**설치환경**

virtualbox(ubuntu16)

기본 메모리, 프로세서, 비디오 메모리 가능한 많이, 프로세서 가능한 많이

--> 속도를 위해서

네트워크

turtlebot과 remotePC는 같은 네트워크에 있어야 한다.

가상환경에서는 무선랜카드를 잡을 수 없으니

virturlbox 설정에서 네트워크를 바꿔준다.

어댑터2 -> 어댑터에 브리지 -> 무선랜카드

**간편설치**

$ sudo apt-get install gcc g++ gedit python git # 필요라이브러리 다운

$ git clone https://github.com/ROBOTIS-GIT/ros\_tutorials.git;

$ sudo apt-get install -y chrony ntpdate;

$ sudo ntpdate -q ntp.ubuntu.com;

$ sudo apt-get update

$ sudo apt-get upgrade

$wgethttps://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros\_kinetic.sh && chmod 755 ./install\_ros\_kinetic.sh && bash ./install\_ros\_kinetic.sh

터틀봇의 경우 robotis에서 제공하는 라즈비안을 받으면 ros가 자동으로 설치된다

<https://www.robotis.com/service/download.php?no=1738>

**종속 ros 패키지설정**

#remotePC

$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers

$ cd ~/catkin\_ws/src/

$ git clone https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git

$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/turtlebot3.git

$ git clone https://github.com/ROBOTIS-GIT/hls\_lfcd\_lds\_driver.git

$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git

$ cd ~/catkin\_ws && catkin\_make

#turtlebot

$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers

$ cd ~/catkin\_ws/src/

$ git clone https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git

$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/turtlebot3.git

$ git clone https://github.com/ROBOTIS-GIT/hls\_lfcd\_lds\_driver.git

$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git

터틀봇에서 필요하지않은 패키지 삭제

$ cd ~/catkin\_ws/src/turtlebot3

$ rm -r turtlebot3\_description/ turtlebot3\_teleop/ turtlebot3\_navigation/ turtlebot3\_slam/ turtlebot3\_example/

**[TurtleBot]** 종속 패키지 설치

$ sudo apt-get install ros-kinetic-rosserial-python ros-kinetic-tf

**[TurtleBot]** 패키지 설치 후 Raspberry Pi 3를 재부팅

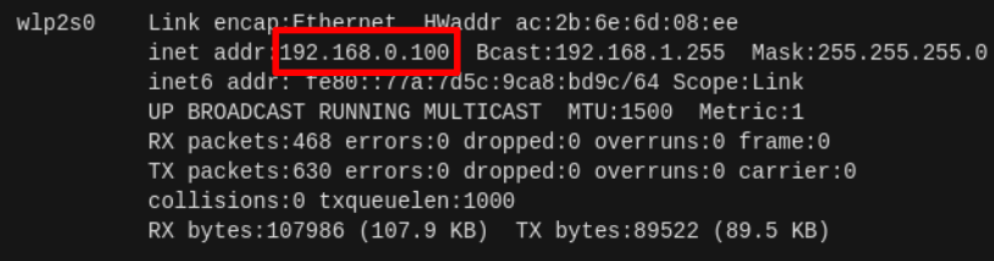
**[TurtleBot]** 패키지 빌드

$ source /opt/ros/kinetic/setup.bash

$ cd ~/catkin\_ws && catkin\_make -j1

**bashrc의 uri수정**

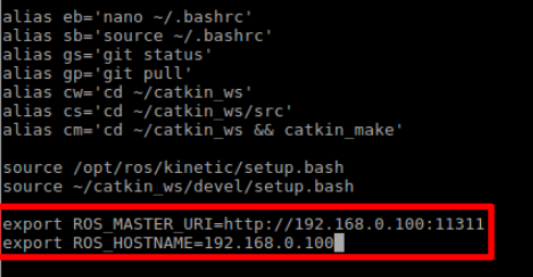
ifconfig로 ip주소를 확인하고



bachrc를 수정해준다

무선환경이라면 무선랜카드의 inet addr로 한다.

gedit ~/.bashrc로 bashrc들어갈수있다.





해당 그림처럼

remotepc에선 ROS\_MASTER\_URI, ROS\_HOSTNAME 둘 다 remotePC ip로 설정해 준다.

turtlebot에선 ROS\_MASTER\_URI는 remotePC ip로 HOSTNAME은 turtlebot ip로 설정한다.

(turtlebot은 nano를 편집기로 사용한다 → nano ~/.bashrc)

수정 후 remote, turtlebot둘다

$ source ~/.bash

를 해준다.

**opencr 설정**

<https://emanual.robotis.com/docs/en/platform/turtlebot3/opencr_setup/#opencr-setup>

메뉴얼대로 하면된다.

