
12-Day Embedded Firmware Development Training Program

Day 1: Embedded Systems Foundations

1. **Introduction to Embedded Systems**
    - **Overview of Embedded Systems in IoT, Automotive, and Consumer Electronics**
    - **Key Components: Microcontrollers, Sensors, Actuators**
    - **Real-Time Constraints and Requirements**
2. **Development Tools for Embedded Systems**
    - **IDEs: STM32CubeIDE, Keil, VS Code**
    - **Debuggers: JTAG, SWD**
    - **Simulators and Emulators**
3. **ARM Cortex-M Architecture**
    - **Features: Low-Power Design, Interrupts, Exceptions**
    - **System Control Block (SCB) and Fault Handling**
4. **Introduction to Bare-Metal Programming**
    - **Setting Up Toolchains (GCC, Keil, STM32Cube)**
    - **Writing and Debugging Bare-Metal Applications**

Day 2: Bare-Metal Programming Fundamentals

1. **GPIO and Timer Basics**
    - **GPIO Pin Configuration and Blinking LEDs**
    - **Timer Configuration for Delays**

2. **Interrupt Programming**
    - **NVIC (Nested Vectored Interrupt Controller) Overview**
    - **Writing ISRs (Interrupt Service Routines)**
3. **Communication Protocols – Part 1**
    - **UART: Configuration and Data Transmission**
    - **Hands-On: UART Loopback**
4. **Communication Protocols – Part 2**
    - **SPI and I2C Overview**
    - **Multi-Device Communication**

Day 3: Advanced Bare-Metal Programming

1. **Power Management**
    - **Low-Power Modes: Sleep, Stop, Standby**
    - **Power-Saving Techniques**
2. **Advanced Interrupt Management**
    - **Nested Interrupts and Priorities**
3. **Memory Management**
    - **Flash, Stack, Heap, and Optimization Techniques**
4. **Debugging Techniques**
    - **Using SWD, Serial Debugging Tools**

Day 4: RTOS Foundations

1. **RTOS Introduction**
    - **Why Use an RTOS? Benefits Over Bare-Metal Programming**
    - **Key Concepts: Tasks, Scheduling, Context Switching**
2. **FreeRTOS Basics**
    - **Setting Up FreeRTOS**
    - **Task Creation and Management**
3. **Inter-Task Communication in FreeRTOS**
    - **Queues, Semaphores, Mutexes**
4. **Hands-On Debugging with FreeRTOS**
    - **Using Trace Tools for Debugging**

Day 5: Advanced FreeRTOS Programming

1. **Task Management**
    - **Task Priorities, Delays, Yielding**
2. **Real-Time Scheduling**
    - **Static vs Dynamic Priority Scheduling**
3. **Advanced Peripheral Management in FreeRTOS**
    - **Managing UART, SPI, I2C in RTOS**
4. **Low-Power Techniques in FreeRTOS**
    - **Building a Power-Efficient System**

Day 6: Firmware Verification and Project Allocation

1. **Secure Firmware Development**
    - **Implementing Secure Bootloaders**
    - **Basics of OTA Updates**
2. **Firmware Verification Techniques**
    - **Unit Testing, Integration Testing, System Testing**
    - **Tools: Ceedling, Unity, CMock**
3. **Code Quality and Static Analysis**
    - **MISRA-C Guidelines**
    - **Using Tools like SonarQube, PC-lint**
4. **Project Allocation**
    - **Explanation of Project Topics (e.g., RTOS-based device, memory optimization)**
    - **Forming Teams or Individual Work Plans**

Day 7: Advanced RTOS Concepts

1. **Advanced Task Synchronization**
    - **Using Event Groups and Notifications**
2. **Dynamic Memory Allocation in FreeRTOS**
    - **Understanding Heap Management**
3. **Advanced Communication Techniques**
    - **Stream Buffers, Message Buffers**
4. **RTOS Security**
    - **Implementing Secure Task Management**

Day 8: Boot Process and Board Bring-Up

1. **Microprocessor and Microcontroller Boot Process**
    - **How the Boot Process Works in Embedded Systems**
    - **Bootloaders and System Initialization**
2. **Board Bring-Up**
    - **Initializing GPIOs, Clocks, Peripherals**
    - **Debugging a New Board**
3. **Memory Subsystems**
    - **SRAM, ROM, Flash Organization and Usage**
4. **Hands-On: Initializing Hardware Components**
    - **SD Card Integration**
    - **Interfacing EEPROM**

Day 9: Profiling, Optimization, and Hybrid Programming

1. **Debugging Multithreaded Applications**
    - **Advanced Debugging Techniques in FreeRTOS**
2. **Performance Profiling**
    - **Using Tracealyzer, Segger SystemView**
3. **Hybrid Programming Models**
    - **Bare-Metal and FreeRTOS Co-Existence**
4. **Practical Project Progress Checkpoint**
    - **Teams/Individuals to Demonstrate Progress**

Day 10: Advanced Firmware Techniques

1. **Real-Time Data Monitoring and Logging**
    - **Efficient Data Logging in RTOS**
2. **System Integration and Testing**
    - **End-to-End System Testing and Validation**
3. **Hands-On: Debugging Complex Systems**
    - **Logic Analyzers and Oscilloscopes**
4. **Case Studies in Firmware Development**
    - **Automotive, IoT, and Industrial Applications**

Day 11: Extended Advanced Topics

1. **Advanced Memory Debugging**
    - **Memory Leaks Detection and Prevention**
    - **Debugging Heap Fragmentation Issues**
2. **Security in Embedded Systems**
    - **Encryption/Decryption Techniques**
    - **Secure Communication Protocols**
3. **Complex Peripheral Integration**
    - **Interfacing Advanced Sensors and Actuators**
    - **Using PWM and ADC for Motor Control**
4. **Project Fine-Tuning and Final Testing**
    - **Teams/Individuals to Prepare for Presentation**

Day 12: Capstone Project Presentation and Wrap-Up

1. **Capstone Project Presentation**
    - **Teams or Individuals Present Their Completed Projects**
2. **Feedback and Discussion**
    - **Peer and Instructor Feedback**
3. **Future Directions in Embedded Firmware Development**
    - **Emerging Trends in Embedded Systems and RTOS**
4. **Program Conclusion**
    - **Wrap-up**