ARMADA™ 610 Tablet Reference Platform

for Android™ 2.3, Linux® Kernel 2.6.35

Software Release Notes

Doc. No. MV-S302040-00, Rev. -

February 24, 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

THIS PAGE INTENTIONALLY LEFT BLANK

1 Overview

This software release package contains source code for the Alpha 1 version of the Marvell® ARMADA™ 610 Tablet Reference Platform for Android™ 2.3, Linux® kernel 2.6.35.

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/A2 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
Revision 4 • PB number: F00055-400 • ECO level: ECO 1 (0x1)	*Marvell® ARMADA™ 610 Tablet Reference Platform, Revision 4 Engineering Change Orders*	MV-S501286-00

The printed board (PB) number F00055 identifies the board as a Marvell ARMADA 610 Tablet Reference Platform. The -100, -200, -300, and -400 indicate revision 1, revision 2, revision 3, and revision 4, 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 2.3, Linux kernel 2.6.35

1.2 Platform Features

Table 2 lists the platform features for this version of the Marvell ARMADA 610 Tablet Reference Platform for Android 2.3 release package.

Table 2: Platform Features (Sheet 1 of 3)

Features		Support
General	Android Version	2.3
	Linux kernel	2.6.35
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 8
	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.

1.3 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	Stagefright	--	Ready
		MP3	MPEG-4			Not Ready
MPEG-2 PS	.mpg	MP3	MPEG-2	GStreamer	--	Ready
3GPP	.3gp/.3gpp	MP3	H.264	Stagefright	--	Not Ready
		AAC	MPEG-4			Ready
MKV	.mkv	MP3	H.264	Stagefright	--	Not Ready
		AAC	H.264			
AAC	.aac	AAC	No video	GStreamer	--	Not Ready
	.adts	AAC	No video			
MP3	.mp3	MP3	No video	Stagefright	--	Ready

1.4 Board Support Package Features

Board support package features for the ARMADA 610 Tablet Reference Platform for Android 2.3 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 2.6.35
  - 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 10)
  - 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, 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.

1.5 Release Package Contents

The following tables list and describe the release package for the ARMADA 610 Tablet Reference Platform for Android™ 2.3, Linux kernel 2.6.35.

Table 4: Prebuilt Binary Files

Files	Description
ARMADA610_ANDROID_PLATFORM_ALPHA1_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 2.1.5*. Contact your Marvell representative if you have issues about the download.

.

Table 5: Source Files

Files	Description
ARMADA610_ANDROID_PLATFORM_ALPHA1_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).

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 1.5, Release Package Contents, on page 11 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.

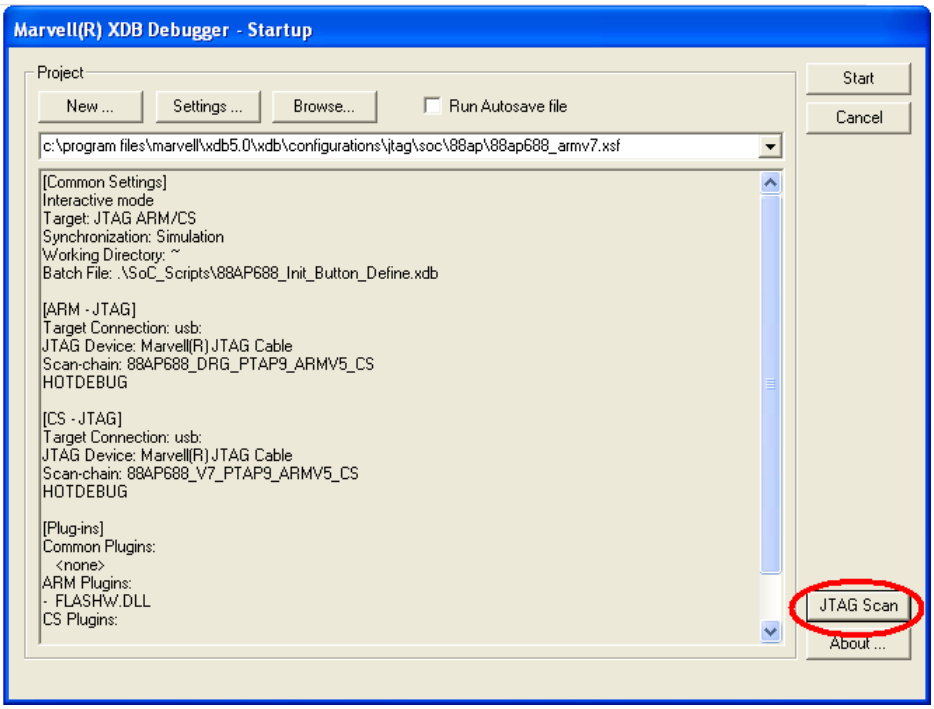
To identify whether the processor is in ARMv6 or ARMv7 mode, use the steps that follow.

