FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

i.MX 6 Series Basics

FTF-CSD-F0066

Enrique Ochoa
MAD SW Release & Trainings Manager

June 2012

Session Introduction

- This session will provide a hands-on walk through of the i.MX 6Quad SABRE Board for Smart Devices (SDB). Providing all the information needed to evaluate our HW solution and the Board Support Packages (BSPs) we offer.

- During this session we will cover the basics of our HW platform, what we deliver with our BSPs and how to use both to deploy a benchmark or a demo application to evaluate our solution.

Session Objectives

- After completing this session you will be able to:
  - Deploy Linux and Android Images in the i.MX 6 SABRE Board for Smart Devices (SDB)
  - Deploy benchmarks and demo applications to the i.MX 6 SABRE SDB to evaluate our System On Chip (SOC) and our BSPs

Tools Used in this Training Session

- iMX 6Quad SABRE Board for Smart Devices
  - Power supply
  - USB cables
- LVDS Display
- Training PC
- SD/MMC Card Reader (USB)
- SD card
- Software:
  - VMWare Player
  - Ubuntu 10.04 Virtual Machine
  - Freescale Linux BSP
  - Freescale Android BSP
  - Demo benchmarks applications

Agenda - i.MX 6 Series Basics

- The i.MX 6 Family, i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

Agenda - i.MX 6 Series Basics

- The i.MX 6 Family, i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

i.MX 6: One Platform, Differentiated Products

Saves **development costs** and **improves time to market**.
Scalability with multiple cores is **key** to implement this strategy.

Smart Device Design

Quad Core

High-End (6Quad)

IPTV
High Performance Tablet
Auto Infotainment

Dual Core

High-End (6Dual)

Low-End (6DualLite)

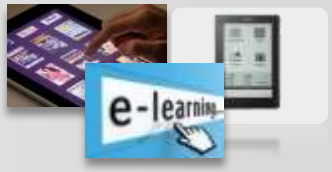
Smart Monitor
Business Tablet
Media Tablet
IP Phone
Tablets for Kids
Mainstream Infotainment
Color eReaders

Single Core

High-End (6Solo)

Low-End (6SoloLite)

Smart Energy
eReaders
Entry Auto Infotainment



freescale™

The i.MX 6 Family

Red indicates change from column to the left

i.MX 6SoloLite

- Single ARM® Cortex™-A9 at 1.0GHz
- 256KB L2 cache, Neon, VFPvd16, Trustzone
- 2D graphics
- 32-bit DDR3 and LPDDR2 at 400MHz
- Integrated EPD controller

i.MX 6Solo

- Single ARM Cortex-A9 at 1.0GHz
- **512KB** L2 cache, Neon, VFPvd16, Trustzone
- **3D graphics** with 1 shader
- 2D graphics
- 32-bit DDR3 and LPDDR2 at 400MHz
- Integrated EPD controller

i.MX 6DualLite

- **Dual** ARM Cortex-A9 at 1.0GHz
- 512KB L2 cache, Neon, VFPvd16, Trustzone
- 3D graphics with 1 shader
- 2D graphics
- **64-bit** DDR3 and 2-channel 32-bit LPDDR2 at 400MHz
- Integrated EPD controller

i.MX 6Dual

- **Dual** ARM Cortex-A9 at 1/**1.2GHz**
- **1 MB** L2 cache, Neon, VFPvd16, Trustzone
- 3D graphics with **4 shaders**
- **Two** 2D graphics engines
- 64-bit DDR3 and 2-channel 32-bit LPDDR2 at **533MHz**
- Integrated **SATA-II**

i.MX 6Quad

- **Quad** ARM Cortex-A9 at 1.2GHz
- 1 MB L2 cache, Neon, VFPvd16, Trustzone
- 3D graphics with 4 shaders
- Two 2D graphics engines
- 64-bit DDR3 and 2-channel 32-bit LPDDR2 at 533MHz
- Integrated SATA-II

i.MX 6 Series Highlights

- ARM Cortex-A9 based solutions ranging up to 1.2GHz
- HD 1080p encode and decode (except 6SL)
- 3D video playback in High definition (except 6SL)
- Low power 1080p playback at 350mW Integrated IO's that include HDMI v1.4, MIPI and LVDS display ports, MIPI camera, Gigabit Ethernet, multiple USB 2.0 and PCI-Express
- SW support: Google Android™, Windows® Embedded CE, Ubuntu, Linux®, Skype™

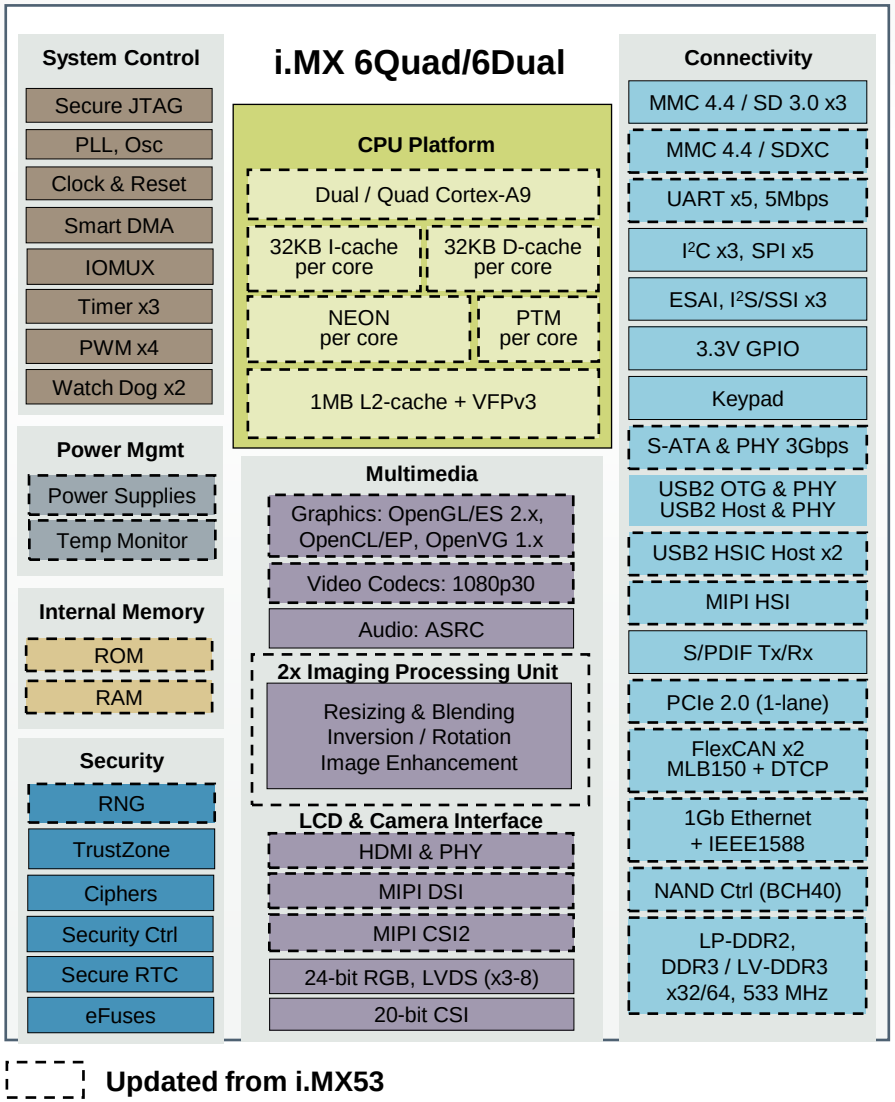
Features vary by product family

freescale™

i.MX 6Quad/6Dual Applications Processor

- **Specifications**
  - **CPU:** i.MX 6Quad 4x Cortex-A9 @1.2 GHz, 12000 DMIPS
  - i.MX 6Dual 2x Cortex-A9 @1.2 GHz, 6000 DMIPS
  - **Process:** 40nm
  - **Core Voltage:** 1.1V
  - **Package:** 21x21 0.8mm Flip-chip BGA
  - 12x12 PoP (LP-DDR2, NAND)

- **Key Features and Advantages**
  - Multi-core architecture for high performance, 1MB L2 cache
  - 64-bit LP-DDR2, DDR3 and raw / managed NAND
  - S-ATA 3Gbps interface (SSD / HDD)
  - Delivers rich graphics and UI in HW
    - OpenGL/ES 2.x 3D accelerator with OpenCL EP support and OpenVG 1.1 acceleration
  - Drives high resolution video in HW
    - Multi-format HD1080 video decode and encode
    - 1080p60 decode, 720p60 encode
    - High quality video processing (resizing, de-interlacing, etc.)
  - Flexible display support
    - Four simultaneous: 2x Parallel, 2x LVDS, MIPI-DSI, or HDMI
    - Dual display up to WUXGA (1920x1200) and HD1080
  - MIPI-CSI2 and HSI
  - Increased analog integration simplifies system design and reduces BOM
    - DC-DC converters and linear regulators supply cores and all internal logic
    - Temperature monitor for smart performance control
  - Expansion port support via PCIe 2.0
  - Car network: 2xCAN, MLB150 with DTCP, 1Gb Ethernet with IEEE1588



