

## 2.2 Download official image and create Jetbot firmware

### 1. Component Environment Analysis:

```
NVIDIA Jetson NANO/TX1 - Jetpack 4.2 [L4T 32.1.0]

- Up Time: 0 days 0:46:54
- Board:
  * Name:          NVIDIA Jetson NANO/TX1
  * Type:          NANO/TX1
  * Jetpack:       4.2 [L4T 32.1.0]
  * GPU-Arch:     5.3
  * SN:            04213190337910400400
- Libraries:
  * CUDA:          10.0.166
  * cuDNN:         7.3.1.28-1+cuda10.0
  * TensorRT:      5.0.6.3-1+cuda10.0
  * VisionWorks:   1.6.0.500n
  * OpenCV:        3.3.1 compiled CUDA: NO
- Hostname: jetbot
- Interfaces
  * l4tbr0:        192.168.55.1
  * wlan0:         192.168.1.67
```

The version information shown in the above figure is the component environment of the official Jetson nano image that was updated from the NVIDIA official website.

It is also the official firmware component environment of the Yahboom jetbot robot car and this component version environment has passed the full-featured test.

If the user needs to follow the tutorial given by Yahboom officially to build the new firmware environment from the official to build the Jetbot firmware environment.

In order to avoid the situation of wasting user time due to version compatibility problems in the setup environment, the user time is wasted.

**Please use the officially provided image on May 31th, 2019 or image before this date.**

For example: the following figure shows the component environment of the Jetson Nano official image that was updated by the official version of Nevada on July 16th, 2019. The Jetpack, CUDA, cuDNN and TensorRT versions are upgraded in the environment firmware updated on May 31th, 2019.

When using the actual functionality of Jetbot robot car, the code cannot find the `create_execution_context()` object property function using the installed TensorRT occurrence function.

Jetson nano Nvidia official developer kit SD card image download address:

<https://developer.nvidia.com/embedded/downloads#?search=Jetson%20Nano>

```
NVIDIA Jetson NANO/TX1 - Jetpack 4.2.1 [L4T 32.2.0]
- Up Time: 0 days 0:18:37
- Board:
  * Name:          NVIDIA Jetson NANO/TX1
  * Type:          NANO/TX1
  * Jetpack:       4.2.1 [L4T 32.2.0]
  * GPU-Arch:     5.3
  * SN:            04213190337910400400
- Libraries:
  * CUDA:          10.0.326
  * cuDNN:         7.5.0.56-1+cuda10.0
  * TensorRT:      5.1.6.1-1+cuda10.0
  * VisionWorks:   1.6.0.500n
  * OpenCV:        3.3.1 compiled CUDA: NO
Hostname: yahboom
- Interfaces
  * l4tbr0:        192.168.55.1
  * wlan0:         192.168.1.67
```

## 2. Burn image into Micro SD card

We need to prepare a PC and a Micro SD card reader.

**Path of image : [Jetbot-AI Car] --> [8.System Image]**

### 2.1 About image:

2.1.1 Yahboom\_jetbot\_image is an environment that has been configured with an environment and can be used directly.

2.2.2 NVIDIA's official image is an image that is not configured with an environment. Users need to try to set up the environment.

### 2.2 About burn image:

2.2.1 Download the Jetson Nano Developer Kit SD card image according to the address provided above and note where it is stored on your computer.

2.2.2 Depending on the type of computer you are using: Windows, Mac or Linux, follow the instructions below to write the image to the Micro SD card.

#### Windows:

Before writing the system image, you need to format your SD card by SD Card Formatter.

