



How to update U-Boot



A quality version of this page, approved on 1 February 2021, was based off this revision.

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

Contents

| | |
|--------------------------------------------------------|---|
| 1 To update an SD card with the Linux dd command | 3 |
| 1.1 U-Boot environment | 3 |
| 1.2 SD card update example | 3 |
| 1.3 SD card update example with SPL as FSBL | 4 |
| 2 Update of eMMC with the Linux dd command | 5 |
| 2.1 On a Linux console | 5 |
| 2.2 On a U-Boot console | 6 |
| 3 References | 7 |



1 To update an SD card with the Linux dd command

When a Linux console has access to the SD card device partitions:

- on a Linux PC
 - with a card reader of this PC
 - through a USB connection to the target and the `ums` command executed on a U-Boot console
- on target, with Linux console.

The 3 first GPT partitions on the SD card are:

1. FSBL1
2. FSBL2
3. SSBL

See [Boot_chain_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 or target Linux console
- `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.

1.1 U-Boot environment

The U-Boot environment is saved at the end of the U-Boot partition, named "ssbl": ID = **3**.

To clear this environment, erase the U-Boot partition before any update; for example, by writing 0 to this partition:

```
PC $> dd if=/dev/zero of=/dev/mmcblk<X>p3 conv=fdatasync
```

```
PC $> dd if=/dev/zero of=sd<X>3 conv=fdatasync
```

1.2 SD card update example

The internal card reader is `/dev/mmcblk0` or for a target Linux console, GPT 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=/dev/zero of=/dev/mmcblk0p3 conv=fdatasync
PC $> dd if=u-boot.stm32 of=/dev/mmcblk0p3 conv=fdatasync
```

Alternatively, with U-Boot console, `dev = 0` (SD card device on ST Microelectronics board), GPT partition <n> is `/dev/sda<n>`:



```
Board $> mmc dev 0
Board $> ums 0 mmc 0

PC $> dd if=tf-a.stm32 of=/dev/sda1 conv=fdatasync
PC $> dd if=tf-a.stm32 of=/dev/sda2 conv=fdatasync
PC $> dd if=/dev/zero of=/dev/sda3 conv=fdatasync
PC $> dd if=u-boot.stm32 of=/dev/sda3 conv=fdatasync
```

1.3 SD card update example with SPL as FSBL

The USB card reader is /dev/sdb, GPT 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=/dev/zero of=/dev/sdb3 conv=fdatasync
PC $> dd if=u-boot.img of=/dev/sdb3 conv=fdatasync
```



2 Update of eMMC with the Linux dd command

The same command, dd, can be used to update eMMC memory mapping:

- SSBL U-Boot is the first GPT partition in the eMMC user area
- FSBL = TF-A (or SPL) is saved at the beginning of the eMMC boot partition

The user needs to select the eMMC hardware partition to update: user data, boot1, or boot2.

2.1 On a Linux console

If dev = **mmcbk1** for eMMC device (default on ST Microelectronics board)

The boot partitions are available in **/dev/mmcbk1boot0** and **/dev/mmcbk1boot1** ^[1].

The user perhaps needs to allow access, for example with:

```
Board $> echo 0 > /sys/class/block/mmcbk1boot0/force_ro
```

The mmc tools allow the boot partition to be selected ^[2].

The STM32MP15x ROM code requires:

- <send_ack> =1
- the eMMC boot configuration is: 1 wire configuration and 25 MHz, it is done with the command:

```
Board $> mmc bootbus set single_backward x1 x1 dev/mmcbk1
```

To update TF-A in boot1 and select this boot partition:

```
Board $> dd if=tf-a.stm32 of=/dev/mmcbk1boot0 conv=fdatasync
Board $> mmc bootpart enable 1 1 /dev/mmcbk1
```

To update TF-A in boot2 and select this boot partition:

```
Board $> dd if=tf-a.stm32 of=/dev/mmcbk1boot1 conv=fdatasync
Board $> mmc bootpart enable 2 1 /dev/mmcbk1
```

To update U-Boot in the first GPT partition:

```
Board $> dd if=/dev/zero of=/dev/mmcbk1p1 conv=fdatasync
Board $> dd if=u-boot.stm32 of=/dev/mmcbk1p1 conv=fdatasync
```

See also ^[3].



2.2 On a U-Boot console

The eMMC update is done with the `ums` command and the targeted eMMC HW partition is selected in U-Boot by 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 (eMMC 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)
```

And on a PC, the mass storage is mounted as `/dev/sda`

To update FSBL=TF-A, select the boot1 HW partition and come back to the user area at the end.

```
Board $> mmc dev 1
Board $> mmc partconf 1 1 1 1
Board $> ums 0 mmc 1

PC $> dd if=tf-a.stm32 of=/dev/sda conv=fdatasync

Board $> mmc partconf 1 1 1 0
```

To update SSBL = U-Boot in the first GPT partition in the user partition

```
Board $> mmc dev 1
Board $> ums 0 mmc 1

PC $> dd if=u-boot.stm32 of=/dev/sda1 conv=fdatasync
```

Before the first boot, select the eMMC correct boot configuration (1 wire, 25 MHz) with the command:

```
Board $> mmc bootbus 1 0 0 0
```



3 References

Please refer to the following links for additional information:

- <https://www.kernel.org/doc/Documentation/mmc/mmc-dev-parts.txt>
- <https://manpages.debian.org/buster/mmc-utils/mmc.1.en.html>
- <https://www.emcraft.com/som/stm32mp1/booting-linux-from-emmc>

SD memory card (<https://www.sdcard.org>)

MultimediaCard

Linux[®] is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))

Second Stage Boot Loader

First Stage Boot Loader

GUID Partition Table

Secondary Program Loader, *Also known as **U-Boot SPL***

Trusted Firmware for Arm Cortex-A

Read Only

Read Only Memory

former spelling for e•MMC ('e' in italic)