ARMADA™ 610 Tablet Reference Platform

for Android™ 2.2, Linux® Kernel 2.6.32

Software Release Notes

Not Approved by Document Control. For Review Only.

Doc. No. MV-S302001-00, Rev. A

January 6, 2011

CONFIDENTIAL

Document Classification: Proprietary Information

Document Conventions

	**Note:** Provides related information or information of special importance.
	**Caution:** Indicates potential damage to hardware or software, or loss of data.
	**Warning:** Indicates a risk of personal injury.

Document Status

Doc Status: Released	

For more information, visit our website at: www.marvell.com

Table of Contents

1 Overview

This software release package contains source code for the Beta 2 version of the Marvell® ARMADA™ 610 Tablet Reference Platform for Android™ 2.2, Linux® kernel 2.6.32.

The release package includes:

- Prebuilt binaries – Use to flash the ARMADA 610 Tablet Reference Platform; these binaries are ready for immediate use.
- Source code – Customize and build the code to create new binaries.

THIS SOFTWARE IS PREPRODUCTION SOFTWARE. IT CANNOT BE USED "AS IS" FOR PRODUCTION SYSTEMS. NO WARRANTY, EXPRESSED OR IMPLIED, IS TO BE ASSOCIATED WITH THIS SOFTWARE AND THE USER ASSUMES ALL RISKS.

For the latest Marvell documentation and the latest Marvell software and hardware updates, contact your Marvell representative.

1.1 System Requirements

1.1.1 Hardware Requirements

This release requires the Marvell ARMADA 610 Tablet Reference Platform with A0/A1 stepping of the ARMADA 610 processor and 1 GB of DDR3 memory.

All applicable Engineering Change Orders (ECOs) for a particular hardware revision must be applied to ensure that the software operates properly.

For the ECO level supported by this release and for ECO documentation, see Table 1.

Table 1: ECO Information for ARMADA 610 Tablet Reference Platform

ECO Level Requirement	Document	Document Number
Revision 1 • PB number: F00055-100 • ECO level: ECOs 1 through 22 (0x3FFFFF)	*Marvell® ARMADA™ 610 Tablet Reference Platform, Revision 1 Engineering Change Orders*	MV-S501271-00
Revision 2 • PB number: F00055-200 • ECO level: ECOs 1 through 12 (0xFFF)	*Marvell® ARMADA™ 610 Tablet Reference Platform, Revision 2 Engineering Change Orders*	MV-S501274-00
Revision 3 • PB number: F00055-300 • ECO level: ECOs 1 through 8 (0xFF)	*Marvell® ARMADA™ 610 Tablet Reference Platform, Revision 3 Engineering Change Orders*	MV-S501275-00

The printed board (PB) number F00055 identifies the board as a Marvell ARMADA 610 Tablet Reference Platform. The -100, -200, and -300 indicate revision 1, revision 2, and revision 3, respectively, of the unpopulated board and schematics.

For the ECO level, an ECO label is applied to the board that encodes the numbers of the ECOs added to the board into a hexadecimal number. Refer to the appropriate Engineering Change Orders document for detailed ECO information.

1.1.2 Software Requirements

This release version requires:

- Host PC with operating system – Ubuntu® 8.04 or 9.04

Note Marvell routinely tests the Android build on Ubuntu 8.04 and 9.04. There are reports of build errors in Ubuntu 9.10 and a required patch for the Android code to build successfully. Therefore, Marvell highly recommends using Ubuntu 8.04 or 9.04.

Note Follow the instructions on http://source.android.com/source/download.html to set up the Android build environment on Ubuntu Linux.

- Android 2.2, Linux kernel 2.6.32

1.2 Release Differences

1.2.1 Differences from the Previous Release (Beta 1)

Differences between Beta 2 and Beta 1 releases follow:

- **BSP Differences**
  - Enables FM audio on the 88W8787 module
  - Enhances the LCD frame rate from 40 frames per second (FPS) to 58 FPS
  - Enables the CyWee motion sensors
  - Keeps the DDR clock at 400 MHz for each production point
- **Android Differences**
  - Provides VideoPlayer support for suspend/resume
  - Enables AudioCache Sink
  - Enables FM audio
  - Enables the CyWee motion sensors
  - Enables Marvell Scalable Power Management (MSPM)
- **Other Differences**
  - Includes the Marvell Vmeta™ hardware abstract layer (HAL) upgrade to version 1126
  - Upgrades the graphics controller (GC) libraries to version 0.8.0.2547
  - Enables silent reset, eglSwapInterval and MRVL_READ_PIXEL_2D
  - Disables skip 2D context, replaces idle_event with broadcast, and adds YUY2 format support for post buffer optimization

1.2.2 Differences for Other Previous Releases

1.2.2.1 Beta 1 Release

Differences between Beta1 and Alpha5 releases follow:

- **BSP Differences**
  - Enables the Video DMA (VDMA) interface by default
  - Adds a 1GHz operation point
  - Enables the OV5642 camera
  - Supports the on-board digital microphone
  - Uses a USB 480 MHz PLL as the GC and Vmeta clock source
- **Android Differences**
  - Unloads the driver when the Wi-Fi®/Bluetooth®/FM module is disabled
  - Chooses IPP SW MEG4/H.264 SW decoder for VT
  - Enables RIL support PCIe TD data modem in the following fields: dialing/accepting call (no voice), sending/receiving SMS and ppp data connection
  - Fixes camera cannot work in low-profile mode issue
- **Other Differences**
  - Includes support for the Wireless Trusted Module (WTM) firmware upgrade to version 2.1.3

    The WTM firmware needs to be upgraded to version 2.1.3. Download it separately from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ARMADA 610 (MMP2)/Software/WTM/Version 2.1.3*.
  - Includes the Marvell Vmeta HAL upgrade to version 1029
  - Fixes most seek issues
  - Includes the Vmeta encoder support for JPEG encoding

1.2.2.2 Alpha5 Release

Differences between Alpha5 and Alpha4 releases follow:

- **BSP Differences**
  - Supports the ARMADA 610 processor A0 and A1 stepping
  - Adds an additional LCD mode
  - Adds the VDMA interface
  - Updates the AXI bus clock from 200 MHz to 266 MHz on the ARMADA 610 Tablet Reference Platform with the A0/A1 stepping of the ARMADA 610 processor
  - Supports the ARMADA 610 Tablet Reference Platform
  - Enables the keypad and touch for the ARMADA 610 Tablet Reference Platform
  - Enables the USB charger for the ARAMDA 610 Tablet Reference Platform
  - Enables the AUO 1280 x 720 LCD panel
- **Android Differences**
  - Fixes the issue in which the ARMADA 610 processor cannot work in an environment where both Bluetooth and Wi-Fi coexist
  - Fixes the Bluetooth/Wi-Fi bug of exceeding the Java Native Interface (JNI) global reference
  - Fixes the HDMI™ bug of no display after switching the display
  - Adds Wi-Fi host sleep and Marvell Mobile Hotspot (MMH) support
  - Integrates dual display support

- **Other Differences**
  - Changes the release platform from the ARMADA 610 Reference Platform to the ARMADA 610 Tablet Reference Platform
  - Upgrades the GC libraries to version 0.8.0.2300
  - Fixes several bugs for games, streams, and graphics controller unit (GCU)
  - Enables GC reset, idle profiling and refines the GC power logic
  - Upgrades the Vmeta hardware abstraction layer to the 0901 version
  - Supports reordering in video playback for all formats

1.2.2.3 Alpha4 Release

Differences between Alpha4 and Alpha3 releases follow.