SABRE Board for Smart Devices (SDB)

i.MX 6Quad 1Ghz Cortex-A9 Processor
- Can be configured as i.MX 6Dual
- Freescale MMPF0100 PMIC
- 1 GB DDR3 memory (non terminated)
- 3” x 7” 8-layer PCB

Display connectors
- 2x LVDS connectors
- Connector for 24 bit 4.3” 800x480 WVGA with 4-wire touch screen
- HDMI Connector

Audio
- Wolfson Audio Codec
- Microphone and headphone jacks

Expansion Connector
- Camera CSI port signals
- I2C, SSI, SPI signals

Part Numbers: MCIMX6Q-SDB ($399)

Display (9.7”): MCIMX-LVDS1 ($499)
Display (4.3”): MCIMX28LCD ($199)

Tools Support
- Lauterbach, ARM (DS-5), Macraigor debug/IDE tool chain

Connectivity
- Full-size SD/MMC card slot
- 7-pin SATA data connector
- 10/100/1000 Ethernet port
- 1x high-speed USB host port
- PCI-e connector

Debug
- JTAG connector
- Serial to USB connector

Additional Features
- 3-axis Freescale accel
- Power supply- USB plug
- No battery charger

OS Support
- Linux and Android IceCream Sandwich from Freescale;
- Others: support by 3rd parties

Agenda - i.MX 6 Series Basics

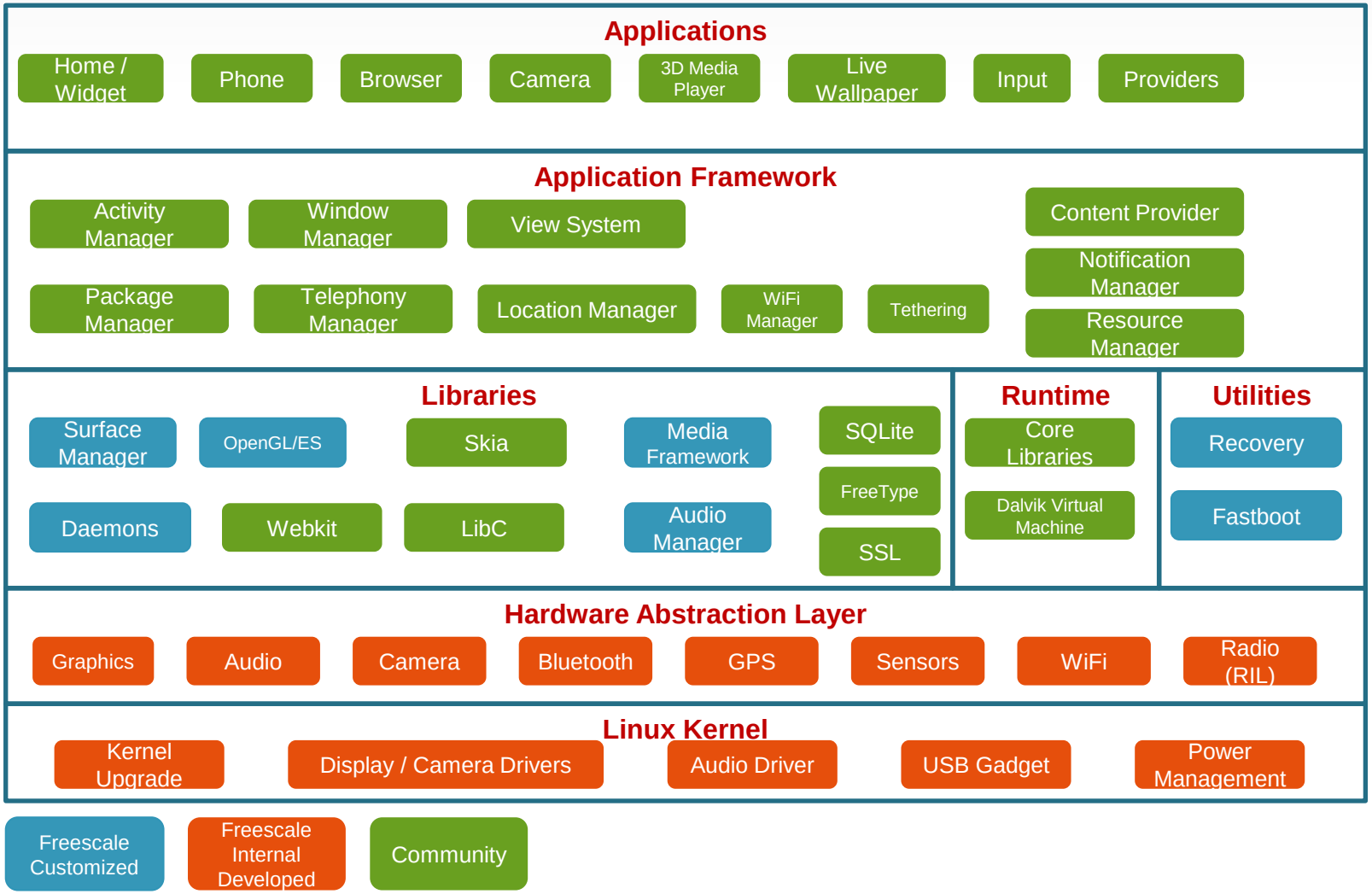
- The i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

i.MX 6Quad Android BSP

What we deliver with the BSP

- Freescale patches for Google source code
- Linux kernel source code
- U-boot source code
- Multimedia Codecs binary files
- Documentation
- Demo images
  - U-boot image
  - Boot image
  - Android system images (System root, recovery)
- Manufacturing Tool (MFG Tool)

Origin of Android Components



Agenda - i.MX 6 Series Basics

- The i.MX 6 Family, i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

Android demo image deployment

1. Decompress the android demo image package, ~Desktop\FTF-CSD-F0066\MX6_BSP\image_imx-android-r13.3_6qsabresd.tar.gz
2. Decompress the MFG tool package (~Desktop\FTF-CSD-F0066\MX6_BSP\Mfgtools-Rel-12.04.01_ER_MX6Q_UPDATER.tar.gz)
3. Copy the SABRE SDB images files (u-boot.bin, boot.img, recovery.img system.img and uramdisk.img) to the i.MX6Q Linux MFG tool profile folder (~\Mfgtools-Rel-12.04.01_ER_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\android).

Android demo image deployment

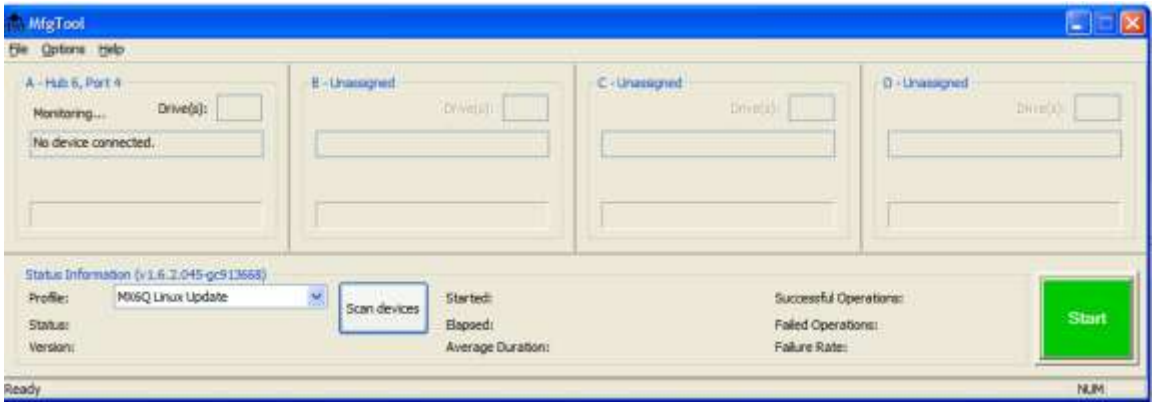
4. Connect the UBS OTG port (J505, bottom) from the SABRE SD to the computer.
5. Connect the USB to SERIAL port (J509, bottom) from the SABRE SD to the computer.
6. Open HyperTerminal on the Windows PC (start->All Programs->Accessories->Communications->Hyper Terminal)
7. Name your new connection IMX6 and click OK
8. Click OK on the Connect to windows

