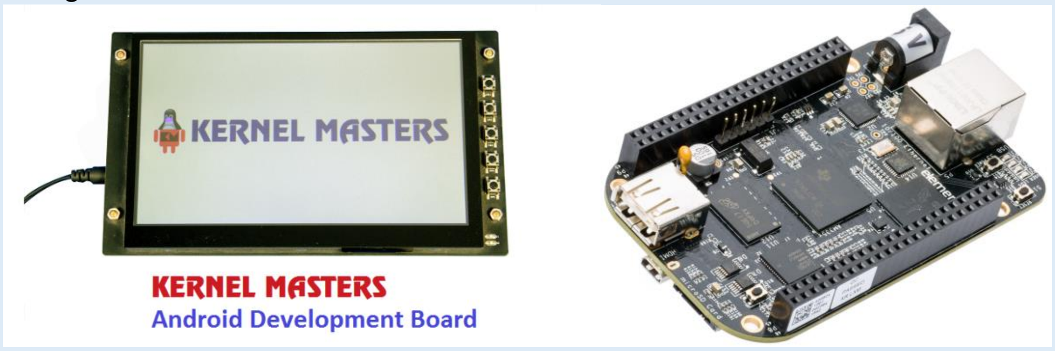
INDIA's one & only
INDUSTRIAL Embedded Systems Training Institute
KERNEL MASTERS

Embedded Android

with Beagle Bone Black

Hardware used

- The practical exercises will be run on a Beagle Bone Black (BBB) with a Cortex ARM.
- All exercises will be applicable to any other type of board supported by Linux.
- Online practical demonstration for android porting on BBB. Later on you can buy and practise, support will be provided. Material will be provided with step by step procedure for lab guidance.



Prerequisites:

- We assume that attendees are fully fluent in C, data structures and that the Linux/Unix command line is a familiar environment.
- Embedded systems development knowledge.
- Linux System Programming & Device Drivers.

Audience:

- This session is mainly intended for those looking to start their career in embedded Android or for those already working in Linux porting.

End of the course you'll be able to (Session Highlights):

- Build a customized Android kernel and u-boot for a designated target.
- Understand Android framework and build process.
- Porting a new device in to HAL.
- Configure a bootloader for booting a customized kernel with its root filesystem from a solid-state storage device.

Note: All Sessions are highly interactive hands-on-sessions.

Embedded Android Syllabus Summary: (Detailed agenda below)

Session 1: Introduction to Embedded Android
Session 2: Setup Android build system & Development Environment
Session 3: Android Kernel and Linux Kernel differences
Session 4: Android Porting to New Hardware
Session 5: Porting native C, C++ library HAL for different SoC
Session 6: Android Linux Kernel Driver development Process
Session 7: Android Debugging, Optimizations for Performance

Session 1: Introduction to Embedded Android (Prebuilt Images)

Objective: To understand Embedded Android platform and cross compilation toolchain. Understand Beagle Bone Black (BBB) specifications, Block Diagram and Memory Mapping and also AM3358 Boot Sequence. Understand KM-BBB Specifications and Bock Diagram. How to install Android Prebuilt images on KM-BBB.

1. **Introduction to Embedded Linux**
   1.1. Embedded Hardware and Software.
   1.2. C libraries. Building a cross-compiling tool chain.
   1.3. BBB Specifications and Block Diagram
   1.4. AM3358 Specifications, Block Diagram and Memory Mapping
2. **Introduction to Embedded Android**
   2.1. Android Framework
   2.2. Understanding Android Services
   2.3. The Role of Binder and AIDL
   2.4. Android Boot Sequence. (AM3358 Boot Sequence)

Hands-On-Session:

- Android kit kat Prebuilt Images porting on KM-BBB.

Session 2: Setup Android Build system & Development Environment (Own Built Images)

Objective: To understand how to setup android build system.

2.1 Android Build Environment Setup.
2.2 Understanding and developing .mk scripts.
2.3 Creating packages aad modules using Android Blueprint (Android.bp) and Android Makefiles (Android.mk)
2.4 AOSP Structure details
2.5 Building a product from packages and modules
2.6 Android U-boot & Kernel Development Setup

Hands-On-Session:
Walkthrough Android source tree and understand its importance.
Built Own boot loader and Kernel and Root File System.

Session 3: Android Kernel and Linux Kernel differences

Objective: To Learn, difference between and Android kernel and Linux Kernel.

3.1 Binder IPC
3.2 Boot Procedure
3.3 Power Management
3.4 Kernel Logs
3.5 Init

Session 4: Android Porting to New hardware

Objective: To Learn, GPIO Framework from Bottom to Top. Understand how to develop GPIO API's using system calls and application programming using Shell Program, 'C' and Python GPIO modules.

Example device 1: Power Button

4.1 Enable Power button Mux Configuration in U-Boot source.
4.2 Enable Power Key button in Device tree source code.
4.3 Mapping Kernel Power key button with android key mapping in HAL.

Example device 2: Accelerometer Sensor

4.4 Enable I2C Mux Configuration in U-Boot source.
4.5 Add Accelerometer sensor I2C slave device information in to device tree source code.
4.6 Enable Accelerometer sensor driver in Linux source code.
4.7 Mapping Kernel accelerometer sensor information in to android event file in HAL.

Hands-On-Session:
Test Power button and accelerometer sensor devices in HAL.

Session 5: Porting native C, C++ library HAL for different SoC.

5.1 Procedure to port OSS libraries to Android.
5.2 Demo application to use ported library.
5.3 HAL & HIDL [HAL Interface Definition Language]
5.4 Run time linking and the Vendor Native Development Kit [VNDK]
5.5 Port HAL/HIDL on SoC.
5.6 Integrated any external HAL code with a processor SDK.
5.7 Modify android device tree structure and corresponding HAL for porting a new device to an SoC.

Session 6: Android Linux Kernel Driver development Process

6.1 Linux Device model
6.2 Mapping physical to Kernel Space memory
6.3 Detecting Device
6.4 Interrupt Handler, Locking Mechanisms
6.5 Data sharing with applications and other sub-systems
6.6 How does operating system know which HAL to load for a corresponding application?

Session 7: Android Debugging, Optimizations for Performance

7.1 adb [android debug bridge]
7.2 ddms
7.3 android emulator
7.4 logcat to view and filter messages
7.5 Debugging native code with gdb

Note: For more No. of Drivers (I2C, UART, SPI and Network), attend separate sessions offered by Kernel Masters.

Authored and Compiled By: Boddu Kishore Kumar
Email: kishore@kernelmasters.org
Reach us online: www.kernelmasters.org
Contact: 9949062828