- **BSP Differences**
  - Changed the reference platform from the ARMADA 610 Reference Platform to the ARMADA Tablet Reference Platform
  - Enlarged the program memory (PMEM) from 32 MB to 48 MB
  - Enabled the eMMC and SD slot on the ARMADA 610 Tablet Reference Platform
  - Enabled the LCD on ARMADA 610 Tablet Reference Platform
  - Added an additional LCD mode
  - Added the VDMA interface
  - Enabled the OmniVision® OV5642 Preview on the ARMADA 610 Tablet Reference Platform
  - Updated the AXI bus clock from 200 MHz to 266 MHz on the ARMADA 610 Tablet Reference Platform with A0 stepping on the ARMADA 610 processor
- **Android Differences**
  - Enabled simultaneously different video content playback on the HDMI and LCD.
  - Disabled the video/UI rotation for no G-Sensor hardware
  - Enabled the keypad backlight
  - Added USB keyboard support
- **Other Differences**
  - Temporarily removed OBM this release; OBM to be added in a following release
  - Upgraded GC libraries to version 0.8.0.2046

1.2.2.4 Alpha3 Release

Differences between Alpha3 and Alpha2.5 releases follow.

- **BSP Differences**
  - Printed ARMADA 610 processor stepping in U-Boot boot
  - Used the DDR memory calibration hardware method to reduce latency
  - Enlarged the PMEM from 16 MB to 32 MB
  - Set the external SD slot clock to a maximum of 24 MHz
- **Android Differences**
  - Upgraded the Android version from 2.1 to 2.2
  - Built the Android image with Vector Floating Point (VFP) and ARMv7 support
  - Enabled audio record
  - Removed the integration of the Color Management Unit (CMU)

- Did not provide support for the .3gpp and .mp4 high resolution (720p and above) video since Marvell Integrated Performance Primitives (Marvell IPP) Codec with Vmeta technology is not integrated in OpenCore
- Added suspend/resume support; letting the device enter suspend requires echo disable_suspend > /sys/power/wake_unlock to remove the disable_suspend wakelock

■ **Other Differences**

- Upgraded the GC800[1] to version 0.8.0.1739
- Updated the Marvell Code Performance Analyzer tool to version 2.3.54
- Used arm-marvell-linux-gnueabi-vfp-4.2.0 to build OBM/U-Boot
- Used google arm-eabi-4.4.0 tool chain to build the kernel/Android.

1.2.2.5 Alpha2.5 Release

Differences between Alpha2.5 and Alpha2 releases follow.

■ **BSP Differences**

- Switched the core to an ARMv7 mode NTIM
- Added support for Micron® 16 GB eMMC flash memory
- Added audio record support
- Added ON-KEY support
- Added power off command support

■ **Android Differences**

- Integrated the Power button
- Changed the MP3 and AAC playback engine from GStreamer to OpenCore

■ **Other Differences**

- Updated GC800 to version 0.8.0.1620
- Updated the Marvell Code Performance Analyzer tool to version 2.3.38
- Added a requirement for a separate download of the WTM firmware image from the Marvell Extranet at *My Products/ Cellular & Handheld Solutions/ Applications Processors/ ARMADA 610 (MMP2)/ Software/ WTM/ Version 1.2.1*; contact your Marvell representative if you have any issues regarding the download.

1.2.2.6 Alpha2 Release

Differences between Alpha2 and Alpha1.1 releases follow.

■ **BSP Differences**

- Upgraded the kernel to 2.6.32
- Enabled CMU
- Enabled ON-KEY and Reset
- Enabled multi-touch
- Enabled Dynamic Voltage and Frequency Management (DVFM) and processor sleep mode
- Enabled the SD8787
- Enabled the Light and Proximity sensor
- Enabled HDMI Extended Display Identification Data (EDID)
- Enabled the compass sensor
- Enabled Vmeta multi-instance support

1. GC800 refers to the Vivante Corporation GCCORE Graphics Processing Unit IP architecture.

- **Android Differences**
  - Integrated the G-Sensor, Light Sensor, and Proximity Sensor
  - Integrated the CMU function and added the CmuSettings.apk for color settings
  - Integrated the Image Rotation Engine (IRE) to the overlay-hal, support for video rotate
  - Added internal storage (7G) support
  - Added Gallery3D, Bluetooth, CertInstaller, DeskClock, LiveWallpapers, FileManager, CmdLine and CmuSettings application
  - Integrated Wi-Fi and Bluetooth
  - Enabled LCD backlight and auto change based on the light sensor
  - Added optimized OpenSSL support
- **Other Differences**
  - Enabled Marvell WTM and provided cryptographic accelerator and trusted computing support
  - Added Marvell Code Performance Analyzer support
  - Added Marvell IPP library and Vmeta multi-instance support

1.2.2.7 Alpha1.1 Release

Differences between Alpha1.1 and Alpha1 releases follow.

- **BSP Differences**
  - None
- **Android Differences**
  - Upgraded the Vmeta HAL to version 2010-4-7
  - Upgraded the GC800 to version 1843
  - Integrated GC800 optimization for Android surface flinger
  - Added IRE user space library source code
  - Added Vmeta user IO library source code
  - Fixed the bug that HDMI video playback does not fit the screen
- **Other Differences**
  - Added requirement for Marvell Extreme Debugger 5.0 Alpha2 to flash OBM and U-Boot
  - Added eMMC auto boot support in OBM
  - Provided both eMMC and NAND auto boot prebuilt binaries

1.2.2.8 Alpha1 Release

Differences between Alpha1 and PreAlpha2 releases follow.

- **BSP Differences**
  - Added support for the ARMADA 610 processor with Z1 stepping
  - Added support for Elpida 512 MB LPDDR2 memory
  - Added support for Synaptics TM1414 touch
  - Added UBIFS support
  - Enabled IRE
  - Enabled VDMA in the LCD pipeline
  - Enabled WM8994 audio playback
  - Enabled USB Host and USB OTG
  - Enabled SDIO

- Enabled Multi-Channel Direct Memory Access (MDMA)
- Enabled the BMA150 G sensor

■ **Android Differences**

- Upgraded the Android framework to 2.1 r2
- Enabled H.264 Main Profile and AAC OpenMAX with Vmeta technology for OpenCore
- Enabled ADB tools
- Integrated the volume up/down key
- Enabled the audio path and integrated the speaker
- Enabled the SD-Card as massive storage to the PC
- Integrated USB 3G dongle support; see “Wireless” in Table 2 on page 11
- Added dual display support with Video application: HDMI for video playback, LCD for graphics; Dynamic switching of audio/video playback on HDMI and LCD

■ **Other Differences**

- Released U-Boot and OBM source code
- Added requirement to use the Marvell Extreme Debugger 5.0 with a flash patch for burning OBM and U-Boot rather than using EMU2.1
- Changed the kernel burning address from NAND 0x980000 to eMMC 0x4c00

1.3 Platform Features

Table 2 lists the platform features for this version of the Marvell ARMADA 610 Tablet Reference Platform for Android 2.2 release package.

