
- STM32 Series (F0, F4, H7)
- Pinout, Memory Map
- Toolchain Setup

Lab Practical

- STM32 pinout identification
- Boot mode testing
- Toolchain setup verification

Week 3 – C & Embedded C

- Pointers, Structures
- Bitwise Operations
- volatile, const, static
- Makefile & Linker Script Basics

Lab Practical

- Bitwise operations
- Register access practice
- Makefile build flow

Week 4 – STM32 Labs

- GPIO LED/Button
- Timers & Interrupts
- UART Serial Communication
- ADC Sensor Reading (Optional)
- **Mini Project:** Sensor Value Display via UART

Outcome (1-Month Program)

- Able to program STM32 MCU
- Understand Embedded C
- Ready for next-level automotive courses