



Embedded Systems

Connecting Engineers and Developers with Practical Skills and the Latest Technologies

Certification in Embedded Systems

The Course at a Glance

Course Contents

Level-I

Introduction to Embedded Systems

- Embedded System Features, Current Trends and Challenges in Technology
- Embedded System Product Life Cycle
- Embedded Software Development Environment
- Processor, Language, Tool chain,
- RTOS Selection and their Impact in Design and Development
- Debugging Techniques

C Programming

- Machine Language through High-Level Languages
- Understanding Tools-Assembler, Compiler, Linker, Loader, Locator etc.,
- Executable File Format and Storage Sections Allocation-Text, Data, Stack, Heap, BSS etc. and Runtime Behavior
- Fundamental Data Storage and Manipulation-Types and Variables, Statements and Expressions
- Flow of Control, Boolean Expressions, Conditional Statements, and Loops
- Modular Programming: Functions with and without Return Values, Actual and Formal Parameters, etc.
- Arrays, Matrices, and Vectors. One- and Two-dimensional Arrays, Simple Sort Algorithms, Matrix Addition, Subtraction, Multiplication, Transposition, etc.
- Introduction Pointers, Indices, Recursion, Objectives-Timing Comparisons, Memory Comparisons, Array with Pointers
- Structure, Union, Nested Structure, Implementation-Array/Linked, Ordered/Unordered, Pointers and Addressing, Dynamic Memory Allocation
- Stacks and Queues, Linked Lists (Linked Memory, Recursive List Methods, and Merge Sort on Linked Lists)
- Trees, Pointer Implementation, Traversal, Binary-trees
- Binary Search Trees-Definition, Searching, Creation and Insertion, Good and Bad Trees, Deletion
- Searching and Sorting Algorithms: Linear and Binary Search
- Selection Sort, Insertion Sort, and Quick Sort - Introduced via Recursive Versions
- Hashing-Initial Hash, Collisions, Separate Chaining, Heap as Priority Queue, Heap Sort
- ASCII Data Files, File Pointers, End-of-file (Stream and Structural Handling)

MCS-51 Architecture and Programming

- MCS-51 Architecture and Memory Organization
- Hardware and Electrical Characteristics
- Addressing Modes and Instruction Set
- I/O Port Construction and Programming
- Timer/Counter Programming
- Serial Communication
- Interrupt and Services

Interfacing Devices and Device Driver

- Interfacing Requirement Analysis, EEPROM and SRAM Interfacing

- LCD, Stepper Motor
- Sensors and Transducers
- ADC/DAC

Firmware Development

- Firmware Architecture and Target System Specification
- Bringing Command Line Interface
- Remote Debugging and Debug Monitor
- File/Data Transfer Protocol
- Multitasking Firmware



Embedded Networking

- Data Communication Network Architecture, Protocols and Standards
- Standard Organization for Data Communication, Layered Network Architecture
- Open System interconnection (OSI) Model
- TCP/IP Protocol Suite, Addressing, TCP/IP Version
- Local Area Networks, IEEE 802.3 Ethernet Interface, Token Ring, Wireless LANs, Connecting Devices, Repeaters, Bridges, Routers, Switches
- IP Addresses, Classful Addressing, Unicast, Multicast and Broadcast Addresses, Subnetting, Supernetting, Connection oriented Vs. Connectionless, Static, Dynamic Routing, Routing Table
- ARP, RARP
- Internet Protocol (IP), Datagram, Fragmentation, ICMP, IGMP
- UDP, UDP Datagram and Operation, TCP, TCP Services, Flow Control, Socket Interface, BOOTP and DHCP, FTP, TFTP

Networking with Rabbit-3000

- Rabbit 3000 Architecture and CPU Registers
- I/O Instructions, External I/O and Auxiliary I/O Bus
- Input Capture Channels, Quadrature Encoder Inputs, Pulse Width Modulation Output
- Peripheral Device I/O Registers and Internal I/O Registers
- Rabbit Parallel and Serial Ports
- Timers, External Interrupts, Interrupt Vector, Watchdog Timer, Real-Time Clock
- Memory Management, Memory Mapping Unit (MMU), Memory Interface Unit (MIU), Separate I and D Space
- Dynamic C, TCP/IP APIs and Network Application Development

