COGA-OJT
3rd Week

박성준 인턴

INDEX

COGA OJT 3rd Week Notice

유선 통신: 싱글 보드와 PC(데탑 or 랩탑) 간 내부(유선) 통신망을 구축하여, 카메라 영상을 배포/취득 하는 ROS node 구현.

무선 통신: 싱글 보드와 PC(데탑 or 랩탑) 간 외부(무선) 통신망을 구축(라우터 이용)하여, 카메라 영상을 배포/취득 하는 ROS node 구현.

세부 조건 :

(1) 영상 배포/취득 하는 모든 node들을 하나의 ROS launch 파일로 통합 실행 시켜야 함.

(2) 각 통신 방법에 대하여, publish되는 영상 토픽들의 Hz를 체크 및 비교하여 결과에 포함 시킬 것.

사용된 기기 목록 :

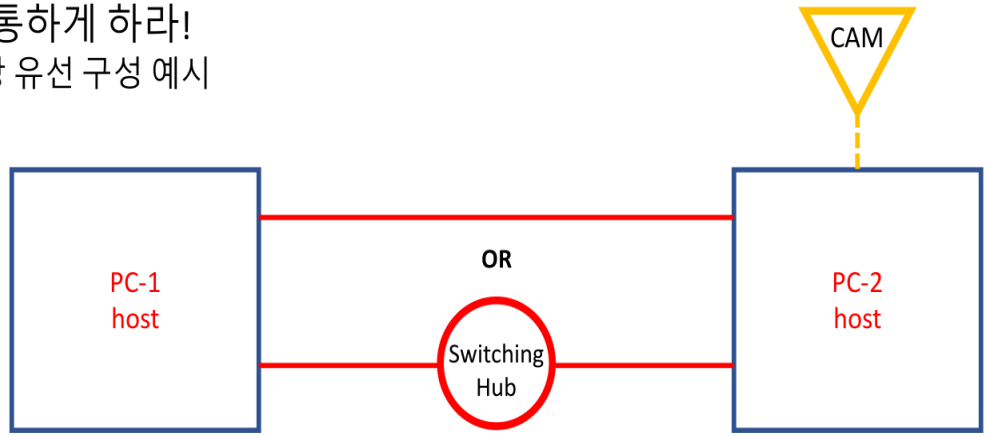
PC 1대(운영체제: Mint 20 or Ubuntu 20.04)

인텔 NUC 누크 11세대 타이거캐년 i7 프로세서 미니PC 베어본 NUC11TNHi7(운영체제: Ubuntu 20.04)

