VadaTech AMC502

Software User Manual

July 01, 2016
Version 2.0.1

THE POWER OF VISION

Notice

While reasonable efforts have been made to assure the accuracy of this document, VadaTech, Inc. assumes no liability resulting from any omissions in this document or from the use of the information obtained herein. VadaTech reserves the right to revise this document and to make changes periodically and the content hereof without obligation of VadaTech to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to the VadaTech Incorporated Web site. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of VadaTech, Inc.

It is possible that this publication may contain reference to or information about VadaTech products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that VadaTech intends to announce such products, programming, or services in your country.

Trademarks

Revision History

Doc Rev	Description of Change	Revision Date
1.0.0	Document created	10/03/2015
2.0.0	Updated document to match latest partition mapping	19/05/2016
2.0.1	Corrected information about repartitioning sizes	01/07/2016

Table of Contents

Tables

1 Document Overview

This document describes the AMC502 board software including the Linux BSP for the iMX6 Quad Core CPU. This document also provides details on how to program AMC502 FPGA image to QSPI flash.

1.1 Document References

- VadaTech AMC502 Datasheet (http://www.vadatech.com)
- VadaTech FMCs User Manual
- PICMG® AMC.0 AdvancedMC Mezzanine Module (http://www.picmg.org)
- PICMG® AMC.1 AdvancedMC PCI Express and AS (http://www.picmg.org)
- PICMG® AMC.2 AdvancedMC Ethernet (http://www.picmg.org)
- PICMG® AMC.4 AdvancedMC Serial RapidIO (http://www.picmg.org)
- Xilinx Kintex-7 Datasheets and User's Guides
- ANSI/VITA 57.1 FPGA Mezzanine Card (FMC) Standard (http://www.vita.com)

1.2 Acronyms Used in this Document

Acronym	Description
AMC	Advanced Mezzanine Card
BAR	Base Address Register
BIST	Built-In Self Test
BPI	Byte Peripheral Interface
BSP	Board Support Package
C2M	Carrier-to-Mezzanine (signal)
CGND	Chassis Ground
CLK	Clock
CPU	Central Processing Unit
DAC	Digital to Analog Converter
DDR3	Dual Data Rate 3 SDRAM
DIP	Dual In-line Package
DMA	Direct Memory Access
DMUX	De-multiplexer
DR	Data Ready
FMC	FPGA Mezzanine Card
FMC10G	Refers to VadaTech FMC106/107
FMC1G	Refers to VadaTech FMC102/103/104/105
FPGA	Field Programmable Gate Array
FRU	Field Replaceable Unit
GbE	Gigabit Ethernet
GND	Signal Ground

GTX	Virtex-6 Gigabit Transceiver
HPC	High Pin Count (FMC connector)
ioctl	Input/Output/Control
IP	Intellectual Property
IPMI	Intelligent Platform Management Interface
JSM	JTAG Switch Module
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LPC	Low Pin Count (FMC connector)
LVCMOS	Low-Voltage Complementary Metal Oxide Semiconductor
LVDS	Low Voltage Differential Signaling
M2C	Mezzanine-to-Carrier (signal)
MAC	Media Access Controller
MB	Megabyte (2^20 bytes)
MIG	Memory Interface Generator
M-LVDS	Multi-point Low Voltage Differential Signaling
mmap	Memory Map
MMC	Module Management Controller (IPMI controller of AMC)
MMIO	Memory Mapped Input/Output
MUX	Multiplexer
n.c.	No connection
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PHY	Physical Layer Device
PICMG	PCI Industrial Computer Manufacturer's Group
PIO	Programmed Input/Output
PLL	Phase Locked Loop
SDRAM	Synchronous Dynamic Random Access Memory
SERDES	Serializer/Deserializer
SGMII	Serial Gigabit Media Independent Interface
SOL	Serial-Over-LAN
SRIO	Serial RapidIO
SSIF	SMBus System Interface
TCLK	Telephony Clock
USB	Universal Serial Bus
VADJ	Adjustable Voltage (power rail)
VIO	I/O Voltage (power rail)
VREF	Reference Voltage (power rail)
XAUI	Ten Gigabit Attachment Unit Interface

Table 1: Acronyms

2 Linux BSP Software Overview

The AMC502 comes pre-configured with a boot loader (U-Boot) and Linux kernel/file system in flash memory.

The CPU RS-232 port can be used to gain access to the serial console. The serial port setup is 115200-8-N-1-NOFLOW.

2.1 Linux Boot Sequence

When the board boots up the sequence is as follows:

1) U-Boot loader starts up and initializes SDRAM, Flash, GbE ports
2) U-Boot script automatically loads/starts the ‘active’ Linux kernel from flash
3) Linux initializes devices and mounts ‘active’ root file system from flash
4) File system init scripts run
5) The Linux login prompt is presented

2.2 Memory Map and Linux Device Nodes

The 8GB NAND Flash (u65) memory map and corresponding Linux device files are as follows:

Address	Size	Part	Description	Linux Device
0x000000000000	128 MB	Nand Flash (U65)	Reserved	/dev/mtd0
0x000008000000	8 MB	Nand Flash (U65)	Reserved	/dev/mtd1
0x000008800000	64 MB	Nand Flash (U65)	Linux Kernel Image A	/dev/mtd2
0x00000c800000	2500 MB	Nand Flash (U65)	Root File System A	/dev/mtd3
0x0000a8c00000	8 MB	Nand Flash (U65)	Reserved	/dev/mtd4
0x0000a9400000	64 MB	Nand Flash (U65)	Linux Kernel Image B	/dev/mtd5
0x0000ad400000	2.500 MB	Nand Flash (U65)	Root File System A	/dev/mtd6
0x000149800000	2920 MB	Nand Flash (U65)	Reserved	/dev/mtd7