Android demo image deployment

9. Configure the Port Settings as shown and click OK

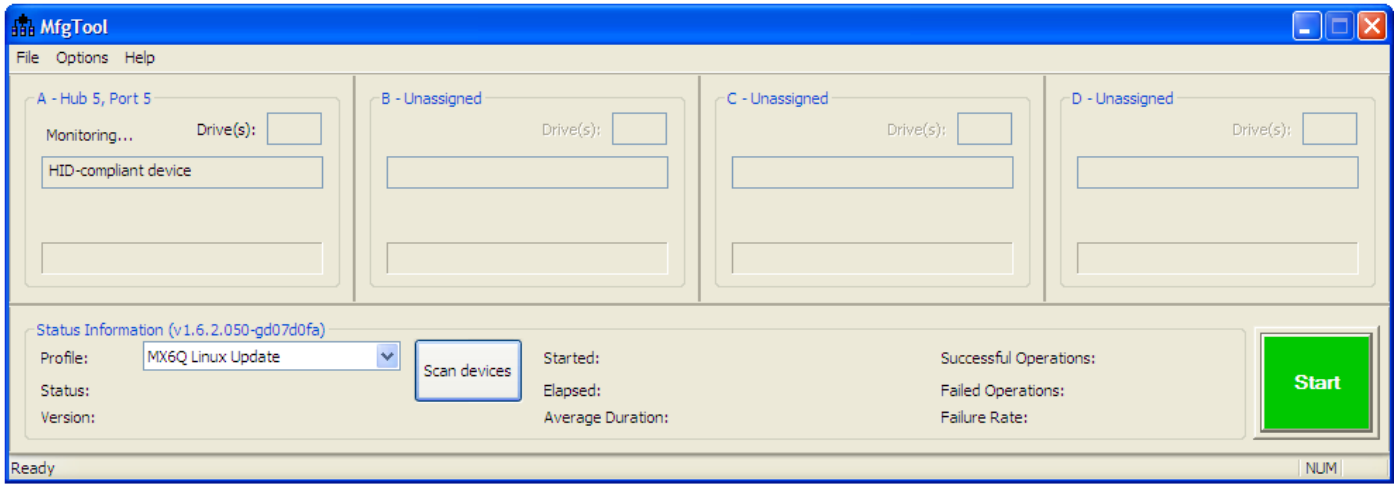
Android demo image deployment

10. Set the board to the serial download mode, Change Boot Switch(SW6) to 00001100 (from 1-8 bit).
11. Plug in the Power supply to the board.
12. Start the MFGTool by clicking "~\Mfgtools-Rel-12.04.01_ER_MX6Q_UPDATER\MFGTool.exe" The user interface of this tool should as below:



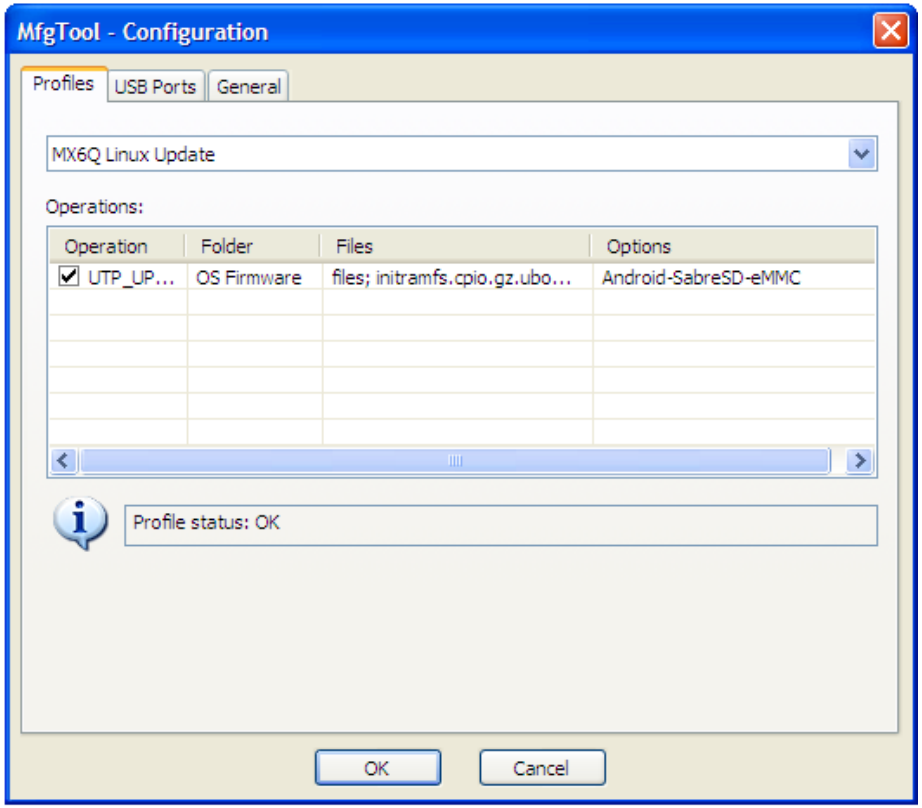
Android demo image deployment

13. Click Scan devices, which will update the dialog to HID-compliant device as show on the next image.



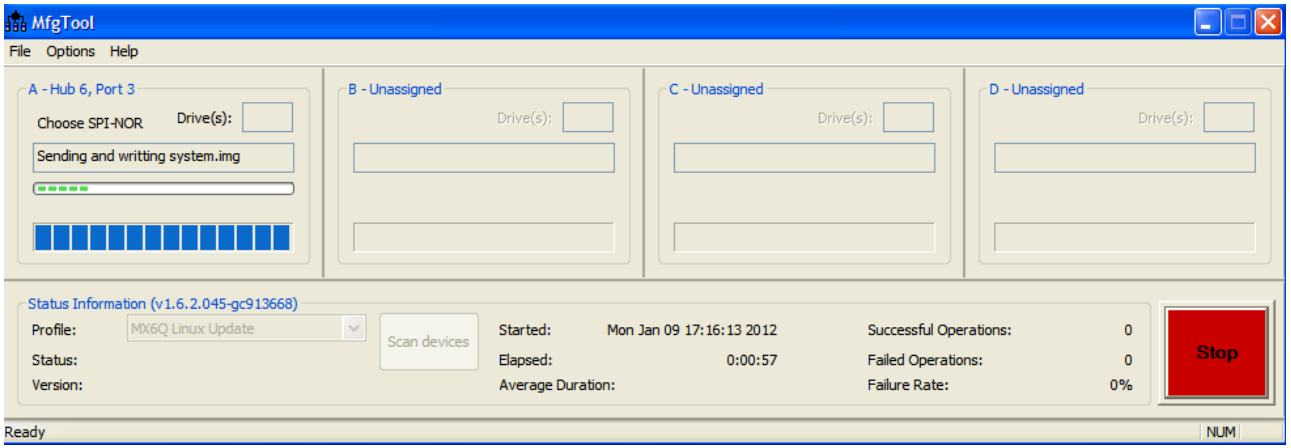
Android demo image deployment

14. Configure the profile, go to Option -> Configuration.. -> Profiles and choose the Android-SabreSD-eMMC profile on the Options section and click OK



Android demo image deployment

15. Click start to start image downloading.
16. During the downloading process, you will see the tool become below image, and the status bar will show downloading status. You can also see the progress on the hyper terminal window.



Android demo image deployment

17. Wait until the download process is done, the MFG tool will display the Operation Compete message when is done.
18. Click Stop and disconnect the USB cable.
19. Change Boot Switch(SW6) to 11100110 to switch the board back to eMMC boot mode.
20. Reset the board to start the boot process.
21. The Android image should boot and the Android GUI will be displayed on the LVDS display.

Agenda - i.MX 6 Series Basics

- The i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

Android benchmark application deployment

- PC set up to support ABD
  - Note the Android SDK is already installed on training machine but can get on this web page http://developer.android.com/sdk/index.html

1. Power-up the board and go to the settings section.
2. Got to the Apps section by clicking on the upper right corner
3. Click on the Settings icons
4. Set display->sleep to 30min.
5. Enable Developer option -> USB debugging.
6. Connect the USB OTG to the computer.
7. For the Android device browse for the driver in the host PC and select the this folder C:\Program Files\Android\android-sdk\extras\google\usb_driver.

Android benchmark application deployment

8. For the MTP device, select Don't search. I will choose the driver to install and select the Android phone option.
9. Open Command Prompt by clicking start-Run-cmd and move to the following directory C:\Program Files\Android\android-sdk\platform-tools.
10. Verify the device is connected to the abd server using this command: adb devices, the command should show at least one device connected.
11. Copy the fsl_mem_benchmark.apk from this location ~Desktop\FTF-CSD-F0066\ to this folder C:\Program Files\Android\android-sdk\platform-tools.
12. Deploy the benchmark apk using the following command, adb install fsl_mem_benchmark.apk.

