AM57xx: Flashing Images to eMMC

The Development Kit is delivered with a pre-flashed bootloader. The following instructions for flashing images from SD card will be useful if you want to:

- Flash images because eMMC is empty
- Upgrade to a new release
- Use custom built images

The images to be flashed will need to be copied to the */boot* or /rootfs/boot/ partition of a properly formatted SD card as described in the Creating a Bootable SD Card instructions.

> ⚠ The Linux commands listed in this section will only work correctly if Linux is booted from SD card.

Partitioning eMMC

Partition eMMC from u-boot

Write a GPT partition table to eMMC. Create UUIDs for the disk and each partition by executing the following on the **host** machine:

Host (Linux)

```
uuidgen
<first UUID>

uuidgen
<second UUID>

uuidgen
<third UUID>
```

After making all required connections, power on the board and enter U-Boot. Set the UUIDs for the disk and rootfs to the generated values:

Target (U-Boot)

```
setenv uuid_gpt_disk <first UUID>
setenv uuid_gpt_rootfs <second UUID>
setenv uuid_gpt_env <third UUID>
gpt write mmc 1 ${partitions}
reset
```

The gpt partition will be visible after a reset. (Note that mmc0 corresponds with the SD card slot interface, while mmc1 corresponds with eMMC):

Target (U-Boot)

```
mmc dev 1
mmc part
```

Partition eMMC from Linux

Boot into Linux from the SD card, then use fdisk with the following options to write a new GPT partition table to eMMC:

Target (Linux)

```
fdisk /dev/mmcblk1
        g        GPT partition table
        n        new partition
        1        partition number
        2048     first sector
        4096     last sector
        n        new partition
        2        partition number
        6144     first sector
        <enter>  use default value
        w        write table to disk and exit

partprobe
```

Flashing the boot loader

Flashing the boot loader from U-Boot

Copy the MLO and u-boot.img from the /boot partition of the SD card (connector X2, mmc0 in U-Boot) to eMMC (mmc1 in U-Boot):

Target (U-Boot)

```
mmc dev 1

fatload mmc 0 ${loadaddr} MLO
echo $filesize
# In the following command, take the hex value from the above command, divide it by 0x200, and round up to the nearest whole number
mmc write ${loadaddr} 0x100 #FILESIZE#

fatload mmc 0 ${loadaddr} u-boot.img
echo $filesize
# In the following command, take the hex value from the above command, divide it by 0x200, and round up to the nearest whole number
mmc write ${loadaddr} 0x300 #FILESIZE#
```

Flashing the boot loader from Linux

Boot into Linux from the SD card and run the following commands to copy MLO and u-boot.img to eMMC:

Target (Linux)

```
dd if=/run/media/mmcblk0p1/MLO of=/dev/mmcblk1 seek=256
dd if=/run/media/mmcblk0p1/u-boot.img of=/dev/mmcblk1 seek=768
```

Flashing the root filesystem

⚠ If rootfs.ext4 is larger than the size of the DDR3, it can only be flashed in Linux. The default rootfs.ext4 for BSP-Yocto-TISDK-AM57xx-PD19.1.0 is larger than the default DDR3 size (2GB).

⚠ The rootfs.ext4 image is not loaded to the card by default. Copy it to the root of the rootfs partition on the SD card.

Flashing the root filesystem from Linux

Boot into Linux from the SD card, then copy the root filesystem to eMMC:

Target (Linux)

```
dd if=/rootfs.ext4 of=/dev/mmcblk1p2 bs=1M
```

Flashing the root filesystem from u-boot

This assumes the SD card was created with TI's **create-sdcard.sh** script. If the SD card is formatted differently, the *ext4load* command may need to be replaced by *fatload*.

Copy the root filesystem from the /rootfs partition of the SD card (connector X2, mmc0 in U-Boot) to eMMC (mmc1 in U-boot):

Target (U-Boot)

```
mmc dev 1
ext4load mmc 0:2 ${loadaddr} rootfs.ext4
mmc write ${loadaddr} 0x1800 [rootfs.ext4 size in bytes divided by 512, in hex]
```