Course Name	**Certificate Course in Embedded Linux - Intermediate**

Target Audience	Engineering students & graduates in EC, EE, CS, IT / MSc / MCA / BCA

Duration	60 Hours ● 50 % for Lecturers ● 50 % for hands-on

Course Syllabus

Embedded System - Practical Approach

- Understanding Embedded System Usecases
- Why Embedded System is different & How
- CPU & Peripheral Interfaces
- Understanding hardware interfacing
- Microprocessor & Microcontroller Embedded System
- Embedded system with ARM platform

Basic of C programming required for Embedded programming

- Data Types, Variables, Constants, Storage Classes, Operators
- Conditions, For, While Loops, Functions
- Strings, Structures and Unions
- Arrays & Pointers
- File Input / Output
- Development of C programs on Linux

Operating System Fundamentals

- Introduction to Embedded Operating Systems
- Process Management and Inter Process Communication, Memory Management, I/O sub - system
- POSIX Thread Programming (Multithreading), POSIX Semaphores, Mutexes
- Interrupts handlers, Timers
- Filesystem
- Kernel Mode Vs User Mode and its security aspects

Linux as development OS

- Installing Ubuntu Linux on desktop
- Basics of Linux command line and commands
- Basic of using Editors
- Managing software packages
- Understanding Host & Target platforms

Embedded Linux - Practical Approach

- Embedded Linux System Architecture (Stack)
- Introduction to Boot loaders and Board Support Packages
- Understanding Linux Booting from PowerON to Application
- Middleware Libraries
- Application communications with system libraries and hardware
- Understanding Busybox
- Embedded File Systems

Introduction to Linux Kernel & device drivers

- Embedded Linux Kernel Internals - Understanding Source Architecture
- Embedded Linux Device Drivers
- Linux Kernel Modules
- Char device driver

Development Software Tools

- Makefiles
- Shell Scripts & Automation
- Binutils, Compilers, Debuggers
- Embedded Toolchain

Building and Booting Embedded Linux on ARM Platform

- Cross compiling Linux Kernel
- Booting Linux kernel
- Setting up file system
- Remote Login to embedded platform
- Logging mechanisms
- Cross compiling Application Software

Embedded Linux Testing and Debugging

- Linux Kernel Debugging Techniques
- Device Driver Debugging Techniques
- Application software debugging techniques

Best Practices of Embedded Linux development as used in Industry

- Coding standard
- Git - Source code management for distributed team

Hardware Used for Training

Platform : Raspberry Pi3
SoC: Broadcom BCM2837
CPU: 4× ARM Cortex-A53, 1.2GHz
GPU: Broadcom VideoCore IV
RAM: 1GB LPDDR2 (900 MHz)
Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage: microSD
GPIO: 40-pin header, populated
Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)