

# Jetson Nano 환경설정 & Raspberry Pi Cam 연동

인턴 송진호/이우림

2021.01.20

# Jetson Nano 환경설정

---

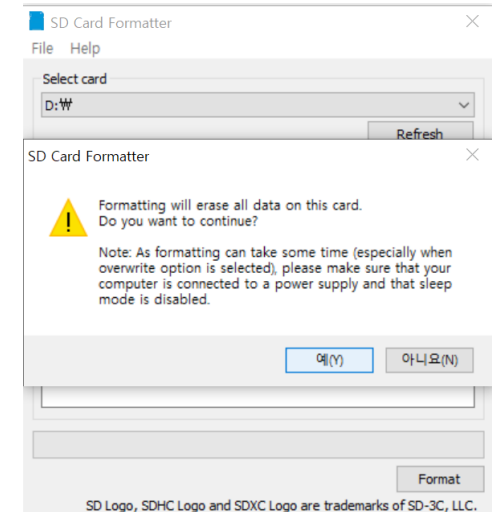
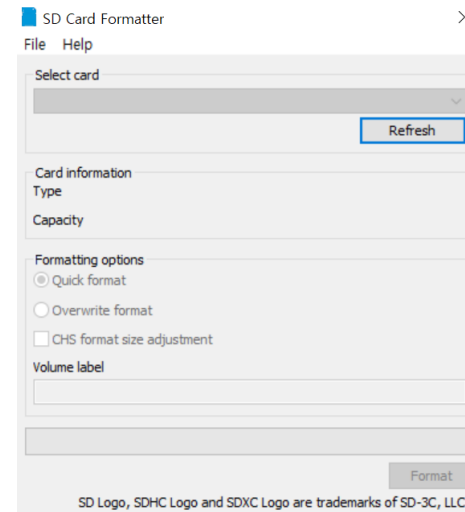
- SD card 포맷
- Jetson Nano
- 메모리 환경 늘리기
- OpenCV 설치
- Python 버전 변경

# SD card 포맷

- SD card formatter 다운로드

: [SD Memory Card Formatter for Windows Download - SD Association \(sdcard.org\)](https://www.sdcard.org/downloads/formatter_4/)