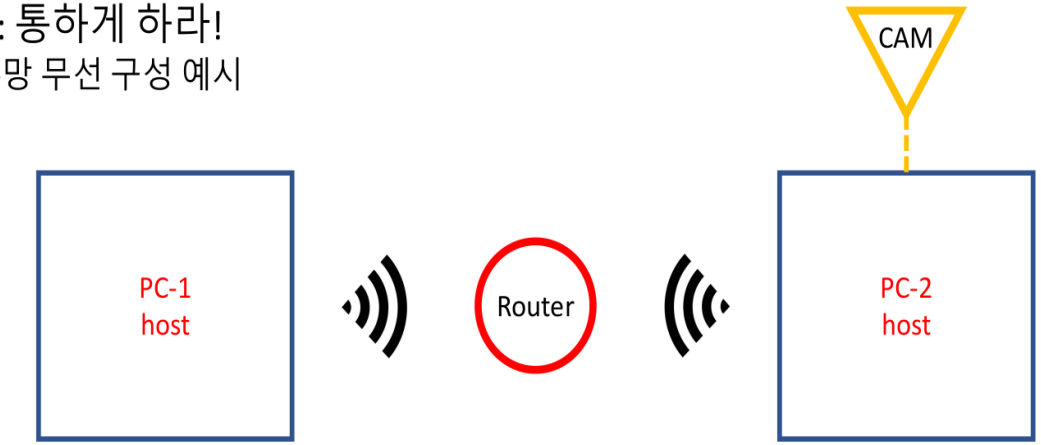
PengCa Web Camera PCWEB400

유선/무선 구성 예시

- 과제명: 통하게 하라!
  - 내부망 유선 구성 예시



- 과제명: 통하게 하라!
  - 외부망 무선 구성 예시



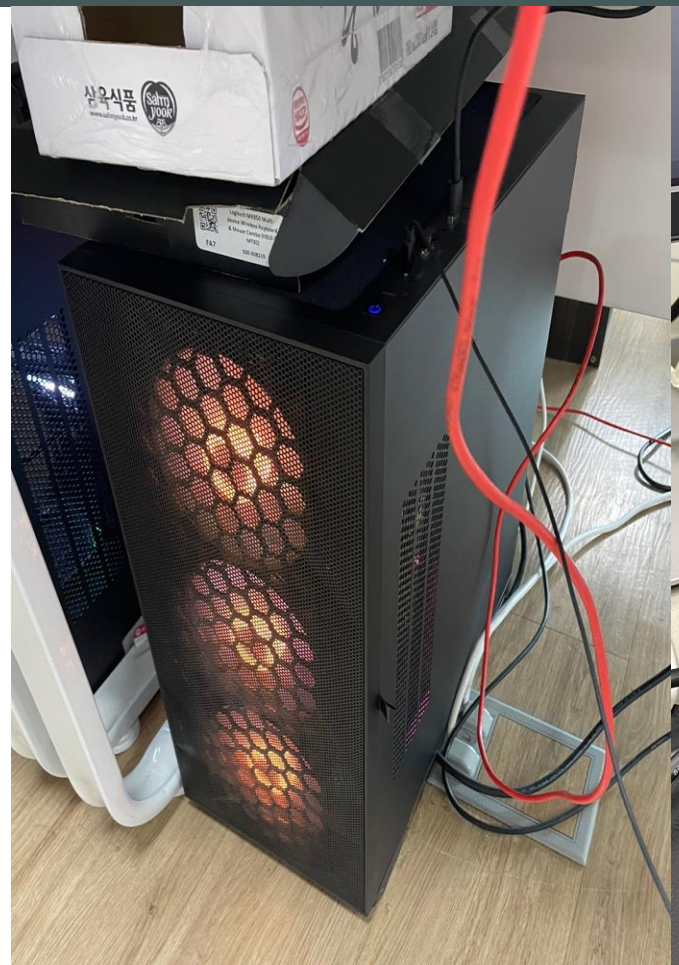
유선 통신

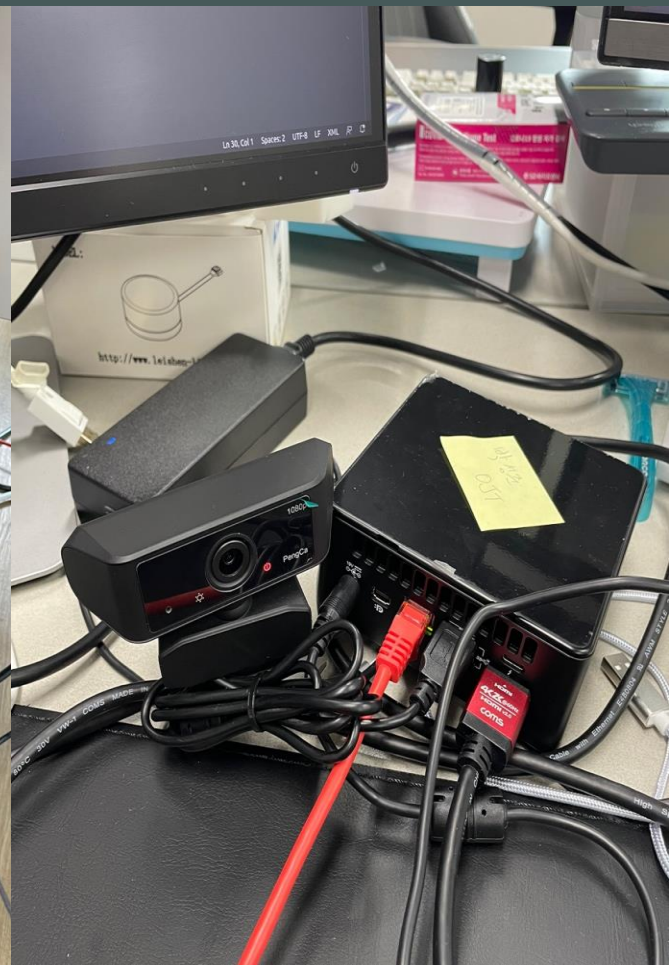
ROS MASTER :

데스크톱(Ubuntu 20.04 or Mint 20)

REMOTE PC :

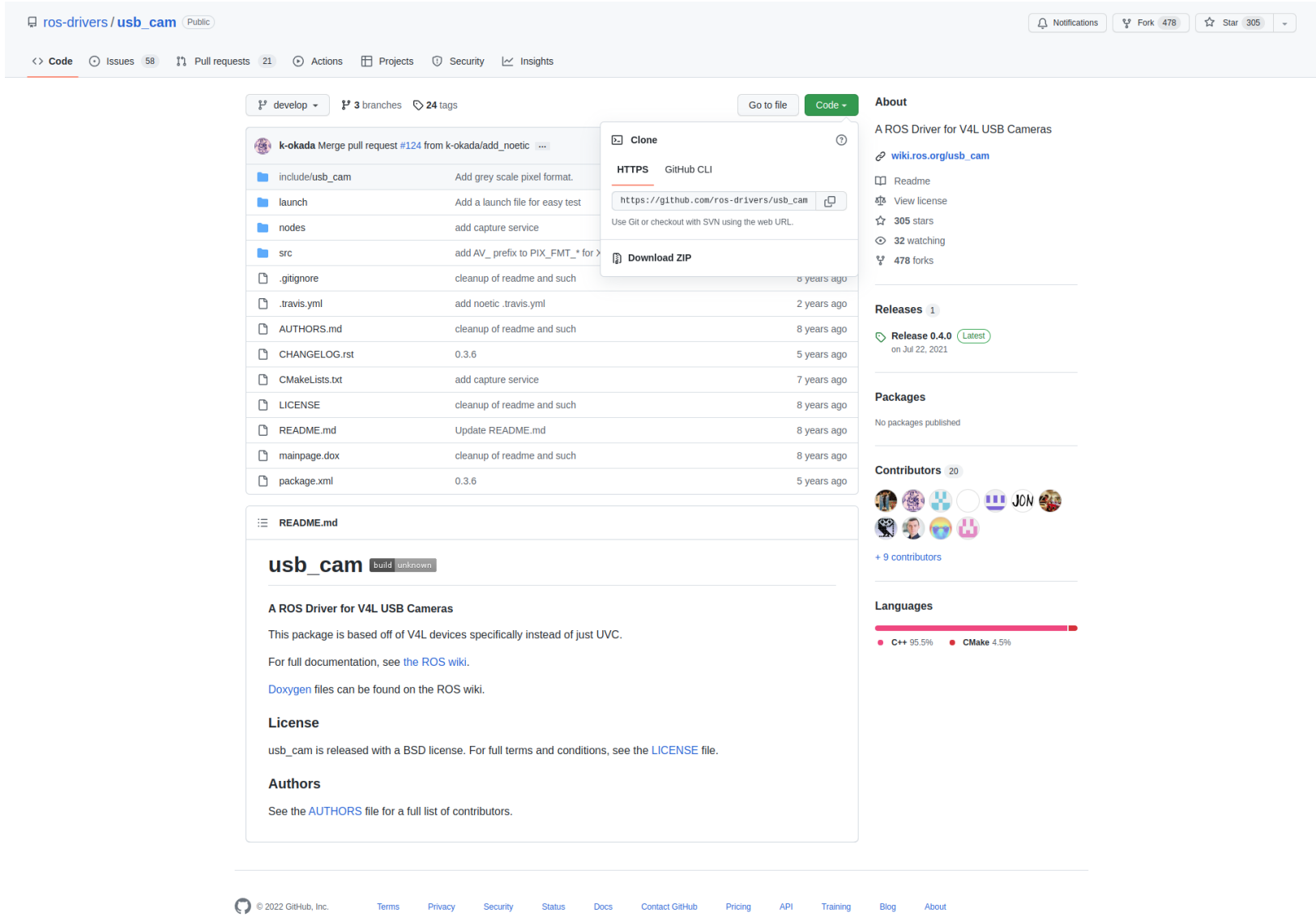
NUC11TNHi7(Ubuntu 20.04)