Android benchmark application deployment

13. Start the memory benchmark application in the application section of the Android device.
14. Configure the source and destination buffer size and start the benchmark.
15. Stop the benchmark to see the results.

Agenda - i.MX 6 Series Basics

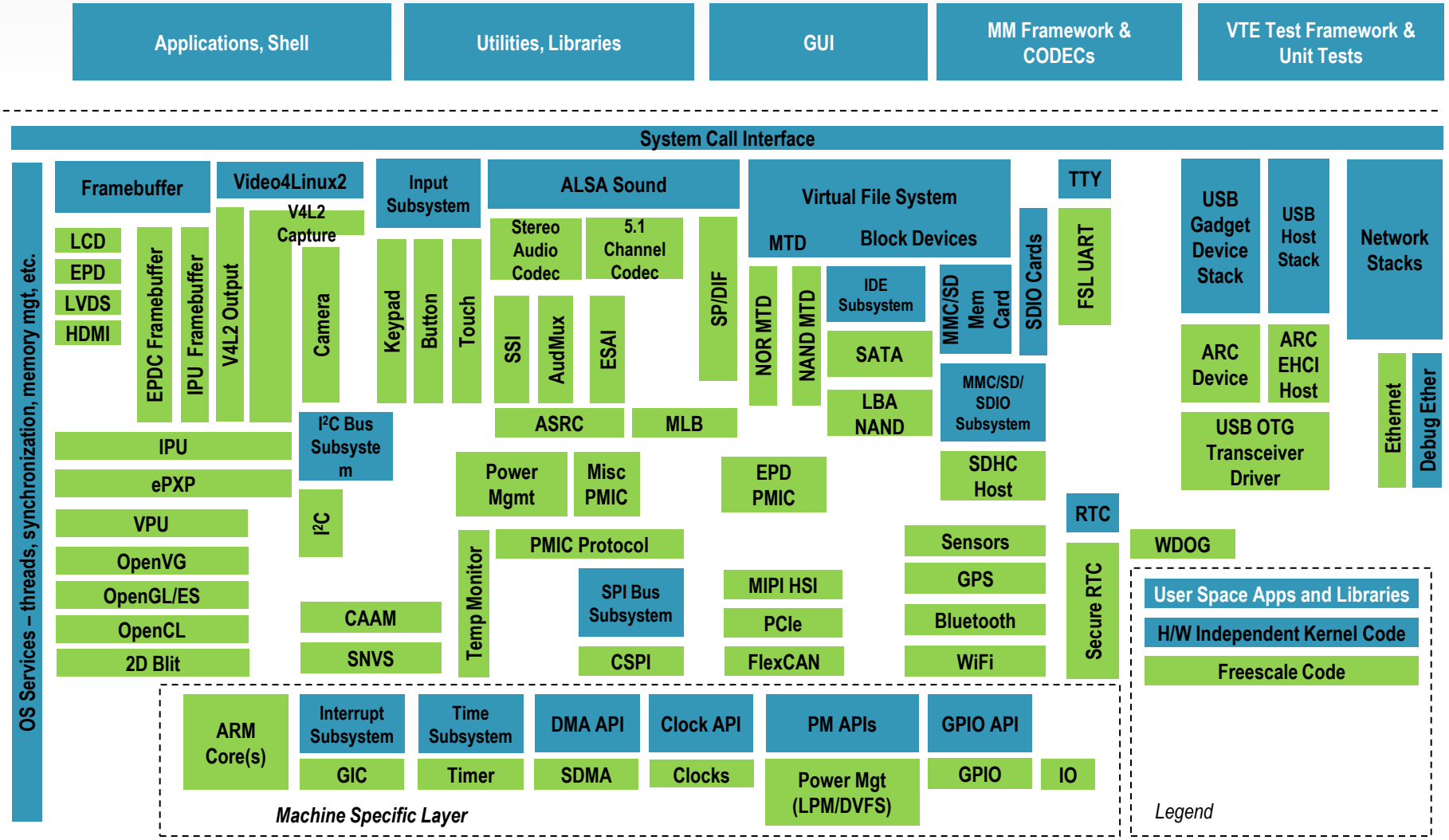
- The i.MX 6 Family, i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

The i.MX 6 Linux BSP

What we deliver with the BSP

- Linux BSP Source Code
- Linux BSP Documentation package
- Linux BSP Demo Image package
  - U-boot image
  - Linux Kernel image
  - Root File System
- Linux Multimedia Codecs binary files
- Linux Multimedia Codecs Documentation
- Manufacturing Tool (MFG tool) package

Driver delivered with the i.MX 6Quad Linux BSP



Agenda - i.MX 6 Series Basics

- The i.MX 6 Family, i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

Linux demo image deployment

1. Decompress the Linux demo image (~Desktop\FTF-CSD-F0066\MX6_BSP\L3.0.15_12.04.01_ER_images_MX6X.tar.gz)
2. Copy the Ubuntu SABRE SD images files

   ~\L3.0.15_12.04.01_ER_images_MX6X\u-boot-mx6q-sabresd.bin,

   ~\L3.0.15_12.04.01_ER_images_MX6X\uImage

   ~\ oneric.tgz

   to the i.MX6Q Linux MFG tool profile folder (~\Mfgtools-Rel-12.04.01_ER_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files).

Linux demo image deployment

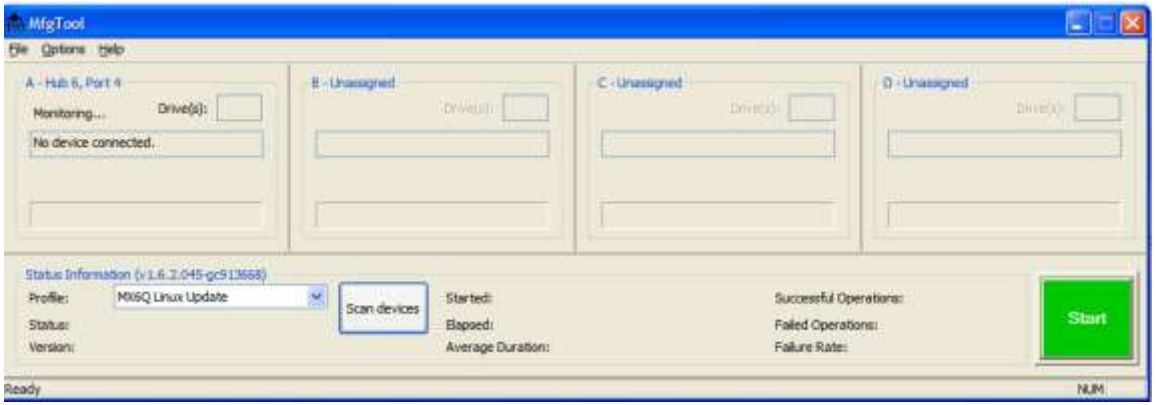
3. Connect the UBS OTG port (J505, bottom) from the SABRE SD to the computer.
4. Connect the USB to SERIAL port (J509, bottom) from the SABRE SD to the computer.
5. Open HyperTerminal on the Windows PC (start->All Programs->Accessories->Communications->Hyper Terminal)
6. Name your new connection IMX6 and click OK
7. Click OK on the Connect to windows