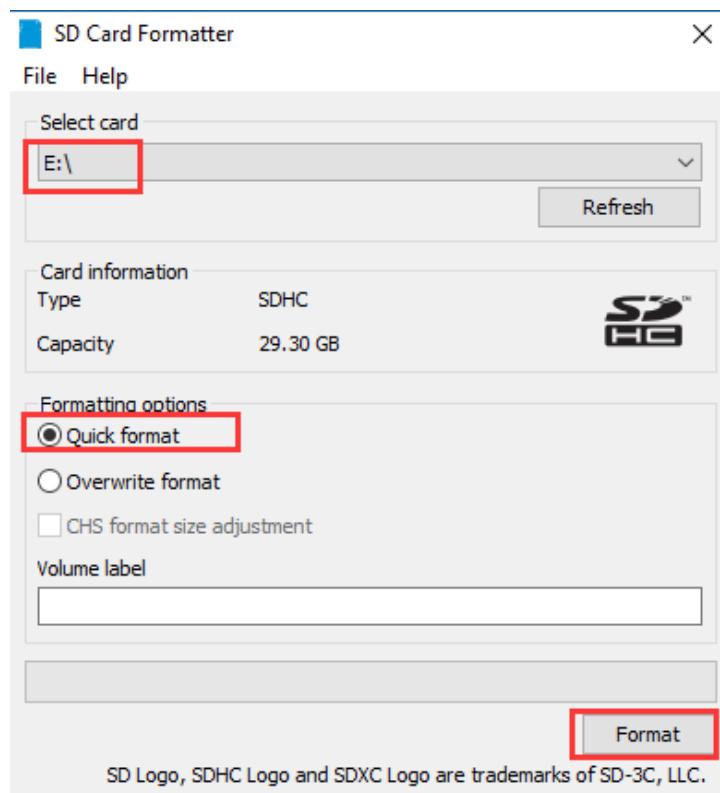
#### Install Formatter:

**Path of package : [Jetbot-AI Car] --> [Annex] --> [Tools]--> [SD Card Formatter]**

#### Format SD card:

1)Insert the SD card into the computer by the card reader, you will see that the

computer has one more drive letter.



**Note:**If any warnings appear, please select "Yes/OK".

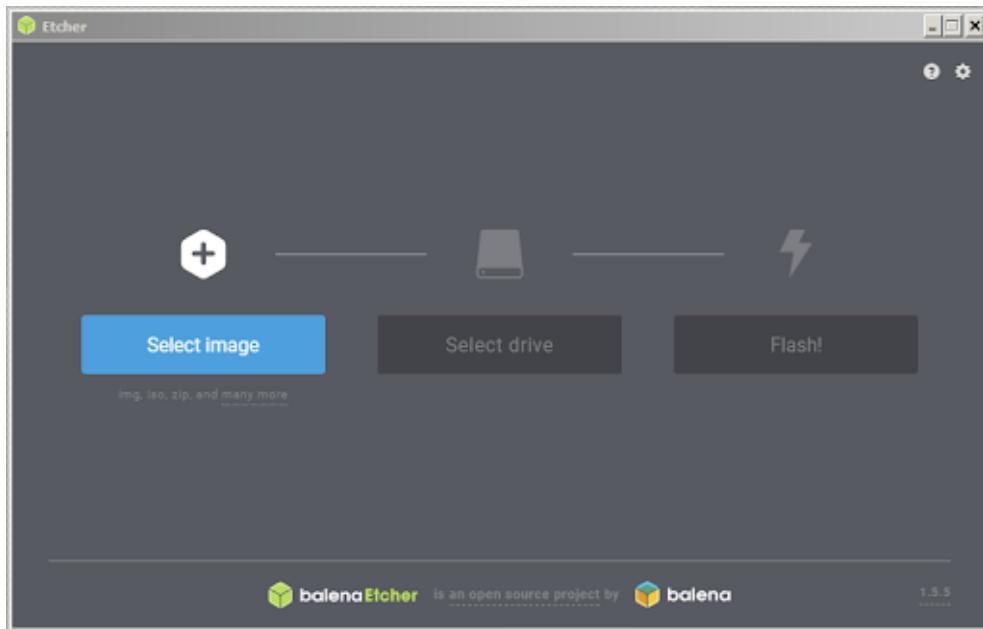
2) Write the Jetson Nano Developer Kit SD card image to the Micro SD card using Etcher.

