

“Weekend Workshop on Embedded Linux Porting Advanced” by Pradeep Tewani

Day 1

+ **Session 1: Bootloaders**

- Understanding the Code flow for X-Loader (First Stage Bootloader)
- Understanding the Code flow for U-Boot (Second Stage Bootloader)
- Booting with the NFS (Networked File system)
- Understanding the BSP in X-Loader and U-Boot
- Porting the X-Loader/U-Boot for Beagle board

+ **Session 2: File System for Embedded Devices**

- W's of a File System
- Choosing a File System
- Configuring and Compiling *busybox* for Beagle board
- Creating & Setting up a (Root) File System for Beagle board
- Creating a *ramdisk* for Beagle board

Day 2

+ **Session 3: Platform Drivers**

- Understanding the Platform Drivers
- Writing a Simple Platform Driver
- Writing a Platform driver for GPIO

+ **Session 4: Kernel Porting & Device Tree Blob (DTB)**

- Understanding the code flow for Kernel
- Understanding the Porting and BSP
- Getting into the Beagle board BSP/LSP in Kernel
- Understanding the Device Tree Concept in Kernel
- Understanding the Device Tree Blob (DTB) Framework in Kernel
- Linking the device driver with Device Tree Blob (DTB)

+ **Wrap Up**

- Conclusion
- What Next?

Caution: All sessions are highly interactive & hands-on with Beagle Bone Black.

Hands-On Details

+ **U-Boot/X-Loader**

- Modifying the X-loader code for controlling the LEDs with switch
- Playing around with Environmental Variables
- Configuring the U-Boot to Save the Environment Variables in MMC
- Configuring the U-Boot to Save the Environment Variables in EEPROM
- Tweaking the U-Boot code to support Multi Boot System

+ **Filesystem for Embedded Systems**

- Configuring & Compiling *busybox*
- Creating a Root Filesystem
- Creating *ramdisk*
- Booting with a *ramdisk* Image
- Booting with NFS

+ **Platform Drivers**

- Writing a Simple Platform Driver
- Adding a Platform Driver for GPIO

+ **Kernel Porting & Device Tree Blob (DTB)**

- Adding the DTB support for GPIO Driver