Table 2: Platform Features (Sheet 1 of 3)

Features		Support
General	Android Version	2.2
	Linux kernel	2.6.32
Power Management	Android power integration	Yes
	Battery information	No
	Suspend/Resume	Yes
Video Playback	Video output to HDMI™	Yes
	Video output optimized by overlay	Yes
	Simultaneous video output to HDMI and UI operation on LCD	Yes
	Simultaneous different video content playback on HDMI and LCD	Yes
	Video rotation	No
	Video output through graphics controller	No
Audio Playback	Headset switch detection	Yes
	Audio driver integration	Yes
	Audio to headset	No
	Audio to speaker	Yes
	Audio to HDMI	Yes

Table 2: Platform Features (Sheet 2 of 3)

Features		Support
Audio Recording	AMR-NB encoding	Yes
	AAC encoding	Yes
	Sound recorder integration	Yes
Video Recording	Camera stack integration	No
	Camera sensor tuning	No
Touch	Single touch	Yes
	Multiple touch	Yes
Bluetooth® technology	Bluetooth base stack	Yes
	Advanced Audio Distribution Profile (A2DP)	Yes
Sensors	Gravity sensor	No
	Light sensor	Yes
	Proximity sensor	Yes
Light	LCD backlight	Yes
	Keypad backlight	Yes
Alarm	Trigger alarm from Standby	No
Wireless	Wi-Fi® access without password	Yes
	Wi-Fi Protected Access (WPA)/WPA2	Yes
	Wi-Fi connection stress test	Yes
	3G USB Dongle	Yes
Multimedia	GStreamer	See Table 3, Supported Media Format and Codecs, on page 13
	OpenCore	
Video Output	Output through graphics controller	No
	Output through overlay	Yes
Tools	Android Debug Bridge (ADB) integration	Yes
	Marvell Code Performance Analyzer integration	Yes
Graphics	2D/3D graphics controller (GC800[1])	Yes
User Storage	SD card	Yes
	Internal storage partition	Yes
Boot Storage	eMMC	Yes
	SD	Yes
System Update	Fast boot protocol	No
	SD upgrade	No

Table 2: Platform Features (Sheet 3 of 3)

Features		Support
Security	Wireless Trusted Platform Service Package (WTPSP)	Yes
	Wireless Trusted Module (WTM) Adapter for Linux Kernel Crypto Framework	Yes
	Optimized OpenSSL	Yes

1. GC800 refers to the Vivante Corporation GCCORE Graphics Processing Unit IP architecture.

1.4 Multimedia Features

This release supports the media formats and the codecs listed in Table 3.

Table 3: Supported Media Format and Codecs

Containers	Extensions	Audio/Video Combinations		Playback Engine	GST Demuxer Plugins	Status
		Audio	Video			
ASF	.asf	WMA	MPEG-4	GStreamer	asfdemux/gst-plugins-ugly	Ready
	.wmv	WMA	WMV			
	.wma	WMA	No video			
AVI	.avi	MP3	H.264	GStreamer	avidemux/gst-plugins-good	Ready
		MP3	MPEG-4			
		MP3	H.263			
MOV	.mov	AAC	H.264	GStreamer	qtdemux/gst-plugins-good	Ready
		AAC	MPEG-4			
MP4	.mp4	AAC	H.264	OpenCore	--	Ready
		MP3	MPEG-4			Not Ready
MPEG-2 PS	.mpg	MP3	MPEG-2	GStreamer	--	Ready
3GPP	.3gp/.3gpp	MP3	H.264	OpenCore	--	Not Ready
		AAC	MPEG-4			Ready
MKV	.mkv	MP3	H.264	GStreamer	--	Ready
		AAC	H.264			
AAC	.aac	AAC	No video	OpenCore	--	Ready
	.adts	AAC	No video			
MP3	.mp3	MP3	No video	OpenCore	--	Ready

1.5 Board Support Package Features

Board support package features for the ARMADA 610 Tablet Reference Platform for Android 2.2 are as follows.

- U-Boot
  - NAND, non-trusted boot
  - USB Ethernet download
  - zImage format support
  - Support for burning Yet Another Flash File System (YAFFS) image
  - Support for burning an image into eMMC
- Linux Kernel 2.6.32
  - L1 cache
  - L2 cache
  - Interrupt controller
  - Peripheral DMA (PDMA) controller
  - Memory controller
  - Real-Time Clock (RTC)
  - Operating System Timer (OST)
  - Intel® Wireless MMX™[1] technology
  - General purpose Input Output (GPIO) interrupt request (IRQ)
  - Clock management
  - Single level cell (SLC) NAND flash memory
  - OneNAND flash memory
  - Journaling Flash File System, version 2 (JFFS2) support
  - Unsorted Block Image File System (UBIFS) support
  - Yet Another Flash File System (YAFFS)
  - MultiMediaCard (MMC3.2 and MMC4.0)
  - Secure Digital (SD)/SDIO (SD1.1 and SD2.0)
  - UART
  - HDMI Audio
  - SSPA drivers
  - ALSA framework
  - Keypad
  - I2C - Normal I2C (see, I2C Stability, on page 15)
  - DSI LCD panel (base frame, overlay)
  - HDMI LCD TV path
  - Audio DMA (ADMA) controller
  - Marvell Wireless Memory Management technology
  - Performance Monitor Unit (PMU) – used by the Linux Oprofile tool
  - USB client
  - Maxim MAX8925 Power Management Integrated Circuit (PMIC) and MAX8649 regulator
  - Battery driver
  - USB charger

1. Intel and MMX and related marks are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

- USB HOST and USB On-the-Go (OTG)
- Video DMA (VDMA) controller
- WM8994 codec
- Dynamic Voltage and Frequency Management (DVFM)
- Multi touch
- On key and Reset
- Wi-Fi, Bluetooth on SD8787
- CM3623 Light and Proximity sensor
- HDMI EDID
- Marvell Vmeta™ technology, a multiple format high-definition video codec supporting multi instances
- ARMv7 mode
- Audio record
- Power off command
- OmniVision® OV5642 5-megapixel SoC sensor support
- LCD additional mode
- Capacity keypad
- TPK800 touch
- AUO panel
- FM audio
- CyWee motion sensor

I2C Stability

The power on sequence can impact the I2C stability.

On the Marvell ARMADA 610 Tablet Reference Platform, Revision 3, use the following steps to power on.

1. Unplug the USB cable and the power cable to make sure power is off to the board.
2. Insert the power cable.
3. Press the on-key for about 2 seconds to turn on power to the board.
4. Insert the USB cable.

Following these steps ensures that the board is powered by the DC power, not by the USB cable VBUS.

1.6 Release Package Contents

The following tables list and describe the release package for the ARMADA 610 Tablet Reference Platform for Android™ 2.2, Linux kernel 2.6.32.

Table 4: Prebuilt Binary Files

Files	Description
ARMADA610_ANDROID_PLATFORM_BETA2_PREBUILT_BIN.zip	Prebuilt bin binaries
• mbr	Main Bootable Record (MBR) table for eMMC
• system_ext3.img	Android system image
• userdata_ext3.img	Android user data image
• zImage.android	Kernel image
• ramdisk_ext3.img	RAM disk image
• u-boot.bin	U-Boot
• /nand/ntim_mmp2_nand_bbu_ddr.bin	NAND version NTIM header
• /nand/ntim_mmp2_nand_ddr3_elipda_lg.txt	NAND version NTIM description file
/nand/MMP2_NTLOADER_3_2_17.bin	NAND version NT loader
• /emmc/ntim_mmp2_nand_bbu_ddr.bin	eMMC version NTIM header
• /emmc/ntim_mmp2_emmc_ddr3_elipda_lg.txt	eMMC version NTIM description file
/emmc/MMP2_NTLOADER_3_2_17.bin	eMMC version NT loader
• /emmc/partition.bin	eMMC partition image
• /emmc/partition.txt	eMMC partition description file

Note

The WTM firmware image must be downloaded separately from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ARMADA 610 (MMP2) Software/WTM/Version 2.1.3*. Contact your Marvell representative if you have issues about the download.

.

Table 5: Source Files

Files	Description
ARMADA610_ANDROID_PLATFORM_BETA2_SRC	Source code tarball (patch based source code)
• setup_android.sh	Script help to set up the Android code base from the xxx_src.tgz and xxx_patches.tgz
• android_patches.tgz	Marvell patches to the Android Projects
• android_src.tgz	Source code for projects added by Marvell
• marvell_manifest.xml	Manifest xml file to download the Android source code from Google as a base
• kernel_patches.tgz	Marvell patches to kernel_src.tgz
• kernel_src.tgz	Kernel base source code
• uboot_src.tgz	U-Boot base source code
• uboot_patches.tgz	Marvell patches to uboot_src.tgz
• obm_src.tgz	OEM Boot Module (OBM) source code

Caution

The non-trusted image module (NTIM) and BootLoader (OEM boot module) files provided are designed and customized for use with the associated Marvell hardware platform. Use these files as a reference. You MUST create the NTIM and BootLoader with the correct parameters for your design.