**Install Etcher:**

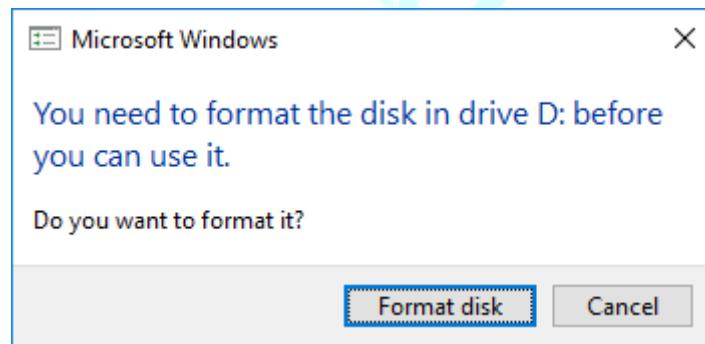
**Path of package :** [Jetbot-AI Car] --> [Annex] --> [Tools]--> [balenaEtcher-Portable]

**Burn image:**

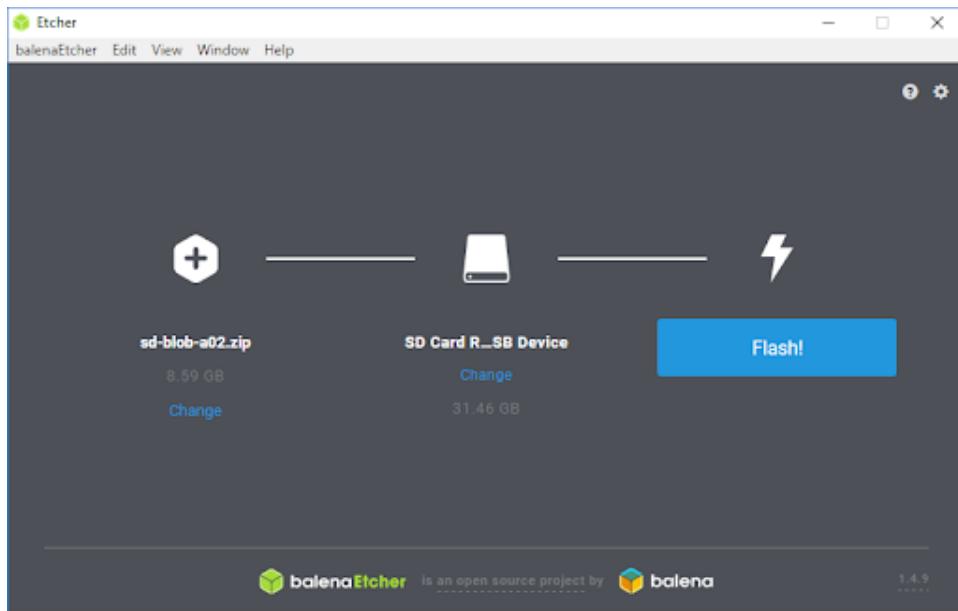
Insert the formatted SD card into the computer through the card reader.



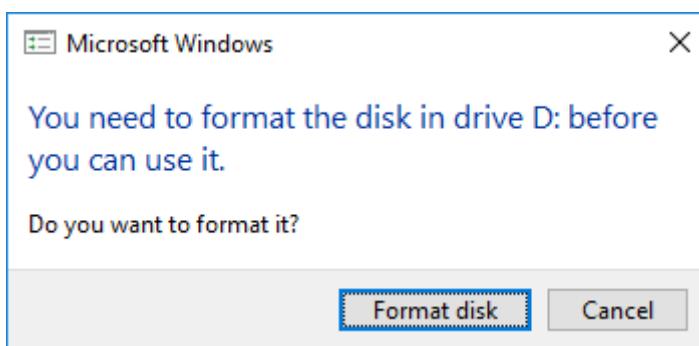
- 1) Click "Select image" and select the compressed image file you downloaded earlier.
- 2) If need to insert microSD card. If Windows prompts you to use the following dialog, please click "Cancel" (according to this description):



- 3) Click "Select Drive" and select the correct device.



