

# Raspberry Pi SD card 이미지 복제하고 재설치하기

출처 : <https://kyubot.tistory.com/132>

raspberry pi 나 tegra board 같은 임베디드 플랫폼에서 개발하다 보면 대부분의 리눅스 개발 환경이 그렇듯 잘 되다가도 가끔씩 시스템이 꼬여서 화면이 안나오거나, 계속해서 빌드 에러가 발생한다던지 해서 재설치 해야되는 곤란한 경우가 생기기 마련이다.

그때마다 다시 처음부터 이미지를 굽고 다시 설치하는 과정에서 많은 시간을 낭비하는데 그래서 sd card의 이미지를 복제하고 다시 복원하는 과정을 정리해 보았다.

사실 메모리 전체 이미지를 복사하는 것은 매우 쉽지만 만약 큰 용량의 sd카드, 예를들어 64기가나 128기가 같은 용량의 메모리를 사용하는 경우 복제한 이미지도 똑같은 용량을 차지해버리므로 이동, 보관이 용이하지 않다.

raspbian OS 이미지의 경우 4기가바이트 정도의 용량으로 사용자 sd카드 용량에 따라 설치 후 확장되는 방식을 적용하는데 여기서도 마찬가지로 실제로 사용하고 있는 용량 만큼의 이미지만 만들고 복원한 다음에는 사용자의 sd 카드 용량에 맞춰서 expand하는 방법을 정리하였다.

이 과정을 위해서는 micro sd카드를 읽을 수 있는 linux PC 가 필요하다.

## Linux PC의 설치 준비

apt install로 gparted와 pv를 설치한다.

```
$ sudo apt install gparted pv dcfldd
```

## Disk image 만들기

라즈베리파이에서 SD 카드를 꺼내서 PC에 삽입 후 df 명령어로 디스크 위치를 확인한다.

```
$ df -h
Filesystem Size Used Avail Use% Mounted on
udev 3.9G 0 3.9G 0% /dev
tmpfs 786M 9.4M 777M 2% /run
/dev/sdb6 231G 63G 157G 29% /
tmpfs 3.9G 318M 3.6G 9% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 3.9G 0 3.9G 0% /sys/fs/cgroup
/dev/sda1 96M 29M 68M 30% /boot/efi
tmpfs 786M 100K 786M 1% /run/user/1000
```

```
/dev/mmcblk0p2 59G 5.0G 52G 9% /media/dronedev/rootfs
/dev/mmcblk0p1 44M 23M 22M 52% /media/dronedev/boot
```

이중에서 micro SD카드와 관련된 항목은 /dev/mmcblk0p1 과 /dev/mmcblk0p2 이다.

이제 카드를 PC에 끼운 상태에서 umount로 위 장치를 마운트 해제한다.

```
$ sudo umount /dev/mmcblk0p*
```

dd 명령어로 sdimage.img 라는 이미지 파일을 만든다.

```
$ sudo dd if=/dev/mmcblk0 of=sdimage.img status=progress
1611215360 bytes (1.6 GB, 1.5 GiB) copied, 39 s, 41.3 MB/s
```

용량에 따라 매우 오랜 시간이 걸린다. 64기가 기준으로 대략 1시간 정도 걸린 듯 하다.

이제 sync 명령으로 sd카드를 안전하게 제거한다.

```
$ sudo sync
```

## image 용량 줄이기

위에서 생성된 img 파일은 root권한으로 생성되었기 때문에 파일의 소유자를 사용자로 변경한다.

```
$ sudo chown username.username sdimage.img

$ ls -l

-rw-r--r-- 1 username username 64087916544 2월 20 12:23 sdimage.img
```

이 파일을 fdisk 명령어로 살펴보면 아래와 같이 나타나는 것을 확인할 수 있다.

