"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: Kernel Porting & Drivers*

- Understanding the code flow for Kernel
- Understanding the basics of Porting and BSP
- Getting into the Beagle board BSP/LSP in Kernel
- Platform Drivers on Beagle board

+ *Session 4: Embedded Applications*

- What's special about Embedded Applications?
- Various OSS Applications: busybox, ...
- Porting the Applications on Beagle board

+ *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
- 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

+ *Kernel Porting & Drivers*

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

+ *Embedded Applications*

- Blinking an LED on Beagle board
- Interfacing a Web Cam to a Beagle board