Failure to correctly implement the NTIM or BootLoader may result in a boot failure or cause an unreliable operation of your device.

For information and assistance in correctly setting up your NTIM and BootLoader, see the Marvell Boot ROM or Marvell Wireless Trusted Tool Package documentation or contact your Marvell Applications Engineer or Field Applications Engineer.

Note

For detailed information about the WTPTP release package, see the *Marvell® Wireless Trusted Platform Tool Package for Application Processors Software Release Notes* (MV-S301673-00).

THIS PAGE INTENTIONALLY LEFT BLANK

2 Installation

This section provides procedures for

- Identifying an ARMv6 or ARMv7 mode boot
- Programming the binaries onto flash memory with the Marvell eXtreme Debugger
- Downloading Android onto flash memory
- Setting up the Android working directory
- Building Android
- Using the optimized OpenSSL for Marvell platforms

Note

See Section 1.6, Release Package Contents, on page 16 for a description of the release package contents.

2.1 Use the Prebuilt Binaries

The following procedures provide information for programming binaries onto flash memory using the Marvell eXtreme Debugger.

2.1.1 Identifying an ARMv6 or ARMv7 Mode Boot

The non-trusted image module (NTIM) is updated to switch the processor core from ARMv6 to ARMv7 mode. If the new NTIM image is burned into flash correctly, the boot ROM switches the core from ARMv6 to ARMv7 mode during the boot. Without a new NTIM, the processor still boots in the ARMv6 mode.

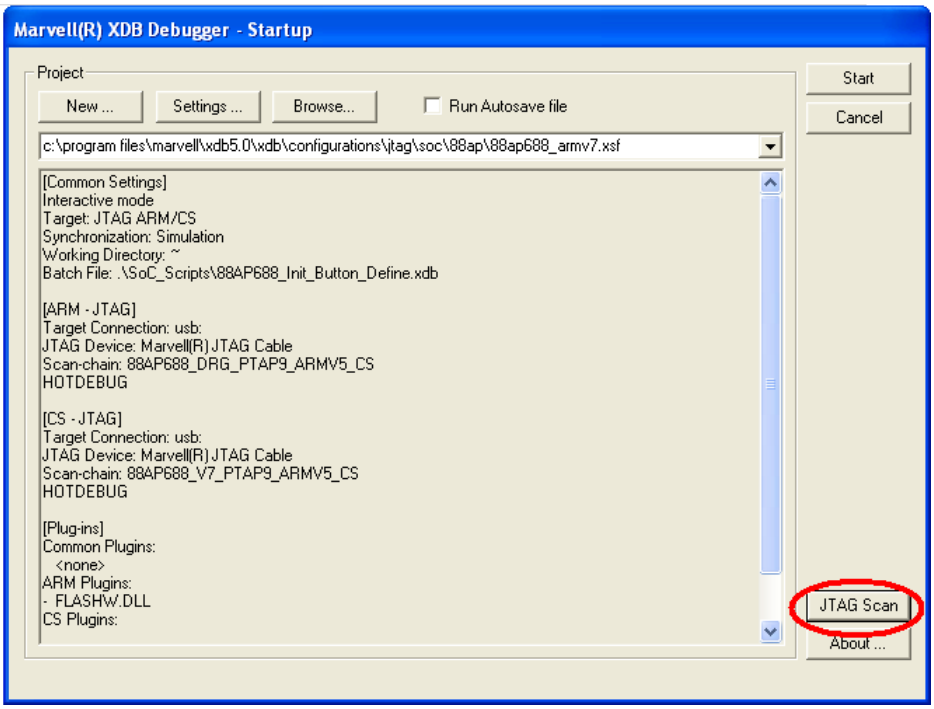
To identify whether the processor is in ARMv6 or ARMv7 mode, use the steps that follow.

Note

You need the Marvell eXtreme Debugger, version 5.1.018 or higher. Download the Marvell eXtreme Debugger from the Marvell Extranet or contact your Marvell representative.

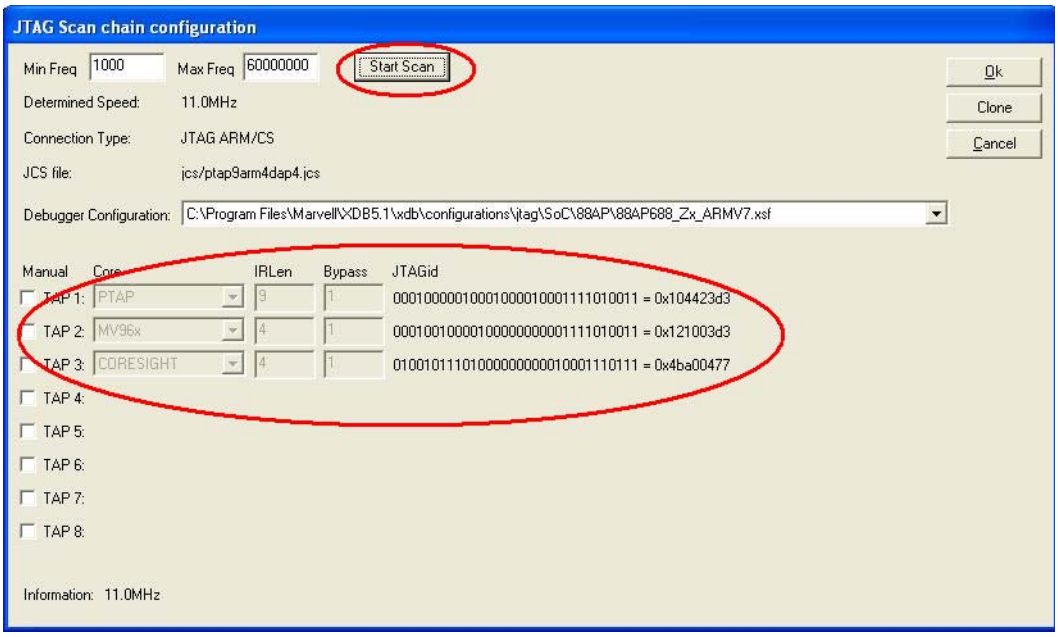
1. With the Marvell eXtreme Debugger (XDB) running, click the **JTAG Scan** button on the Startup screen.

Figure 1: XDB Debugger JTAG Scan Button



2. Click the **Start Scan** button.
   Figure 2 shows a scan chain that indicates the core mode is in ARMv7 mode.
   - Three Test Access Ports (TAPs) indicate the core is in ARMv7 mode.
   - Four TAPs indicate the core is in ARMv6 mode

Figure 2: Start Scan Button and JTAG Scan Chain



2.1.2 Burning the WTM Firmware, OEM Boot Module, and U-Boot Binaries to NAND Using JTAG

Perform the steps that follow to burn the WTM firmware, OEM boot module (OBM), and U-Boot binaries to NAND flash memory using JTAG.

Note You need the Marvell eXtreme Debugger, version 5.1.018 or higher. Download the Marvell eXtreme Debugger from the Marvell Extranet or contact your Marvell representative.

1. Use the Marvell eXtreme Debugger (XDB), version 5.1.018 and the appropriate configuration file for the steps that follow:
   - If the processor boots in the ARMv6 mode, use the `pxa688 _a0_armv6.xsf` file.
   - If the processor boots in the ARMv7 mode, use the `pxa688 _a0_armv7.xsf` file.

   Leave all settings at their default settings.

