How to flash image

Jack YC Lee

Before flashing image to NVIDIA Jetson Family, you should prepare the following items:

1.**Host PC** with **Ubuntu** operating system.

2.A good quality **OTG cable (micro-usb/USB type C)** for connecting to NVIDIA Jetson Family (**Client PC**).

Complete the following steps on the **Client PC** before performing the host PC steps.

Client PC

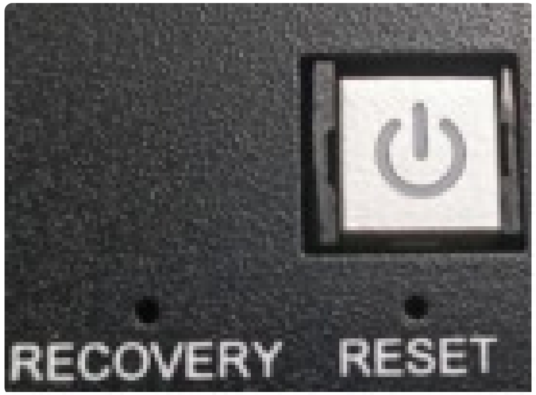
1. Connect the OTG cable to the USB port.
2. Enter Recovery Mode

    Method 1: (Power on)

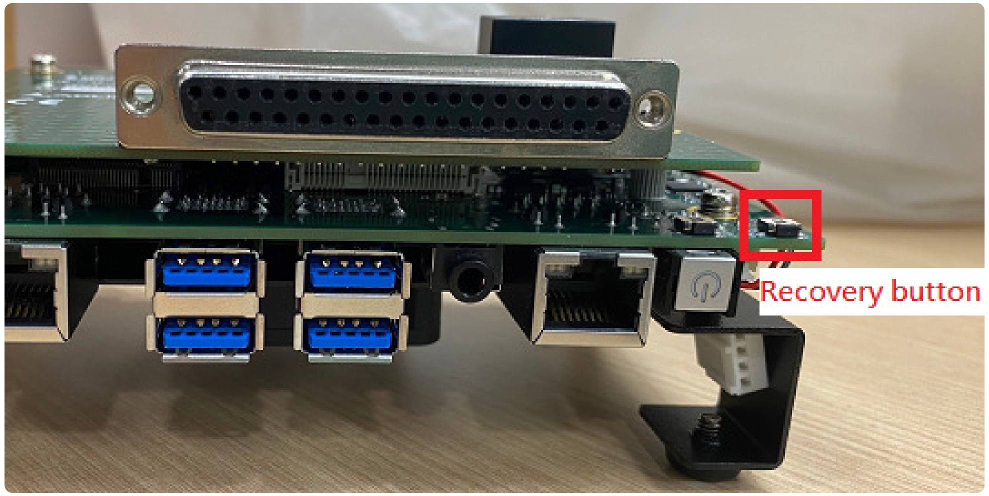
    ```
    $ sudo reboot --force forced-recovery
    ```

    Method 2: (Power off)

    1. Press and hold recovery button.
    2. Press the power switch.
    3. Press reset button once and release then wait for 3-5 seconds release recovery button.



The location of Recovery button



The location of Recovery button on Development Kit

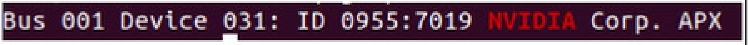
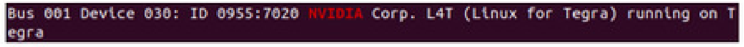
Host PC

- The host must have the following dependencies:

```
$ sudo apt install libxml2-utils simg2img network-manager abootimg sshpass device-tree-compiler nfs-kernel-server
```

1. Connect the host PC to the Client PC via OTG cable.
2. Open a terminal, run following command and check out the mode on Host PC:

```
$ lsusb | grep NVIDIA
```

Recovery mode will show APX.	Normal mode will show L4T.
 Bus 001 Device 031: ID 0955:7019 NVIDIA Corp. APX	 Bus 001 Device 030: ID 0955:7020 NVIDIA Corp. L4T (Linux for Tegra) running on T egra

- **We need to make sure the client PC in recovery mode before flash image.**
- **If client PC doesn't recognize or not in recovery mode, double-check the OTG cable connection and perform troubleshooting as needed until the client is recognized and in recovery mode.**

3. Download mfi file image into the Host PC.

```
ex:  mfi_jetson-orin-nano-dlap211-onx_16-35.3.1-2v1.2.tbz2
```

4. Unzip mfi file that we download previously.

```
ex: $ sudo tar -jxf mfi_jetson-orin-nano-dlap211-onx_16-35.3.1-2v1.2.tbz2
```