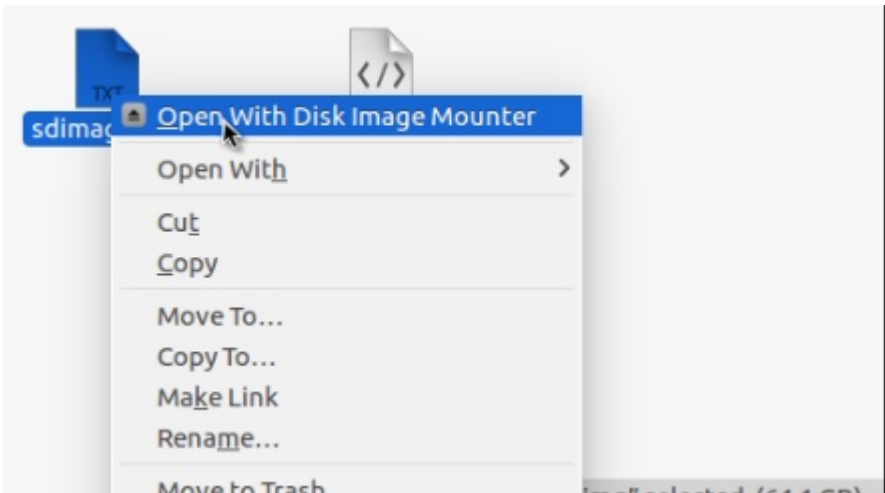
```
$ sudo fdisk -l sdimage.img
Disk sdimage.img: 59.7 GiB, 64087916544 bytes, 125171712 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc8ba1b7d

Device          Boot Start      End  Sectors  Size Id Type
sdimage.img1    8192      97890    89699  43.8M  c W95 FAT32 (LBA)
sdimage.img2    98304 125171711 125073408 59.7G  83 Linux
```

이제 아래와 같이 losetup 명령어를 사용하여 이미지를 /dev/loop0에 마운트 한다.

```
$ sudo losetup /dev/loop0 sdimage.img
```

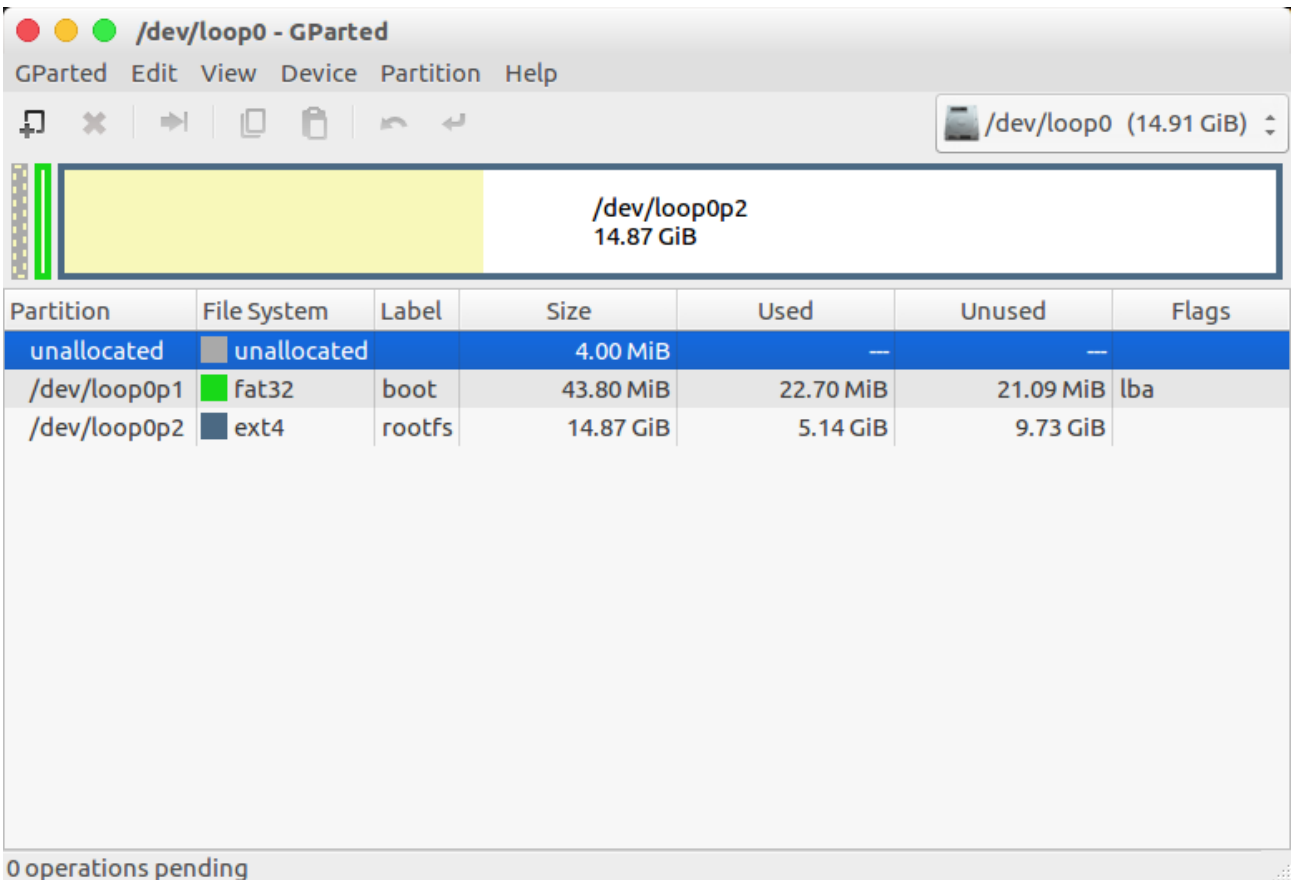
또는 Finder 에서 disk image mounter 를 사용해도 된다.