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.

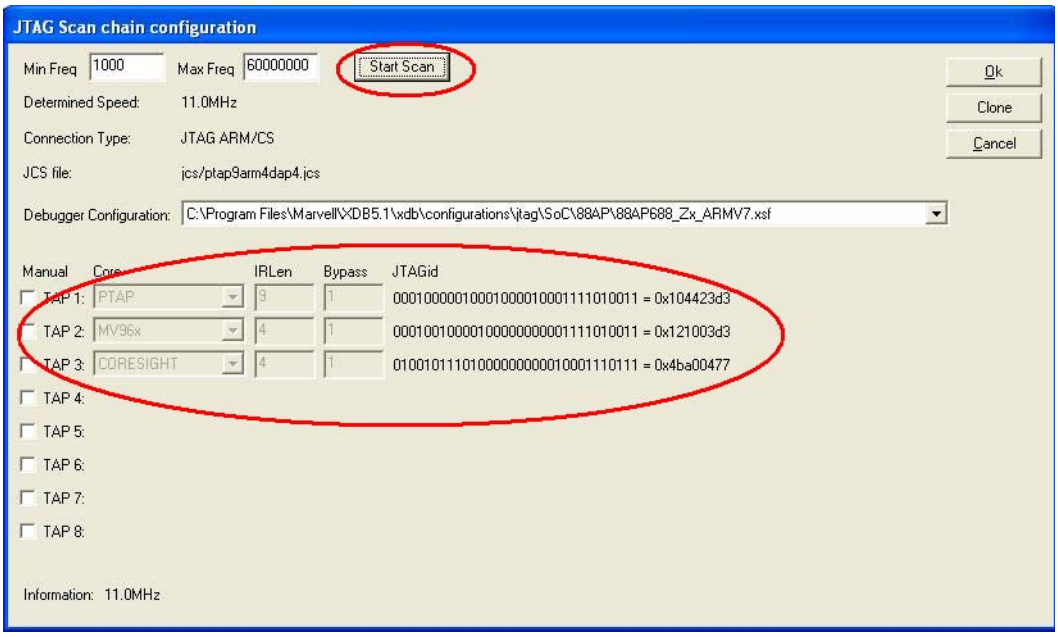
1. With the Marvell eXtreme Debugger (XDB) running, click the **JTAG Scan** button on the Startup screen.

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.

