Tutorial : Configuring a micro SD memory card as both “swap” and “storage” area for the DragonBoard 410c (DB410C)

[1] Introduction

The unboxing experience of the **DragonBoard 410c** is very enjoyable and straightforward. At first, one must connect it to an HDMI monitor, keyboard and mouse and then power it on. In its factory configuration, the **DB410C** is shipped with Android operating system. Nevertheless, it's usual that some users may prefer to use other operating systems available for download. This tutorial is useful to those who opt for utilizing the Debian (Linaro) operating system.

After reflashing the **DB410C** with a new operating system, the next configuration step is likely to be the WiFi network parameters. With that accomplished, the **DB410C** is ready for use.

However, the overall user experience with the **DB410C** will significantly enhance, should the user follow the instructions of this tutorial, which aim the add a micro SD memory card as both “swap” and extra “storage” for the board.

[2] Required Items

- 1 micro SD memory card (preferably, a fast one, such as the “SanDisk Ultra”) with 16GB or larger capacity.

[3] Procedure

In this example, I'll be using a micro SD card with a capacity of 16GB (empty and previously formatted as FAT32).

[3.1] Open a new Terminal Window (System Tools -> LXTerminal)

[3.2] Check the device names

$ **ls /dev/mmc***

The result will contain a list of names starting with "**/dev/mmcblk0**”.

[3.3] Insert the micro SD card in the DC410C slot (the micro SD card will probably be mounted automatically)

[3.3.1] Recheck the device names

$ **ls /dev/mmc***

At this time, the result must contain a list of several names starting with either “**/dev/mmcblk0**” or “/**dev/mmcblk1**”. Note the difference between this result and that one from item [3.2].

Probably, the difference are those names starting with “**/dev/mmcblk1**” (which are the disk partitions related to the newly inserted micro SD card).

[3.3.2] Check the mounted disks names

$ ls /media/linaro
The result will contain a list of names of mounted disks in the **DB410C**.

[3.3.3] Check the link between devices and disks

$ df -hT

The result will contain a table with 7 columns and many rows. The columns of interest are the first one (**FileSystem**) and the last one (**Mounted on**).

The line related to the micro SD card will likely be the last one of this table. In my case, what I see is the following :

FileSystem	**Type**	**Size**	**Used**	**Avail**	**Use%**	**Mounted on**
/dev/mmcblk1p1	vfat	15G	2.7M	15G	1.00%	**/media/linaro/FAT**

This means that the micro SD card has a device name "/**dev/mmcblk1**" and its "**p1**" partition is mounted in a directory named "/**media/linaro/FAT**" (this name may differ a little, case-by-case).

[3.3.4] Eject (unmount) the micro SD card

$ sudo umount /media/linaro/FAT

[3.3.5] Verify whether the "unmounting" really happened

$ df -hT

Note that, at this time, the last line (related to the micro SD card) must have been gone.

[3.4] Repartition the micro SD card

The name "**/dev/mmcblk1p1**" means that the device "**/dev/mmcblk1**" has a partition "**p1**". Other partitions may also exist ("**p2**", "**p3**" , etc). In my case, what I see is a unique partition named "**/dev/mmcblk1p1**".

On the next steps, we'll repartition the micro SD. The goal is to create two new partitions: one named "**/dev/mmcblk1p1**" and another named "**/dev/mmcblk1p2**".

Partition name	**Size**	**Type of partition**	**Function**
/dev/mmcblk1p1	3 / 4 of the available space	Linux (83)	Storage
/dev/mmcblk1p2	1 / 4 of the available space	Swap Linux (82)	Swap

We'll be using the command **fdisk**, which must be run as superuser (**sudo**). For this reason, be **very** careful while using it.

$ sudo fdisk /dev/mmcblk1

The main options of the command **fdisk** are:

option	**description**
m	Help
p	Show partitions
d	Delete a partition
n	Add a new partition
t	Change the type of a partition
w	Save results and exit

[3.4.1] Delete the existent partition

type **d**

[3.4.2] Create a new partition

type **n**

type **p** (type of partition : primary)

type **1** (id of partition : 1)

type **2048** or press enter (first sector : 2048)

At this time, a number will be shown (as default). In my case, the number is 31116287. Read this number and multiply it by **3/4**. The result wil be 23337215. Type this number.

type **23337215** (last sector : 23337215)