Linux demo image deployment

8. Configure the Port Settings as shown and click OK

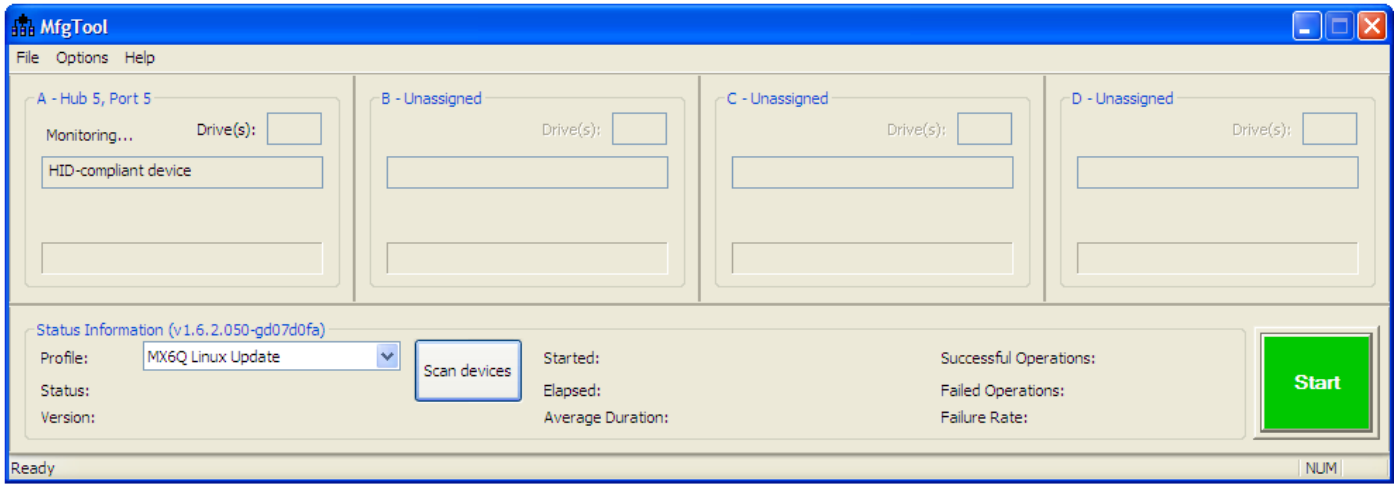
Linux demo image deployment

9. Set the board to the serial download mode, Change Boot Switch(SW6) to 00001100 (from 1-8 bit).
10. Plug in the Power supply to the board.
11. Start the MFGTool by clicking "~\Mfgtools-Rel-12.04.01_ER_MX6Q_UPDATER\MFGTool.exe" The user interface of this tool should as below:



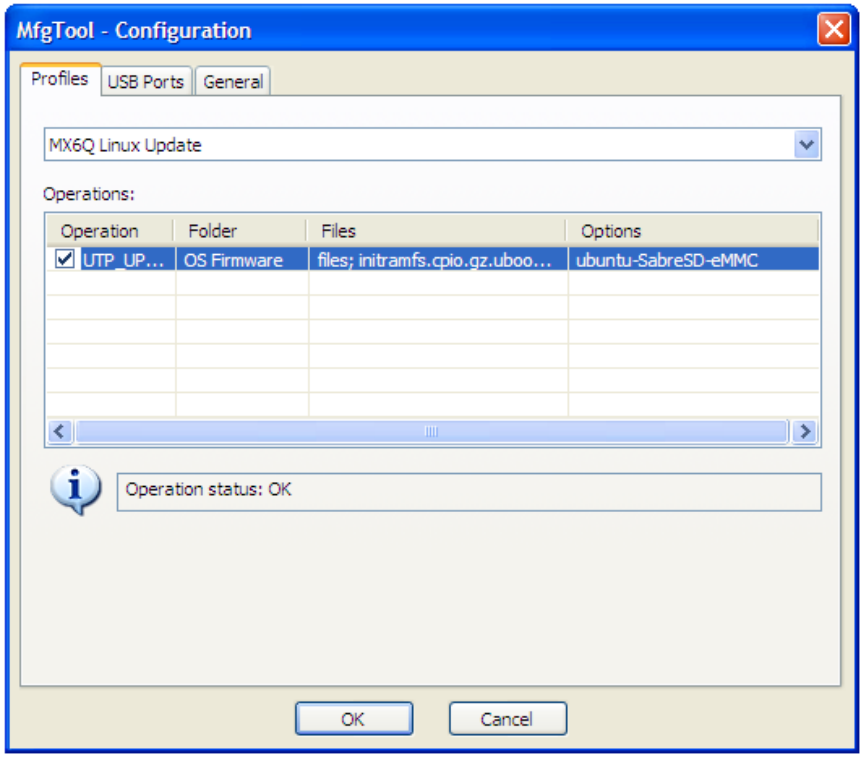
Linux demo image deployment

12. Click Scan devices, which will update the dialog to HID-compliant device as show on the next image.



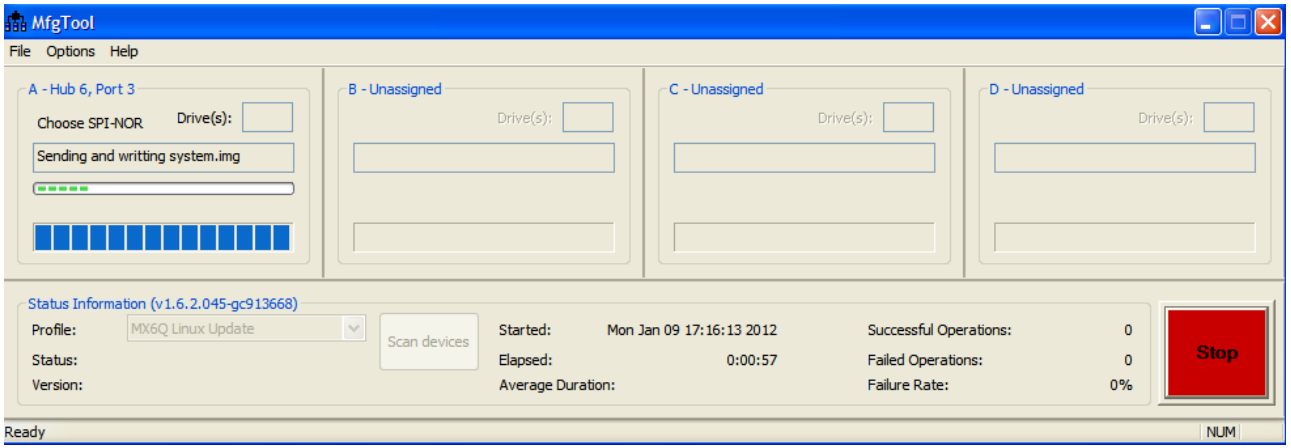
Linux demo image deployment

13. Configure the profile, go to Option -> Configuration.. -> Profiles and choose the ubuntu-SabreSD-eMMC profile on the Options section and click OK



Linux demo image deployment

14. Click start to start image downloading.
15. During the downloading process, you will see the tool become below image, and the status bar will show downloading status. You can also see the progress on the hyper terminal window.



Linux demo image deployment

16. Wait until the download process is done, the MFG tool will display the Operation Compete message when is done.
17. Click Stop and disconnect the USB cable.
18. Change Boot Switch(SW6) to 11100110 to switch the board back to eMMC boot mode.
19. Power down the board and disconnect the LVDS cable
20. Connect the LVDS cable to the LVDS0 port
21. Power up the board to start the boot process.
22. The Ubuntu image should boot and the Ubuntu GUI will be displayed on the LVDS display.

Agenda - i.MX 6 Series Basics

- The i.MX 6 Family, i.MX 6Quad SOC and the i.MX 6Quad SABRE Board for Smart Devices
- i.MX 6Quad Android BSP
- Android demo image deployment
- Android benchmark application deployment
- i.MX 6Quad Linux BSP
- Linux demo image deployment
- Linux benchmark compiling and deployment

Linux benchmark compiling and deployment

- Notes:
  - The Linux BSP for the i.MX 6Q SABRE SDB is already installed on the training virtual machine. For more information about installing the BSP in a host machine please follow the i.MX 6Dual/Quad SABRE SD Linux User Guide.
  - The Eclipse IDE for C/C++ developer is already installed on the training virtual machine. For more information about installing Eclipse in a host machine, refer to the following link http://www.eclipse.org/

Linux benchmark compiling and deployment

- The SW benchmark used in this example is the FSL Memory Copy Benchmark, which is a very simple application that test various memcpy operations under Linux.
- To compile and deploy the benchmark perform the following steps:
  1. Start the VMware player located in start->All Programs->VMware->VMware Player
  2. Open the Ubuntu virtual machine, File->Open a Virtual Machine->~/Desktop/FTF-CSD-F0066/vm_Ubuntu/Ubuntu.vmx
  3. Click on Play virtual machine