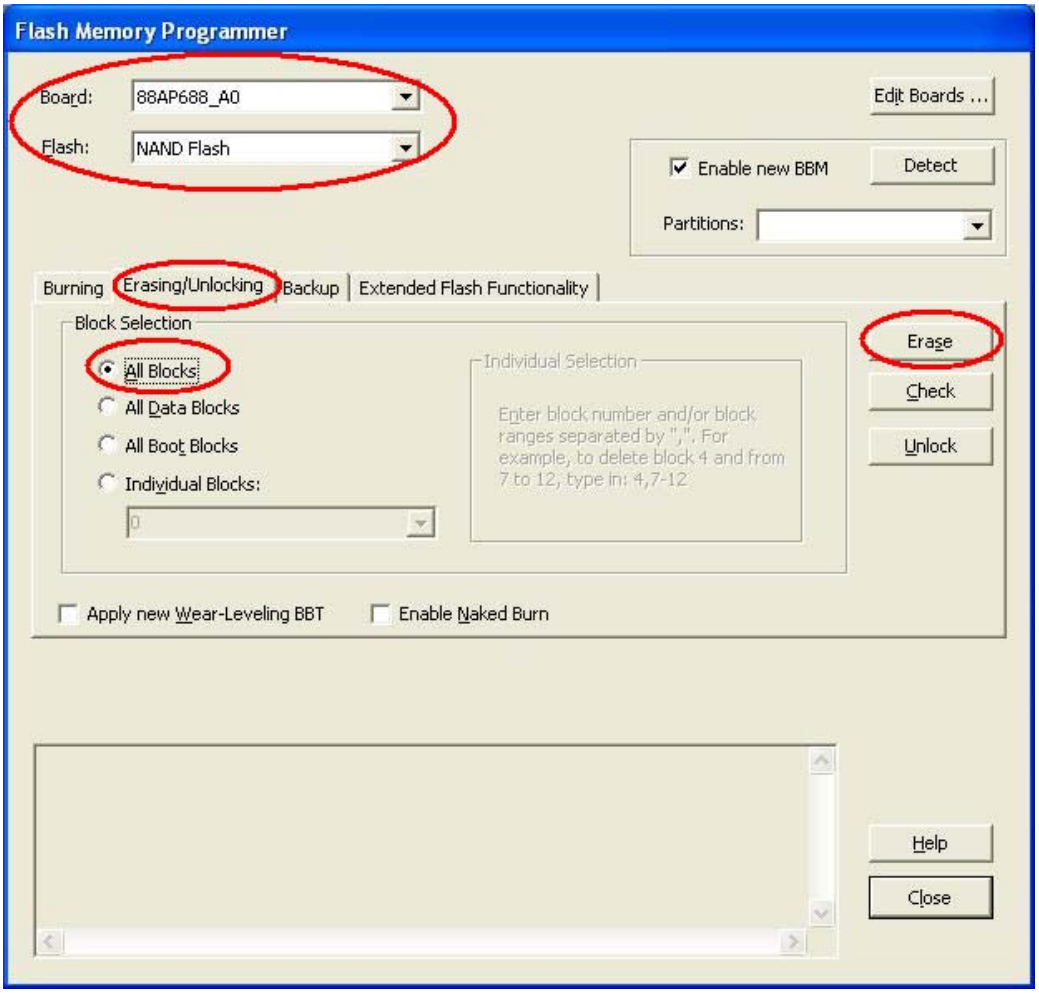
Note To identify in which mode your processor boots, see Section 2.1.1, Identifying an ARMv6 or ARMv7 Mode Boot, on page 13.