Github Repository

`usb_cam`



ifconfig
아이피 확인
(Master PC)

```
seongjun@seongjun-desktop:~/pc-side$ ifconfig
enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.42.0.88  netmask 255.255.255.0  broadcast 10.42.0.255
        inet6 fe80::a1df:746:d9c8:a249  prefixlen 64  scopeid 0x20<link>
        ether a8:a1:59:93:96:cb  txqueuelen 1000  (Ethernet)
        RX packets 5692403  bytes 8498991929 (8.4 GB)
        RX errors 0  dropped 34  overruns 0  frame 0
        TX packets 715779  bytes 100653604 (100.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xa3300000-a3320000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 169170  bytes 14180955 (14.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 169170  bytes 14180955 (14.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlx705dccffb1c5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.161  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::918c:917e:e6e8:913f  prefixlen 64  scopeid 0x20<link>
        ether 70:5d:cc:ff:b1:c5  txqueuelen 1000  (Ethernet)
        RX packets 173798  bytes 185005108 (185.0 MB)
        RX errors 0  dropped 5979  overruns 0  frame 0
        TX packets 44156  bytes 5964519 (5.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Bashrc 파일 (Master PC)

```
source /opt/ros/noetic/setup.bash
source ~/pc-side/devel/setup.bash

# Set ROS Network
export ROS_HOSTNAME=10.42.0.88
export ROS_MASTER_URI=http://10.42.0.88:11311

export ROSLAUNCH_SSH_UNKNOWN=1
```

Launch 파일 (Master PC)

src > usb_cam > launch > usb_cam-test.launch

```
<launch>
  <!-- <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
  </node>
  -->
  <!-- [TEST] -->
  <!-- machine을 통해 보드의 카메라를 켜는 코드가 여기에 들어가야 함 -->
  <group>
    <machine name="board" address="10.42.0.1" env-loader="/opt/ros/noetic/env.sh" user="cona0901" password="conacona
  </group>

  <node machine="board" name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen">
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
  </node>
  <!-- [TEST] -->

  <node name="image_view" pkg="image_view" type="image_view" respawn="false" output="screen">
    <remap from="image" to="/usb_cam/image_raw"/>
    <param name="autosize" value="true" />
  </node>
</launch>
[EOF]
```

원격 PC에서 카메라 구동

PC에서 카메라 데이터 읽어옴

Others (Master PC)

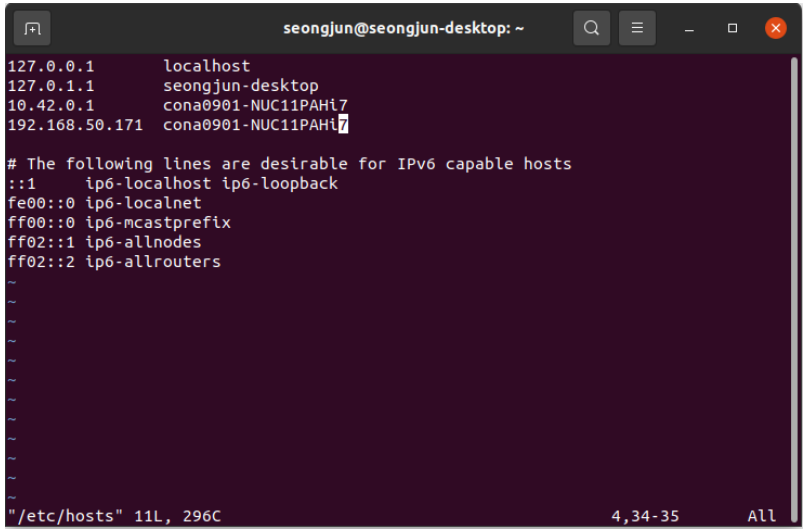
sudo ufw allow from 10.0.1.0/24

sudo ufw allow to 10.0.1.0/24

ssh cona0901@10.42.0.1

sudo vim /etc/hosts

```
10.42.0.1     cona0901-NUC11PAHi7
```



ifconfig
아이피 확인
(Client Board)

```
cona0901@cona0901-NUC11PAHi7:~$ ifconfig
enp89s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.42.0.1  netmask 255.255.255.0  broadcast 10.42.0.255
        inet6 fe80::1af4:9aeb:4080:c01a  prefixlen 64  scopeid 0x20<link>
        ether 1c:69:7a:ae:d7:ad  txqueuelen 1000  (Ethernet)
        RX packets 724173  bytes 102435692 (102.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5734631  bytes 8526500288 (8.5 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0x6a200000-6a2fffff