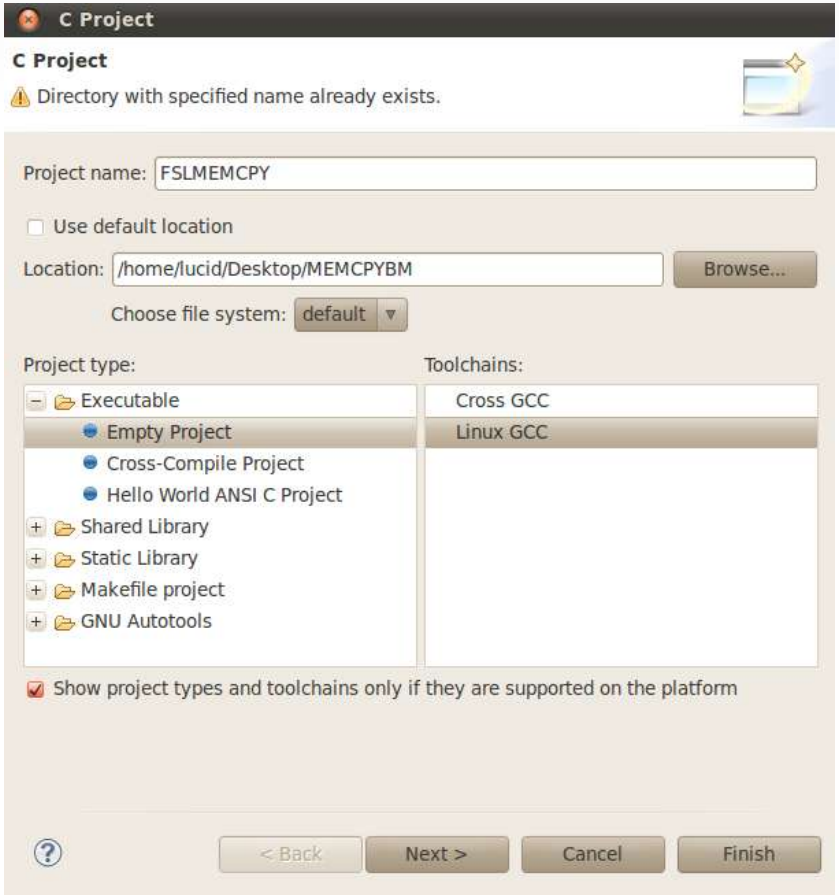
Linux benchmark compiling and deployment

4. Wait until the Ubuntu finish to boot and log-in with this credentials:
5. User: lucid Password: lucid
6. Open the Eclipse IDE with a double click on the eclipse executable file located in the desktop eclipse folder

7. Click Ok on the default workspace
8. Create a new project. Click on File ->New C Project
9. Add a project name, example FSLMEMCPY

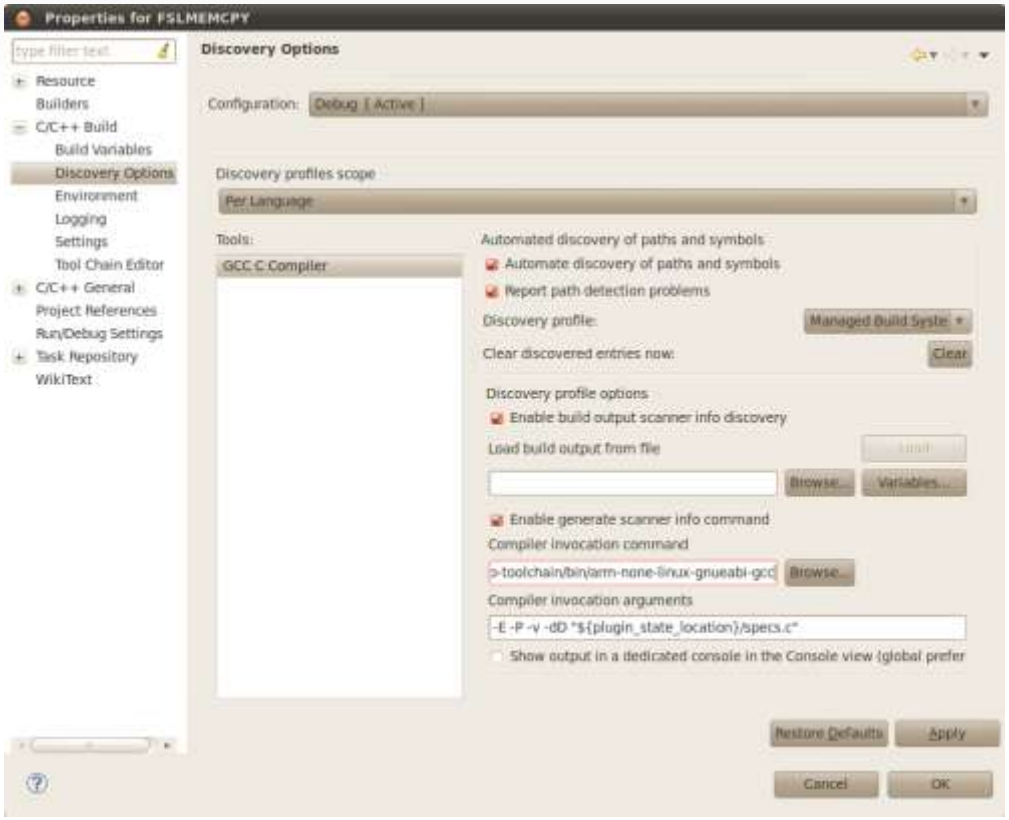
Linux benchmark compiling and deployment

10. Uncheck the Use default location and use the Browse button to find the location of the benchmark source, located in this path /home/lucid/Desktop/MEMCPYBM
11. Select the Empty Project and Linux GCC toolchain, then click on Next >
12. Click on Advanced settings



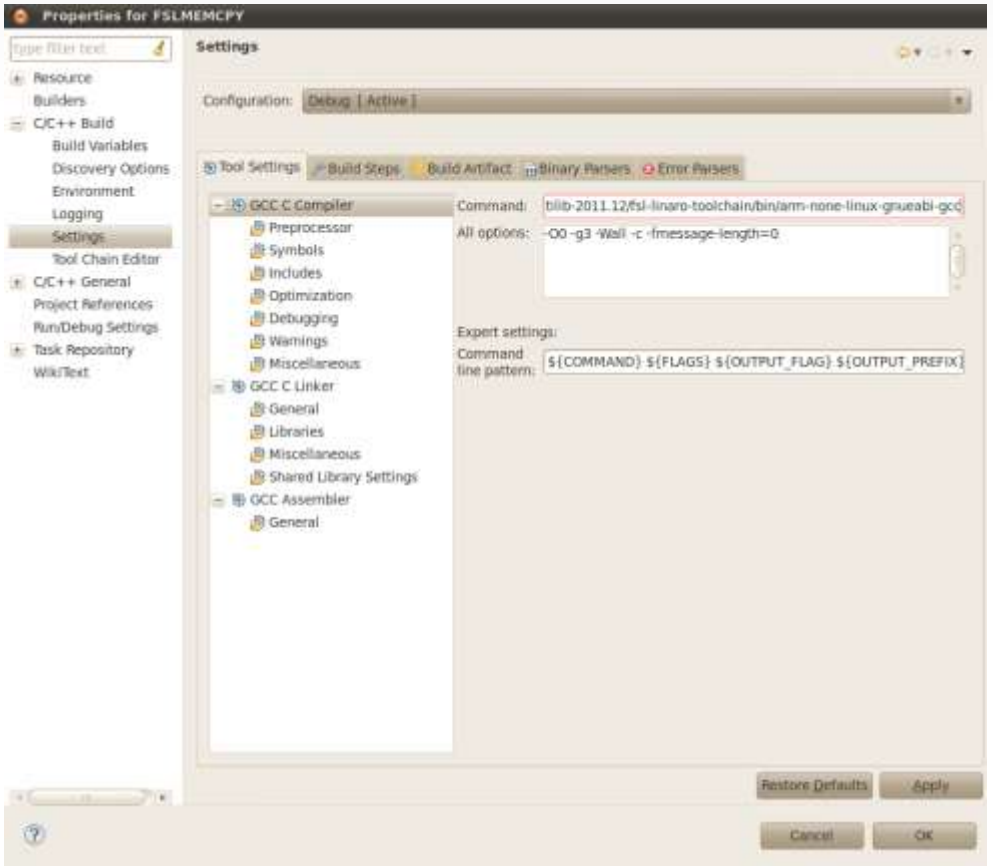
Linux benchmark compiling and deployment

13. Select C/C++ Build -> Discovery Options and browse for the Compiler invocation command selecting this command /opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-gcc
14. Click on Apply