리모트 PC

$ sudo dpkg --add-architecture armhf

$ sudo apt-get update

$ sudo apt-get install libc6:armhf

터틀봇에서 설정

$ export OPENCR\_PORT=/dev/ttyACM0

$ export OPENCR\_MODEL=waffle

$ rm -rf ./opencr\_update.tar.bz2

$ wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencr\_update.tar.bz2 && tar -xvf opencr\_update.tar.bz2 && cd ./opencr\_update && ./update.sh $OPENCR\_PORT $OPENCR\_MODEL.opencr && cd ..

**remotePC에서 turtlebot 접속**

remotepc의 터미널에서

$ ssh [turtlebotID]@[turtlebotIP]

로turtlebot에 접속가능

**bringUP**

로봇을 활성화

#remotePC

$ roscore

#turtlebot

$ roslaunch turtlebot3\_bringup turtlebot3\_robot.launch

#remotePC

$ export TURTLEBOT3\_MODEL=waffle\_pi

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

로 키보드로 turtlebot 조종가능

**터틀봇 카메라**

#turtlebot

$ git clone https://github.com/UbiquityRobotics/raspicam\_node.git

$ sudo apt-get install ros-kinetic-compressed-image-transport ros-kinetic-camera-info-manager

$ cd ~/catkin\_ws && catkin\_make

#remotePC

$ roscore

#turtlebot

$ roslaunch turtlebot3\_bringup turtlebot3\_rpicamera.launch

#remote

$ rqt\_image\_view

**캐브레이션 : 카메라 보정**

#remotePC

$ roscore

#turtlebot

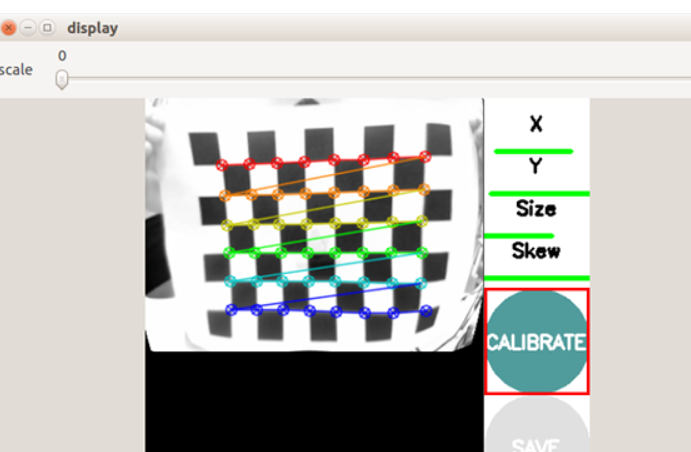
$ **roslaunch turtlebot3\_autorace\_traffic\_light\_camera turtlebot3\_autorace\_camera\_pi.launch**

$ export AUTO\_IN\_CALIB=calibration

$ export GAZEBO\_MODE=false

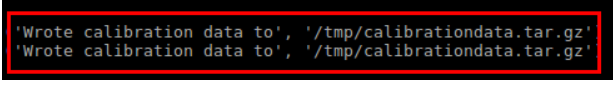
$ roslaunch turtlebot3\_autorace\_traffic\_light\_camera turtlebot3\_autorace\_intrinsic\_camera\_calibration.launch

을 하면 카메라를 환경에맞게 설정할 수 있다.



조정중

save버튼으로 저장하면



해당 위치에 파일이 생긴다

#remotePC

$ cd/tmp

$ tar -xvzf calibrationdata.tar.gz

로 압축을 풀고

$ mv ost.txt ost.ini

$ rosrun camera\_calibration\_parsers convert ost.ini camera.yaml

$ mkdir ~/.ros/camera\_info

$ mv camera.yaml ~/.ros/camera\_info/

ost.txt파일을 ost.ini 파일로 이름을 바꿈

그 후 yaml을 생성,

camera\_info에 저장한다.

그러면 파일 보정에 대한 경고가 나오지 않는다.

**slam**

#remotePC

$ rescore

# turtlebot

$ roslaunch turtlebot3\_bringup turtlebot3\_robot.launch

#remotePC

$ export TURTLEBOT3\_MODEL=waffle\_pi

$ roslaunch turtlebot3\_slam turtlebot3\_slam.launch slam\_methods:=gmapping

#remotePC

$ export TURTLEBOT3\_MODEL=waffle\_pi

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

로 이동하면서 맵핑을 한다.

# 맵저장

rosrun map\_server map\_saver -f ~/map

**네비게이션**

#remotePC

$ rescore

#turtlebot

$ roslaunch turtlebot3\_bringup turtlebot3\_robot.launch

#remotePC

$ export TURTLEBOT3\_MODEL=waffle\_pi

$ roslaunch turtlebot3\_navigation turtlebot3\_navigation.launch map\_file:=$HOME/map.yaml