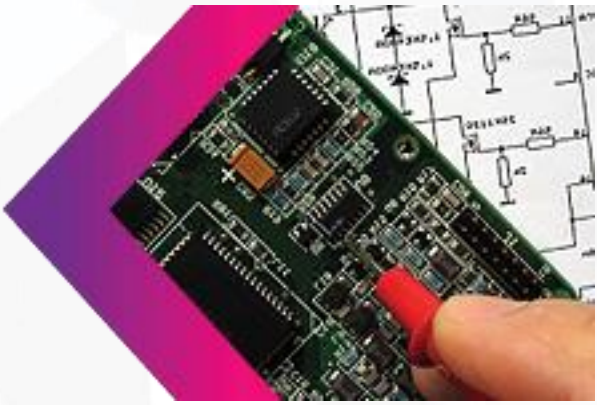
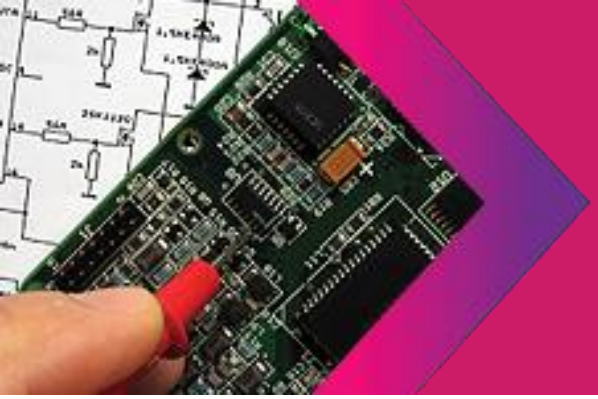
EMERTXE

"Here to take you beyond"

EMBEDDED LINUX ON ARM9

Weekend Workshop





NOW FOR SIMPLIFIED SOLUTIONS

Embedded Linux on ARM9 - Weekend workshop

- **Objectives:**
  - ✓ Get you exposed with various trends in Embedded OS
  - ✓ Leverage Opensource tools to build a complete Embedded system/product, including bootloader, Linux kernel, startup-scripts and applications.
  - ✓ Deploy and Debug Embedded Linux Applications on the target
  - ✓ Setup and automate a build environment for customizing Embedded Linux kernel, bootloader and applications.
  - ✓ Be familiar with kernel configuration, compilation and tweaks.
  - ✓ 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

- **Overview:**

  A unique workshop combines various previous modules you have learnt by combing Linux administration, Hardware knowledge, Linux as OS, C/Computer programming areas.Every session is backed by discussion and topic related assignments, demo by instructor and practice by you.

  Ideally designed for working professionals to gain understanding about "arm-port" by attending weekend sessions. At the end of the workshop you should be build your own embedded linux os from scratch.

- **Duration:**

  4 days (Two weekends)

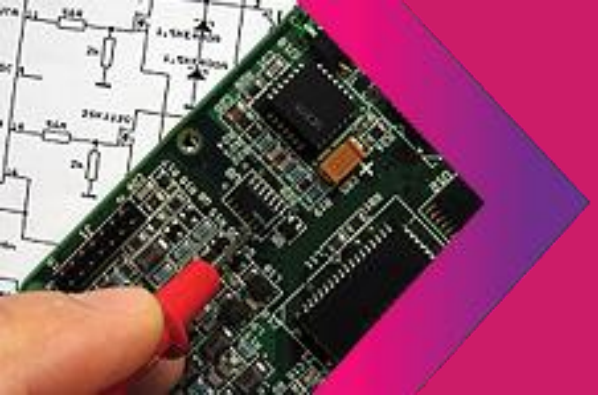
- **Platform:**

  Target board: BeagleBone Black (RevC / ARM Cortex A8)
  Host OS: Linux (Ubuntu preferred)
  Target OS: Custom Embedded Linux

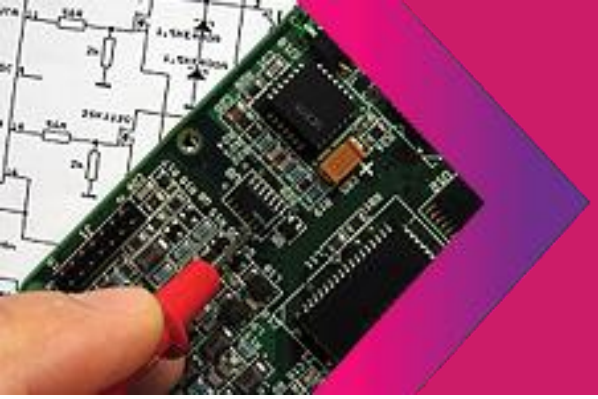
- **Delivery method:**

  Workshop based approach delivered in a fast-track

- **Pre-requisites:**
  - ✓ Basic C Programming Skills
  - ✓ Basic Linux Usage, and OS fundamentals
  - ✓ Good to have (Not mandatory): Knowledge about Makefiles, gdb