1. SD card 드라이브를 선택한 후 Quick format 을 선택
2. Volume label을 비워 두고 format을 클릭 한 후 '예' 클릭

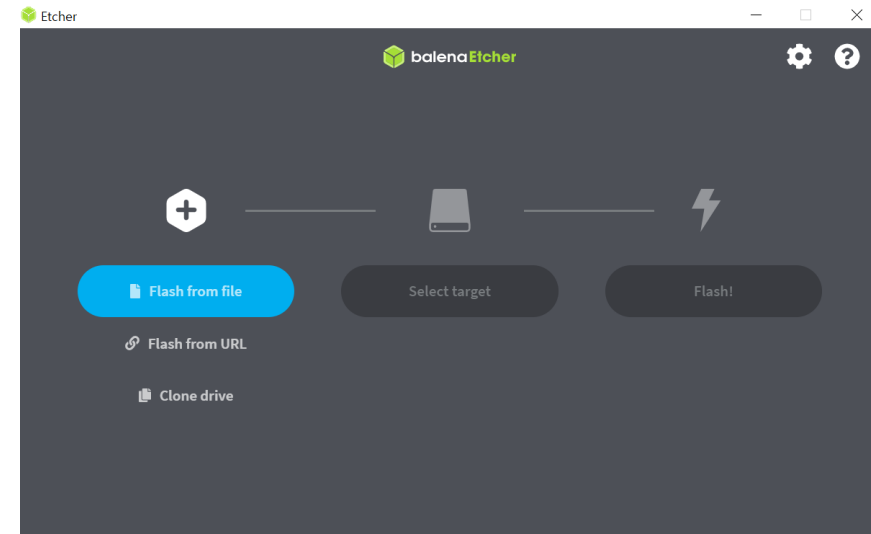


- 이미지(JetPack)/Etcher 다운로드

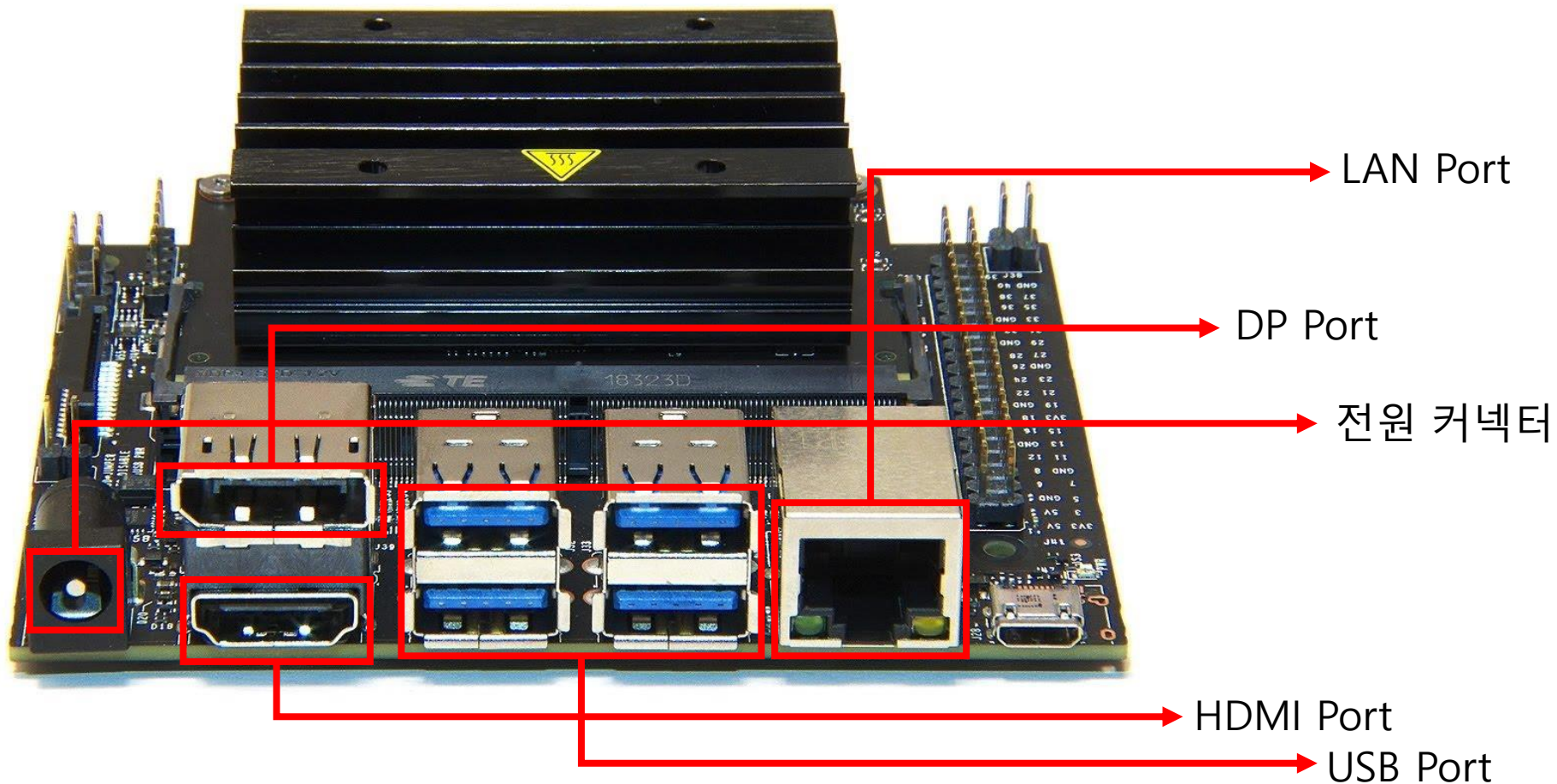
: <https://developer.nvidia.com/jetson-nano-sd-card-image-r3231>