Level-II

Computer Organization, GNU Tools, PowerPC Architecture and MPC850

- Basic Structures of Computers and Functional Units, Bus Structures, Storage Representation, Memory Locations & Addresses, Combinational Logic Networks, Arithmetic/Logic/Sequential Units
- CPU Design-Clock, Datapath and Control Unit, Basic Pipelining and Branch Prediction, PowerPC Architecture, Core Registers
- Instruction Sets, Addressing Modes, Linking and Loading, Subroutines, etc. ALU and Basic Processor Design
- MPC850 Overview and Memory Map, MPC850 Instruction Set, Introduction to Cygwin
- GNU-X-Tools Utilities-as, ar, nm, ld, etc., Compiler and Libraries, Makefiles
- Exceptions, MMU and Instruction and Data Cache
- System Interface Unit (SIU)
- Memory Controller, Linkers, Memory Map and Linker Scripts
- Communication Processor Architecture and Configuration
- Serial Management Controller (SMC) Configuration and Configuring SMC as UART
- Serial Communication Controller (SCC)
- System Debugging and Testing Support

OS Concepts and Embedded Linux

- Introduction to Linux, Kernel Architecture, Operating System Components-Process Management, Memory Management, IPC, File System. I/O Management, etc.
- Processes and Concurrent Programming, Threads, Process Management, States, Transitions, Processor Scheduling, Creating and Destroying Process.
- Memory Addresses, Segmentation and Paging, Linux Memory Management, Real and Virtual Memory, Paging and Segmentation, Fetch, Placement, and Replacement Algorithms, Thrashing
- Interrupts and Exceptions, Interrupt Signals, Nested Exception and Interrupt Handlers, Interrupt Descriptor Table
- Synchronization, Mutual Exclusion, Synchronization, Monitors, etc. Inter Process Communication, Semaphore, Pipes, Message Queue, etc.
- Deadlock and Indefinite Postponement, Prevention, Avoidance, Detection, Recovery, etc., Priority Inversion and Priority Inheritance, etc.
- Data Structure, System Calls and Signals, etc.
- Ext2, Ext3, JFFS2 and Cramfs File Systems
- Disk Space Management and Allocation, Seek and Rotational Optimization, Blocking and Buffering, File Systems, Directory Structures, Access Methods, Access Control

Linux Device Driver

- Compiling Issues and Kernel Version, Linux Module Programming, Modules Vs Programs

- User Space Vs Kernel Space, Name Space, Code Space
- Device Drivers, Major and Minor Numbers, Character Device Driver, File Structure
- Registering/Unregistering Devices, Block, Network Device Driver Concepts
- Writing Modules for Multiple Kernel Versions, proc File System and IOCTLs
- Serial and Parallel Device Driver Implementation

Running Linux on PowerPC MPC850

- Linux Boot Process and Boot Configuration, System Memory Layout
- Cross Platform Development Tools Setup, Building GNU Tools, Different C Libraries
- Kernel Selection, Configuration and Customization and Command line Arguments
- Root File System Structure
- Libraries, Device Files, Custom Applications, System Initialization,
- File System Selection, NFS-Mounted Root File System, Architecture Dependent Code Modification, Board Specific Data

Real Time Systems Fundamentals and RTLinux

- Real Time Systems, Release Time, Absolute and Relative Deadline
- Hard and Soft Real Time Systems
- Asynchronous, Synchronous and Isochronous Event
- Predictable and Deterministic Systems, Race Condition
- Critical Section of Code, Resource, Shared Resource, Multitasking, Task, Context Switch (or Task Switch)
- Atomic Operation, Kernel, Scheduler, EFD and LST, Hierarchical Scheduling, Scheduling Jitter
- RMA and RMS, Non-Preemptive Kernel, Preemptive Kernel, Reentrancy. Round-Robin Scheduling
- Real Time Terms and Scheduling Analysis, Architecture of RTLinux, RTCore OS, Interrupt Emulation
- POSIX Real-time Extensions, The POSIX PSE 51 Standard
- Thread Architecture and Multithreading, Thread Creation, Thread Joining, Thread Destruction, Thread Management, Thread Attribute Functions
- Synchronization, POSIX Spinlocks, Mutexes, Conditional Variables, Semaphores, API Namespace, Resource Cleanup, Deadlocks, Synchronization-induced Priority Inversion, Atomic Operation
- Basic Communication, Real-time FIFOs, Signaling and Multithreading, Memory Management, Simple Memory Allocators/Deallocators
- Shared Memory, Soft-IRQs
- IRQ-Control, Debugging Hints, Device Driver

Course Duration

3 Months, 68 Working Days, Weekly 6 Classes, 4 Hrs. a Day



Nurture

Technologies