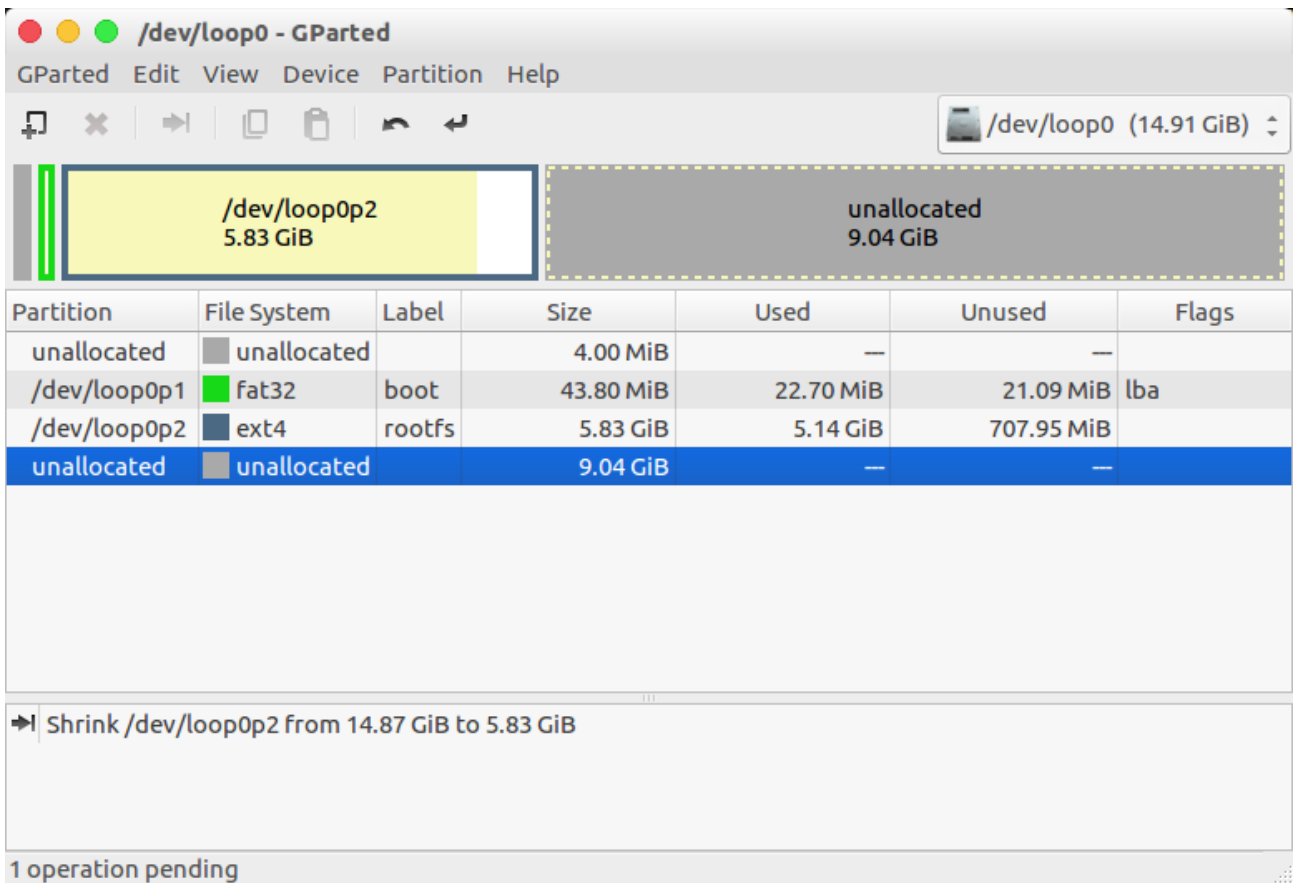
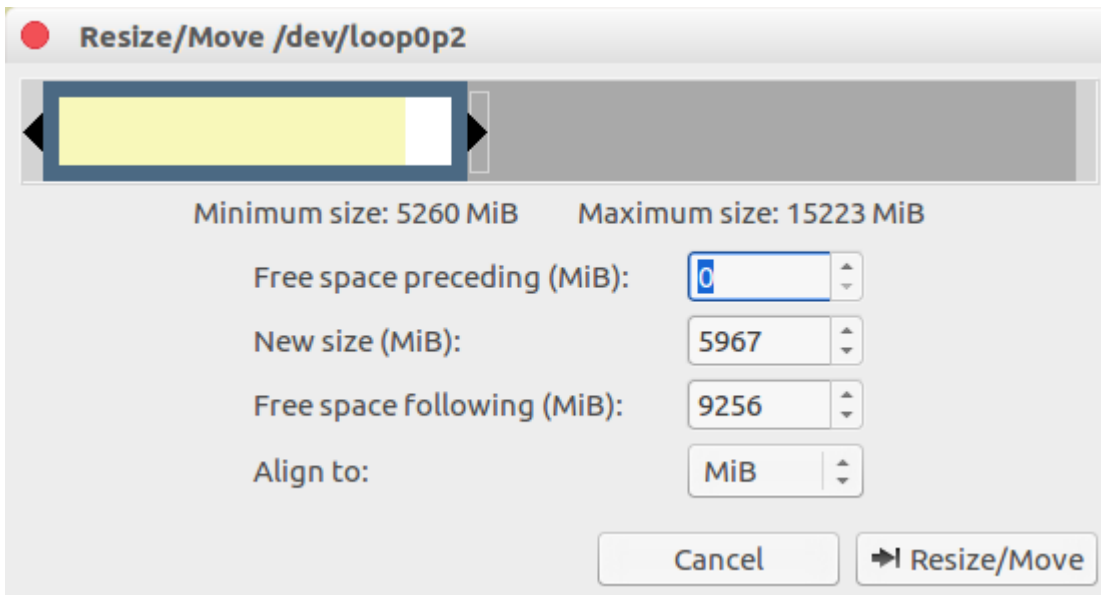
이제 마운트한 이미지를 gparted에서 열어서 용량을 변경해보자. 이 시점에서 만약을 위해 img 파일을 백업해 놓는것도 좋다.

```
$ sudo partprobe /dev/loop0
$ sudo gparted /dev/loop0
=====
libparted : 3.2
=====
```