: <https://www.balena.io/etcher>

1. PC에 micro SD card 삽입
2. 이미지 압축파일 선택 후 Select target에서 sd card 선택

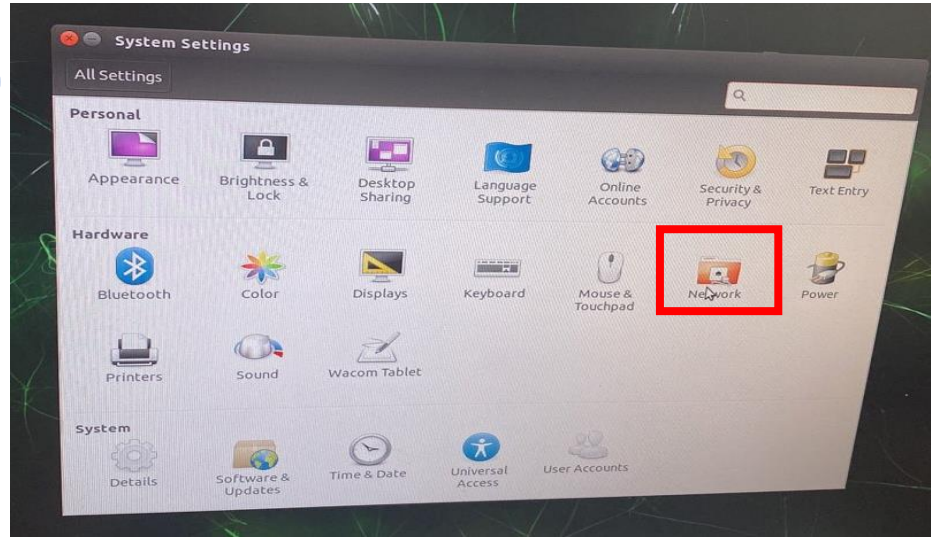


# Jetson Nano

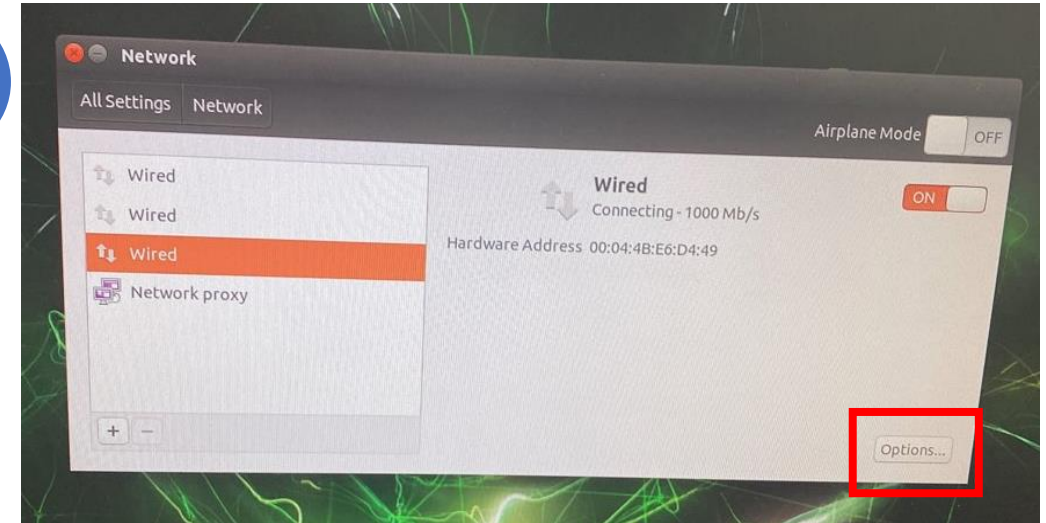


# Jetson Nano

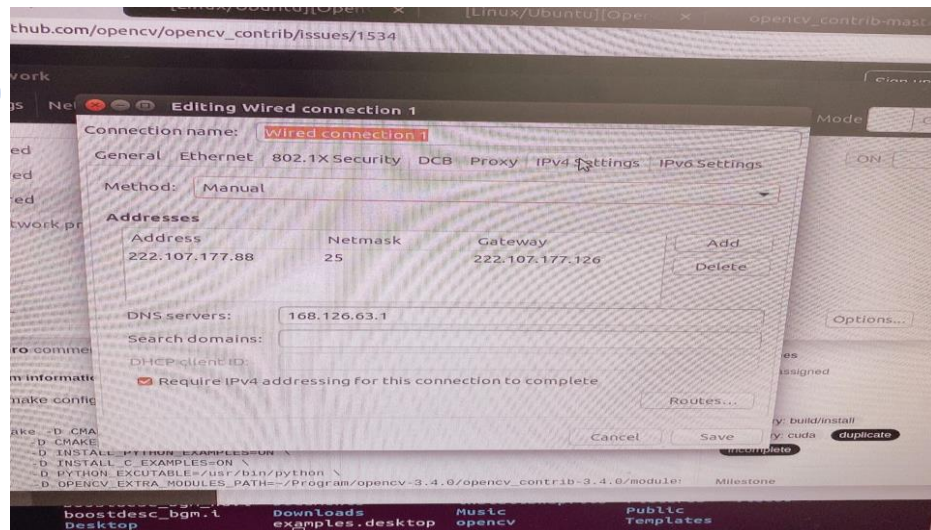
1



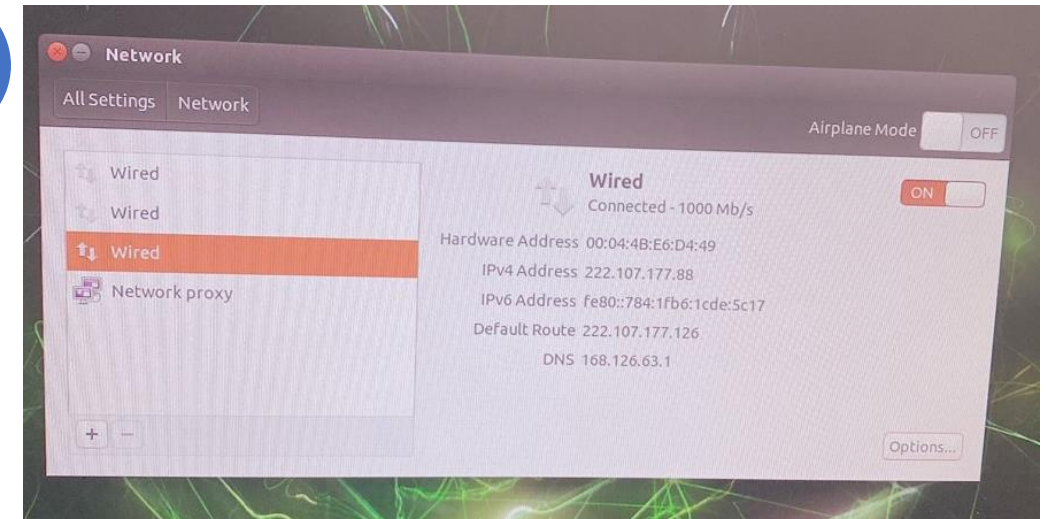
2



3



4





# 메모리 환경 늘리기

---

- OpenCV 설치 과정 중 Jetson Nano가 작동되지 않을 수 있어 가상메모리(swapfile)를 통해 용량을 늘려준다

```
$ git clone https://github.com/JetsonHacksNano/installSwapfile  
$ cd installSwapfile  
$ ./installSwapfile.sh
```

# OpenCV 설치

- Ubuntu 18.04에 기본(Default) OpenCV 제거

```
$ sudo apt-get remove libopencv*  
$ sudo apt-get autoremove  
$ sudo find /usr/local/ -name "*opencv*" -exec rm {} \;
```

- Prerequisites 설치

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ sudo apt-get install build-essential cmake unzip pkg-config
```

# OpenCV 설치

- Libraries 설치

```
$ sudo apt-get install libjpeg-dev libpng-dev libtiff-dev  
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev v4lutils  
libxvidcore-dev libx264-dev libxine2-dev  
$ sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev  
$ sudo apt-get install libgtk-3-dev  
$ sudo apt-get install mesa-utils libgl1-mesa-dri libgtkgl2.0-dev libgtkglext1-dev  
$ sudo apt-get install libatlas-base-dev gfortran libeigen3-dev
```

- Python 설치

```
$ sudo apt-get install python2.7-dev python3-dev python-numpy python3-numpy
```



# OpenCV 설치

- OpenCV 3.4.0 다운로드

```
$ mkdir opencv  
$ cd opencv  
$ wget -O opencv.zip https://github.com/opencv/opencv/archive/3.4.0.zip  
$ wget -O opencv_contrib.zip https://github.com/opencv/opencv\_contrib/archive/3.4.0.zip  
$ unzip opencv.zip  
$ unzip opencv_contrib.zip
```

- Build & install OpenCV

```
$ cd opencv-3.4.0  
$ mkdir build  
$ cd build
```

# OpenCV 설치

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE ₩  
-D CMAKE_INSTALL_PREFIX=/usr/local ₩  
-D WITH_TBB=OFF ₩  
-D WITH_IPP=OFF ₩  
-D WITH_1394=OFF ₩  
-D BUILD_WITH_DEBUG_INFO=OFF ₩  
-D BUILD_DOCS=OFF ₩  
-D INSTALL_C_EXAMPLES=ON ₩  
-D INSTALL_PYTHON_EXAMPLES=ON ₩  
-D BUILD_EXAMPLES=OFF ₩  
-D BUILD_TESTS=OFF ₩  
-D BUILD_PERF_TESTS=OFF ₩  
-D WITH_QT=OFF ₩
```