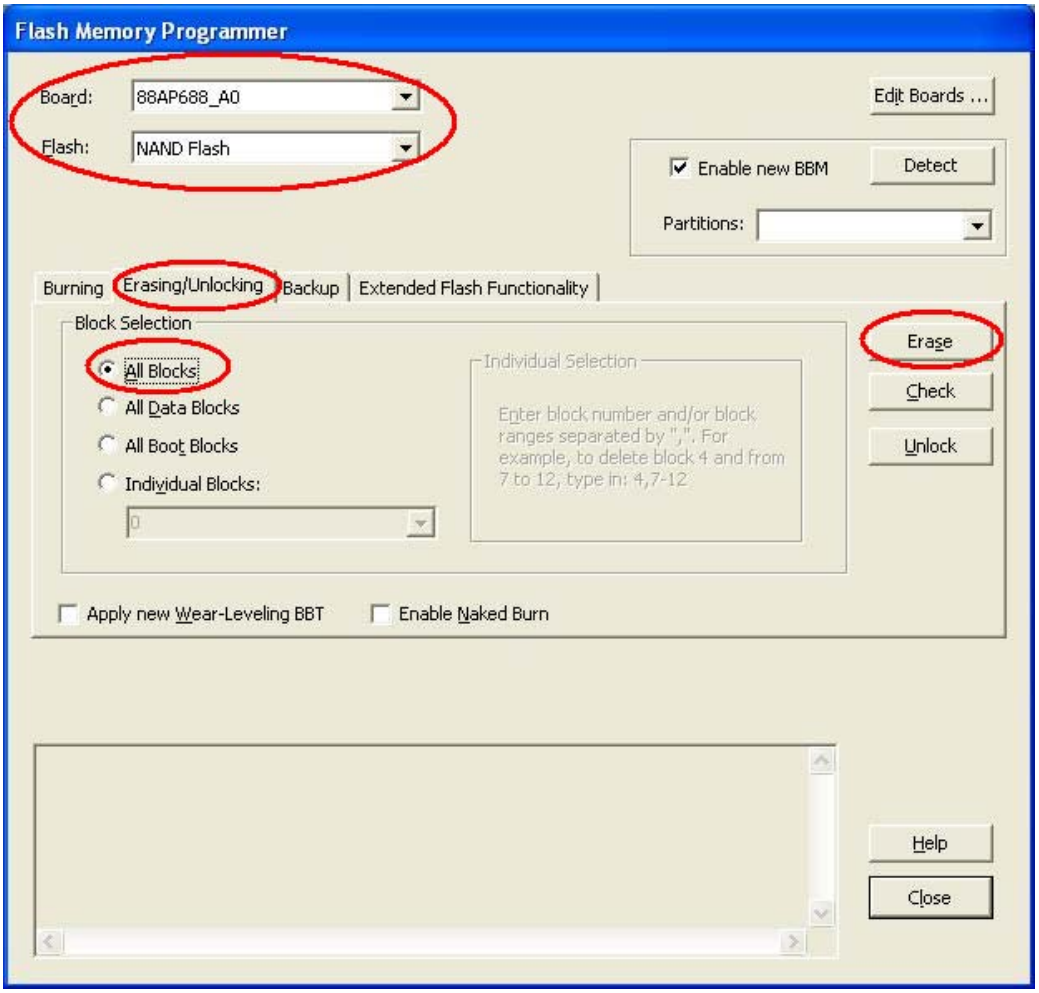
Note To identify in which mode your processor boots, see Section 2.1.1, Identifying an ARMv6 or ARMv7 Mode Boot, on page 19.

2. With XDB running, select **Flash** on the toolbar menu.
3. Click **Burn Flash** on the drop-down menu.
4. In the **Board** field, select **88AP688_A0**, and in the **Flash** field, select **NAND Flash** (see Figure 3, NAND Flash Options).

Note If you are using the 88AP688 Z0/Z1 stepping processor, select **88AP688_ZX** in the **Board** field instead.

5. Select the **Erasing/Unlocking** tab and click the **All Blocks** option button. Click the **Erase** button to erase all blocks.

Figure 3: NAND Flash Options



6. Select the **Burning** tab (see Figure 4). Burn the following files to the addresses as follows:

```
ntim_mmp2_nand_bbu_ddr.bin        --> 0x0
MMP2_NTLOADER_3_2_17.bin          --> 0x80000
WtmUnresetPJ4.bin                 --> 0xc0000
u-boot.bin                        --> 0x100000
```

The `WtmUnresetPJ4.bin` image is a loadable WTM kernel firmware binary image that is executed by the ARMADA 610 on-chip secure processor. This image provides the cryptographic services for both Federal Information Processing Standard (FIPS) and non-FIPS mode operations.

To acquire the image, download the `WTM_Firmware_ARMADA610_2.1.3.zip` file from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ARMADA 610 (MMP2)/Software/WTM/Version 2.1.3*.

Once downloaded, extract both `wtm_rel_mmp2_virtualOTP_2.1.3.bin` and `wtm_rel_mmp2_realOTP_2.1.3.bin` images from the zip file. Both binary images support the same set of WTM primitive functions with the same API definition. However, the `wtm_rel_mmp2_virtualOTP_2.1.3.bin` binary image performs the device RKEK/EC521-DK provision with primitives using the buffers within the secure SRAM to emulate the provisioning over the FUSE/OTP macro. On the other hand, the `wtm_rel_mmp2_realOTP_2.1.3.bin` image performs the device key provision with primitives directly operating over the FUSE/OTP macro. With real OTP operation, the performed platform provision becomes permanent.

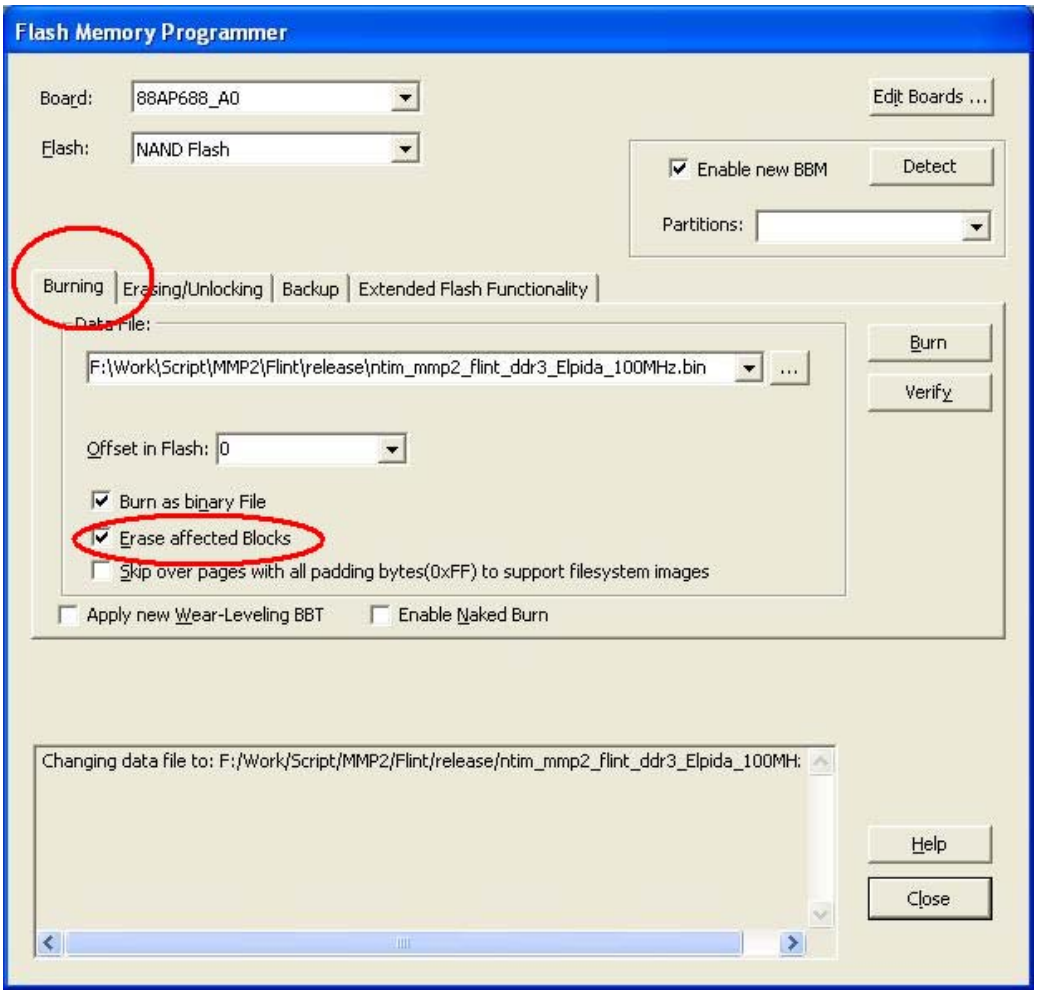
It is recommended to use the virtual OTP version of the WTM kernel binary image for platform software development. To do this, change the binary image file name `wtm_rel_mmp2_virtualOTP_2.1.3.bin` to `WtmUnresetPJ4.bin`.