아래와 같은 화면에서 마우스 오른쪽 클릭하여 Resize/Move를 선택한 다음,



세로 바를 움직여서 적당한 크기로 만들거나 직접 입력한다. 최소 용량에서 200MB 정도 여유를 주는 게 좋다.



이제 Edit -> Apply All Pperations를 눌러 적용한다. **Save details**를 누르면 세부 사항이 저장된다.

모두 닫고 losetup -d 명령으로 장치를 제거한 다음 fdisk -l 로 이미지의 정보를 확인한다.

```
$ sudo losetup -d /dev/loop0

$ fdisk -l sdimage.img
```

```
Disk sdimage.img: 14.9 GiB, 16012804096 bytes, 31275008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc8ba1b7d

Device Boot Start End Sectors Size Id Type
sdimage.img1 8192 97890 89699 43.8M c W95 FAT32 (LBA)
sdimage.img2 98304 12257279 12158976 5.8G 83 Linux
```

truncate 명령어를 사용하여 사용되지 않는 영역을 삭제한다. 여기서 END는 2번째 파티션의 마지막 섹터, 즉 12257279 이다.

```
$ truncate --size=$((12257279+1)*512) sdimage.img
```

이제 etcher와 같은 툴을 사용하여 sd 메모리에 저장 후 새로운 라즈베리파이로 부팅하면 성공이다.

## 라즈베리파이에서 파티션 늘리기

라즈베리파이에서 처음 부팅하게 되면 줄어든 용량만큼만 사용하게 되므로 sd카드의 나머지 영역을 사용하도록 확장하기 위해 raspi-config를 실행한다.

```
$ sudo raspi-config
```

여기서 Advanced Options -> A1 Expand Filesystem 을 선택하여 전체 sd카드 용량을 사용하도록 선택한다.

설정을 적용하여 Finish 후 재부팅하면 된다.

재부팅 후 df -h 를 실행하면 아래와 같이 16G 용량으로 변경된 것을 확인할 수 있다.

```
$ df -h

Filesystem Size Used Avail Use% Mounted on /dev/root 15G 5.0G 9.1G 36% / devtmpfs 460M
0 460M 0% /dev tmpfs 464M 0 464M 0% /dev/shm tmpfs 464M 13M 452M 3% /run tmpfs 5.0M
4.0K 5.0M 1% /run/lock tmpfs 464M 0 464M 0% /sys/fs/cgroup /dev/mmcblk0p1 44M 23M 22M
52% /boot tmpfs 93M 0 93M 0% /run/user/1000
```

---

🔄Revision #5

★Created 2023-07-11 08:01:20 UTC by Hyeon Su Ryu

✎Updated 2023-07-11 08:10:10 UTC by Hyeon Su Ryu