# EMBEDDED LINUX

# SD card setup

**Brief**
This chapter presents how to set-up an SD card. This includes erasing the existing partition table to create a new one, and building the filesystem of an ext4 partition. Well that just sounds like formatting with extra steps.

**Author**
[dimitri.boudier@ensicaen.fr](mailto:dimitri.boudier@ensicaen.fr)
[hugo.descoubes@ensicaen.fr](mailto:hugo.descoubes@ensicaen.fr)

**Source**
Debian: Getting started with the BeagleBone Black.
by Robert C. Nelson

**Reminder**
/!\ Understand all commands before running them! /!\

---

# RUN THE START-UP SCRIPT

Run the `elinux/tp/disco/script/start_elinux.sh` script that moves you to your working directory

```
source scripts/start_elinux.sh
```

---

# ERASE MEDIA

__ /!\ Do not plug your SD card in yet! /!\ __

## Erase MMC SD card by deleting hold partitions layout and labels

Identify your computer/laptop drive(s) by their device name (first column)

```
lsblk -f
sudo sfdisk -l
```

- What devices exist on your computer (only physical drives)? What partitions exist?

>

Connect your MMC microSD card to computer with USB adapter, and identify its device name

```
lsblk -f
sudo sfdisk -l
```

- What is the full name of your SD card? Do you see any partition? Keep a copy of the output as a proof.

>

**WARNING - You have to be sure of the device name**
**Be sure to never use *ical/dev/sda* device: this may be your hard drive!**

```
export DISK=/dev/*your_Device_Name*
echo $DISK
```

Erase partition layout and labels (MBR, Master Boot Record)

```
sudo dd if=/dev/zero of=${DISK} bs=1M count=10
sudo sfdisk -l
```

- Is there any partition now? Keep a copy of the output as a proof.

>

Dump the SD card content into a local file.

```
cd $DISCOPATH/mass
sudo dd if=${DISK} of=./sdcard_dump_step0_erase bs=1 count=1M
xxd sdcard_dump_step0_erase > sdcard_dump_step0_erase.txt
gedit sdcard_dump_step0_erase.txt&
```

- What has been done? What do you see in the `txt` file?

>

---

# CREATE PARTITION TABLE

- What is a partition table? What is MBR?

>

Create a new partition layout (MBR)

```
sudo sfdisk ${DISK} <<__EOF__
> 4M,,L,*
> __EOF__
```

- What does `sfdisk` do? What the does parameters do? Check `man sfdisk`.

*Note: you might have to unmount your device first*

```
lsblk -f
umount *your_device_mountpoint*
```

Dump the new content of the SD card into a local file.

```
sudo dd if=${DISK} of=./sdcard_dump_step1_mbr bs=1 count=1M
xxd sdcard_dump_step1_mbr > sdcard_dump_step1_mbr.txt
diff sdcard_dump_step0_erase.txt sdcard_dump_step1_mbr.txt -y --suppress-common-lines
gedit sdcard_dump_step1.txt_mbr &
```

- What has been done? What do you see in the `txt` file?

You should now see a partition 1 on your device (sdb1 for example)

```
sudo sfdisk -l ${DISK}
lsblk -f
```

## MBR (Master Boot Record) description

Informations about MBR on [Wikipedia](Wikipedia).

MBR 2,192Tb max has been replaced by GPT since 2013 (GUID Partition Table).
Addressing mode: LBA (Logical Block Addressing) vs old CHS (Cylinder Head Sector).
On a mass storage device, the first sector (512 bytes) is used for the MBR.

### Structure of a MBR

| Address | Size | Description |
|---|---|---|
| 0x0000 | | --- first sector --- |
| 0x0000 | 440 | Boot routine |
| 0x01B8 | 4 | Sign (optional) |
| 0x01BC | 2 | Null |
| 0x01BE | 64 | Partition table (4 entries of 16 bytes) |
| 0x01FE | 2 | 0xAA55 (MBR sign) |
| 0x0200 | | --- second sector --- |

More info on the MBR Wikipedia article, more specifically:

- the "Structure of a modern standard MBR" table,
- the "Layout of one 16-byte partition entry" table.

---

# CREATE FILE SYSTEM

Create an ext4 technology file system on partition 1 (sdb1 for example)

```
sudo mkfs.ext4 -L rootfs ${DISK}1
lsblk -f
```

- What have you done?

---

# CREATE MOUNT POINT

- What is a mountpoint? This is a Linux specific question.

## Automatically create a mount point on the host's file system (/media)

Remove your SD card from your computer and connect it back again without removing USB adapter
Linux system will automatically create a mount point on /media/*user_Name*

```
lsblk -f

export MEDIA=/media/*user_Name*
echo $MEDIA
```

## Manually create a mount point on the host's file system (/media)

WARNING - Do this only if Linux has not automatically created a mount point

```
sudo mkdir -p ${MEDIA}/rootfs
sudo mount ${DISK}1 ${MEDIA}/rootfs
lsblk -f
```

## Unmount partition when done

**/!\ WARNING - Never forget to unmount your partition after files copies and before removing your microSD /!**

```
sync
sudo umount ${MEDIA}/rootfs
```

- Why do you need to `sync` before unmounting the device?

Verify that your mounting point `${MEDIA}/rootfs/` is correctly unmounted.

```
lsblk -f
```

Now look at this system file to show static (non dynamic) file system information.

```
cat /etc/fstab
```

- What can you see?

---

# UPDATE THE start_elinux.sh SCRIPT

Update the `elinux/tp/disco/scripts/start_elinux.sh` script by adding the definition of the variables `DISK` and `MEDIA`.

```
export DISK=/dev/*your_Device_Name*
echo $DISK

export MEDIA=/media/*user_Name*
echo $MEDIA
```

---

# CREATE A SCRIPT FOR SETTING UP THE SD CARD

Create a new script `elinux/tp/disco/scripts/sdcard_setup.sh`. It should contain only the necessary commands to set up the SD Card.

- Copy here the content of your script.