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
Wtm_rel_mmp2.bin                  --> 0xc0000
u-boot.bin                        --> 0x100000
```

The `Wtm_rel_mmp2.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_2.1.5.zip` file from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ ARMADA 610 (MMP2)/Software/WTM/Version 2.1.5*.

Once downloaded, extract both `wtm_rel_mmp2_virtualOTP_2.1.5.bin` and `wtm_rel_mmp2_realOTP_2.1.5.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_2.1.5.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_2.1.5.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.

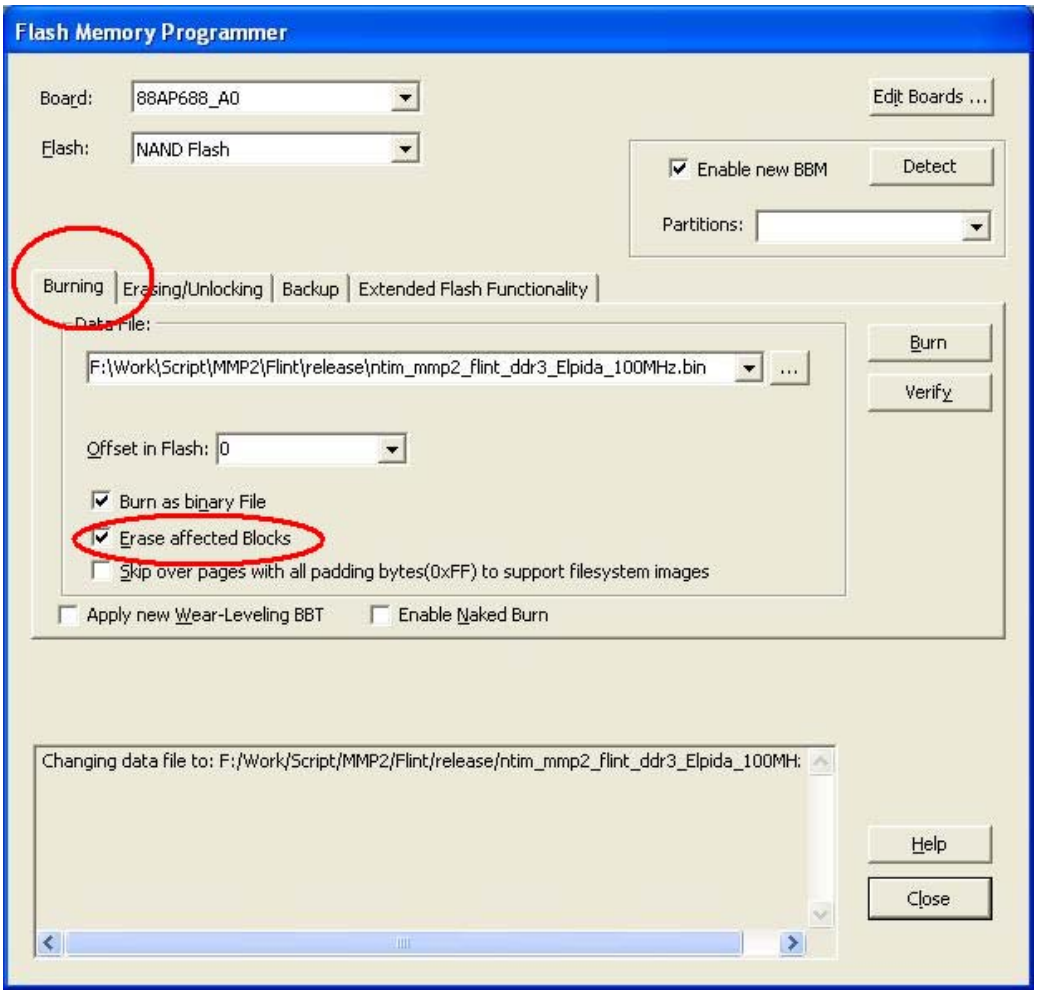
It is recommended to use the virtual OTP version of the WTM kernel binary image for platform software development. To do this, change the binary image file name `wtm_rel_mmp2_virtualOTP_2.1.5.bin` to `Wtm_rel_mmp2.bin`.