# OpenCV 설치

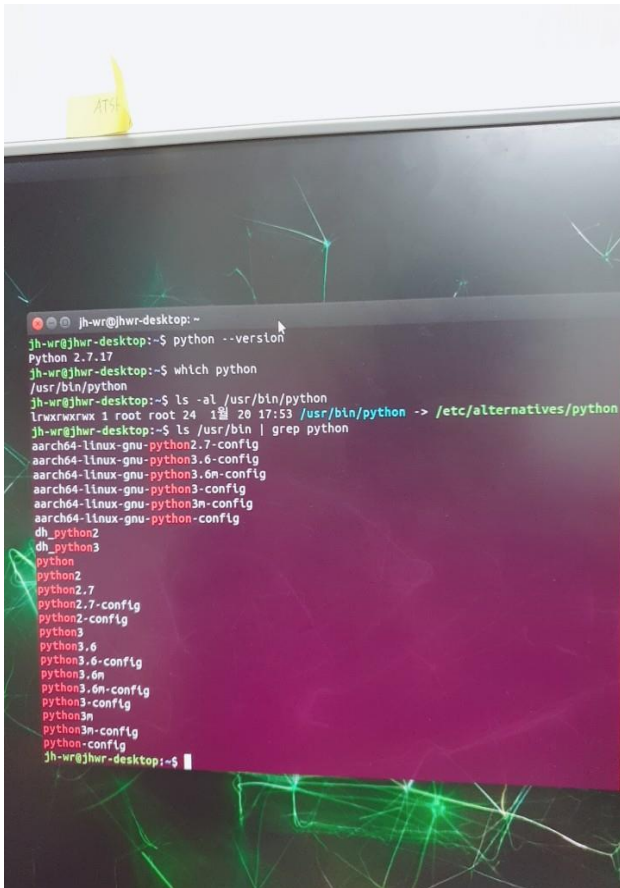
```
-D WITH_GTK=ON ₩  
-D WITH_OPENGL=ON ₩  
-D OPENCV_EXTRA_MODULES_PATH=../opencv_contrib-3.4.0/modules ₩  
-D WITH_V4L=ON ₩  
-D WITH_FFMPEG=ON ₩  
-D WITH_XINE=ON ₩  
-D BUILD_NEW_PYTHON_SUPPORT=ON ₩  
-D PYTHON2_INCLUDE_DIR=/usr/include/python2.7 ₩  
-D PYTHON2_NUMPY_INCLUDE_DIRS=/usr/lib/python2.7/distpackages/numpy/core/include/ ₩  
-D PYTHON2_PACKAGES_PATH=/usr/lib/python2.7/dist-packages ₩  
-D PYTHON2_LIBRARY=/usr/lib/x86_64-linux-gnu/libpython2.7.so ₩  
-D PYTHON3_INCLUDE_DIR=/usr/include/python3.6m ₩  
-D PYTHON3_NUMPY_INCLUDE_DIRS=/usr/lib/python3/distpackages/numpy/core/include/ ₩  
-D PYTHON3_PACKAGES_PATH=/usr/lib/python3/dist-packages ₩
```

# OpenCV 설치

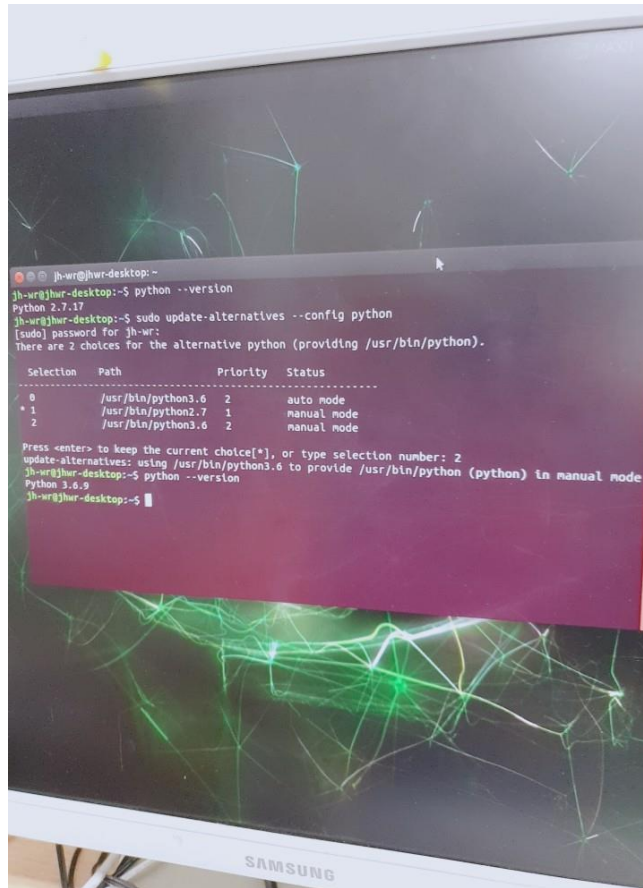
---

```
-D PYTHON3_LIBRARY=/usr/lib/x86_64-linux-gnu/libpython3.6m.so ₩  
../  
$ make -j4
```