- 4) Click on "Flash!" If your microSD card is connected via USB3, Etcher will take about 10 minutes to write and verify the image.
- 5) After the Etcher is complete. If Windows prompts you to use the following dialog, please click "Cancel" (according to this description):



- 6) Physically remove the microSD card from the computer.

## MAC

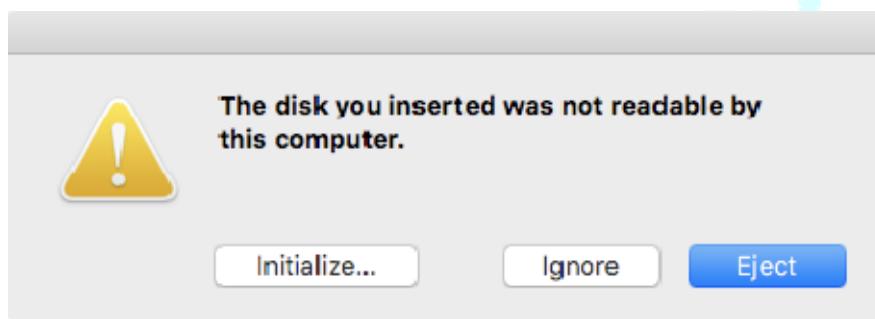
You can write an SD card image using a graphics program such as Etcher or a command line.

### Method 1:

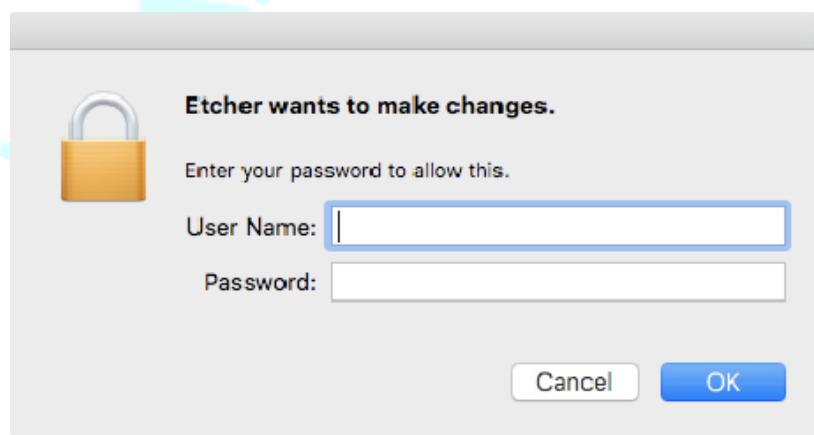
- 1) **Do not insert a microSD card.**
- 2) Download, install and start the Etcher.



- 3) Click "Select Image" and select the compressed image file you downloaded earlier.
- 4) **Insert the micro SD card.** If your Mac displays this window, click "Ignore":

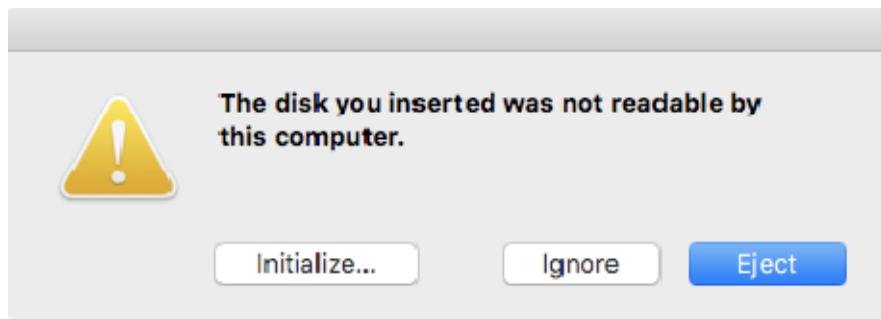


- 5) If you are not connected to another external drive, Etcher will automatically select the microSD card as the target device. Otherwise, click "Select Drive" and select the correct device.
- 6) Click on "Flash!" Your Mac may prompt you for a username and password before allowing Etcher to continue.



