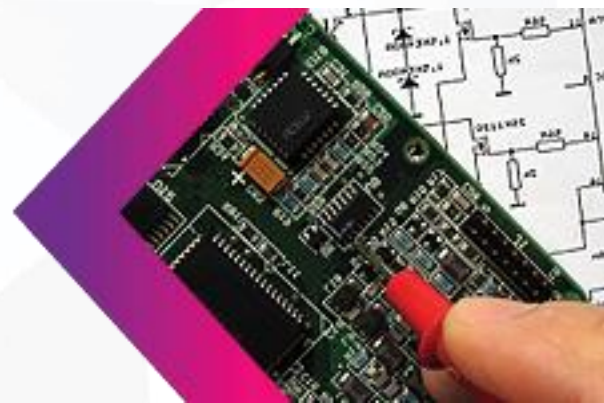
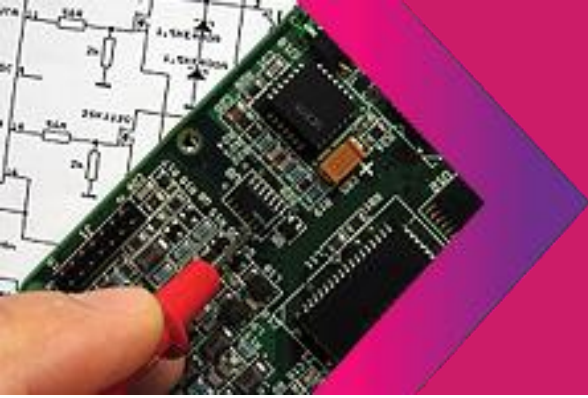




# ECEP - Course syllabus





NOW FOR SIMPLIFIED SOLUTIONS

## Module: 1/7

### Module name: Linux Systems

- **Objectives:**

- ✓ To get familiar with Linux Operating system - Commands, tools and editors
- ✓ Enable you to write Shell scripts
- ✓ To understand different stages involved in code compilation
- ✓ Enable you to create projects and automate the build using Make

- **Overview:**

This kick-start module is intended to get new programmers up and running with Linux embedded environment. Starting with basics of Linux and its features, this module dives into essential commands, Shell scripting and using tools like Make, GDB, Vim. By getting to know powerful Linux commands for redirection and pipes, lay strong foundation to automate routine tasks. The module then deals with compilation stages, coding guidelines and creating new projects using Make.

- **Duration:**

2 weeks

- **Platform:**

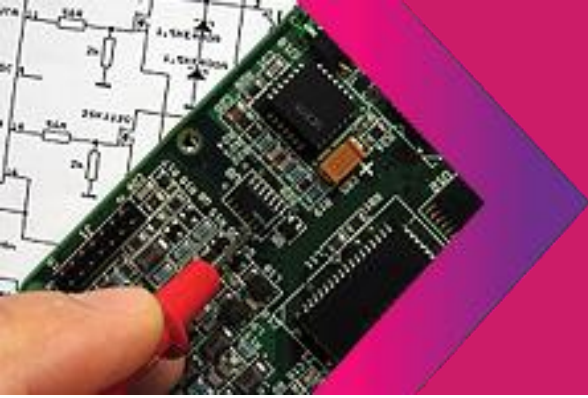
Linux (Fedora / Mandriva / Ubuntu)

- **Delivery method:**

Instructor-lead, Worksheets, Assignments with hands-on

- **Module topics:**

- *Overview of Linux OS*
- *Command Line Interface*
- *Linux Shell programming*
- *Environment variables, shell commands, filtering commands*
- *User accounts, remote login, redirection, pipes, Directory and File system structure*
- *Visual editor(VIM)*
- *Makefiles*
- *Code compilation stages, coding guidelines and creation of project and libraries*



NOW FOR SIMPLIFIED SOLUTIONS

## Module: 2/7

### Module name: Advanced C

- **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!

- **Overview:**

Lot of book tells about what is programming. Many also tell how to write a program, but very few cover the critical aspect of translating logic into a program. Specifically, in this fast paced industry, when you don't have time to think to program, this course comes really handy. It builds on the basics of programming, smooth sailing through the advanced nitty-gritty's of the Advanced C language by translating logic to code. Every class is backed by discussion and topic related assignments. Finally a project letting you apply most of the concepts learned throughout the module

- **Duration:**

1+ month

- **Platform:**

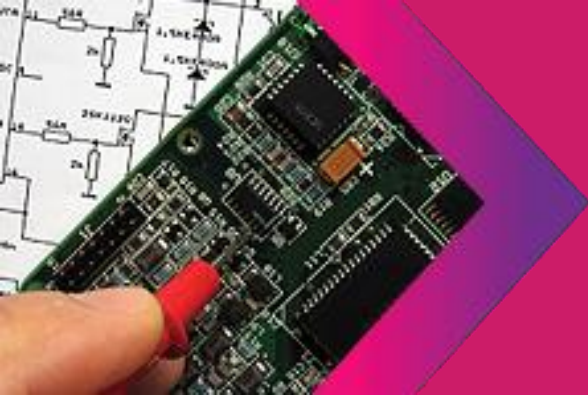
Linux (Fedora / Mandriva / Ubuntu)