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_2.1.5.bin` to `Wtm_rel_mmp2.bin`.

Note

Uncheck the **Erase affected blocks** check box when burning `MMP2_NTLOADER_3_2_17.bin`, `Wtm_rel_mmp2.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.

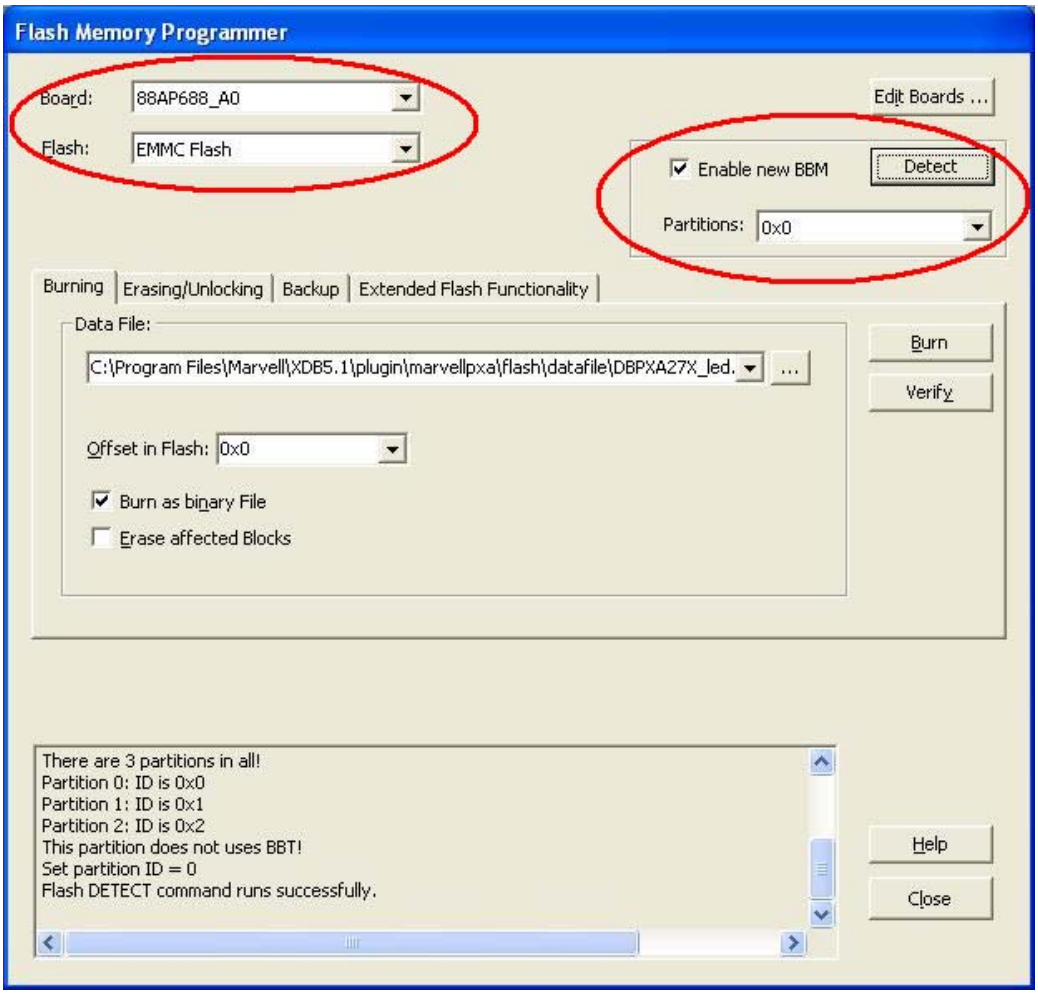
Note To identify in which mode that your processor boots, see Section 2.1.1, Identifying an ARMv6 or ARMv7 Mode Boot, on page 13.

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 21).

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
Wtm_rel_mmp2.bin                    --> 0x1000
MMP2_NTLOADER_3_2_17.bin            --> 0x32000
u-boot.bin                          --> 0x52000
```