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 5288  bytes 561867 (561.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5288  bytes 561867 (561.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.171  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::7cd6:6d7e:ff92:6fec  prefixlen 64  scopeid 0x20<link>
        ether 80:38:fb:5d:37:e8  txqueuelen 1000  (Ethernet)
        RX packets 374952  bytes 401167422 (401.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 113333  bytes 64776773 (64.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Bashrc 파일 (Client Board)

```
source /opt/ros/noetic/setup.bash
source ~/board-side/devel/setup.bash

export ROS_MASTER_URI=http://10.42.0.88:11311
export ROS_HOSTNAME=10.42.0.1
```

Others (Client Board)

sudo ufw allow from 10.0.1.0/24

sudo ufw allow to 10.0.1.0/24

sudo vim /etc/hosts

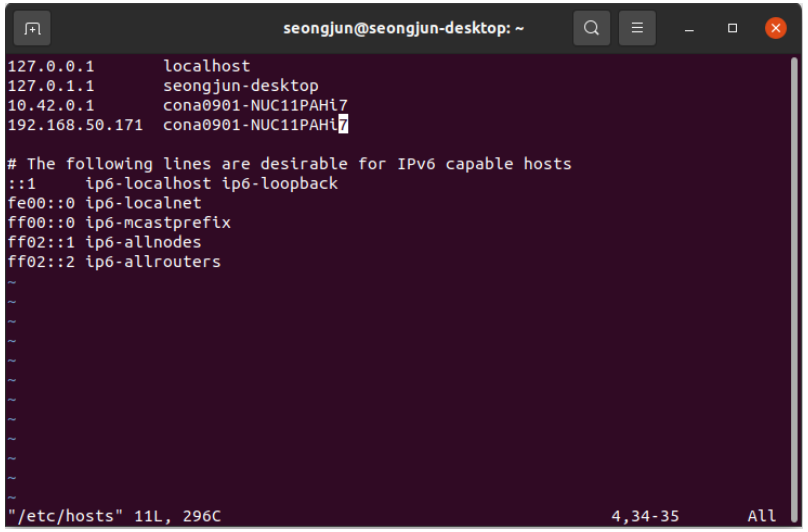
```
10.42.0.88 cona0901-NUC11PAHi7
```

ls -ltr /dev/video*

sudo apt install v4l-utils

v4l2-ctl -list-formats-ext

sudo apt install ros-noetic-usb-cam



유선
Roslaunch
실행 녹화

무선통신

ROS MASTER :

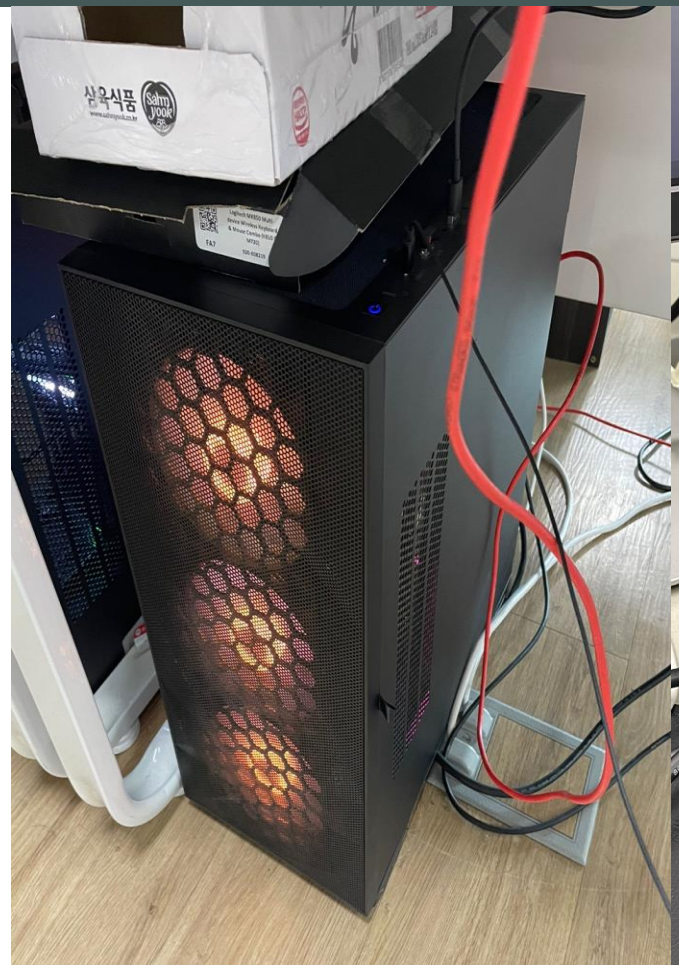
데스크톱(Ubuntu 20.04 or Mint 20)

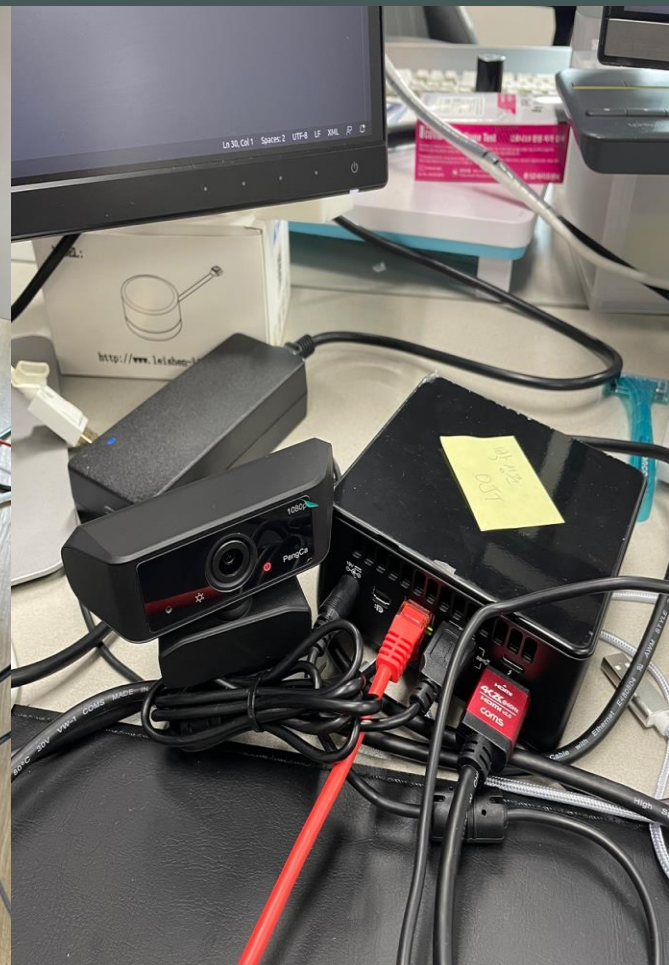
REMOTE PC :