- **Delivery method:**

Instructor lead, Specific assignments and Mini Project

- **Module topics:**

- *Basics of C - Operators, Conditionals, Arrays, Pointers*
- *Structures, Unions, Functions, Files, Preprocessor 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*



NOW FOR SIMPLIFIED SOLUTIONS

## Module: 3/7

### Module name: Data structures and Logic analysis

- **Objectives:**

- ✓ Review methods for problem solving and algorithm analysis
- ✓ Develop an understanding of Abstract Data Types (ADT) and their implementation
- ✓ Understand the importance of information hiding, data abstraction, and modular design
- ✓ Recognize programming needs - cost and benefits of each data structure
- ✓ Enable you to select the data structure for the job

- **Overview:**

A firm understanding of data structures provides a basis for writing more efficient code. The four hallmarks of good programming are proper design, clear coding, good organization of data, and correct algorithm selection. This course is intended to provide an understanding of data specification and abstraction, how that leads to the specification of data structures, and how data structures are implemented as late as possible, based on the specified structures.

- **Pre-requisites:**

Proficiency in C

- **Duration:**

1+ month

- **Platform:**

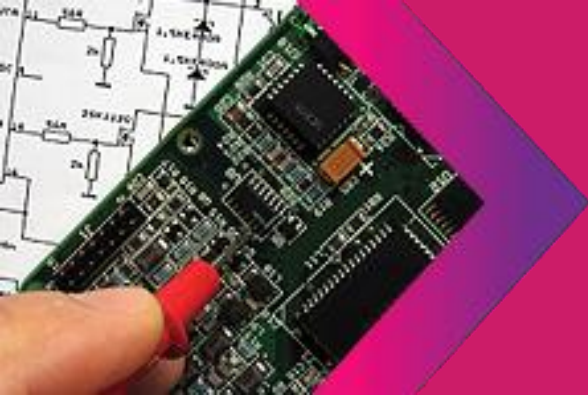
Linux (Fedora / Mandriva / Ubuntu)

- **Delivery method:**

Instructor lead, Assignments, Workshops and Project

- **Module topics:**

- *Introduction to Data Structures & Algorithm Analysis*
- *Abstract Data Types (ADT)*
- *Stacks, Queues and Linked Lists*
- *Trees, Recursion*
- *Sorting Algorithms*
- *Hashing*



NOW FOR SIMPLIFIED SOLUTIONS

## Module: 4/7

### Module name: Linux Internals & Networking

- **Objectives:**

- ✓ To gain strong knowledge of OS programming
- ✓ Proficiency on the Linux API's and system calls
- ✓ To get the knowledge of high performance and secure coding by using OS capabilities
- ✓ Enable you to write Network protocols using Sockets

- **Overview:**

Currently most of the embedded systems are built over an operating system due to performance requirements and the resource complexity. This adds responsibility to developers to understand the make use of the operating system capabilities to build a secure, high performance and crash free system. This course gives complete understanding of the Operating system concepts and Linux internals (Interfaces, API's and system calls). This module helps the audience to move to the next level of programming by considering other factors in the system. This module is industrial aligned and provides ample practical classes to provide good exposure to Linux programming.

- **Pre-requisites:**

Proficiency in C programming, Fundamentals of Operating system

- **Duration:**

1+ month

- **Platform:**

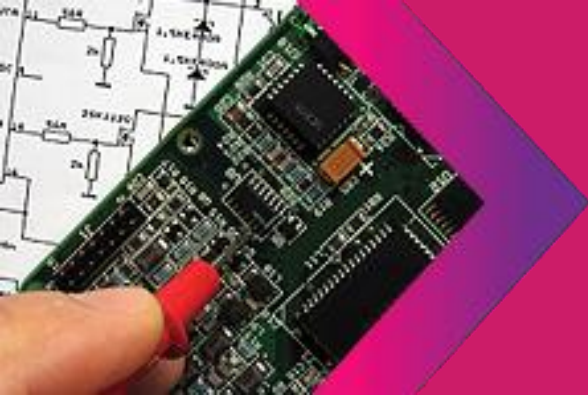
Linux (Fedora / Mandriva / Ubuntu)

- **Delivery method:**

Instructor lead, Assignments, Workshops and project

- **Module topics:**

- *OS Basics - Process, CPU scheduling, Dead lock & starvation, priority*
- *Components of Linux - Kernel structure, Shell basics, Linux file system-ext2 & ext3*
- *POSIX Threads - Multi threaded programming, p-thread API's*
- *Synchronization - Race condition & mutex, Semaphores*
- *IPC - Pipes, FIFO, shared memory, System V-semaphores*
- *Sockets - TCP and UDP sockets, Client-server programming*



NOW FOR SIMPLIFIED SOLUTIONS

## Module: 5/7

### Module name: Micro controllers

- **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

- **Overview:**

A complete module deals with writing an Embedded C program for Hardware with any micro-controller on it. This module gives you first-hand experience of running your program on target hardware, thereby giving you the “Embedded” feel. With Emertxe custom designed self-learning kit boards (based on 8051 / AVR / PIC) you will learn how to write optimized Embedded programs.