If your microSD card is connected via USB3, Etcher will take about 10 minutes to write and verify the image.

- 7) After the Etcher is finished, if Mac prompts you to use the following dialog, please click "Ignore" (according to this description):



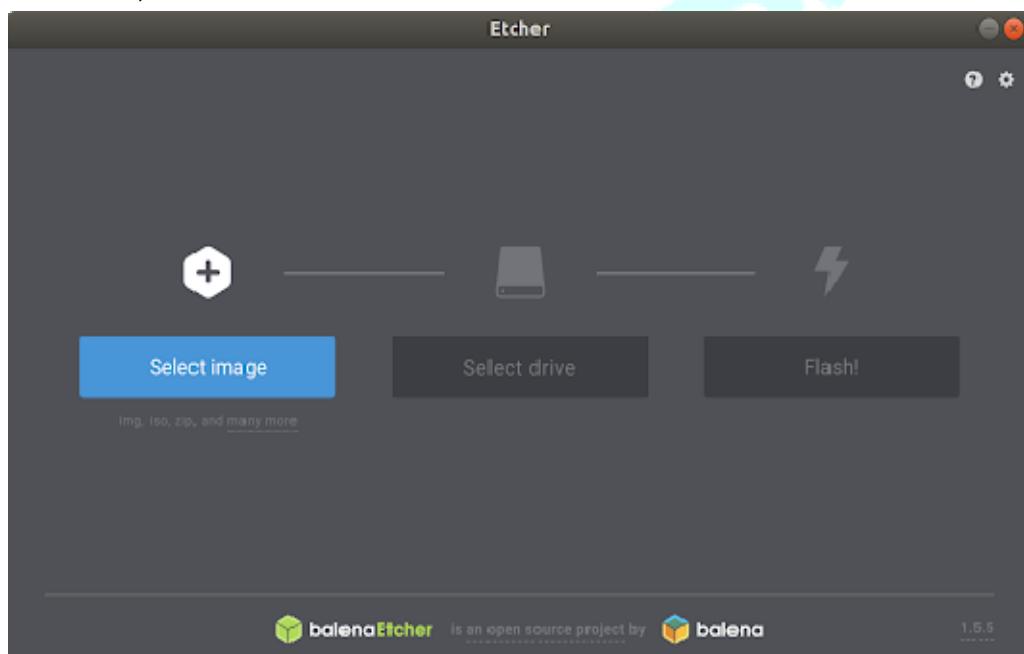
- 8) Physically remove the microSD card from the computer.

## LINUX

You can write an SD card image using a graphics program such as Etcher or a command line.

