Spring, 2021

지능로봇

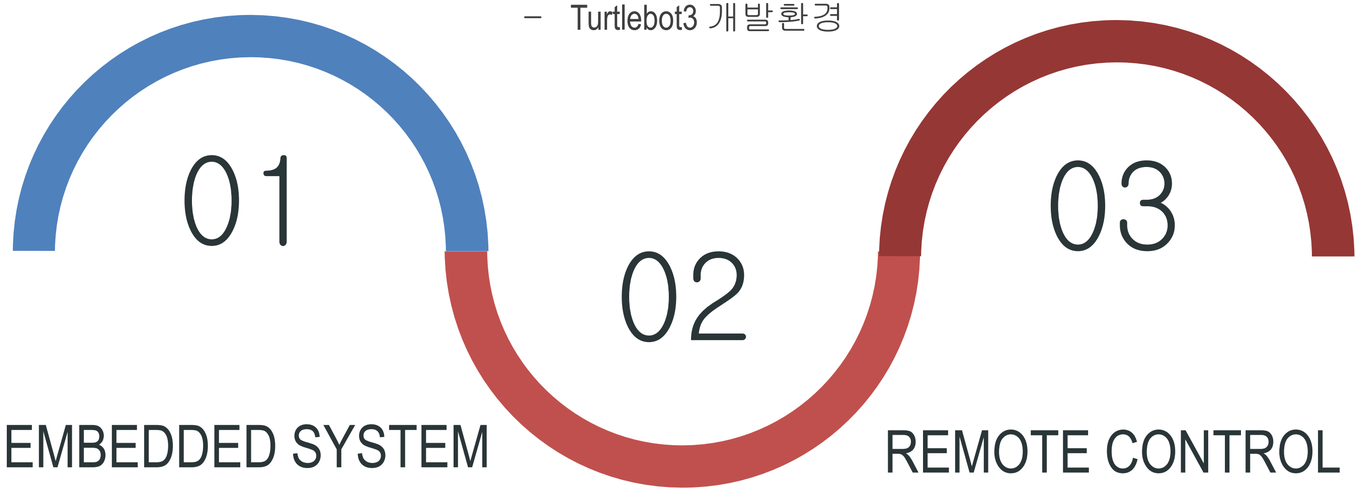
EMBEDDED SYSTEM
TURTLEBOT3
REMOTE CONTROL

메카트로닉스공학과

한국산업기술대학교

TURTLEBOT3

- Turtlebot3
- Turtlebot3 개발환경



EMBEDDED SYSTEM

- Embedded System in ROS
  - rosserial
- rosserial restrictions
  - OpenCR

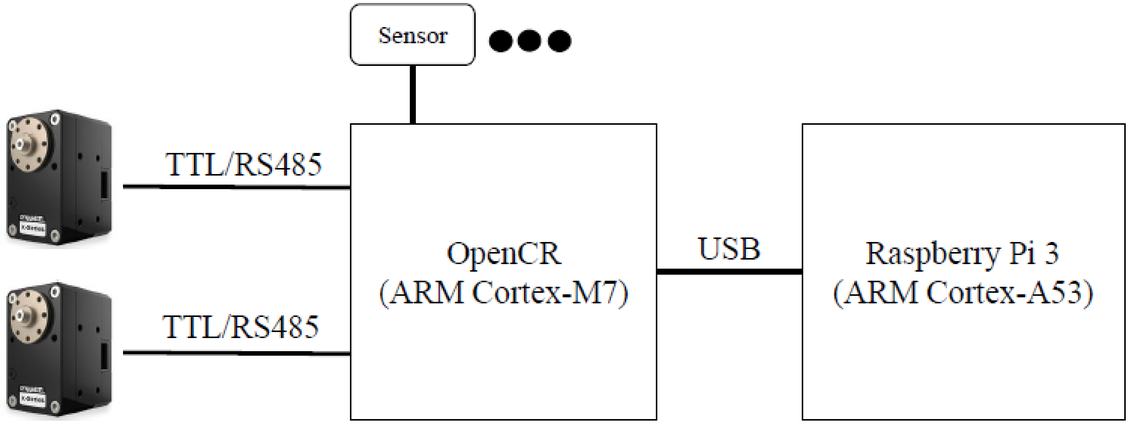
REMOTE CONTROL

- Turtlebot3 원격제어
- Turtlebot3 시각화
- Turtlebot3 Toic / TF

01. EMBEDDED SYSTEM

– Embedded System in ROS

- PC와는 달리 임베디드 시스템에서는 **ROS 설치가 불가능**
- **실시간성 확보** 및 **하드웨어 제어**를 위해서는 ROS가 설치된 PC와 임베디드 시스템 간의 연결이 요구됨