The `Wtm_rel_mmp2.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_2.1.5.zip` file from the Marvell Extranet at *My Products/Cellular & Handheld Solutions/Applications Processors/ ARMADA 610 (MMP2)/Software/WTM/Version 2.1.5*.

Once downloaded, extract both `wtm_rel_mmp2_virtualOTP_2.1.5.bin` and `wtm_rel_mmp2_realOTP_2.1.5.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_2.1.5.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_2.1.5.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_2.1.5.bin` to `Wtm_rel_mmp2.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_2.1.5.bin` to `Wtm_rel_mmp2.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 Setting 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_2.1.5.bin` from `WTM_Firmware_ARMADA610_2.1.5.zip` which you can get from the Marvell Extranet. Rename it as `Wtm_rel_mmp2.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 2.1.5*. 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`.

```
#ifdef _ARM_
#ifndef OPENSSL_MRVL
#define OPENSSL_MRVL /* enable marvell crypto support */
#endif
#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 the Alpha 1 version of the ARMADA 610 Tablet Reference Platform for Android 2.3.

Problem: [MMP2PV-237] Copying a file about the size of 8MB to a USB drive causes an IO error. (Reading a file from a USB drive is OK.)

Workaround: Use a USB OTG interface instead of a USB High-Speed Inter-Chip (HSIC) interface.

Problem: [MMP2PV-416] A kernel error occurs when calling the ioctrl of streaming_ off after a polling call timeout occurs 3 times.

Workaround: None

Problem: [MMP2PV-419] The system cannot resume from suspend mode during a 2D graphics controller (GC2D) multi-thread operation.

Workaround: None

Problem: [MMP2PV-455] The system cannot enter a suspend mode after a resume from suspend mode stress test via a Real Time Clock (RTC) wakeup source after 2 hours.

Workaround: None

Problem: [MMP2PV-474] When a binder error occurred during the Graphics Abstract Layer 2D (GAL2D) stress test and LCD+SD+WIFI+USB stress testing, the system was unable to restore the Android GUI.

Workaround: None

Problem: [MMP2PV-478] The TouchPerf test can produce about 58 events per second on the Marvell PXA920 platform, while only producing about 22 events per second on the ARMADA 610 Tablet Reference Platform.

Workaround: None

Problem: [MMP2PV-483] An I2C read/write operation via Audio/Light Sensor/Camera with a resume from suspend mode causes the system to hang.

Workaround: None

Problem: [MMP2PV-497] A JPEG capture does not work. (There is no data output at the polling call; the YUV format works.)

Workaround: None

Problem: [MMP2PV-499] The screen displays abnormally with flickering and some stripes.

Workaround: None

Problem: [MMP2PV-507] The system hangs when it resumes from being in suspend mode longer than 4 hours during SD/eMMC read/write operations.

Workaround: None

Problem: [MMP2PV-510] Running 15 threads of read/write eMMC and audio/video playback (720P H.264) causes the system to hang.

Workaround: None

Problem: [MMP2PV-516] When running the Wi-Fi iPerf stress test longer than 4 hours, a resume form suspend mode causes kernel panic.

Workaround: None

Problem: [MMP2PV-541] Some video clips cannot be switch to HDMI display.

Workaround: None

Problem: [MMP2PV-542] In some streams, audio cannot be switched back from HDMI to the target.

Workaround: None

Problem: [MMP2PV-543] The first few frames (about 2 seconds) do not play smoothly after switching some streams to HDMI.

Workaround: None

Problem: [MMP2PV-545] The system cannot enter suspend mode after performing a power test.

Workaround: None

Problem: [MMP2PV-546] eMMC/SD read/write performance on the Alpha 1 version of the platform with Android 2.3 degrades more than10% compared with the Beta 2.1 RC version of the platform with Android 2.2.

Workaround: None

Problem: [MMP2PV-547] The camcorder function does not work.

Workaround: None

Problem: [MMP2PV-549] The system cannot enter suspend mode when the on-key is pressed during the video playback.

Workaround: None

Problem: [MMP2PV-551] The 720p/1080p UYVY format preview displays with streaks.

Workaround: None

Problem: [MMP2PV-554] The IOZONE read/write performance test on eMMC causes the system to hang.

Workaround: None

Problem: [MMP2PV-555] The camera preview screen in the camera application sometimes hangs. (The failure rate is 10 percent.)

Workaround: None

Problem: [MMP2PV-567] Wi-Fi WPA2 TCP throughput expectations failed.

Workaround: None

Problem: [MMP2PV-568] Some audio track tests hang after an "AudioTrack.write" call.

Workaround: None

Problem: [MMP2PV-569] Only some channels (about 3 or 4) can be scanned in the ARMADA 610 Tablet Reference Platform.

Workaround: None

Problem: [MMP2PV-570] The FM application sometimes does not respond when adjusting the channel.

Workaround: None

Problem: [MMP2PV-573] Some video clips require 10 to 15 seconds to switch the video to HDMI display.

Workaround: None

Problem: [MMP2PV-574] Some video clips have no audio output during video playback through HDMI display.

Workaround: None

Problem: [MMP2PV-575] Some video clips cannot play when HDMI is connected.

Workaround: None

Problem: [MMP2PV-587] Copying files between the WIFI NFS and USB Ethernet NFS stress tests causes the system to hang.

Workaround: None

Problem: [CQ00129534] Gallery 3D loads images very slowly; if there are more than 100 images in one folder, some pictures require about 1 minute to load.

Workaround: None

Problem: [CQ00138514] Tank Recon 3D.apk does not run on the ARMADA 610 Tablet Reference Platform.

Workaround: None

Problem: [CQ00138997] GPUBench-1.0.0 does not run on the ARMADA 610 Tablet Reference Platform.

Workaround: None

Problem: [CQ00140959] While running Talking Tom Cat and launching Android Market, pressing the button to "add 4 extra animations" forces Talking Tom Cat to close.

Workaround: None

Problem: [CQ00146442] Dungeon Defenders cannot run on the ARMADA 610 Tablet Reference Platform.

Workaround: None

Problem: [CQ00147434] Using the BenchmarkSuite_v0.3_20100514.apk (from SamSung), the application auto-exits after running a while.

Workaround: None

Problem: [CQ00148581] In Quadrant-3D, the earth color is wrong.

Workaround: None

THIS PAGE INTENTIONALLY LEFT BLANK

A Revision History

Date	Revision	Description
February 24, 2011	-	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