### Method 1--- Etcher

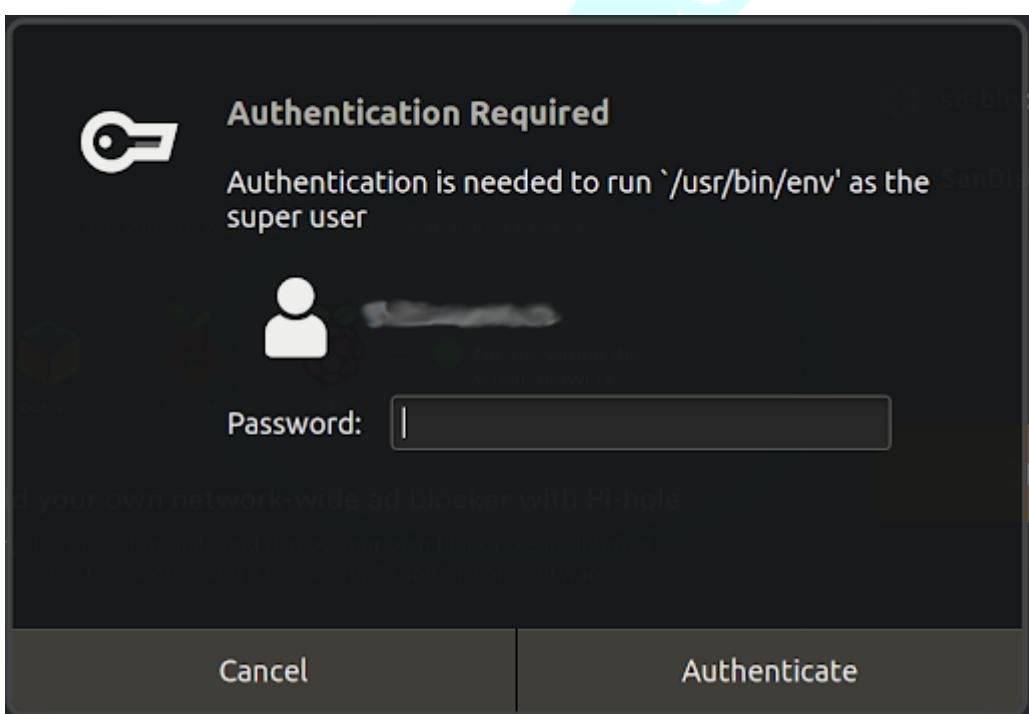
- 1)Download, install and start the Etcher.



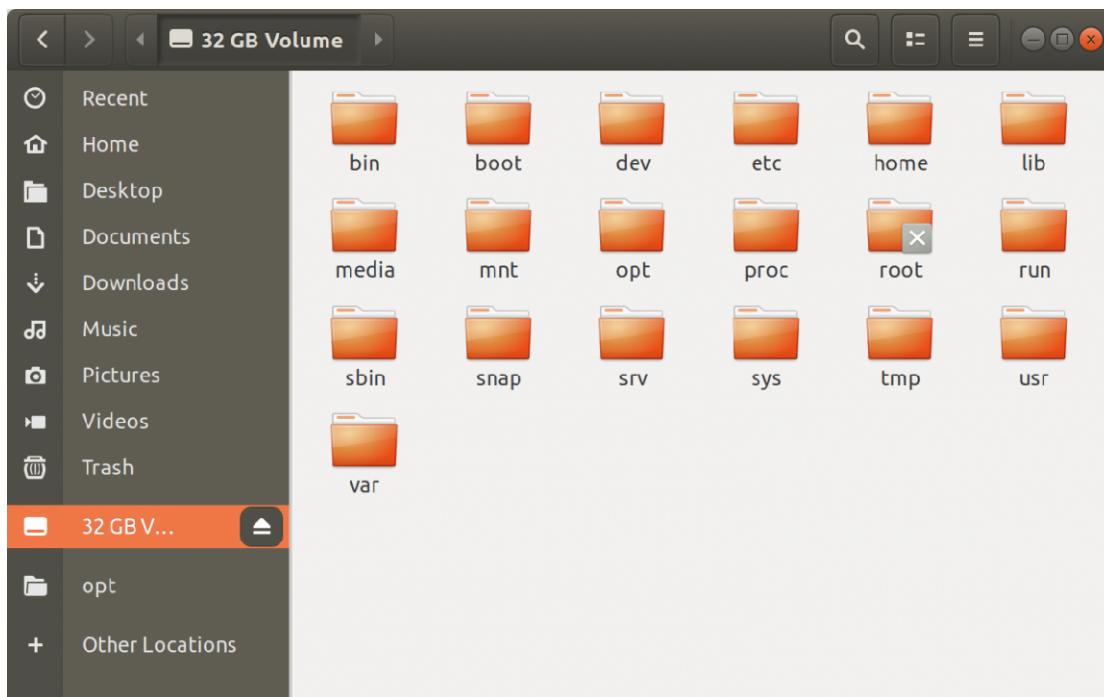
- 2)Click "Select Image" and select the compressed image file you downloaded earlier.
- 3)**Insert the micro SD card.** If you are not connected to another external drive, Etcher will automatically select the microSD card as the target device. Otherwise, click "change" and select the correct device.