[3.4.3] Create a second partition

type **n**

type **p** (type of partition : primary)

type **2** (id of partition : 2)

<u>press enter</u> (first sector : default = 23339008)

<u>press enter</u> (last sector : default = 31116287)

[3.4.4] Change the types of partitions

type **t**

type **1** (id of partition: 1)

type **83** (type of partition : Linux)

type **t**

type **2** (id of partition: 2)

type **82** (type of partition : Linux Swap / Solaris)

[3.4.5] Verify results

type **p**

The result will be a table with 8 columns :

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/mmcblk1p1		2048	23337215	23335168	11.1G	83	Linux
/dev/mmcblk1p2		23339008	31116287	7777280	3.7G	82	Linux Swap / Solaris

[3.4.6] Save results and exit

type **w**

[3.4.7] Check the names of the new devices

$ ls /dev/mmc*

Note that at the tail of the list there will be the following devices :

/dev/mmcblk1 (micro SD card)
/dev/mmcblk1p1 (partition 1 : Linux)
/dev/mmcblk1p2 (partition 2 : Linux Swap)

[3.5] Format Linux partition

We'll format the partition **p1**, create a mounting directory (**/media/sdcard**) and mount the device in this directory.

$ sudo mkfs.ext3 /dev/mmcblk1p1	(format)
$ sudo mkdir /media/sdcard	(create directory)
$ sudo mount -t ext3 /dev/mmcblk1p1 /media/sdcard	(mount)

[3.6] Configure the Swap Linux partition

We'll prepare the partition **p2** as Swap area and mount it:

$ sudo mkswap /dev/mmcblk1p2	(prepare)
$ sudo swapon /dev/mmcblk1p2	(mount)
$ sudo swapon -s	(show status)

[3.7] Create scripts for automating the mounting of partitions

Using your preferred text editor, create the following file (saving it as "**dragon_up.sh**") in a "utility scripts" folder :

In my case, the "utility scripts" folder is : **/home/linaro/Documents/utils**

dragon_up.sh

```
#! /bin/sh

sudo mount -t ext3 /dev/mmcblk1p1  /media/sdcard
sudo swapon /dev/mmcblk1p2
sudo swapon  -s
```

Similarly, create the file below (saving it as "**dragon_down.sh**") :

dragon_down.sh

```
#! /bin/sh

sudo umount /media/sdcard
sudo swapoff /dev/mmcblk1p2
```

Transform them into executable files :

```
$sudo chmod +x ./dragon_up.sh
$sudo chmod +x ./dragon_down.sh
```

[3.8] The new partitions are ready for use

Whenever needed, run the script "**dragon_up.sh**" to activate the new partitions, or the script "**dragon_down.sh**" to deactivate them.

One interesting idea is to include the calling of the script "**dragon_up.sh**" in the Linux initialization (boot).

[3.9] Automate the calling of the script just after the Linux boot (**Optional**)

Certainly, your project will need to execute initialization scripts along with the boot (or just after it).

In this example, we'll create a new script name "**mybootscript**" with this goal.

Should your project already have a similar script, suffice to include a calling to "**dragon_up.sh**" into it.

[3.9.1] Create the script "**mybootscript"**

Using your preferred text editor, create the following file, saving it as "**mybootscript"** in the folder **/etc/init.d**

/etc/init.d/mybootscript

```
#! /bin/sh

cd /home/linaro/Documents/utils
sudo ./dragon_up.sh
```

$sudo chmod 755 /etc/init.d/mybootscript	(transform into executable)
$sudo update-rc.d mybootscript defaults	(configure the service)
$sudo service --status-all	(show services status)

From now on, just after the **DB410C** boot, the script "**mybootscript"** will run. In this example, this script calls the "**drangon_up.sh**" script which, in its turn, activates the partitions of the micro SD card. In this same "**mybootscript**" file, it's possible to include other callings to different scripts of your project.

[4.0] Conclusion

Your **DragonBoard 410c** will have an optimized performance due to the new "swap" and "storage" partitions (directory **/media/sdcard**) of the micro SD card. You'll promptly notice the performance enhancement. With these new resources, it'll be possible to build large software packages from source code directly (or natively) in the **DB410C,** and much more swiftly. The user experience of the DB410C will be more enjoyable due to the possibility of several terminal windows or applications be simultaneously opened, with minimal overload to the operating system.

Cezar Menezes
cezar.menezes@live.com