

Armada 8040 MACHIATOBin U-Boot and ATF

[a8040](#), [u-boot](#), [uboot](#), [bootloader](#)

Introduction

Armada 8040 MACHIATOBin community board uses U-boot wrapper in ATF (ARM Trusted Firmware) in order to boot the board.

Following are build instructions where the target is to build flash-image.bin

Build Instructions

Following are instructions on how to build U-Boot followed by ATF for the Armada 8040 MACHIATOBin community board -

```
git clone git@github.com:MarvellEmbeddedProcessors/u-boot-marvell.git -b u-
boot-2015.01-armada-17.02
git clone git@github.com:MarvellEmbeddedProcessors/atf-marvell.git -b atf-
v1.2-armada-17.02
git clone git@github.com:MarvellEmbeddedProcessors/mv-ddr-marvell.git atf-
marvell/drivers/marvell/mv_ddr/
export CROSS_COMPILER=<path to AARCH64 toolchain>
cd u-boot-marvell
make clean
make mvebu_armada80x0_customer0_mcbin_defconfig
make
cd ..
cd atf-marvell
make PLAT=a80x0_cust clean
make PLAT=a80x0_cust all fip
```

The result image file is at build/a80x0_cust/debug/flash-image.bin

Flashing instructions

The above flash-image.bin can be used to boot from Micro SD, SPI, eMMC and through UART as xmodem

Micro-SD Flashing Instructions

For flashing u-boot on an empty micro SD perform the following on a Linux PC -

- Insert a micro SD into your Linux PC and then run
- `dd if=flash-image.bin of=/dev/sdX bs=512 seek=1`

This will flash u-boot.mmc to the sector after the MBR where the Armada 38x bootrom expects it to

be.

Note - /dev/sdX is where the micro SD is mapped in your Linux machine, change the 'X' to your corresponding mapped volume. If you set /dev/sdX to a wrong volume then you might risk erasing a hard drive or different volume than the designated micro SD.

eMMC Flashing Instructions

Short Summary:

- Booting the device in UART
- Getting up Uboot
- Copy the operationsystem onto a USB-Drive (get it from images.solid-build.xyz)
- Using Uboot to write the operation-files onto the emmc. (emmc write)
- (depends on which board you use - just update the dtb file)

The main different between booting u-boot on eMMC and MicroSD is that in MicroSD u-boot resides in second sector of the SD card, where in eMMC it must reside in first block in either the main eMMC storage or it's first or second boot partitions.

So if the main physical partition of eMMC is used then it will override the partition table. So it's recommended to use the boot partitions for that purpose -

- Boot u-boot via UART
- Load the built u-boot.mmc described above to memory; assuming address 0x02000000. It can be loaded from TFTP (sadly USB does not work at this point)
- Set working partition to be the boot partition #1 by running 'mmc dev 0 1'. You can run 'help mmc' to see different u-boot commands
- Flash u-boot into the FIRST block 'mmc write 0x02000000 0 0x800'. Notice that you can modify 0x800 to the actual u-boot size in 512bytes blocks

TFTP sample:

```
dhcp
tftpboot 0x02000000 192.168.178.68:u-boot.mmc
mmc write 0x02000000 0 ${filesize}
```

Alternatively you can boot Linux and run

- dd if=flash-image.rom of=/dev/mmcblk0boot0

Booting the board using UART xmodem

The Armada 8040 can be booted through UART xmodem. Even if the processor is indicated to boot from SPI, Micro SD or eMMC the bootrom inside the processor first checks if there is a pattern on it's UART RX and decides if to continue booting from the designated boot sources, or use UART for that.

This is mostly used for system manufacturing, unbricking etc...

The flash-image.bin that is built above is a ready to use xmodem protocol transferable image to Armada 8040 processor. But first the processor needs to be redirected from it's default boot vector.

This can be achieved by running the 'download-serial.sh' script that is part of SolidRun Armada 38x u-boot -

<https://github.com/SolidRun/u-boot-armada38x/blob/u-boot-2013.01-15t1-clearfog/download-serial.sh>

An example is as follows -

```
./download-serial.sh /dev/ttyUSB0 flash-image.bin
```

Embedded in download-serial.sh a small C program that gets built every time the script runs and requires curses libraries.

SPI flash flashing instructions

There are multiple ways how to flash the SPI flash on the board. Following is one example

- Set the DIP switches to boot from SPI flash
- Boot the board through uart xmodem as described in the above section
- When booted, stop the boot count down
- Setup 'ipaddr' and 'serverip' environment variables; and setup a TFTP server that has flash-rom.bin
- On the console run 'bubt'; this will download flash-rom.bin from the TFTP server and write it down on SPI flash

Verified Boot

TBD Read more here: [Verified Boot](#)

From:

<https://wiki.solid-run.com/> - Wiki | SolidRun

Permanent link:

https://wiki.solid-run.com/doku.php?id=products:a8040:software:development:u-boot_atf

Last update: **2017/03/23 06:13**