Table 2: Nand Flash Memory Map

The Boot Flash (U63) memory map and corresponding Linux device files are as follows:

Address	Size	Part	Description	Linux Device
0x000000000000	512 KB	Boot Flash (U63)	U-boot	/dev/mtd8
0x000000080000	64 KB	Boot Flash (U63)	U-boot Environment A	/dev/mtd9
0x000000090000	64 KB	Boot Flash (U63)	U-boot Environment B	/dev/mtd10
0x000000200000	3456 KB	Boot Flash (U63)	Etc file system	/dev/mtd11

Table 3: Boot Flash Memory Map

The FPGA QSPI Flash (U24) memory map and corresponding Linux device files are as follows:

Address	Size	Part	Description	Linux Device
0x000000000000	64 MB	Boot Flash (U24)	FPGA Image	/dev/mtd12

Table 4: FPGA QSPI Memory Map

2.3 Default IP Addresses

eth0 (AMC Port 0): `192.168.1.252`

These IP addresses can be used to **`ssh`** into the board to get a command prompt or to use **`scp`** for transferring files. To change them edit the **`/etc/rc.d/rc.conf`** file.

2.4 Linux BSP Field Upgrade

The BSP binary release includes a field upgrader package which can be used to upgrade the board from the Linux command prompt. This upgrader uses a ping-pong partition style upgrade using the 'A' and 'B' partitions seen in the memory map above. This ensures a consistent upgrade with minimal risk should a failure occur. But it does mean that any customizations that you've applied to the file system will need to be re-applied after the upgrade except **`/etc`** file system which is standalone partition and all files changed in this location will be preserved during upgrade.

To use the upgrade package, transfer the upgrade tar file to the **`/upgrade`** directory using **`scp`**. Then untar the package using the following command:

```
tar xvzf amc502-bsp-upgrade-releaseX.tgz
```

Then to invoke the upgrade type:

```
./upgrade
```

To verify the BSP version after rebooting the board, type:

```
bsp-version
```

3 Programming BSP via U-Boot

Programming the board via U-Boot is recommended only when some other factor is preventing upgrade via Linux.

In order to program via U-Boot, power up the board and interrupt the auto-boot sequence by pressing ENTER. This will drop you to a U-Boot prompt. Then you will need to setup the U-Boot networking environment.

To set up the U-Boot networking, use the following commands from the U-Boot prompt:

```
=> setenv ipaddr <board ip address>
=> setenv netmask <board netmask>
=> setenv gatewayip <optional gateway router>
=> setenv serverip <tftp server ip>
=> setenv ethact XXX    (XXX is Gem.e000b000, Gem.e000c000)
```

Next, ping through the selected port first to the gateway router (if you have one) and next to the TFTP server to ensure that you have network connectivity.

```
=> ping <gateway router>
=> ping <tftp server>
```

Then proceed on to placing the upgrade images onto your TFTP server and using U-Boot to download them to the board and program the partitions. Use extreme care when entering commands and do not progress from one step to the next if any failure occurs.

WARNING: If the U-Boot binary partition of the flash becomes erased or corrupted and the board is reset/power-cycled before reprogramming a valid image.

3.1 Programming the U-Boot Image

- **run up_uboot**

This will download amc502/u-boot.imx image from TFTP server and reprogram it.

NOTE: If the board does not already have a valid U-Boot image programmed on it then the board will not boot.

3.2 Programming the Linux Kernel Image

- **run up_kernela (for upgrading Kernel-A)**
- **run up_kernelb (for upgrading Kernel-B)**

This will download amc502/uImage image from TFTP server and reprogram it.

3.3 Programming the Root File System

- **run up_roota (for upgrading Rootfs-A)**
- **run up_rootb (for upgrading Rootfs-B)**

This will download amc502/rootfs.ubifs.img image from TFTP server and reprogram it.

3.4 Programming the Etc File System

- **run up_etc**

This will download amc502/ etcfs.jffs2 image from TFTP server and reprogram it.

3.5 Completing an Upgrade via U-Boot

After the necessary partitions are upgraded please issue the **`reset`** command from U-Boot to ensure that the upgrades take effect. The **`bsp-version`** command can be used within Linux to check the BSP version and confirm the success of the upgrade.

4 Programming FPGA image from iMX6 CPU

In order to program FPGA image from iMX6 CPU download FPGA image to **`/upgrade`** directory and do the following:

- `[root@amc502 upgrade]# fpga_upgrade amc502_fpga.bin`

```
NOTE: This upgrade script DOES upgrade/downgrade
the FPGA Configuration in QSPI flash......

Please ensure that you have specified the proper image.
Do you wish to continue with the upgrade (y/n)? y

s25fl512s spi3.0: s25fl512s (65536 Kbytes)
Creating 1 MTD partitions on "s25fl512s":
0x000000000000-0x000004000000 : "s25fl512s"
Writing FPGA image to /dev/mtd12...
Erasing blocks: 12/12 (100%)
Writing data: 2987k/0k (100%))
Verifying data: 2987k/0k (100%))
FPGA Configuration ... SUCCESS
```

The `fpga_upgarde` script will attach QSPI flash to iMX6 CPU, program provided image into it and trigger FPGA to configure itself from the QSPI Flash using the new image. Refer to AMC502 Hardware reference manual for more information about FPGA QSPI Flash Upgrade Sequence.