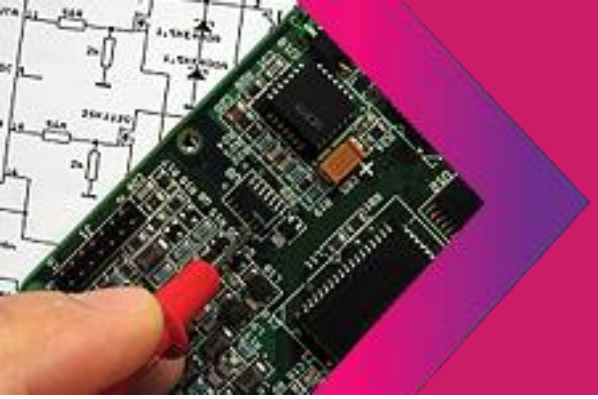
- ## Detailed course contents:
  - ✓ **Introduction**
    - W's of Embedded Systems & The Real Time Aspect
    - Open Source & Free Software Fundamentals
  - ✓ **Linux as an Embedded OS**
    - Architecture
    - Software components
    - Choices to make
    - Applying Patches
    - Source Code Browsing Techniques
  - ✓ **Target Hardware Architecture: Overview**
    - Processes Architecture -ARM
    - Hardware Peripherals available
    - Schematics
    - Memory Mappings
    - Connectivity for board bring-up
  - ✓ **Linux booting sequence**
    - Desktop Linux booting sequence
    - Embedded Linux booting sequence
    - Processor/Target board specific booting sequence
  - ✓ **Choosing the Software Components for Embedded Linux**
    - Requirements & Setup
    - The Embedded Environment Tools
    - Tool-chain (Cross Compilers & friends)
    - Debug-ability
  - ✓ **Tool-chain: Configuration and Cross-compilation**
    - What is a tool-chain
    - Native vs. cross-compilation
    - Toolchain Components
    - Toolchain choices
    - Using buildroot to build the toolchain
    - Configuration options
    - Adding path variables to startup scripts (.bashrc)
    - The CROSS_COMPILE variable
    - Validating the cross-compiler
    - Alternative: Getting a pre-compiled toolchain



- ✓ **Booting sequence and U-boot in detail**
  - Boot-loader Phases
  - U-boot – Embedded boot loader
  - What does u-boot do?
  - Navigating the u-boot sources
  - Configuring and Cross-compiling u-boot
  - Installing u-boot on the target
  - Understanding u-boot commands
  - Changing environment variables to setup kernel booting
  - Transferring files to the target using tftp

- ✓ **Embedded Linux Kernel**
  - Kernel Features
    Kernel Subsystems
    - Memory Manager
    - Scheduler
    - Embedded Storage
    - I/O Subsystem
    - Network Subsystem
  - Navigating the kernel sources
  - Kernel Configuration
  - Kernel Compilation
  - Booting the kernel using u-boot
  - Module compilation and Installation to RootFS
  - Procedure for adding a new driver to the kernel
  - Applying patches

- ✓ **Creating a custom Root File System**
  - Introduction to File system
  - Linux directory structure
    - Organization and Important directories
    - /dev file system
  - What next after kernel booting
    - init and startup scripts
  - Creating the RootFS
    - Busybox
    - Adding additional packages
  - RootFS Storage choices
    - Option 1: initramfs: RootFS in memory
    - Re-building the kernel with initramfs
    - Option 2: RootFS in Flash/SD Card storage
      - Creating Partitions
      - Filesystem choices



- Formatting Partitions
- Copying RootFS to partition
- Updating kernel boot params from u-boot

✓ **Linux Device Drivers Overview**
- Character Drivers
- Hardware Access
- Demo: a GPIO based LED driver
- Demo: Driver for user switch using interrupts

✓ **Embedded Application Programming**
- Compiling and Running a multi-threaded embedded application
- High level code navigation of application

✓ **Remote Debugging Embedded Applications using GDB**
- How remote debugging works
- Setting up tools for remote debugging using buildroot
- Hands on session to debug remote application

By the end of the workshop:

✓ Know “What” is Open Source and “How it benefits” the Industry
✓ Be able to “Make the choices for Embedded Linux”
✓ Be equipped to “Setup” an ARM-based Platform
✓ Understand the “Linux Boot Process & Role of a Bootloader”
✓ Be able to “Build your own tool-chain for ARM” platforms
✓ Be able to “Make your own root file-systems for ARM-based Embedded Linux”
✓ Be able to “Install standard applications for Embedded Linux”
✓ Be able to “Configure & Compile your own Linux Kernel for Linux on ARM”
✓ Be able to “Write a basic char driver”
✓ Write your own applications for Embedded Linux on ARM

Workshop Highlights:

- *Work 1:1 with ARM9 - Boards (1 per participant)*
- *Kernel Source Version: 4.1 or higher versions*
- *Live Examples: From ARM-based industry product(s)*
- *Hands-On: Linux Porting, Cross Development Tools, uboot, Kernel Internals, Basic Drivers, Embedded Application Programming*

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

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

www.emertxe.com