

# Command Line

## ClusterCTRL: How do I USBBOOT a Compute Module and use eMMC as local storage/swap?

A Compute Module will boot as a USB device if either USBBOOT is enabled or it can't find bootcode.txt in the first FAT partition on the eMMC.

Replace X in the following steps with X of the pX device you're configuring (for X=1 for p1, X=5 for p5, etc).

First setup the filesystem and Compute Module to [boot as a USB Device](#).

Once USBBOOT is working power on the Compute Module "clusterctrl on pX" and wait for it to boot - (until you can "ping pX.local", etc).

On the controller you will now need to disable USBBOOT.

```
clusterctrl usbboot off pX
```

To permanently allow access to the eMMC as local storage you will need to save the power on state (see the USBBOOT instructions on how to do this).

Log into the Compute Module node (via either "ssh pi@pX.local" or "screen /dev/ttypiX").

You can now access the eMMC as /dev/mmcblk0 which can be partitioned with fdisk/etc.

The existing partitions can be viewed with fdisk

```
sudo fdisk -l /dev/mmcblk0
```

To ensure the Compute Module boots as a USB Device (and not from the eMMC) you will need to ensure the first FAT partition does not contain bootcode.bin file (either by removing the file or not having a FAT partition at all).

For example to use it as a 1GB swap partition and the rest as a /data partition you could do the following.

Using fdisk remove the existing partitions (this will remove all existing data so **backup** first).

```
sudo fdisk /dev/mmcblk0
```

# Command Line

Press "p" and return to see the existing partitions.

For each existing partition press "d" and return, enter the partition number or press enter to delete the last one each time.

Once the existing partitions are removed you can add the new partitions.

Type "n" and return, use "p" for a primary partition, use partition number 1, it will start with the first available sector so you can press enter, for the size enter "+1G". To set the type to "Linux Swap" press "t" return, then type "82" (Linux Swap) and press return.

Then add the data partition, again type "n" and return, use "p" for a primary partition, use partition number 2, go with the default first and end sectors by pressing enter twice.

You can then use "p" and enter to see the partitions.

If you're happy use "w" and return to write the changes and exit or "q" to quit without changing the partition table.

Reload the partition table

```
sudo partprobe
```

Setup swap and create the filesystem

```
sudo mkswap /dev/mmcblk0p1
```

This will show you the UUID for the swap partition, keep this for later.

```
sudo mke2fs -j /dev/mmcblk0p2
```

Again this will show the UUID you will need later.

Create the mount point for the filesystem, here I'm using /data

```
sudo mkdir /data
```

Edit /etc/fstab to setup using both swap and the new filesystem

# Command Line

```
sudo nano /etc/fstab
```

Append new lines for both swap and data partitions (replacing with your UUID from above).

```
UUID=029577ad-b1d2-43ad-  
b1f9-348e37969049          swap      swap defaults 0 0  
UUID=e5dd1581-8baf-43d7-b49b-  
a24cb93da970              /data    ext4  defaults 0 0
```

Write the file and exit nano.

Check the swap partition is setup OK in /etc/fstab

```
sudo swapon -a
```

You should now see the added swap space if you run "free".

Mount the filesystem.

```
sudo mount -a
```

You should see the mounted filesystem by running "df -h".

After rebooting the Compute Module it will still boot via USBBOOT and use the new local swap and /data filesystem.

Unique solution ID: #1105  
Author: Chris Burton  
Last update: 2019-08-06 12:33