# Python 버전 변경



```
jh-wr@jhwr-desktop:~  
jh-wr@jhwr-desktop:~$ python --version  
Python 2.7.17  
jh-wr@jhwr-desktop:~$ which python  
/usr/bin/python  
jh-wr@jhwr-desktop:~$ ls -al /usr/bin/python  
lrwxrwxrwx 1 root root 24 1월 20 17:53 /usr/bin/python -> /etc/alternatives/python  
jh-wr@jhwr-desktop:~$ ls /usr/bin | grep python  
aarch64-linux-gnu-python2.7-config  
aarch64-linux-gnu-python3.6-config  
aarch64-linux-gnu-python3.6n-config  
aarch64-linux-gnu-python3-config  
aarch64-linux-gnu-python3n-config  
dh_python2  
dh_python3  
python  
python2  
python2.7  
python2.7-config  
python2-config  
python3  
python3.6  
python3.6-config  
python3.6n  
python3.6n-config  
python3-config  
python3n  
python3n-config  
python-config  
jh-wr@jhwr-desktop:~$
```



```
jh-wr@jhwr-desktop:~  
jh-wr@jhwr-desktop:~$ python --version  
Python 2.7.17  
jh-wr@jhwr-desktop:~$ sudo update-alternatives --config python  
[sudo] password for jh-wr:  
There are 2 choices for the alternative python (providing /usr/bin/python).  


| Selection | Path               | Priority | Status      |
|-----------|--------------------|----------|-------------|
| 0         | /usr/bin/python3.6 | 2        | auto mode   |
| * 1       | /usr/bin/python2.7 | 1        | manual mode |
| 2         | /usr/bin/python3.6 | 2        | manual mode |

  
Press <enter> to keep the current choice[*], or type selection number: 2  
update-alternatives: using /usr/bin/python3.6 to provide /usr/bin/python (python) in manual mode  
jh-wr@jhwr-desktop:~$ python --version  
Python 3.6.9  
jh-wr@jhwr-desktop:~$
```