- **Duration:**

1+ month

- **Platform:**

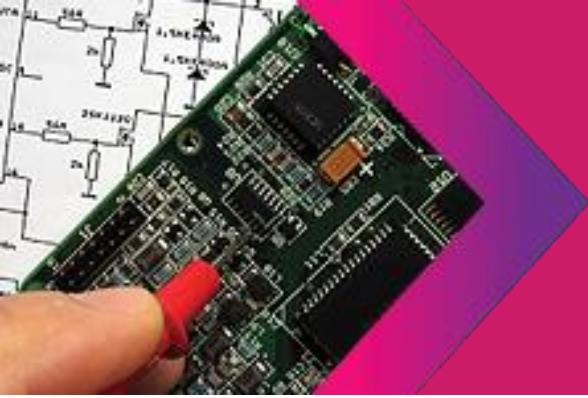
Emertxe Self Learning Kits (based on 8051 / PIC / AVR micro-controllers)

- **Delivery method:**

Instructor led, Assignments and Mini Project

- **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 - UART, SPI, I2C etc*



NOW FOR SIMPLIFIED SOLUTIONS

## Module: 6/7

### Module name: Embedded Linux on ARM 9

- **Objectives:**

- ✓ Get you exposed with various trends in Embedded OS
- ✓ Making appropriate Open source choices for your Embedded device
- ✓ Get hands-on with Flash memory usage, EEPROMS using development boards
- ✓ Equip you with high end application Embedded development with ARM 9

- **Overview:**

An unique module combining various previous modules you have learnt by combing Linux administration, Hardware knowledge, Linux as OS, C/Computer programming areas. This is a complete module on Embedded OS, as of now no books are written on this with such practical aspects. Every class is backed by discussion and topic related assignments, demo by instructor and practice by you. Finally a project letting you apply most of the concepts learned throughout the module

- **Duration:**

1 month class and lab

- **Platform:**

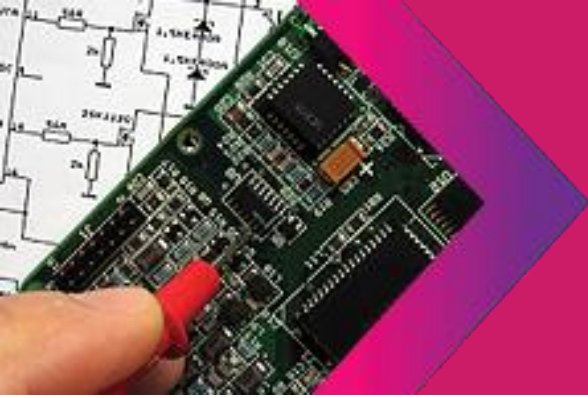
Emertxe ARM development board ExDev9A and ExDev9A SDK  
Host platform may be either Linux / Windows.

- **Delivery method:**

Instructor lead, Assignments and mini project

- **Module topics:**

- *Open Source ecosystem*
- *Making choices for Embedded Linux distribution*
- *Evolution of high end Embedded Systems*
- *Host and Target concepts*
- *U-boot, cross compiling, porting Linux kernel*
- *Emertxe file system on the board, debugging methods, Real-time OS.*



NOW FOR SIMPLIFIED SOLUTIONS

## Module: 7/7

### Module name: Linux device drivers

- **Objectives:**

- ✓ Introduce you to Linux device drivers framework
- ✓ Interpreting data sheets and writing any character device driver
- ✓ Deep dive into Linux architecture, Hardware and memory management, modularization, and the layout of the kernel source
- ✓ To enable you for becoming proficient Linux driver developer for character based devices
- ✓ Equip you with enough tools to recognize and optimize inefficient drivers

- **Overview:**

This course provides a practical overview about the structure of device drivers, in-depth information about the interface between the device driver and the rest of the Linux kernel, and various practical exercises to develop and test device drivers for major devices in a Linux environment. It also covers the initiators and detailed insights of writing a Device Driver in Linux, focused and aligned towards the industry perspective.

- **Duration:**

1 month class and lab

- **Platform:**

Emertxe Linux Device Drivers learning kit SDK

- **Delivery method:**

Instructor lead, Assignments and mini project

- **Module topics:**

- *General function of the device driver*
- *Character drivers, block drivers and file system.*
- *Sleep and wakeup (wait queues), Buffer allocation*
- *Timer handling and Interrupt handling.*
- *Error logging, debugging, the /proc and /dev file systems.*
- *Data transport between user mode and kernel mode, error codes, the ioctl interface.*
- *USB drivers: Mini Project on live hardware*





Emertxe Information Technologies Private Ltd  
83, Farah Towers, 1st Floor,  
MG Road,  
Bangalore 560001

T: +91 809 555 7 333 (M), +91 80 4128 9576 (L)  
E: [training@emertxe.com](mailto:training@emertxe.com)

[www.emertxe.com](http://www.emertxe.com)