5. Navigate to working folder.

```
ex: $ cd mfi_jetson-orin-nano-dlap211-onx_16-35.3.1-2v1.2
```

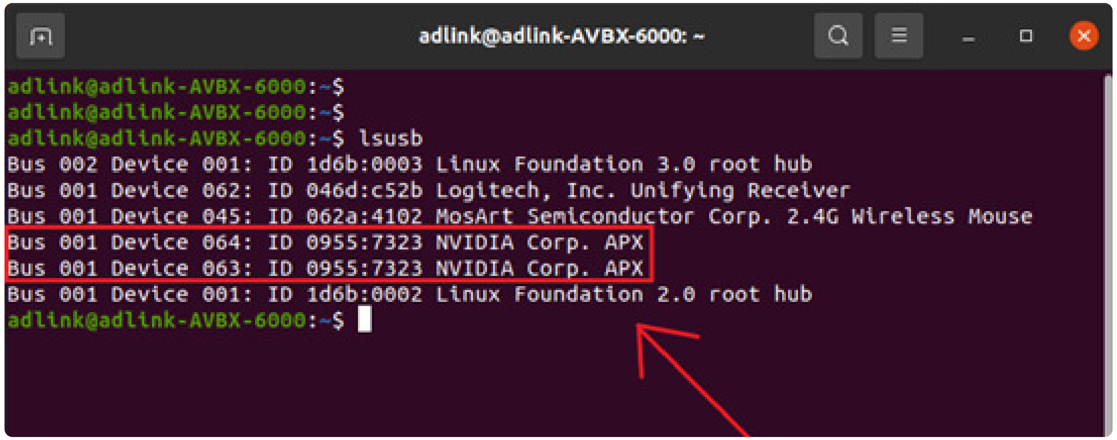
6. Execute flash command

```
ex: $ sudo ./tools/kernel_flash/l4t_initrd_flash.sh --flash-only
```

The following will describe how to do mass flash.

1. Connect **Client PCs** to **Host PC**

2. In terminal, run **lsusb** to check all Client PCs enter into Recovery Mode.

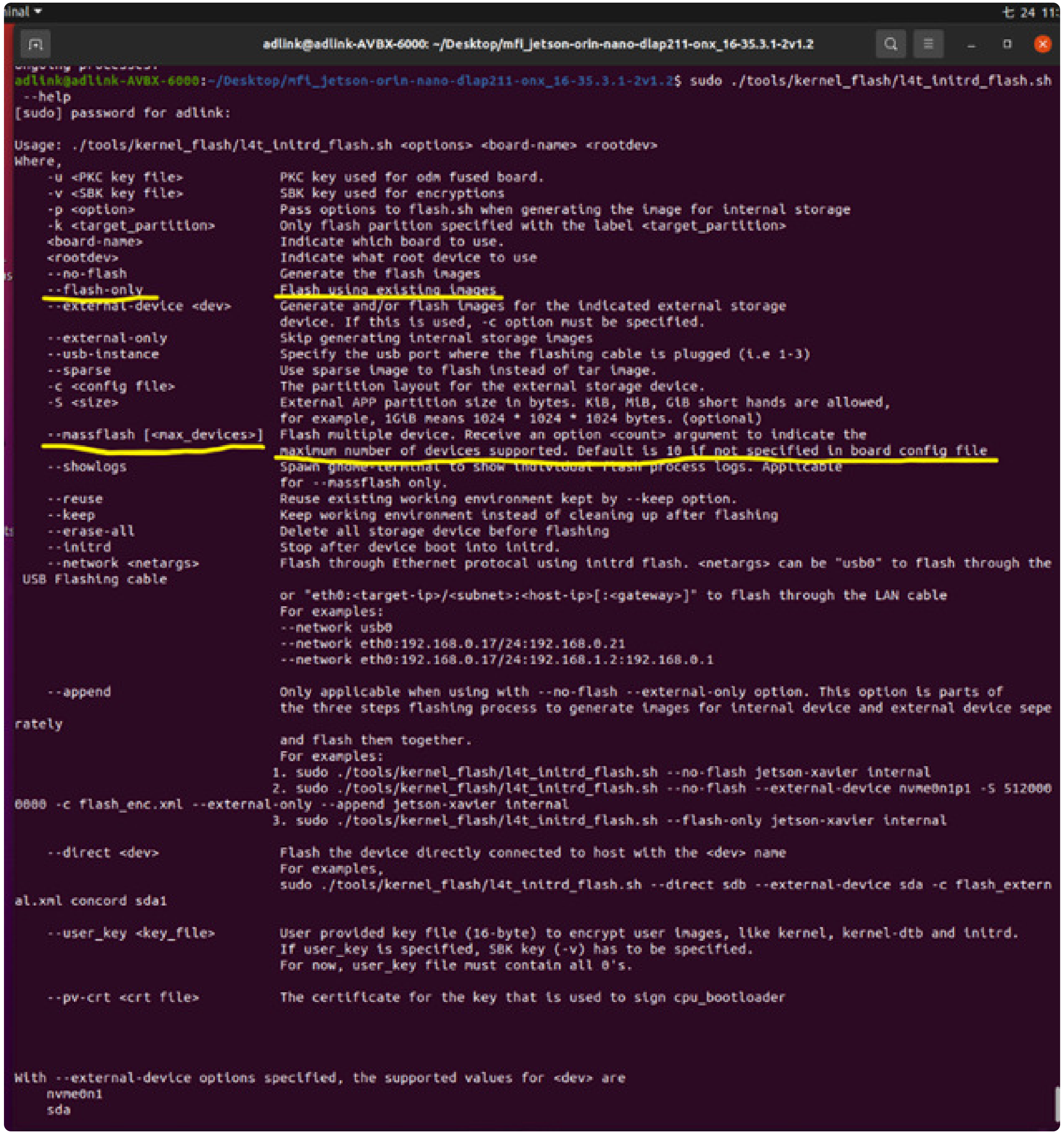


```
adlink@adlink-AVBX-6000:~$
adlink@adlink-AVBX-6000:~$
adlink@adlink-AVBX-6000:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 062: ID 046d:c52b Logitech, Inc. Unifying Receiver
Bus 001 Device 045: ID 062a:4102 MosArt Semiconductor Corp. 2.4G Wireless Mouse
Bus 001 Device 064: ID 0955:7323 NVIDIA Corp. APX
Bus 001 Device 063: ID 0955:7323 NVIDIA Corp. APX
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
adlink@adlink-AVBX-6000:~$
```