`python --version`  
: 파이썬 버전 확인

`Which python`  
: 파이썬이 설치된 위치 확인

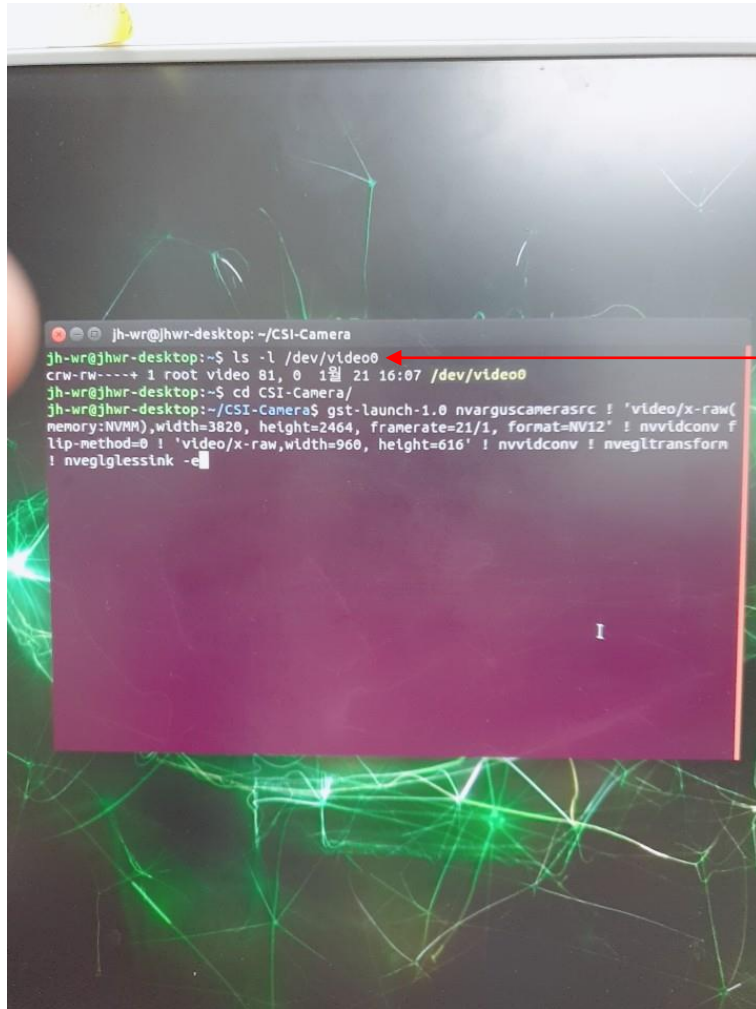
`ls -a` : 숨겨진 파일 및 디렉토리까지 표시

`ls -l` : 권한, 포함된 파일 수, 소유자, 그룹, 파일 크기 등  
파일의 자세한 내용 표시

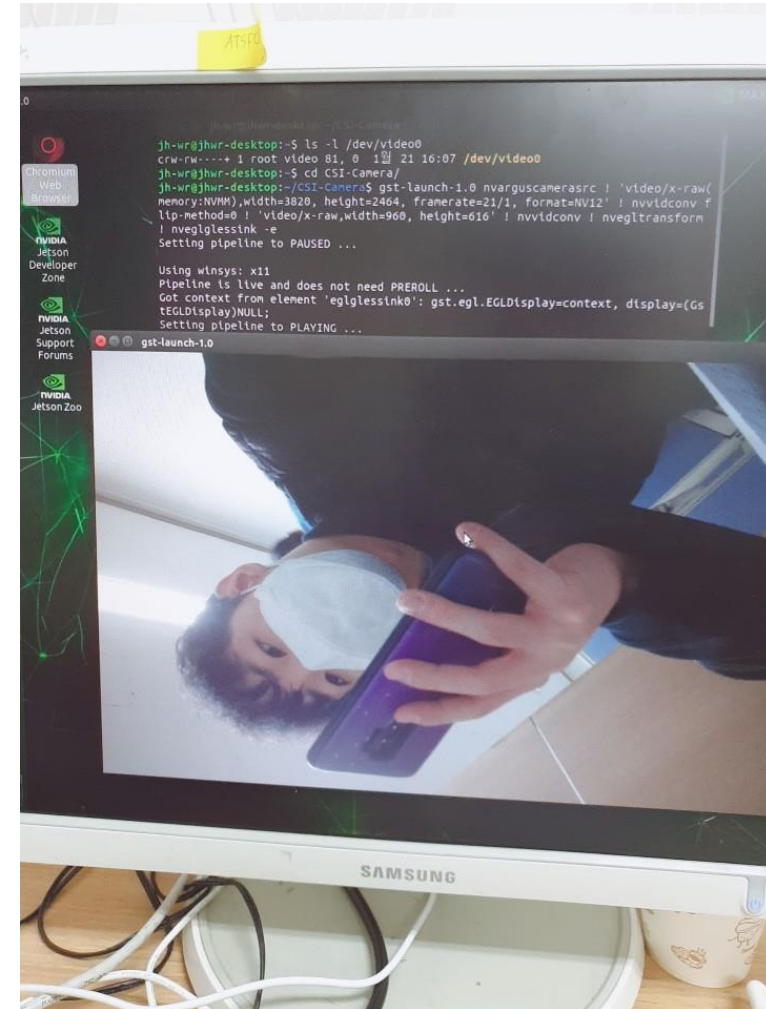
`ls '위치' | grep python` : '위치'에 존재하면서, 파일명에  
python이 들어가는 파일 목록 전체 출력

