

Embedded Systems Bare-Metal Programming

Ademtek's Embedded Bare-Metal Programming training course syllabus and curriculum is systematically designed in sync with industry needs. It will enable you to build a complete Embedded Solutions on your own. Following are

Objectives:

- ✓ Enable you to read and understand data-sheets and hardware manuals
- ✓ Setup, develop and download Embedded applications into a target hardware
- ✓ To provide a basic idea of hardware/electronics aspects of programming
- ✓ Enable you to program any micro controller
- ✓ To get you started with basic Embedded application development with ease

Module topics:

- ✓ *Introduction to embedded systems*
- ✓ *Microprocessors vs. Microcontrollers with intense hardware focus*
- ✓ *Hands-on working with GPIOs, Analog I/Os, Memory usage, interfacing etc.*
- ✓ *Micro controller peripherals usage - Timers, Counters, Interrupts and its sources*
- ✓ *Communication protocols – CAN, UART, SPI, I2C etc*



Modules

Mod1:Microcontroller Embedded C Programming

Program microcontrollers with C programming language

Learn the basics of coding in C, Make a user friendly program

Trace errors in your Code easily and effectively

Mod2: Master bare metal embedded system programming with AVR uC

Learn embedded C, assembly & mixed programming, AVR microcontroller internals, GPIO, EEPROM & more

Mod3: PIC Microcontroller: Architecture & Embedded C Programming

PIC Microcontroller architecture, interfacing real time devices and embedded C programming

Mod4: Embedded Systems Bare-Metal Programming (STM32)

No Libraries used, Professional CMSIS Standard, ARM Cortex, ADC,UART,TIMERS, DMA,SPI,I2C,RTC,GPIO etc.

Mod5: Advanced Embedded Systems Bare-Metal Programming(STM32)

No libraries : RTC, IWDG, WWDG, DMA-ADC, DMA-I2C, DMA-SPI, DMA-UART, DMA-PWM, Standby, Wakeup

Module 1: Advanced / Embedded C Programming

Objectives:

- ✓ Clearly understand concepts of C language
- ✓ To obtain good quality and style in programming
- ✓ Gear you up for programming in Embedded environment
- ✓ To induce confidence in you!

Detailed course contents:

✓ Basics of C:

- Operators
- Conditionals
- Arrays
- Functions

✓ Advanced C programming:

- Structures
- Unions
- Files
- Deep dive into pointers
- Pre-processor directives
- Recursion

- ✓ Project environment - Creating & Building a project, Makefiles
- ✓ Deep dive - Logic to program translation, Creating your own library, Dry-run
- ✓ Introduction to Data Structures

Master bare metal embedded system programming with AVR uC

- 1) AVR Micro-controller architecture & it's working - Instruction execution, pipelining, stack pointer & interrupts
- 2) Arguments and return values passing from a function to another, how memory & CPU registers are allocated during program execution
- 3) Interrupts, interrupt vector table (IVT) and interrupt service routine (ISR), IVT & ISR programming
- 4) Setup in-circuit programmer and USB-Serial adapter, flash & configure ATmega328 micro-controller
- 5) C, AVR assembly programming & implement mixed language projects
- 6) How hardware manages program stack and management of local & global variables
- 7) Build environment setup, GNU tools & utilities & build automation using GNU Make
- 8) GPIO Programming, 4x4 keypad interfacing & driver development

Bare Metal Programming PIC Microcontroller

- 1) PIC Micro-controller architecture & it's working - Instruction execution, pipelining, stack pointer & interrupts
- 2) GPIO Programming
 - 1) GPIO Registers
 - 2) PIC Microcontroller Board and LED Flashing
 - 3) Interfacing of Relay, Buzzer, switch and LEDs
 - 4) MPLAB Software for Simulation of Program
 - 5) Basics of LCD Interfacing
 - 6) 16 x 2 LCD Features and Pin Diagram
 - 7) LCD Interfacing Embedded C Program
 - 8) LCD Interfacing Embedded C Program (Using PPT)
- 3) Timer Theory and Programming
 - a) Timer 0 : Theory and Programming
 - b) Timer 2: Theory and Programming
 - c) PWM Signal Generation :CCP Module : Part 1
 - d) PWM Signal Generation :CCP Module : Part 2 (Program)
 - e) PWM Signal Generation
- 4) ADC Programming
- 5) Serial Protocols
 - a) Basics of Serial Communication
 - b) USART Block Registers
 - c) USART: Theory and Programming
 - d) Serial Data Transmission Program, compilation and Execution using PIC KIT
 - e) Basics of I2C Protocol
 - f) MSSP Block of PIC: I2C Module and Programming
 - g) RTC Interfacing: using I2C
 - h) SPI Protocol
 - i) MSSP Block of PIC: SPI Module and Programming

Mod5:Embedded Systems Bare-Metal Programming(STM32)

- Write firmware using only bare-metal embedded-c
- Write Analog - Digital Converter (ADC) drivers using bare-metal embedded-c
- Write UART drivers using bare-metal embedded-c
- Write Interrupt drivers using bare-metal embedded-c
- Write I2C drivers using bare-metal embedded-c
- Write DMA drivers using bare-metal embedded-c
- Use No third party libraries or header files
- Use the debugger effectively to analyze and resolve any bugs
- Define addresses for the different peripherals
- Create registers from the addresses
- Understand the Cortex-M Architecture
- Write PWM drivers using bare-metal embedded-c
- Write TIMER drivers using bare-metal embedded-c
- Write SPI drivers using bare-metal embedded-c
- Master the ARM-Cortex CMSIS standard
- Build every single line of code from scratch by writing to the microcontroller's memory space directly.
- Analyze the chip documentation