3. Execute flash command,

```
$ sudo ./tools/kernel_flash/l4t_initrd_flash.sh --flash-only --massflash 2 (2 as number of client PCs need flash)
```

Reference:


adlink@adlink-AVBX-6000: ~/Desktop/mfi_jetson-orin-nano-dlap211-onx_16-35.3.1-2v1.2

```
adlink@adlink-AVBX-6000:~/Desktop/mfi_jetson-orin-nano-dlap211-onx_16-35.3.1-2v1.2$ sudo ./tools/kernel_flash/l4t_initrd_flash.sh
 --help
[sudo] password for adlink:

Usage: ./tools/kernel_flash/l4t_initrd_flash.sh <options> <board-name> <rootdev>
Where,
    -u <PKC key file>            PKC key used for odm fused board.
    -v <SBK key file>            SBK key used for encryptions
    -p <option>                  Pass options to flash.sh when generating the image for internal storage
    -k <target_partition>        Only flash parition specified with the label <target_partition>
    <board-name>                 Indicate which board to use.
    <rootdev>                    Indicate what root device to use
    --no-flash                   Generate the flash images
    --flash-only                 Flash using existing images
    --external-device <dev>      Generate and/or flash images for the indicated external storage
                                 device. If this is used, -c option must be specified.
    --external-only              Skip generating internal storage images
    --usb-instance               Specify the usb port where the flashing cable is plugged (i.e 1-3)
    --sparse                     Use sparse image to flash instead of tar image.
    -c <config file>             The partition layout for the external storage device.
    -S <size>                    External APP partition size in bytes. KiB, MiB, GiB short hands are allowed,
                                 for example, 1GiB means 1024 * 1024 * 1024 bytes. (optional)
    --massflash [<max_devices>]  Flash multiple device. Receive an option <count> argument to indicate the
                                 maximum number of devices supported. Default is 10 if not specified in board config file
    --showlogs                   Spawn gnome-terminal to show individual flash process logs. Applicable
                                 for --massflash only.
    --reuse                      Reuse existing working environment kept by --keep option.
    --keep                       Keep working environment instead of cleaning up after flashing
    --erase-all                  Delete all storage device before flashing
    --initrd                     Stop after device boot into initrd.
    --network <netargs>          Flash through Ethernet protocal using initrd flash. <netargs> can be "usb0" to flash through the
 USB Flashing cable

                                 or "eth0:<target-ip>/<subnet>:<host-ip>[:<gateway>]" to flash through the LAN cable
                                 For examples:
                                 --network usb0
                                 --network eth0:192.168.0.17/24:192.168.0.21
                                 --network eth0:192.168.0.17/24:192.168.1.2:192.168.0.1

    --append                     Only applicable when using with --no-flash --external-only option. This option is parts of
                                 the three steps flashing process to generate images for internal device and external device sepe
rately
                                 and flash them together.
                                 For examples:
                                1. sudo ./tools/kernel_flash/l4t_initrd_flash.sh --no-flash jetson-xavier internal
                                2. sudo ./tools/kernel_flash/l4t_initrd_flash.sh --no-flash --external-device nvme0n1p1 -S 512000
0000 -c flash_enc.xml --external-only --append jetson-xavier internal
                                3. sudo ./tools/kernel_flash/l4t_initrd_flash.sh --flash-only jetson-xavier internal

    --direct <dev>               Flash the device directly connected to host with the <dev> name
                                 For examples,
                                 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --direct sdb --external-device sda -c flash_extern
al.xml concord sda1

    --user_key <key_file>        User provided key file (16-byte) to encrypt user images, like kernel, kernel-dtb and initrd.
                                 If user_key is specified, SBK key (-v) has to be specified.
                                 For now, user_key file must contain all 0's.

    --pv-crt <crt file>          The certificate for the key that is used to sign cpu_bootloader



With --external-device options specified, the supported values for <dev> are
    nvme0n1
    sda
```