`sudo update-alternatives --config python`  
: 사용할 파이썬의 버전을 설정

# Jetson Nano & Pi Cam 연동



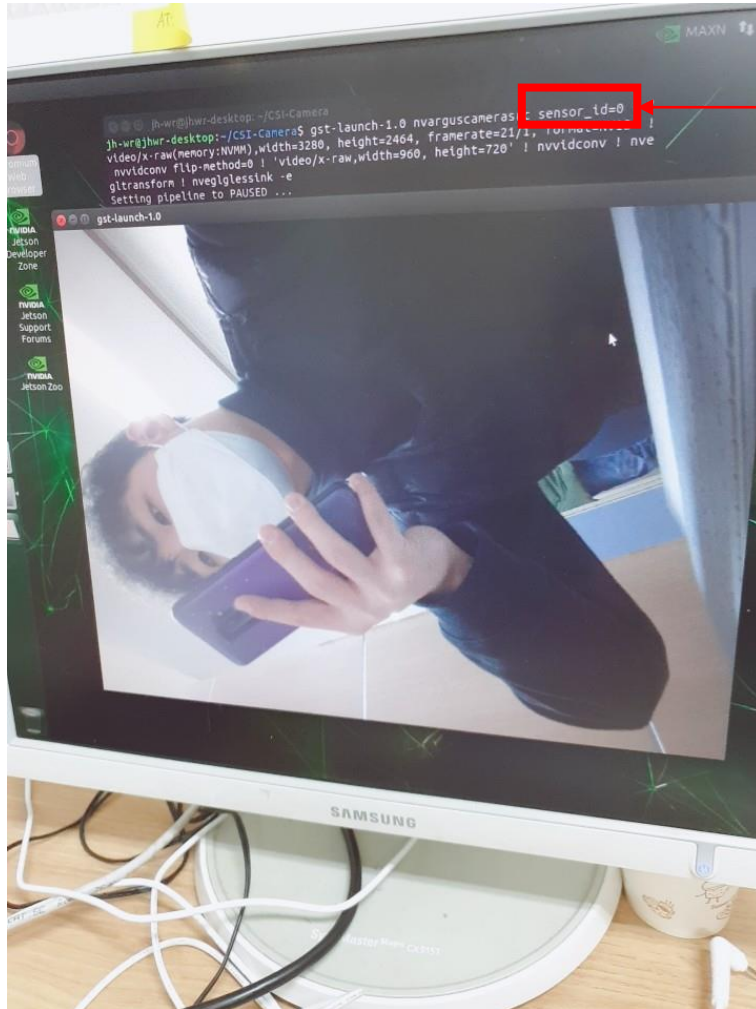
Raspberry Pi Cam을 인식하면 아래 결과를 출력



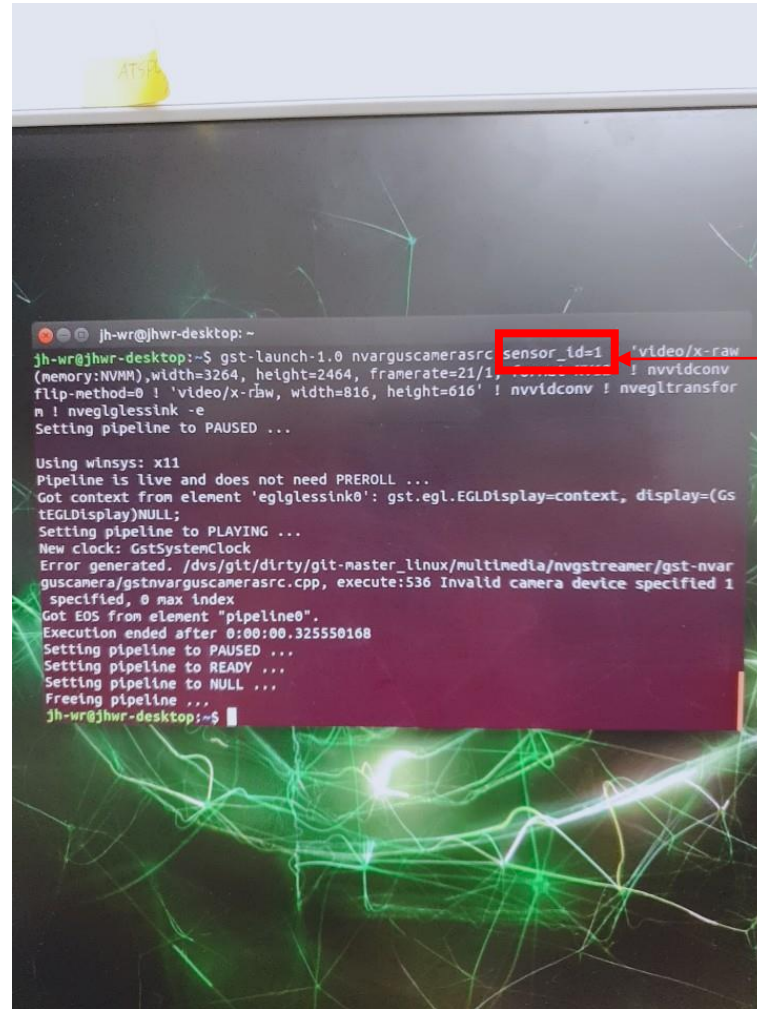
정상 출력 화면



# Jetson Nano & Pi Cam 연동



Sensor\_id가 일치하지  
않으면 카메라 화면  
출력



Sensor\_id가 일치하지  
않으면 카메라 화면이  
출력되지 않고 오류  
발생