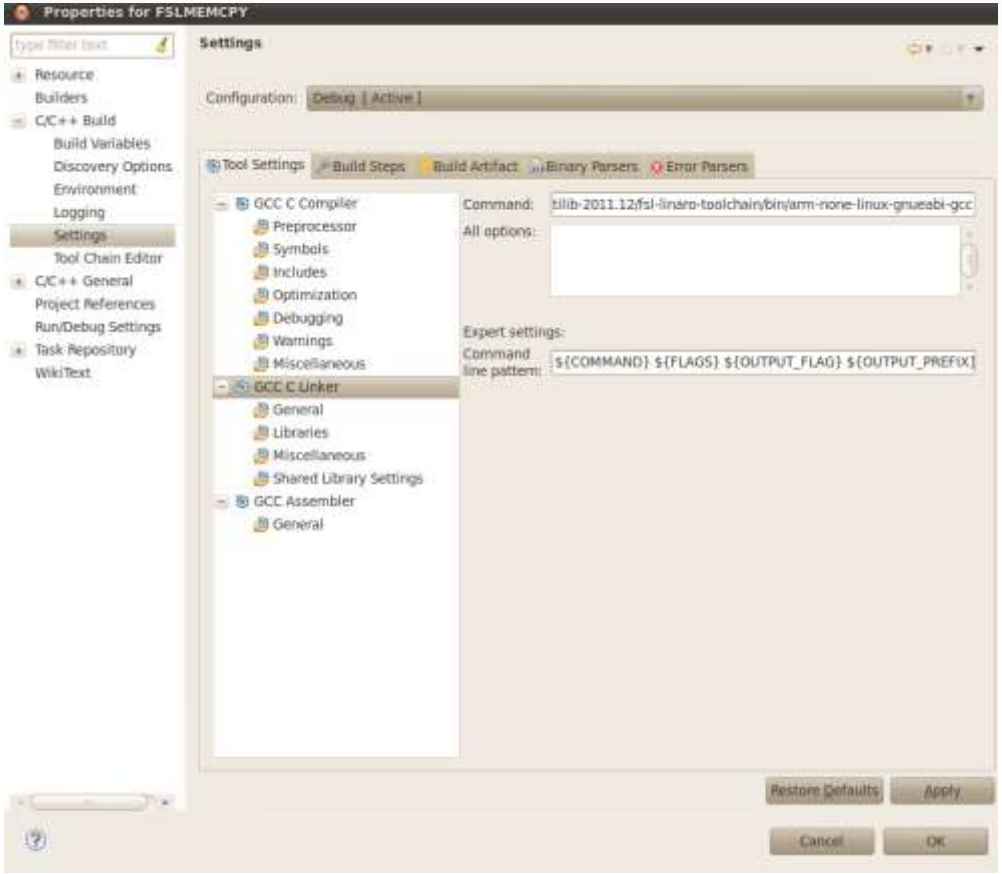
Linux benchmark compiling and deployment

15. Select C/C++ Build -> Settings on the Tool Seetings tab set the GCC C Compiler command with this command /opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-gcc
16. Click on Apply



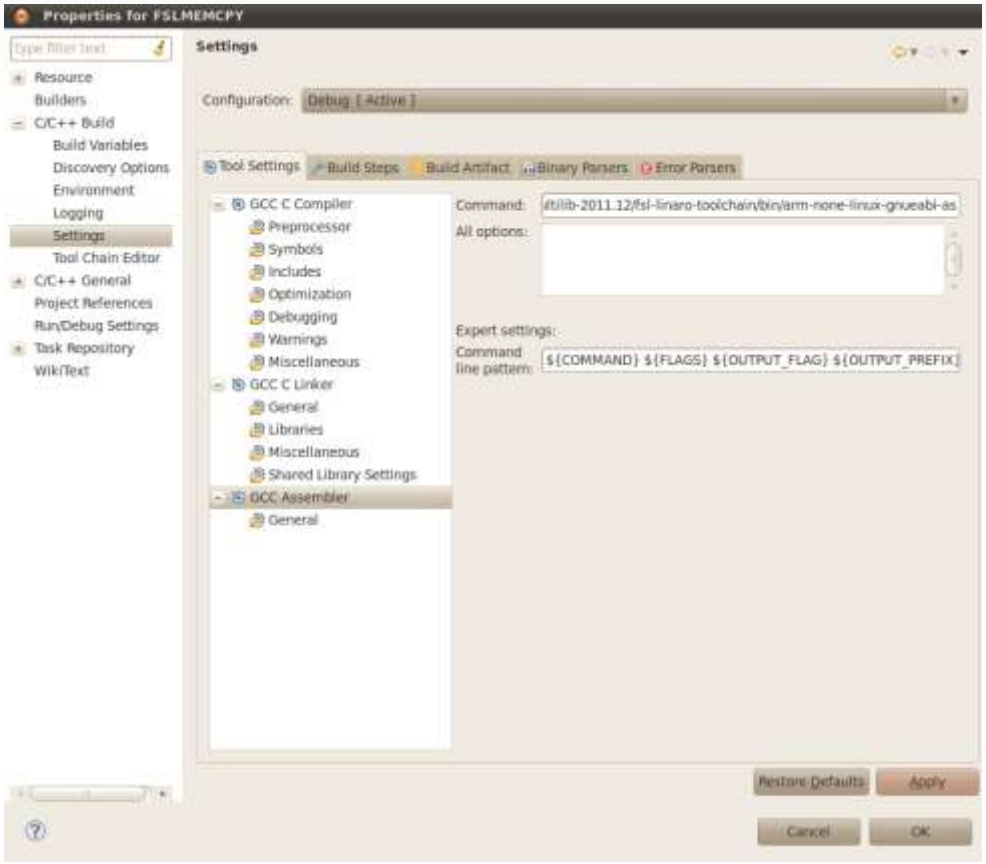
Linux benchmark compiling and deployment

17. Select C/C++ Build -> Settings on the Tool Settings tab set the GCC C Linker command with this command /opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-gcc
18. Click on Apply



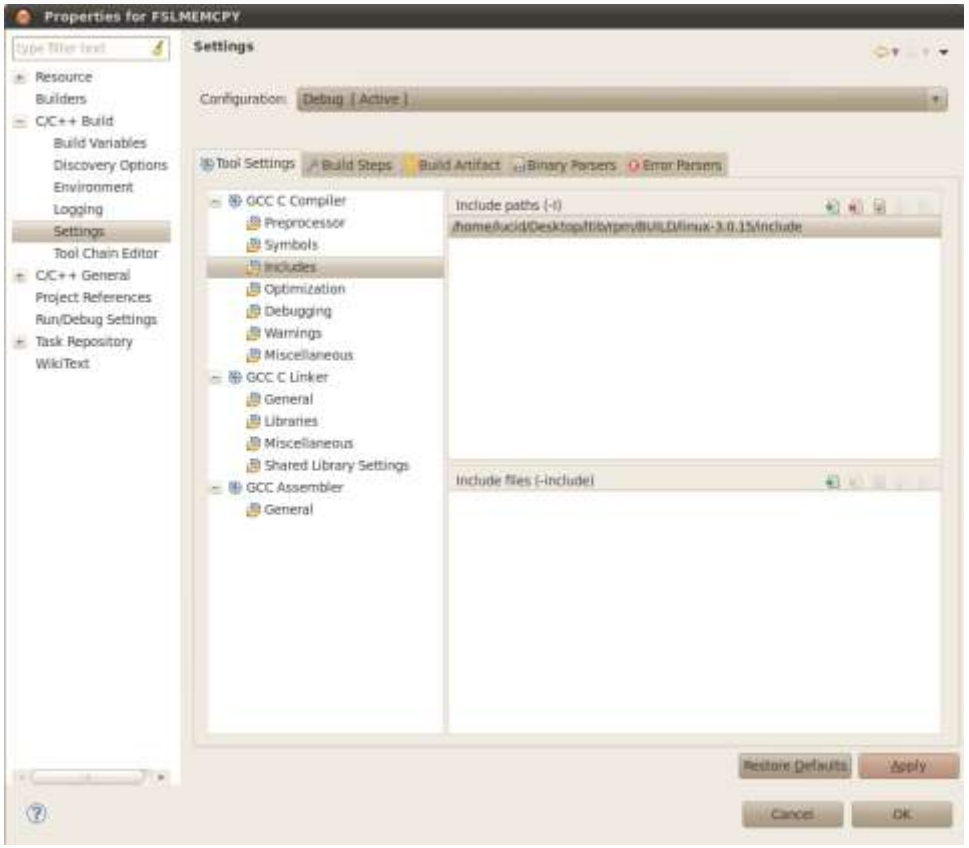
Linux benchmark compiling and deployment

13. Select C/C++ Build -> Settings on the Tool Settings tab set the GCC C Assembler command with this command /opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-as
14. Click on Apply