NUC11TNHi7(Ubuntu 20.04)

Wifi :

SK-COGA-ROBOTICS-5GHz-2





ifconfig
아이피 확인
(Master PC)

```
seongjun@seongjun-desktop:~/pc-side$ ifconfig
enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.42.0.88  netmask 255.255.255.0  broadcast 10.42.0.255
        inet6 fe80::a1df:746:d9c8:a249  prefixlen 64  scopeid 0x20<link>
        ether a8:a1:59:93:96:cb  txqueuelen 1000  (Ethernet)
        RX packets 5692403  bytes 8498991929 (8.4 GB)
        RX errors 0  dropped 34  overruns 0  frame 0
        TX packets 715779  bytes 100653604 (100.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xa3300000-a3320000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 169170  bytes 14180955 (14.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 169170  bytes 14180955 (14.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlx705dccffb1c5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.161  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::918c:917e:e6e8:913f  prefixlen 64  scopeid 0x20<link>
        ether 70:5d:cc:ff:b1:c5  txqueuelen 1000  (Ethernet)
        RX packets 173798  bytes 185005108 (185.0 MB)
        RX errors 0  dropped 5979  overruns 0  frame 0
        TX packets 44156  bytes 5964519 (5.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Bashrc 파일 (Master PC)

```
119 source /opt/ros/noetic/setup.bash
120 source ~/pc-side/devel/setup.bash
121
122 # Set ROS Network
123 export ROS_HOSTNAME=192.168.50.161
124 export ROS_MASTER_URI=http://192.168.50.161:11311
125
126 export ROSLAUNCH_SSH_UNKNOWN=1
127
```

Others (Master PC)

sudo ufw allow from 192.168.0.0/24

sudo ufw allow to 192.168.0.0/24

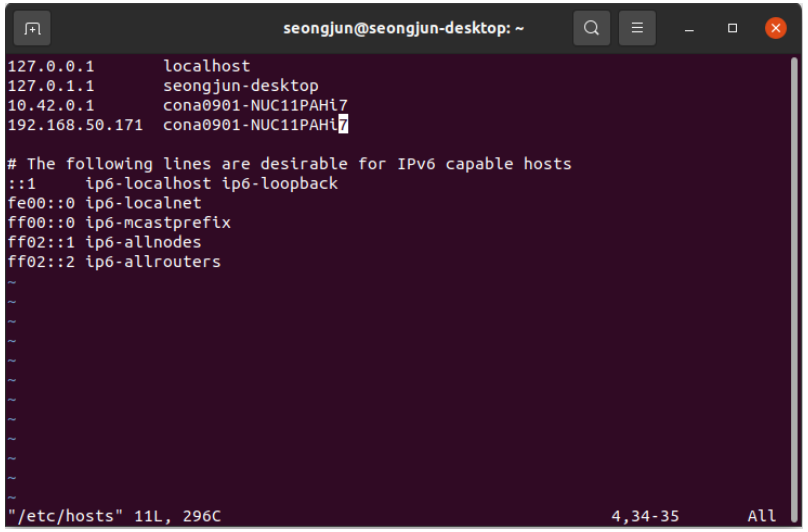
ssh cona0901@192.168.50.171

sudo vim /etc/hosts

```

192.168.50.171  cona0901-NUC11PAHi7

```



ifconfig
아이피 확인
(Client Board)

```
cona0901@cona0901-NUC11PAHi7:~$ ifconfig
enp89s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.42.0.1  netmask 255.255.255.0  broadcast 10.42.0.255
        inet6 fe80::1af4:9aeb:4080:c01a  prefixlen 64  scopeid 0x20<link>
        ether 1c:69:7a:ae:d7:ad  txqueuelen 1000  (Ethernet)
        RX packets 724173  bytes 102435692 (102.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5734631  bytes 8526500288 (8.5 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0x6a200000-6a2fffff

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 5288  bytes 561867 (561.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5288  bytes 561867 (561.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.171  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::7cd6:6d7e:ff92:6fec  prefixlen 64  scopeid 0x20<link>
        ether 80:38:fb:5d:37:e8  txqueuelen 1000  (Ethernet)
        RX packets 374952  bytes 401167422 (401.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 113333  bytes 64776773 (64.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Bashrc 파일 (Client Board)

```
source /opt/ros/noetic/setup.bash
source ~/board-side/devel/setup.bash

export ROS_MASTER_URI=http://192.168.50.161:11311
export ROS_HOSTNAME=192.168.50.171
```