- 4) Click on "Flash!" Your operating system may prompt you for a username and password before allowing Etcher to continue.



- If your microSD card is connected via USB3, Etcher will take about 10 minutes to write and verify the image.
- 5) After the Etcher is finished, use the Files application to eject the SD card:
  - 6) Physically remove the microSD card from the computer.



### Method 2--- command line

- 1) Press **Ctrl+Alt+t** to open the terminal application.
- 2) Insert the microSD card and use the command to display the disk device assigned to it:

```
dmesg | tail | awk '$3 == "sd" {print}'
```

In this example, we can see that the 16GB micro SD card is assigned /dev/sda:

```
Terminal
$ dmesg | tail | awk '$3 == "sd" {print}'
[3932.318223] sd 0:0:0:0: Attached scsi generic sg0 type 0
[3932.869891] sd 0:0:0:0: [sda] 31275008 512-byte logical blocks: (16.0 GB/14.9 GiB)
[3932.870275] sd 0:0:0:0: [sda] Write Protect is off
[3932.870279] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[3932.870733] sd 0:0:0:0: [sda] No Caching mode page found
[3932.870742] sd 0:0:0:0: [sda] Assuming drive cache: write through
[3932.880136] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

- 3) Use this command to write the compressed image to the microSD card:

```
/usr/bin/unzip -p ~/Downloads/jetson_nano_devkit_sd_card.zip | sudo /bin/dd of=/dev/sd<x> bs=1M status=progress
```

E.g:

```
Terminal
$ /usr/bin/unzip -p ~/Downloads/jetson_nano_devkit_sd_card.zip |
> sudo /bin/dd of=/dev/sda bs=1M status=progress
12867141632 bytes (13 GB, 12 GiB) copied, 428 s, 30.1 MB/s
1+171572 records in
1+171572 records out
12884901888 bytes (13 GB, 12 GiB) copied, 428.584 s, 30.1 MB/s
$
```

After the command is completed, eject the disk device from the command line:

```
sudo eject /dev/sd<x>
```

4)Physically remove the microSD card from the computer.

### 3. Install the required components

#### 3.1 Install pip

Because Python 3.6 is already installed in Jetson Nano, installing pip is relatively simple.

Input this command install: `sudo apt-get install python3-pip python3-dev`

After installation, pip is version 9.0.1, you need to upgrade it to the latest version.

Input this command to update: `python3 -m pip install --upgrade pip`

After upgrading, the pip version is 19.1.1.

And there will be a BUG after the upgrading, you need to manually change it.

As shown below:

```
nano@nano-desktop:~$ python3 -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/5c/e0/be401c003291b56efc55
  aeba6a80ab790d3d4cece2778288d65323009420/pip-19.1.1-py2.py3-none-any.whl (1.4MB)
    100% |████████████████████████████████| 1.4MB 84kB/s
Installing collected packages: pip
Successfully installed pip-19.1.1
```

Input this command: `sudo vim /usr/bin/pip3`

#### Original:

```
-----
from pip import main
if __name__ == '__main__':
    sys.exit(main())
-----
```

#### After modification:

```
-----
from pip import __main__
if __name__ == '__main__':
    sys.exit(__main__.main())
-----
```

Save after the modification is completed.

Input this command: `pip3 -V`

It will bring up the interface as shown below:

```
nano@nano-desktop:~$ pip3 -V
pip 19.1.1 from /home/nano/.local/lib/python3.6/site-packages/pip (python 3.6)
```

### 3.2 Install Jetbot Dependency package

Here are the packages that need to be installed when Jetbot is running:

Name	Version
Adafruit-GPIO	1.0.4
Adafruit-MotorHAT	1.4.0
Adafruit-PureIO	0.2.3
Adafruit-SSD1306	1.6.2
Flask	1.1.1
Jetson.GPIO	1.0.0
numpy	1.16.4

#### Local installation:

If you use the command to find the package in the source or the network environment is not good, we have provide the Jetbot dependency package, transfer the local package to the Jetbot file system by WinSCP.

**Path of package : [Jetbot-AI Car] --> [Annex] --> [Jetbot Dependency package]**

Then, we can input: `cd xxxx` jumps to the directory where you need to install the package (the directory with the setup.py file)

Input this command: `sudo python3 setup.py install`

Install the package.

#### Online installation:

Use the following command to install the corresponding package, (xxx is name of package you need to install)

`sudo pip3 install xxx`

If there is a compatibility issue with the new version of the installation, you can use the following command to install the dependencies of the version specified in the above listed package.

`sudo pip3 install package==version0`

(For example, you need to specify numpy 1.16.4 , you need to input command:  
`sudo pip install numpy==1.16.4`)

Install traitlets:

`sudo python3 -m pip install git + https://github.com/ipython/traitlets@master`

Install the cmake software:

`sudo apt-get install cmake`

### 3.3 Modified to Jetbot custom driver library

If you use the `pip install package` command (online installation) to install from

the Ubuntu source, because the Yahboom-Jetbot hardware and the driver package downloaded by Adafruit-MotorHAT are different, you need to modify or replace the installed driver file.

The replacement steps are as follows: :

- ① Modify the Adafruit\_MotorHAT\_Motors.py file permissions in the Adafruit-MotorHAT driver file directory that has been installed.

Input command:

```
sudo chmod 777
```

```
/usr/local/lib/python3.6/dist-packages/Adafruit_MotorHAT-1.4.0-py3.6.egg/Adafruit_MotorHAT/Adafruit_MotorHAT_Motors.py
```

As shown below:

```
sudo chmod 777 /usr/local/lib/python3.6/dist-packages/Adafruit_MotorHAT-1.4.0-py3.6.egg/Adafruit_MotorHAT/Adafruit_MotorHAT_Motors.py
```

- ② Transmits the the custom Adafruit-MotorHAT driver provided in the Jetbot package to the Jetbot file system through WinSCP.

**Path of package : [Jetbot-AI Car] --> [Annex] --> [Jetbot Dependency package]**

- ③ Replace our previously installed Adafruit\_MotorHAT\_Motors.py with the Adafruit\_MotorHAT\_Motors.py Yahboom provided by command:

Eg: the custom Adafruit-MotorHAT driver provided by yahboom officially is placed in the home directory.

So need to input the following command:

```
sudo cp ~/Adafruit_MotorHAT/Adafruit_MotorHAT_Motors.py  
/usr/local/lib/python3.6/dist-packages/Adafruit_MotorHAT-1.4.0-py3.6.egg/Adafruit_MotorHAT/Adafruit_MotorHAT_Motors.py
```

As shown below:

```
sudo cp ~/Adafruit_MotorHAT/Adafruit_MotorHAT_Motors.py /usr/local/lib/python3.6/dist-packages/Adafruit_MotorHAT-1.4.0-py3.6.egg/Adafruit_MotorHAT/Adafruit_MotorHAT_Motors.py
```

### 3.2.4 Making swap space to increase memory

Use the following command to increase the 4G size swap file:

```
sudo fallocate -l 4G /var/swapfile
```

```
sudo chmod 600 /var/swapfile
```

```
sudo mkswap /var/swapfile
```

```
sudo swapon /var/swapfile
```

```
sudo bash -c 'echo "/var/swapfile swap swap defaults 0 0" >> /etc/fstab'
```