It is recommended to use the real OTP version of the WTM kernel binary image for the device that is ready to be deployed as a product. To do this, change the binary image file name `wtm_rel_mmp2_realOTP_2.1.3.bin` to `WtmUnresetPJ4.bin`.

Note

Uncheck the **Erase affected blocks** check box when burning `MMP2_NTLOADER_3_2_17.bin`, `WtmUnresetPJ4.bin`, `u-boot.bin`.

Figure 4: NAND Flash Burning Options



2.1.3 Burning the WTM Firmware, OBM, and U-Boot Binaries to eMMC Using JTAG

Perform the steps that follow to burn the WTM firmware, OBM and U-Boot binaries to eMMC flash memory using JTAG:

Note You need the Marvell eXtreme Debugger, version 5.1.018 or higher. Download the Marvell eXtreme Debugger from the Marvell Extranet or contact your Marvell representative.

1. Use the Marvell eXtreme Debugger (XDB), version 5.1.018 or higher and the appropriate configuration file for the steps that follow:
   - If the processor boots in the ARMv6 mode, use the `pxa688 _a0_armv6.xsf` file.
   - If the processor boots in the ARMv7 mode, use the `pxa688 _a0_armv7.xsf` file.

   Leave all settings at their default settings.

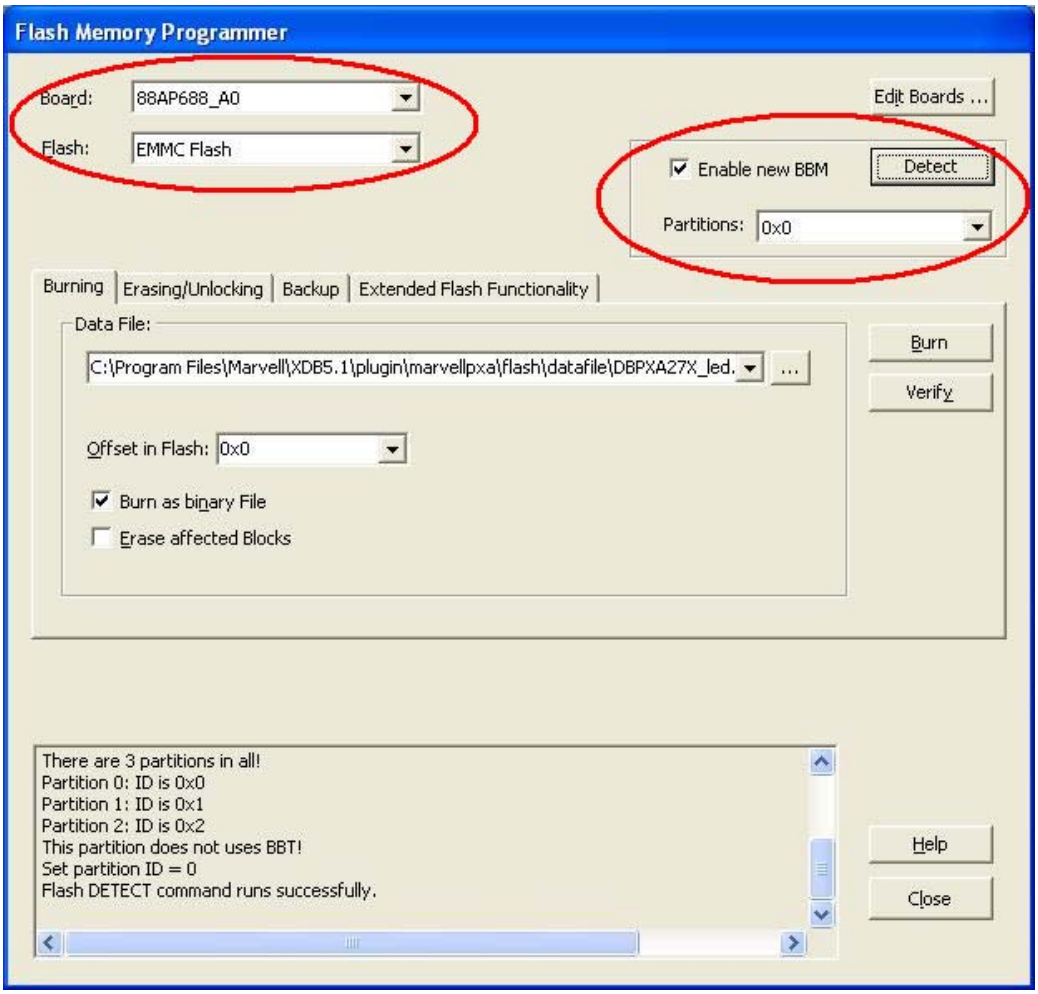
Note To identify in which mode that your processor boots, see Section 2.1.1, Identifying an ARMv6 or ARMv7 Mode Boot, on page 19.

2. Burn OBM and U-Boot into eMMC flash memory using XDB:
   a) In XDB, select **Flash** on the toolbar menu. Click **Burn Flash** on the drop-down menu.
   b) In the **Board** field, select **88AP688_A0**, and in the **Flash** field, select **EMMC Flash** (see Figure 5 on page 27).

Note If you are using the 88AP688 Z0/Z1 stepping processor, select **88AP688_ZX** in the **Board** field instead.

   c) Check the **Enable new BBM** check box and click the **Detect** button to detect the eMMC partition (see Figure 5).

Figure 5: eMMC Flash Detect eMMC Partition



d) Select partition **0x1**. Burn the following binary files to the addresses as follows:

```
ntim_mmp2_nand_bbu_ddr.bin          --> 0x0
partition.bin                       --> 0x600
WtmUnresetPJ4.bin                   --> 0x1000
MMP2_NTLOADER_3_2_17.bin            --> 0x32000
u-boot.bin                          --> 0x52000
```

The `WtmUnresetPJ4.bin` image is a loadable WTM kernel firmware binary image that is executed by the ARMADA 610 on-chip secure processor. This image provides the cryptographic services for both Federal Information Processing Standard (FIPS) and non-FIPS mode operations.

To acquire the image, download the `WTM_Firmware_ARMADA610_2.1.3.zip` file from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ARMADA 610 (MMP2)/Software/WTM/Version 2.1.3*.

Once downloaded, extract both `wtm_rel_mmp2_virtualOTP_2.1.3.bin` and `wtm_rel_mmp2_realOTP_2.1.3.bin` images from the zip file. Both binary images support the same set of WTM primitive functions with the same API definition. However, the `wtm_rel_mmp2_virtualOTP_2.1.3.bin` binary image performs the device RKEK/EC521-DK provision with primitives using the buffers within the secure SRAM to emulate the provisioning over the FUSE/OTP macro. On the other hand, the `wtm_rel_mmp2_realOTP_2.1.3.bin` image performs the device key provision with primitives directly operating over the FUSE/OTP macro. With real OTP operation, the performed platform provision becomes permanent.