Others (Client Board)

sudo ufw allow from 192.168.0.0/24

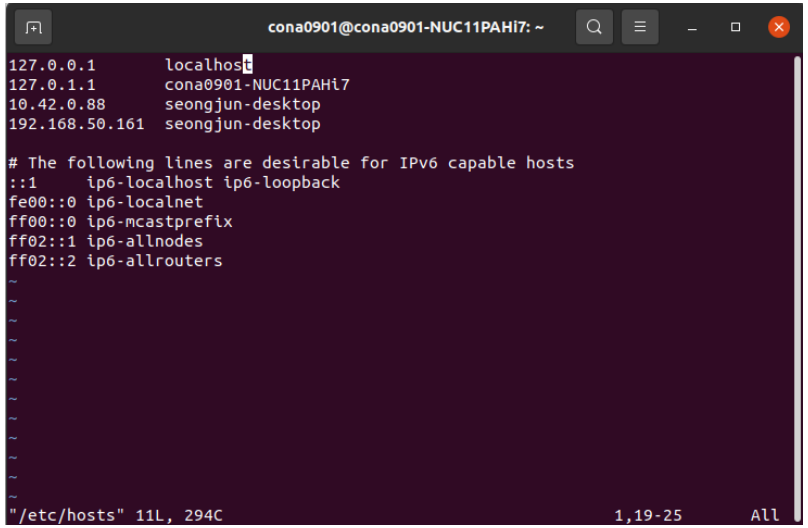
sudo ufw allow to 192.168.0.0/24

sudo vim /etc/hosts

```

192.168.50.161  seongjun-desktop

```



무선 Roslaunch 실행 녹화

이미지 뷰어

by `cv_bridge`

src > cv_image_viewer > src > image_viewing.cpp > ...

```cpp
#include "ros/ros.h"
#include "opencv2/opencv.hpp"
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "cv_bridge/cv_bridge.h"
#include "sensor_msgs/image_encodings.h"
#include "image_transport/image_transport.h"
#include "iostream"

static const std::string OPENCV_WINDOW = "Image window";

class ImageConverter
{
  ros::NodeHandle nh;
  image_transport::ImageTransport imgTransport;
  image_transport::Subscriber imgSub;
  image_transport::Publisher imgPub;

  public:
  // Constructor & Destructor
  ImageConverter(): imgTransport(nh)
  {
    imgSub = imgTransport.subscribe("/usb_cam/image_raw", 1, &ImageConverter::imageCb, this);
    imgPub = imgTransport.advertise("/image_viewing/output_video", 1);
    cv::namedWindow(OPENCV_WINDOW);
  }
  ~ImageConverter()
  {
    cv::destroyWindow(OPENCV_WINDOW);
  }

  // Methods
  void imageCb(const sensor_msgs::ImageConstPtr &msg)
  {
    cv_bridge::CvImagePtr cv_ptr;
    try
    {
      cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
    }
    catch (cv_bridge::Exception &e)
    {
      ROS_ERROR("cv_bridge execption occurred>>> %s", e.what());
      return;
    }

    // Update GUI Window
    cv::imshow(OPENCV_WINDOW, cv_ptr->image);
    cv::waitKey(1);

    // Output modified video stream
    imgPub.publish(cv_ptr->toImageMsg());
  }
};

int main(int argc, char** argv)
{
  ros::init(argc, argv, "image_viewing");
  ImageConverter imageConverter;
  ros::spin();

  return 0;
}
[EOF]
```

Constructor & Destructor

sensor_msgs::Image의 data 속성을 해석해 CvImage로 변환

CvImage를 해석해 sensor::Image로 변환해 publish 함

image_viewing 노드가 추가된 Launch 파일

src > usb_cam > launch > usb_cam-test.launch

```
<launch>
  <!-- <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
  </node>
  -->

  <!-- [TEST] -->
  <!-- machine을 통해 보드의 카메라를 켜는 코드가 여기에 들어가야 함 -->
  <group>
    <!-- 유선 통신 -->
    <!-- <machine name="board" address="10.42.0.1" env-loader="/opt/ros/noetic/env.sh" user="cona0901" password="con
    <!-- 무선 통신 -->
    <machine name="board" address="192.168.50.171" env-loader="/opt/ros/noetic/env.sh" user="cona0901" password="con
  </group>

  <node machine="board" name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen">
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
  </node>
  <!-- [TEST] -->

  <node name="image_view" pkg="image_view" type="image_view" respawn="false" output="screen">
    <remap from="image" to="/usb_cam/image_raw"/>
    <param name="autosize" value="true" />
  </node>

  <node name="cv_image_view" pkg="cv_image_viewer" type="image_viewing" />
</launch>
[EOF]
```

