

Clone / Restore SD Card

the Ubuntu package "pv" needs to be install before running the following commands.

Prerequisites

```
sudo apt install pv
```

you can check where what devices your SD card is using by running "fdisk"

```
$ sudo fdisk -l
```

Copy SD Card to Image

Before cloning the SD Card, remove temporary files by deleting all files within the following directories:

- ~/workspace/.ccache
- ~/workspace/.vscode-server
- ~/workspace/ros_home
- ~/workspace/rtabmap
- ~/workspace/rtk_logs
- ~/data

To limit the size of the cloned file, using GParted to first resize the SD card partition so that it is only as large as needed to contain all the data on the SD card. To resize the partition in GParted, right click the partition and select "unmount" then right click again and select "Resize/Move".

NOTE: if the minimum possible SD card partition size shown in GParted is shown equal to the maximum SD card size (i.e., disallowing shrinking the SD card), then check that the SD card is not mounted / locked by looking for a key symbol on the line showing the SD card information in GParted. If you cannot get the SD card to unmount because it is busy, you may need to start GParted first before inserting the SD card and this should allow the SD card to unlock / unmount. The minimize possible size of the SD card should now allow shrinking the SD card partition.

Then copy the SD card to file:

Copy SD card to Image

```
# ensure to umount the partitions of the SD card after you have connected your host machine
sudo umount /dev/sdb1
sudo umount /dev/sdb2

export SD_CARD_DEV=/dev/sdb
export SIZE=72312 # use SD card partition size in MiB (partition size as shown in GParted)
# to find sd card partition size in gparted, right click the partition and select "Resize/Move". "New size (MiB)" will then initially show the current size of the partition.

# NOTE: use image name without spaces
export IMG_NAME_BACKUP=MPL_TakeHome_V1
# above environment variables are an example to for the next command
sudo sh -c "dd if=$SD_CARD_DEV | pv -s ${SIZE}m | dd of=${IMG_NAME_BACKUP}.img bs=1M count=${SIZE} iflag=fullblock"
```

Remember to resize the SD card partition back to its full size after copying.

NOTE: SD Card issue for Orin Systems:

When creating the SD Card for the Orin devices, the following errors were encountered following a clean formatting of the SD card and a clean rebuild of all programs (i.e., not a rebuild from clone).

Running fsck (and same output obtained from e2fsck):

```
slamr01@slamr01-orin2:~$ sudo fsck /dev/mmcbk1
fsck from util-linux 2.34
e2fsck 1.45.5 (07-Jan-2020)
ext2fs_open2: Bad magic number in super-block
fsck.ext2: Superblock invalid, trying backup blocks...
fsck.ext2: Bad magic number in super-block while trying to open /dev/mmcbk1
```

The superblock could not be read or does not describe a valid ext2/ext3/ext4 filesystem. If the device is valid and it really contains an ext2/ext3/ext4 filesystem (and not swap or ufs or something else), then the superblock is corrupt, and you might try running e2fsck with an alternate superblock:

```
e2fsck -b 8193 <device>
or
e2fsck -b 32768 <device>
```

Found a dos partition table in /dev/mmcb1k1

- It is uncertain if the error above is due to a problem with the SD card itself, or a software problem, or a normal condition for this setup.

Write Image to SD Card

Use gparted to delete all partitions on the SD card, if any.

If overwriting a corrupted SD card, begin by recreating the partition tables in case these were corrupted:

```
# wipe the sd card partition tables
sudo dd if=/dev/zero of=<sd_card_device_path> count=63

# EXAMPLE
sudo dd if=/dev/zero of=/dev/sda count=63
```

Then use gparted to create a new partition table.

```
$ sudo gparted
```

First select the SD Card device from the upper-right drop-down
Device Create Partition Table select ms_dos format

Then use gparted to reformat the SD card to ext4 file filesystem. (Not sure if this is needed)

Write the image to the SD card

Write Image to SD Card

```
# if the file is as ${IMAGE_FILE}.tar.gz
gunzip $IMAGE_FILE

# set $IMAGE_FILE to path to the sd backup image
export IMAGE_FILE=slamr01_orin_ubuntu20_04_jetpack5_0_2_sdcard.img

# BE CAREFUL when identifying the sd card device.
# Use 'fdisk -l' to check
export SD_CARD_DEV=/dev/sdb

sudo sh -c "pv -tpreb $IMAGE_FILE | dd of=$SD_CARD_DEV bs=1M"
```

Using gparted, now resize the SD card partition to the full size of the SD card if needed. (If this fails on the host system or just as a precaution, you can use gparted on the jetson to resize the SD card)

Optionally, disable journaling on SD card to improve IO and reduce number of writes:

```
# disable ext4 journaling
# (replace "/dev/sda1" by your correct device path)
sudo tune2fs -O ^has_journal /dev/sda1
```

Fixing Broke SD Card

In cases when the power of the platform is lost , and the platform is not able to boot up. We found that the following command fixes the issue:

You need to insert the SD card into the host machine to run the following commands.

```

# on the terminal of the host machine

# unmount the sd card if it is automatically mounted at the moment it inserted on the host machine.
$ sudo umount /dev/mmcblk0p1
or
$ sudo umount /media/{user}/{sd_card_name}

# check that it has been unmounted with
$ df -h

# based on this blog : https://forums.developer.nvidia.com/t/stuck-at-nvidia-logo-after-power-outage/238335/2
# Attempt automatic repair (I assume the name is "/dev/mmcblk0", and that the ext4 partition is "/dev/mmcblk0p1", adjust for your case to name the ext4 partition):
$ sudo fsck.ext4 -p /dev/mmcblk0p1
# If damage was sufficient, then you might need to force check. Automatic repair might refuse. This is an indication that damage is more extensive. Any repair (once done) will have
# hints as to what was removed by examining the "lost+found/" subdirectory at whatever mount point the partition is mounted on.

$ sudo fsck.ext4 -fvy /dev/mmcblk0p1

# Usually what gets corrupted is the content which was being written. Damage is not necessarily limited to this though since writing to a file in a directory might update the directory # entry, and the directory itself might have been caught mid-write during the power loss.

```

Another option (can be run on Jetson)

```

# unmount the SD card
sudo umount <sd-card-partition-name>
# Example:
sudo umount /dev/mmcblk1p1

# run disk check on sd card
sudo e2fsck <sd-card-device-name>
# Example:
sudo e2fsck /dev/mmcblk1

```