It is recommended to use the virtual OTP version of the WTM kernel binary image for platform software development. To do this, change the binary image file name `wtm_rel_mmp2_virtualOTP_2.1.3.bin` to `WtmUnresetPJ4.bin`.

It is recommended to use the real OTP version of the WTM kernel binary image for the device that is ready to be deployed as a product. To do this, change the binary image file name `wtm_rel_mmp2_realOTP_2.1.3.bin` to `WtmUnresetPJ4.bin`.

Note

Select these binary files from the eMMC subdirectory rather than from the NAND subdirectory since there are differences between the NAND and eMMC versions.

2.1.4 Burning Android on eMMC Using U-Boot

Use the U-Boot commands that follow to tftp `zImage.android`, `mbr`, `ramdisk_exr2.img`, `system_ext.img`, and `userdata_ext2.img` to eMMC flash memory.

Note

The following procedures require the connection of the USB cable for USB Ethernet between the host Linux PC and the ARMADA 610 Tablet Reference Platform.

After the first TFTP command is issued, use the ifconfig utility to set the host side USB0 connection to `192.168.1.100`. The Network File System (NFS) server is fixed as `192.168.1.100:/nfs/android`. If you want to change it, modify `vendor/marvell/brownstone/rootdir/rdinit` and build the kernel again.

You do not see the USB0 Ethernet interface on the Linux host until AFTER the first TFTP “t” command is issued from U-boot.

First, burn zImage:

```
MMP2       --> mmc sw_part 0
MMP2       --> sw
MMP2       --> t zImage.android
MMP2       --> mmc write 0x4c00 0x2000 0x1100000
```

Second, burn mbr, ramdisk, system, and userdata images:

```
MMP2       --> t mbr
MMP2       --> mmc write 0x4800 0x1 0x1100000
MMP2       --> t ramdisk_ext3.img
MMP2       --> mmc write 0x8C00 0x4000 0x1100000
MMP2       --> t system_ext3.img
MMP2       --> mmc write 0xCC00 0x3C000 0x1100000
MMP2       --> t userdata_ext3.img
MMP2       --> mmc write 0x48C00 0x4B000 0x1100000
```

Note

When you are done, power off the ARMADA 610 Tablet Reference Platform and power it on again to boot from eMMC.

Rebooting by pressing the reset button does not work.

2.2 Use the Patch-based Source Code

The source code package includes the code for the kernel, U-Boot, Android, and everything needed to boot Android on an ARMADA 610 Tablet Reference Platform with the ARMADA 610 processor with A0 stepping and 1GB of DDR3 memory.

The Android code is provided as a group of patches based on a certain version of Android source code. A manifest file is provided to download that version of Android code from the Android Open Source Project (AOSP) at http://source.android.com.

The kernel and U-Boot are provided as a tar ball of base code and a tar ball of patches Marvell made on it.

2.2.1 Set Up the Android Working Directory

Use the steps that follow to set up the code base.

Note

Check the version of your Git. You can do this by typing `git version`. If the Git version is 1.6.x.x. You can go ahead with it.

If the Git version is 1.7.1.x or later, open the `~/.gitconfig` file and add the following section:

```
[am]
    keepcr=true
```

If the Git version is 1.7.0.x, upgrade your Git to a version later than 1.7.1.x. You can download the package from http://git-scm.com/download.

1. Go to http://source.android.com to download the "repo" tool and set up the build environment for Android.
2. Create the Android working directory and download the initial code base.

```
$ mkdir <android_working_dir>
$ cd <android_working_dir>
$ repo init -u git://android.git.kernel.org/platform/manifest -b master
$ repo sync
```

3. Switch the code base specified by `marvell_manifest.xml`.

```
$ cp <installed_source_dir>/marvell_manifest.xml .repo/manifests/
$ repo init -m marvell_manifest.xml
$ repo sync
```

4. Apply the Marvell patches:

```
$ cd <installed_source_dir>
$ ./setup_android.sh <android_working_dir>
```

2.2.2 Building the Source Code

Use the steps that follow to build Android.

1. Edit `vendor/marvell/brownstone/BoardConfig.mk`; set `BOARD_ENABLE_HELIX` to false, as Helix is not included in the package.
2. Build the kernel and modules:

```
$ cd <android_working_dir>
$ cd kernel
$ make all
```

Note The location of the zimage is at `kernel/out/`.

3. Build android:

```
$ cd <android_working_dir>
$ . build/envsetup.sh
$ chooseproduct brownstone
$ export ANDROID_PREBUILT_MODULES=kernel/out/modules/
$ make -j4
```

Note The location of the Android `mbr`, `ramdisk_ext3.img`, `system_ext3.img`, and `userdata_ext3.img` files are at `out/target/product/brownstone`.

4. Build U-Boot and OBM:

   Before building U-Boot and OBM, extract `wtm_rel_mmp2_virtualOTP_2.1.3.bin` from `WTM_Firmware_ARMADA610_2.1.3.zip` which you can get from the Marvell Extranet. Rename it as `WtmUnresetPJ4.bin` and copy it to `boot/obm/binaries`.

Note The WTM firmware image must be downloaded separately from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ARMADA 610 (MMP2) Software/WTM/Version 2.1.3*. Contact your Marvell representative if you have issues about the download.

   Issue the following commands:

```
$ cd <android_working_dir>
$ cd boot
$ make all
```

Note The `u-boot.bin` file is at `boot/out`, the nand auto boot version OBM is located at `boot/out/nand`, the eMMC auto boot version OBM is located at `boot/out/emmc`.

2.2.3 Using the Optimized OpenSSL for Marvell Platforms

An OpenSSL patch is included in this platform release. This patch fixes a SHA384/512 bug in Android OpenSSL, and optimizes the Android OpenSSL cryptographic library for Marvell platforms. By default, the SHA384/512 bug fix is enabled, the optimization is disabled.

To enable the optimization, this patch requires the Marvell Wireless Trusted Platform Service Package (WTPSP) in this platform release. See the WTPSP release notes for information about how to enable it. Contact your Marvell representative if you have any issues with this package.

When the optimization is enabled, the following OpenSSL cryptographic schemes are optimized: SHA1/224/256, MD5 message digest, AES (CBC mode), RC4 and DES (CBC, CBC3 mode).

Note

Whether the optimization is enabled or not enabled, the Android OpenSSL cryptographic API stays unchanged. Thus, applications using the OpenSSL cryptographic library do not need to be modified. However, an application rebuild is required when the optimization is enabled.

To enable Marvell optimization for OpenSSL:

1. Put the Marvell WTPSP middleware library and header file into `<android_working_dir>/external/openssl/crypto/` and rename it as `libwtpsp.a`.
2. Add the following into `external/openssl/include/openssl/opensslconf.h`.

```
#ifndef OPENSSL_MRVL
#define OPENSSL_MRVL /* enable marvell crypto support */
#endif
```

3. Add the following into `vendor/marvell/brownstone/BoardConfig.mk`.

```
USE_MARVELL_CRYPTO := true
```

4. Build the Android system image.

3 Marvell Code Performance Analyzer

The Marvell Code Performance Analyzer v2.3 is supported in this release.

3.1 What's New

- Call Stack sampling data collection and corresponding data analysis
- Remote data collection via the Android Debug Bridge (ADB) for the Android device

3.2 Features

The following sections provide information about the supported and unsupported features for this release.