Roslaunch
실행 녹화

rostopic hz
/usb_cam/image_raw

유선

```
seongjun@seongjun-desktop:~/pc-side$ rostopic hz /usb_cam/image_raw
subscribed to [/usb_cam/image_raw]
no new messages
no new messages
no new messages
average rate: 1.571
        min: 0.636s max: 0.636s std dev: 0.00000s window: 2
average rate: 1.581
        min: 0.559s max: 0.702s std dev: 0.05822s window: 4
average rate: 1.744
        min: 0.424s max: 0.702s std dev: 0.09347s window: 6
average rate: 1.952
        min: 0.203s max: 0.702s std dev: 0.14194s window: 9
average rate: 2.152
        min: 0.203s max: 0.702s std dev: 0.14436s window: 12
average rate: 2.309
        min: 0.203s max: 0.702s std dev: 0.14168s window: 15
average rate: 2.419
        min: 0.203s max: 0.702s std dev: 0.13611s window: 18
average rate: 2.528
        min: 0.203s max: 0.702s std dev: 0.12831s window: 22
average rate: 2.592
        min: 0.203s max: 0.702s std dev: 0.12309s window: 25
average rate: 2.620
        min: 0.203s max: 0.702s std dev: 0.11731s window: 28
average rate: 2.667
        min: 0.203s max: 0.702s std dev: 0.11435s window: 31
average rate: 2.693
        min: 0.203s max: 0.702s std dev: 0.11130s window: 34
average rate: 2.726
        min: 0.203s max: 0.702s std dev: 0.10774s window: 37
average rate: 2.754
        min: 0.203s max: 0.702s std dev: 0.10444s window: 40
average rate: 2.817
        min: 0.203s max: 0.702s std dev: 0.10304s window: 44
average rate: 2.830
        min: 0.203s max: 0.702s std dev: 0.10183s window: 47
average rate: 2.890
        min: 0.194s max: 0.702s std dev: 0.10153s window: 51
average rate: 2.957
        min: 0.194s max: 0.702s std dev: 0.10180s window: 55
average rate: 3.019
        min: 0.194s max: 0.702s std dev: 0.10149s window: 59
average rate: 3.069
        min: 0.194s max: 0.702s std dev: 0.10068s window: 63
average rate: 3.127
        min: 0.194s max: 0.702s std dev: 0.10055s window: 67
average rate: 3.182
        min: 0.185s max: 0.702s std dev: 0.09954s window: 72
average rate: 3.234
        min: 0.185s max: 0.702s std dev: 0.09918s window: 76
average rate: 3.286
        min: 0.185s max: 0.702s std dev: 0.09822s window: 81
average rate: 3.338
        min: 0.185s max: 0.702s std dev: 0.09829s window: 85
average rate: 3.392
        min: 0.163s max: 0.702s std dev: 0.09794s window: 90
average rate: 3.454
        min: 0.163s max: 0.702s std dev: 0.09797s window: 95
average rate: 3.503
        min: 0.163s max: 0.702s std dev: 0.09724s window: 100
average rate: 3.550
        min: 0.163s max: 0.702s std dev: 0.09637s window: 105
average rate: 3.587
        min: 0.151s max: 0.702s std dev: 0.09593s window: 109
average rate: 3.621
        min: 0.151s max: 0.702s std dev: 0.09502s window: 114
average rate: 3.657
        min: 0.151s max: 0.702s std dev: 0.09398s window: 119
average rate: 3.697
        min: 0.151s max: 0.702s std dev: 0.09340s window: 124
average rate: 3.737
        min: 0.151s max: 0.702s std dev: 0.09273s window: 129
^C^Caverage rate: 3.742
        min: 0.151s max: 0.702s std dev: 0.09247s window: 130
seongjun@seongjun-desktop:~/pc-side$
```

무선

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
seongjun@seongjun-desktop:~/pc-side$ rostopic hz /usb_cam/image_raw
subscribed to [/usb_cam/image_raw]
no new messages
no new messages
no new messages
average rate: 0.977
        min: 1.024s max: 1.024s std dev: 0.00000s window: 2
average rate: 1.175
        min: 0.629s max: 1.024s std dev: 0.16499s window: 4
average rate: 1.387
        min: 0.412s max: 1.024s std dev: 0.21632s window: 6
average rate: 1.709
        min: 0.298s max: 1.024s std dev: 0.24666s window: 9
average rate: 1.964
        min: 0.234s max: 1.024s std dev: 0.24567s window: 12
average rate: 2.421
        min: 0.183s max: 1.024s std dev: 0.24885s window: 17
average rate: 2.978
        min: 0.130s max: 1.024s std dev: 0.23846s window: 24
average rate: 3.452
        min: 0.000s max: 1.024s std dev: 0.22815s window: 31
average rate: 3.829
        min: 0.000s max: 1.024s std dev: 0.21396s window: 38
average rate: 4.134
        min: 0.000s max: 1.024s std dev: 0.20138s window: 45
average rate: 4.445
        min: 0.000s max: 1.024s std dev: 0.18997s window: 53
average rate: 4.711
        min: 0.000s max: 1.024s std dev: 0.18002s window: 61
average rate: 4.918
        min: 0.000s max: 1.024s std dev: 0.17251s window: 68
average rate: 5.098
        min: 0.000s max: 1.024s std dev: 0.16472s window: 76
average rate: 5.282
        min: 0.000s max: 1.024s std dev: 0.15823s window: 84
average rate: 5.456
        min: 0.000s max: 1.024s std dev: 0.15245s window: 92
average rate: 5.584
        min: 0.000s max: 1.024s std dev: 0.14788s window: 100
average rate: 5.583
        min: 0.000s max: 1.024s std dev: 0.14471s window: 105
average rate: 5.622
        min: 0.000s max: 1.024s std dev: 0.14027s window: 112
average rate: 5.679
        min: 0.000s max: 1.024s std dev: 0.13714s window: 118
average rate: 5.755
        min: 0.000s max: 1.024s std dev: 0.13336s window: 126
average rate: 5.813
        min: 0.000s max: 1.024s std dev: 0.13009s window: 133
average rate: 5.841
        min: 0.000s max: 1.024s std dev: 0.12733s window: 139
average rate: 5.885
        min: 0.000s max: 1.024s std dev: 0.12442s window: 146
average rate: 5.897
        min: 0.000s max: 1.024s std dev: 0.12211s window: 152
average rate: 5.912
        min: 0.000s max: 1.024s std dev: 0.11955s window: 159
average rate: 5.915
        min: 0.000s max: 1.024s std dev: 0.11737s window: 165
average rate: 5.931
        min: 0.000s max: 1.024s std dev: 0.11539s window: 171
average rate: 5.949
        min: 0.000s max: 1.024s std dev: 0.11320s window: 178
average rate: 5.963
        min: 0.000s max: 1.024s std dev: 0.11138s window: 184
average rate: 5.995
        min: 0.000s max: 1.024s std dev: 0.10949s window: 191
average rate: 6.017
        min: 0.000s max: 1.024s std dev: 0.10759s window: 198
average rate: 6.034
        min: 0.000s max: 1.024s std dev: 0.10615s window: 204
average rate: 6.081
        min: 0.000s max: 1.024s std dev: 0.10445s window: 212
^Cno new messages
seongjun@seongjun-desktop:~/pc-side$
```

* average rate : topic의 publishing 되는 hz
* min/max : 초 단위 hz 계산의 편차
* window : publishing 된 화면의 수

Decoding Compressed Image

/usb_cam/image_raw 의 문제점 :

"화면 부하"

/usb_cam/image_raw 의 문제점에 대한 해결 방안 :

"/usb_cam/image_raw/compressed"

- 적당한 크기로 이미지를 압축하여 크기를 줄임. 이에 따라 화면 부하를 감소를 기대할 수 있음.

* 주의할 점은, ROS가 sensor_msgs::CompressedImage를 지원하더라도,
이를 받는 OpenCV에 해당 메시지를 받을 적정한 파라미터 오버로딩이 없을 수 있다는 것임.

Compressed Image Viewer

src > cv_image_viewer > src > compressed_image_viewing.cpp > ...