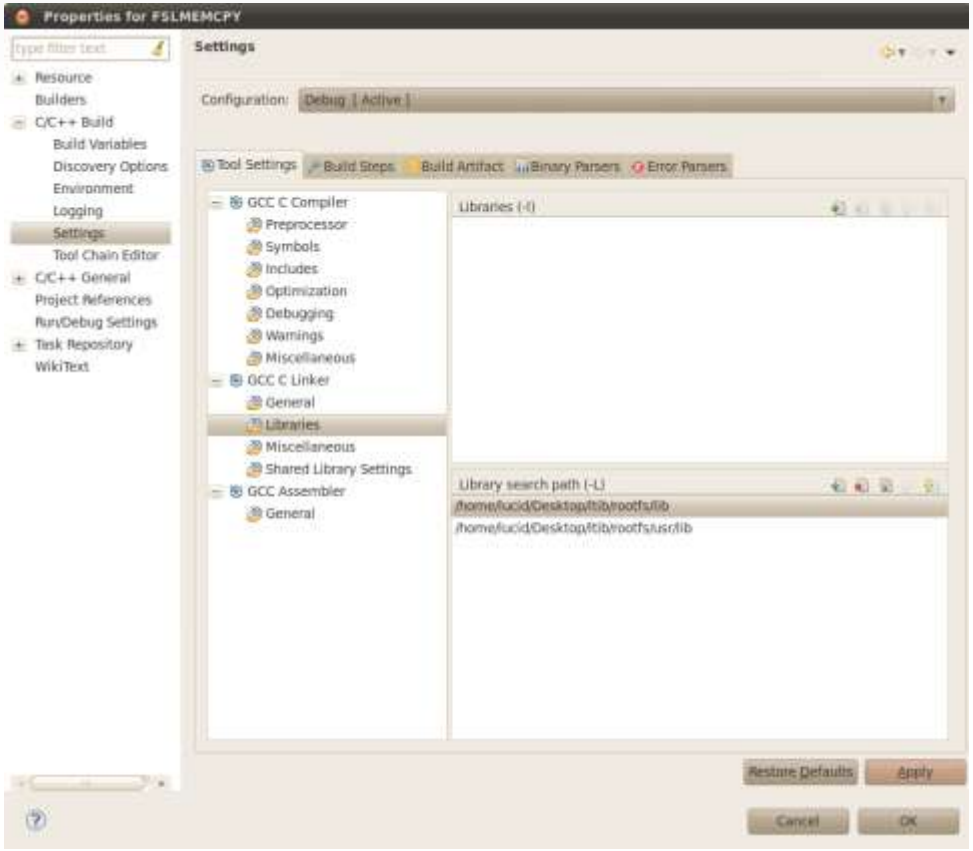
Linux benchmark compiling and deployment

15. Select C/C++ Build -> Settings->GCC C Compiler->Includes and add the next path in the include paths: /home/lucid/Desktop/ltib/rpm/BUILD/linux-3.0.15/include
16. Click on Apply



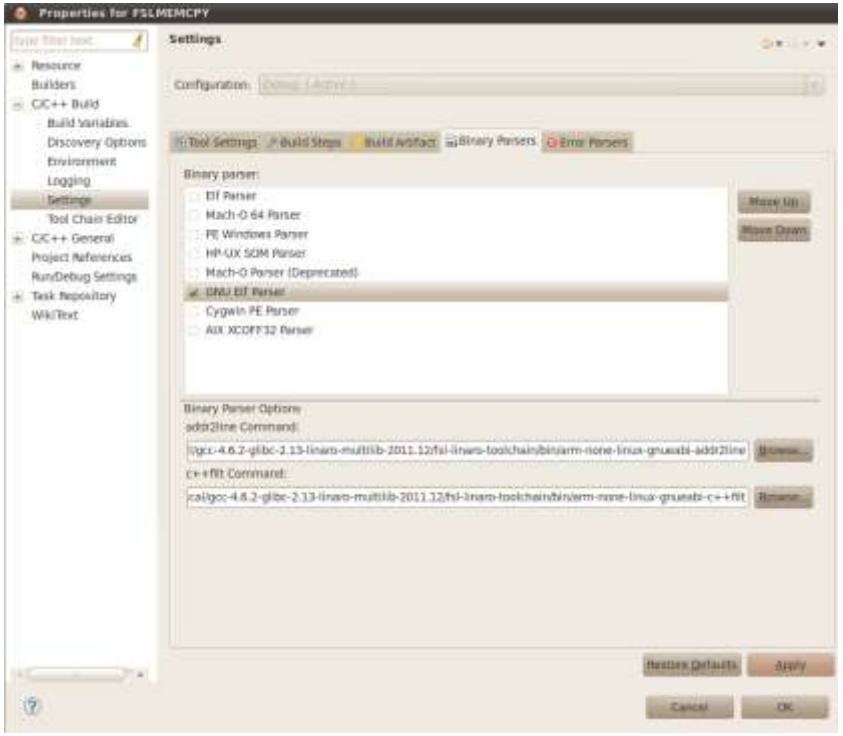
Linux benchmark compiling and deployment

17. Select C/C++ Build -> Settings->GCC C Linker->Libraries and add the next 2 paths in the Library search path:
/home/lucid/Desktop/ltib/rootfs/lib
/home/lucid/Desktop/ltib/rootfs/usr/lib
18. Click on Apply



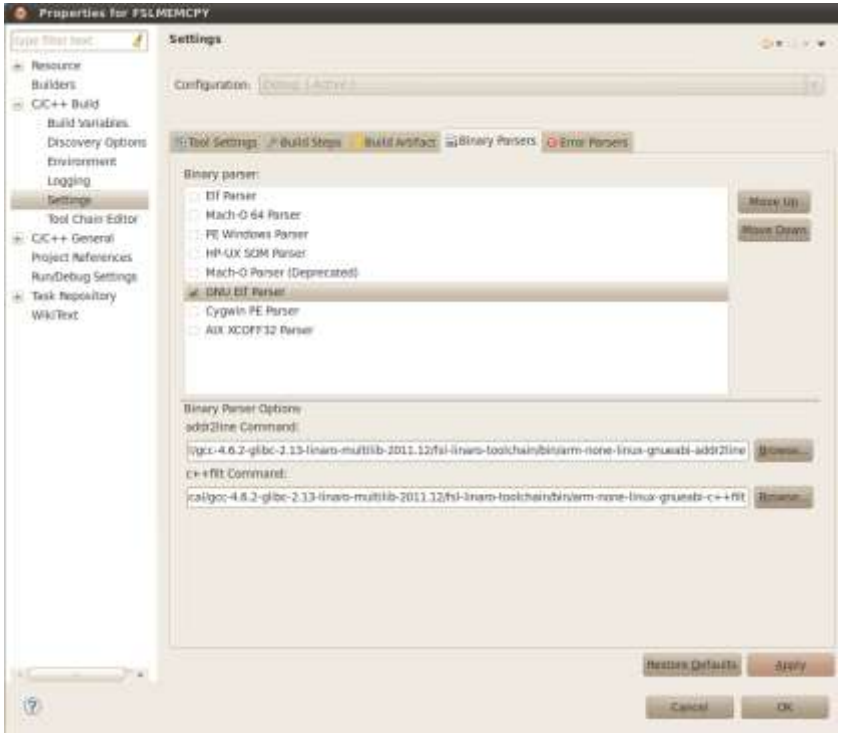
Linux benchmark compiling and deployment

19. Select C/C++ Build->Settings on the Binary Parsers tab uncheck Elf Parser and check the GNU Elf Parser
20. With GNU Elf Parser selected change the addr2line Command with this command: /opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-addr2line



Linux benchmark compiling and deployment

23. With GNU Elf Parser selected change the c++filt Command with this command: /opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-c++filt
24. Click on Apply



Linux benchmark compiling and deployment

25. Close the Advanced Settings windows clicking on OK
26. Click on Finish, to finish the configuration of the project
27. Finally, build the project. Click on Project->Build Project
28. The FSLMEMCPY executable file is now located in this path /home/lucid/Desktop/MEMCPYBM/Debug
29. Connect the USB SD card reader to the training laptop.
30. Enable the use of the USB SD card reader in the virtual machine selecting VM->Removable Devices->Standard Flash Card Reader - > Connect. The SD icon should appear on the desktop.
31. Copy the FSLMEMCPY exe file to the SD card

Linux benchmark compiling and deployment

25. Eject the SD card from the Ubuntu VM using the Eject option doing right clock on the SD icon.
26. Unplug the SD card from the SD card reader and plug the SD card in the SD3 slot (bottom) of the board
34. Power on the board and wait until the command prompt is available in the hyper terminal
35. In the hyper terminal, move to the SD card directory with this command, cd /media/sd
36. Execute the benchmark application with this command, ./FSLMEMCPY 500 2 16384

Q&Q

- Thank you

Facebook.com/Freescale
Tag yourself in photos
and upload your own!

Tweeting?
Please use hashtag
#FTF2012

Session materials will be posted @ www.freescale.com/FTF
Look for announcements in the FTF Group on LinkedIn or follow Freescale on Twitter

freescale™

www.imxcommunity.org

A Freescale supported open web community of developers sharing common interest in transforming i.MX applications processors into practically anything imaginable.

Community Facts at a Glance

- Over 3,800 members and over 200 Freescale engineers and marketers interacting with you
- Support and enablement for i.MX processors and software
- Forums, Groups and Blogs Posts
- News, Photos and Videos
- Training, Events and Promotions


freescale™