

# SD card setup

This chapter shows how to create the SD card. As mentioned in page "[Board bring-up overview for TEI0022](#)" this part is for SD card setup which should be done in the following steps:

- Generate SD card partitions and filesystems
  - SD card structure
  - Partition generation
  - Filesystems generation
  - SD card modification
- Copy files to partitions

The section "Generate SD card partitions and filesystems" describes the process to create the needed partitions and filesystems on the SD card via the Linux tool fdisk. Section "Copy files to partitions" shows the moving from files on the partitions.

## Generate SD card partitions and filesystems

The Intel Cyclone V needs a specific SD card structure which should be realized within the following steps in a linux system. Therefore, the structure shall be introduced in the next subsection. Then, in the second subsection the partitions shall be generated. After that, the filesystems shall be created. Then, in the last subsection, the modification shall be written on the SD card.

### SD card structure

The SD card needs three partitions, one vfat, one linux and one 0xA2 partition. The following table shows the target SD card partition which shall be generated in the next section.

#	Partition	Purpose	Size
1	W95 FAT32 (LBA)	(U-Boot,)* Linux Kernel, Device Tree Blob	> 20 MB
2	Linux ext(2/3 or 4)	Linux Root Filesystem	> 150 MB to 2 GB
3	0xA2	Preloader(, U-Boot)*	>= 2 MB

\* In this guide, the U-Boot shall be located in vFAT partition 1.

### Partition generation

To generate the several partitions, the next steps are necessary:

- Insert the SD card into your linux system.
- Start a terminal.
- Identify the SD card via shell command **lsblk**
- An exemplary output is visible in the next listing. The last line is an additional information, not part of the output.

#### Exemplary lsblk output

```
NAME MAJ:MIN RM   SIZE RO  TYPE MOUNTPOINT
sda    8:0    0  120G  0   disk
sda1   8:1    0   112G  0   part /
sda2   8:2    0    1K  0   part
sda5   8:5    0    8G  0   part [SWAP]
sdc    8:32   1    7,2G  0   disk
sdc1   8:33   1    1M  0   part
sr0    11:0    1  1024M  0    rom

RM = Removable sdc is the SD card with one partition sdc1
```

- **Attention: If the wrong identifier is selected, the following commands can destroy the host system! Therefore, be aware to use the correct identifier.**
- The identifier is similar to sdXXX, where XXX symbols the identified SD card. To detect the SD card, it could help to look for the expected SD card size. After that, remove the SD card, use **lsblk** and look whether the expected device is removed. Then, insert the SD card again and check with **lsblk** which identifier is selected for the SD card.
- Unmount all partitions of the SD card via **sudo umount /dev/<above identified partitions>**
- Start the tool fdisk as root via **sudo fdisk /dev/sdXXX**
- Look at the partition structure via inserting **p** **Enter**
- **Attention: Deleting the partitions in the next step destroys the saved information. If the information are needed save them before continuing!**
- Then delete all partitions via **d** **Enter**
- After all partitions are deleted the SD card structure can be created.

Creating partition #1:

- Create a new partition via inserting **n** **Enter**
- Select the primary partition via inserting **p** **Enter**
- Set the partition number via inserting **1** **Enter**
- Select the first sector via inserting **2048** **Enter**
- Select the last sector via inserting **+20M** **Enter**
- Set partition type via **t** **Enter**
- Select the first partition via **1** **Enter**, if it is not automatically selected.
- Select the partition type "W95 FAT32 (LBA)" via **c** **Enter**

Creating partition #2:

- Create a new partition via inserting **n** **Enter**
- Select the primary partition via inserting **p** **Enter**
- Set the partition number via inserting **2** **Enter**
- Select the first sector via inserting **Enter**
- Select the last sector (adapted to your SD card size) via inserting **+2G** **Enter**
- Set partition type via **t** **Enter**
- Select the first partition via **2** **Enter**
- Select the partition type "Linux, default value" via **83** **Enter**

Creating partition #3:

- Create a new partition via inserting **n** **Enter**
- Select the primary partition via inserting **p** **Enter**
- Set the partition number via inserting **3** **Enter**
- Select the first sector via inserting **Enter**
- Select the last sector via inserting **+20M** **Enter**
- Set partition type via **t** **Enter**
- Select the first partition via **3** **Enter**
- Select the partition type "Altera boot partition" via **a2** **Enter**

Writing SD card modifications:

The previous settings are not written to the SD card at this point. To write the changes to the card the next steps are necessary. **Therefore, if the wrong identifier is selected, the following commands can destroy the host system! Be aware to use the correct identifier! Furthermore, deleting the existing partitions on the card destroys the saved information! If the information are needed save them before continuing!**

- Write new settings to the SD card via **w** **Enter**
- Close the program via **q** **Enter**

## Filesystem generation

The filesystems are created with the linux tool **mkfs**.

- Plug the SD card out and in of the system.
- Create a vFAT filesystem on partition 1 via **sudo mkfs -t vfat /dev/sdXXX1**, whereby XXX symbols the above detected SD card identifier.
- Create an ext3 filesystem on partition 2 via **sudo mkfs.ext3 /dev/sdXXX2**, whereby XXX symbols the above detected SD card identifier.
- No extra filesystem for partition 3 is needed.

## Copy files to partitions

After preparing the SD card for booting, this section describes the copy process for the relevant files with a Linux and a Windows operating system (OS):

- Insert the SD card to a Linux or Windows OS.

vFAT Partition 1:

- Copy the linux kernel **zImage.bin** file to the **vFAT** partition 1 as **zImage** file. That means, the ".bin" extension has to be deleted.
- Copy the device tree blob **socfpga.dtb** file to the **vFAT** partition 1 as **socfpga.dtb** file. The file needs this name.
- Copy the bootloader **u-boot.img** file to the **vFAT** partition 1 as **u-boot.img** file. The file needs this name.

ext3 Partition 2:

- Copy the linux root filesystem to the ext3 partition 2 and decompress it if it is compressed.
  - For Linux this can be done using the following steps:
    - Mount the ext3-partition (in this case 2) to an available folder (e.g. /mnt/mountpoint), if necessary, via ***sudo mount /dev /<correct partition of the SD card> /mnt/mountpoint***
    - Extract the <root filesystem>.tar.gz via ***sudo tar -xvf <path to compressed root filesystem>.tar.gz -C /mnt/mountpoint/***
    - Synchronise the buffers via ***sudo sync***

0xA2 Partition 3:

- Insert the SD card to a Windows OS.
- Start the SoC EDS terminal as administrator. To do that navigate to ...\\intelFPGA\\18.1\\embedded\\ , right click on the file "Embedded\_Command\_Shell.bat", and select "Run as administrator". Click **Yes** in the window "User Account Control".
- Identify one of the SD card letter(s).
- Copy the preloader to the 0xA2 partition 3 via ***alt-boot-disk-util -p <path to the preloader image>preloader-mkpimage.bin -a write -d <SD card letter>***

After doing these steps, the SD card can be inserted into the board and the boot process can be started. If booting runs into linux, the procedure is done.