```cpp
#include "ros/ros.h"
#include "opencv2/opencv.hpp"
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "cv_bridge/cv_bridge.h"
#include "sensor_msgs/image_encodings.h"
#include "sensor_msgs/CompressedImage.h"
#include "image_transport/image_transport.h"
#include "iostream"

static const std::string OPENCV_WINDOW = "Compressed Image window 1";

class ImageConverter
{
  ros::NodeHandle nh;
  image_transport::ImageTransport imgTransport;

  ros::Subscriber imgCompSub;
  ros::Publisher imgCompPub;

  public:
  ImageConverter(): imgTransport(nh)
  {
    imgCompSub = nh.subscribe("/usb_cam/image_raw/compressed", 1, &ImageConverter::imageCb, this);
    imgCompPub = nh.advertise<sensor_msgs::CompressedImage>("/image_viewing/output_video", 1);
    cv::namedWindow(OPENCV_WINDOW);
  }
  ~ImageConverter()
  {
    cv::destroyWindow(OPENCV_WINDOW);
  }

  void imageCb(const sensor_msgs::CompressedImageConstPtr &msg)
  {
    cv_bridge::CvImagePtr cv_ptr;
    try
    {
      cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
    }
    catch (cv_bridge::Exception &e)
    {
      ROS_ERROR("cv_bridge execption occurred>>> %s", e.what());
      return;
    }

    cv::imshow(OPENCV_WINDOW, cv_ptr->image);
    cv::waitKey(1);

    imgCompPub.publish(cv_ptr->toCompressedImageMsg());
  }
};

int main(int argc, char** argv)
{
  ros::init(argc, argv, "compressed_image_viewing");
  ImageConverter imageConverter;
  ros::spin();

  return 0;
}
[EOF]
```

Cv_bridge::toCvCopy의 첫번째 매개변수로 Sensor_msgs::CompressedImageContPtr 타입이 전달됨

Roslaunch
실행 녹화

Decoding Compressed Image

Assume that...

Cv_bridge::toCvCopy 메서드가

sensor_msgs::CompressedImage를 지원하지 않는다면,

Solution is...

toCvCopy에 sensor_msgs::Image를 사용해 compressed image를 표현하고,

해당 타입으로 파라미터를 넘겨줘야 함.

Compressed Image Viewer

```cpp
void imageCb(const sensor_msgs::CompressedImageConstPtr &msg)
{
  cv_bridge::CvImagePtr cv_ptr;
  try
  {
    /*
    CompressedImage has header, format, and data.
    ex)
    header:
      seq: 2632
      stamp:
        secs: 1664445983
        nsecs: 820730634
      frame_id: "usb_cam"
    format: "rgb8; jpeg compressed bgr8"
    data: [255, 216, 255, 224, 0, 16, 74, 70, 73, 70, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 255, ...]
    */

    // cv_bridge::CompressedImage가 지원되지 않는다는 전제
    // 아래의 코드를 직접 구현
    // cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);

    cv::Mat jpegData(1,msg->data.size(),CV_8UC1);
    jpegData.data = const_cast<uchar*>(&msg->data[0]);
    cv::InputArray data(jpegData);
    cv::Mat image = cv::imdecode(data,cv::IMREAD_COLOR);

    sensor_msgs::Image ros_image;
    ros_image.header = msg->header;
    ros_image.height = image.rows;
    ros_image.width = image.cols;
    ros_image.encoding = sensor_msgs::image_encodings::BGR8;
    ros_image.is_bigendian = false;
    ros_image.step = image.cols * image.elemSize();
    size_t size = ros_image.step * image.rows;
    ros_image.data.resize(size);
    if (image.isContinuous())
    {
      memcpy((char*)(&ros_image.data[0]), image.data, size);
    }
    else
    {
      uchar* ros_data_ptr = (uchar*)(&ros_image.data[0]);
      uchar* cv_data_ptr = image.data;
      for (int i = 0; i < image.rows; ++i)
      {
        memcpy(ros_data_ptr, cv_data_ptr, ros_image.step);
        ros_data_ptr += ros_image.step;
        cv_data_ptr += image.step;
      }
    }

    cv_ptr = cv_bridge::toCvCopy(ros_image, sensor_msgs::image_encodings::BGR8);
  }
  catch (cv_bridge::Exception &e)
  {
    ROS_ERROR("cv_bridge execption occurred>>> %s", e.what());
    return;
  }

  cv::imshow(OPENCV_WINDOW, cv_ptr->image);
  cv::waitKey(1);

  imgCompPub.publish(cv_ptr->toCompressedImageMsg());
}
};
```

CompressedImage 의 속성
Header, Format, Data