3.2.1 Supported Features

- Remote data collection via TCP/IP
- Remote data collection via ADB for the Android device
- Target local data collection in a connectionless environment
- Call Stack sampling data collection and corresponding data analysis
- Hotspot sampling data collection and corresponding data analysis
- Counter monitor data collection and corresponding data analysis.
- Real-time counter monitor and post analysis

3.2.2 Unsupported Features

The software development kit (SDK) for dynamic code is not supported in this release.

3.3 System Requirements

This release supports the Marvell Code Performance Analyzer, version 2.3. Download this version from the Marvell Extranet website at http://www.marvell.com/extranet. If you do not have a Marvell Extranet user ID, click on the “register” link at http://www.marvell.com and follow the instructions therein.

3.4 Installation

Before running the data collector, go to the `/system/bin` folder and run the following command to load the kernel driver:

```
$ ./load_mpdc.sh
```

3.5 Known Issues or Limitations

- It is recommended to first turn off the Marvell Scalable Power Management (MSPM). Otherwise, the Marvell Performance Data Collector (mpdc) may not work normally. Use the following commands to turn off MSPM on your target:

  ```
  echo 0 >/sys/power/mspm/mspm
  ```

- When using the command line, if you get the error message "Fail to communicate with daemon:Success", reload your activity and run it to collect the result again.

- When doing remote data collection on the Android target, if the analyzer exits abnormally, the status of the mpdc_svr may still be connected. If this happens, restart the mpdc_svr on the target.
- If the samples/second value is set too large, the events/sample value is adjusted after a calibration to a smaller value, which makes the interrupt happen more frequently. If this happens, the system becomes busy and the mpdc stops after the expected duration is expired. In addition, most of the samples will be located on the process "mpdc_d". In this case, create your activity with a decreased samples/second value and start it.
- When using the command line, multiple activities cannot be started at the same time.
- If you want to make mpdc_svr listen to another port through using the command "mpdc_svr -p <PortNumber>" on the Android target, a segmentation fault occurs.
- Do not set OS_TIMER as the event in the Event-Based Sampling (EBS).

4 Known Issues

This section describes the Known Issues for this version of the ARMADA 610 Tablet Reference Platform for Android 2.2.

4.1 Fixed Issues

The following issues are fixed in this release.

Problem: [MMP2PV-188] When enabled, the Marvell Scalable Power Management and Dynamic Voltage Frequency Management cause kernel panic.

Workaround: None.

Problem: [MMP2PV-199] The HDMI extended display identification data (EDID) 480p setting does not work.

Workaround: None.

Problem: [MMP2PV-291] The ARMADA 610 Tablet Reference Platform may hang if a suspend operation is performed again after a USB cable is inserted to resume the system from the suspend status.

Workaround: None.

Problem: [MMP2PV-311] The Bluetooth function may not work.

Workaround: Enable the Wi-Fi mode first before you enable the Bluetooth mode.

Problem: [MMP2PV-314] The ARMADA 610 Tablet Reference Platform crashes if it is suspended during Wi-Fi data transmission.

Workaround: None.

Problem: [MMP2PV-324] The touch function may have a slow response in the ARMADA 610 Tablet Reference Platform.

Workaround: None.

Problem: [MMP2PV-340] The audio output fails after resuming back from suspend mode on the ARMADA 610 Tablet Reference Platform.

Workaround: None.

Problem: [MMP2PV-354] Continuous VDMA rotation leads to the VDMA working abnormally.

Workaround: None.

Problem: [MMP2PV-364] The system cannot detect the SD card properly sometimes after a replug operation.

Workaround: Reboot the system.

Problem: [MMP2PV-365] The system cannot recognize the SD card after a format operation.

Workaround: Reboot the system.

Problem: [MMP2PV-369] The camera application sometimes exits automatically.

Workaround: Restart the camera application.

Problem: [MMP2PV-376] The ARMADA 610 Tablet Reference Platform can hang if the user plugged in the USB cable at the same time as when the Android home menu displayed.

Workaround: None.

Problem: [MMP2PV-377] The ARMADA 610 Tablet Reference Platform sometimes enters the "safe mode" after it boots.

Workaround: Reboot the system.

Problem: [MMP2PV-413] The system can hang after a still image capture operation in the camera application.

Workaround: None.

Problem: [MMP2PV-415] The camera preview display sometimes shows as a black preview.

Workaround: Restart the camera application.

Problem: [MMP2PV-430] The ARMADA 610 Tablet Reference Platform can hang after recording several video clips with the camcorder.

Workaround: Restart the platform.

4.2 Not Fixed Issues

The following issues have not been fixed in this release.

Problem: [MMP2PV-143] The Color Management Unit (CMU) preset mode and picture-in-picture (PIP) feature do not work.

Workaround: None.

Problem: [MMP2PV-292] The system may hang during the boot phase. (The failure rate is about 5 percent.)

Workaround: None

Problem: [MMP2PV-384] The Vmeta-based H.264 encoder causes errors in other Vmeta-based codec in multi-instance usage.

Workaround: None.

Problem: [MMP2PV-448] A USB High Speed Inter-Chip (HSIC) write to a USB drive does not work.

Workaround: None.

4.3 New Issues

The following issues are new in this release.

Problem: [MMP2PV-431] The camcorder application does not respond if the user stops the video record process in 1 second.

Workaround: Reboot the system.

Problem: [MMP2PV-452] Wi-Fi WPA/WPA2 enterprise mode does not work.

Workaround: None.

Problem: [MMP2PV-484] The system can hang if the user presses the platform ON button during the camera preview mode.

Workaround: None.

Problem: [MMP2PV-485] Some videos cannot switch to HDMI display as they failed in the seek operation.

Workaround: None.

Problem: [MMP2PV-486] The audio output fails to switch to the HDMI in the video player.

Workaround: None.

THIS PAGE INTENTIONALLY LEFT BLANK

A Revision History

Date	Revision	Description
January 5, 2011	A	Changes in this document: Section 1, Overview • Section 1, Overview, on page 5, updated the “Beta1” release description to “Beta2.” • Section 1.1.1, Hardware Requirements, on page 5, Added ECO requirements. • Section 1.2.1, Differences from the Previous Release (Beta 1), on page 6, added a description of the differences between the Beta 2 and Beta 1 releases. • Section 1.2.2.1, Beta 1 Release, on page 7, moved this description under Section 1.2.2, Differences for Other Previous Releases, on page 7 and added download information for the WTM firmware upgrade. • Section 1.3, Platform Features, on page 11, corrected the platform name in the introductory sentence. • Section 1.4, Multimedia Features, on page 13, updated status information. • Section 1.5, Board Support Package Features, on page 14, removed “U-BOOT from the introductory sentence; removed the image rotation feature; added FM audio and CyWee motion sensor features and added a power-on sequence to address I2C stability. • Section 1.6, Release Package Contents, on page 16, updated the file name for prebuilt bin binaries in Table 4 and source code tarball in Table 5. Section 2, Installation • Section 2.1.2, Burning the WTM Firmware, OEM Boot Module, and U-Boot Binaries to NAND Using JTAG, on page 22, updated the section title by adding “WTM Firmware” and added WtmUnresetPJ4.bin image information. • Section 2.1.3, Burning the WTM Firmware, OBM, and U-Boot Binaries to eMMC Using JTAG, on page 26, updated the section title by adding “WTM Firmware” and added WtmUnresetPJ4.bin image information. • Section 2.2.2, Building the Source Code, on page 31, added a note with downloading information for the WTM firmware image. Section 3, Marvell Code Performance Analyzer • Section 3.4, Installation, on page 33, updated the folder name and command to load the kernel driver. Section 4, known Issues • Section 4, Known Issues, on page 35, updated.
November 2010	-	Initial release.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

Marvell Semiconductor, Inc.
5488 Marvell Lane
Santa Clara, CA 95054, USA

Tel: 1.408.222.2500
Fax: 1.408.988.8279

www.marvell.com

Marvell. Moving Forward Faster