How to update U-Boot on an SD card

Stable: 10.10.2019 - 17:14 / Revision: 10.10.2019 - 14:04

This page explains how to manually update the U-Boot binaries on an SD card.

Contents

1 Copying a binary to an SD card with the Linux dd command

When you have access to the device (with the card reader of a PC, or the `ums` command on a target), the 3 first partitions on the SD card are:

1. FSBL1
2. FSBL2
3. SSBL

See Boot_chains_overview for the bootloader definitions.

You can use the Linux `dd` command to copy the FSBL and SSBL directly to the correct partition:

```
PC $>  dd if=<file> of=/dev/<dev> conv=fdatasync
```

<dev> is:

- mmcblk<X>p<n> : PC-embedded card reader case
- sd<X><N> : USB-connected SD card reader case

where <X> is the ID of the device, and <n> the ID of the partition.

Note: the dd option conv=fdatasync is used to force synchronous copying.

2 Trusted boot chain update example

The internal card reader is /dev/mmcblk0, partition <n> is /dev/mmcblk0p<n>:

```
PC $> dd if=tf-a.stm32 of=/dev/mmcblk0p1 conv=fdatasync
PC $> dd if=tf-a.stm32 of=/dev/mmcblk0p2 conv=fdatasync
PC $> dd if=u-boot.stm32 of=/dev/mmcblk0p3 conv=fdatasync
```

3 Basic boot chain udpate example

The USB card reader is /dev/sdb, partition <n> is /dev/sdb<n>

```
PC $> dd if=u-boot-spl.stm32 of=/dev/sdb1 conv=fdatasync
PC $> dd if=u-boot-spl.stm32 of=/dev/sdb2 conv=fdatasync
PC $> dd if=u-boot.img of=/dev/sdb3 conv=fdatasync
```

4 update of *e*•MMC

The same command, `dd`, can be used to update *e*•MMC with `ums` command, but the SSBL = TF-A (or SPL) can't be saved directly in the cached boot partitions, and the indexes of other partitions need to be adapted.

For the user area, we use GPT partitioning. SSBL is the first user partition in *e*•MMC memory mapping :

```
Board $> ums 0 mmc 1
PC $> dd if=u-boot.stm32 of=/dev/sdb1 conv=fdatasync
```

For the boot partition, the user needs to select the targeted partition with the third parameter **partition_access** of command **`mmc partconf`** :

- **0**: user data partition
- **1**: boot partition 1
- **2**: boot partition 2

```
Board $>  help mmc
...
mmc bootbus dev boot_bus_width reset_boot_bus_width boot_mode
- Set the BOOT_BUS_WIDTH field of the specified device
mmc bootpart-resize <dev> <boot part size MB> <RPMB part size MB>
- Change sizes of boot and RPMB partitions of specified device
mmc partconf dev [boot_ack boot_partition partition_access]
- Show or change the bits of the PARTITION_CONFIG field of the specified device
..
```

For example,

- dev = 1 (*e*•MMC device on ST Microelectronics board)
- boot_ack=1 (Boot Acknowledge is needed by ROM code)
- boot_partition = 1 (Boot partition 1 enabled for boot)
- **partition_access** = **0** (No access to boot partition - default)
- or **partition_access** = **1** (R/W boot partition 1)

To copy FSBL with a `ums` command and come back to the user area, we have:

```
Board $> mmc dev 1
Board $> mmc partconf 1 1 1 1
Board $> ums 0 mmc 1
PC $> dd if=tf-a.stm32 of=/dev/mmcblk0p1 conv=fdatasync
Board $> mmc partconf 1 1 1 0
```

Read Only Memory