- ROS에서는 이를 위해 **rosserial**이라는 패키지를 제공
- 다이나믹셀 모터에는 모터드라이버가 내장되어 있음

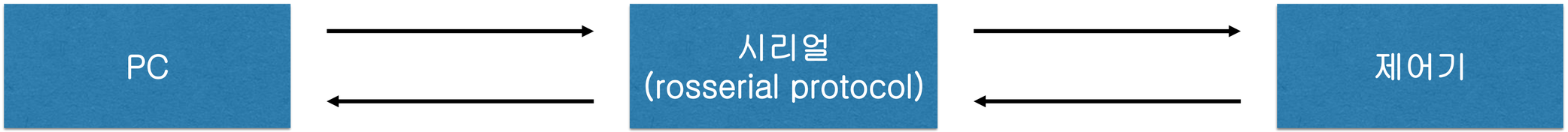


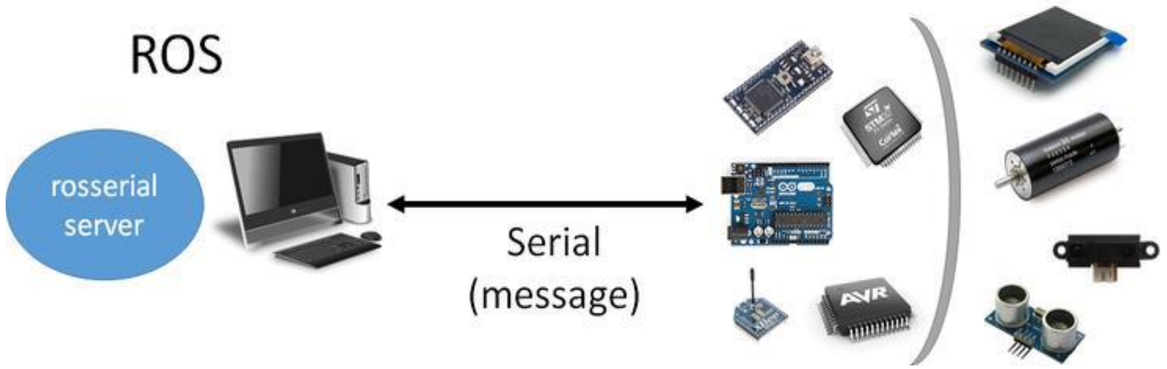
[터틀봇3에서의 PC와 임베디드 시스템의 구성]

01. EMBEDDED SYSTEM

– rosserial

- PC와 제어기 간의 메시지 통신을 위해 **중계자 역할**을 수행하는 ROS 패키지





01. EMBEDDED SYSTEM

– rosserial restrictions

- 메모리

—퍼블리셔, 서브스크라이버 개수 및 송신, 수신 버퍼의 크기를 미리 정의해야 함

- Float64

—마이크로컨트롤러는 64비트 실수연산을 지원하지 않아 32비트형으로 변경함

- Strings

—문자열 데이터를 String 메시지 안에 저장하지 않고 외부에서 정의한 문자열 데이터의 포인터 값만 메시지에 저장함

- Array

—메모리 제약사항으로 배열의 크기를 지정해서 사용

- 통신 속도

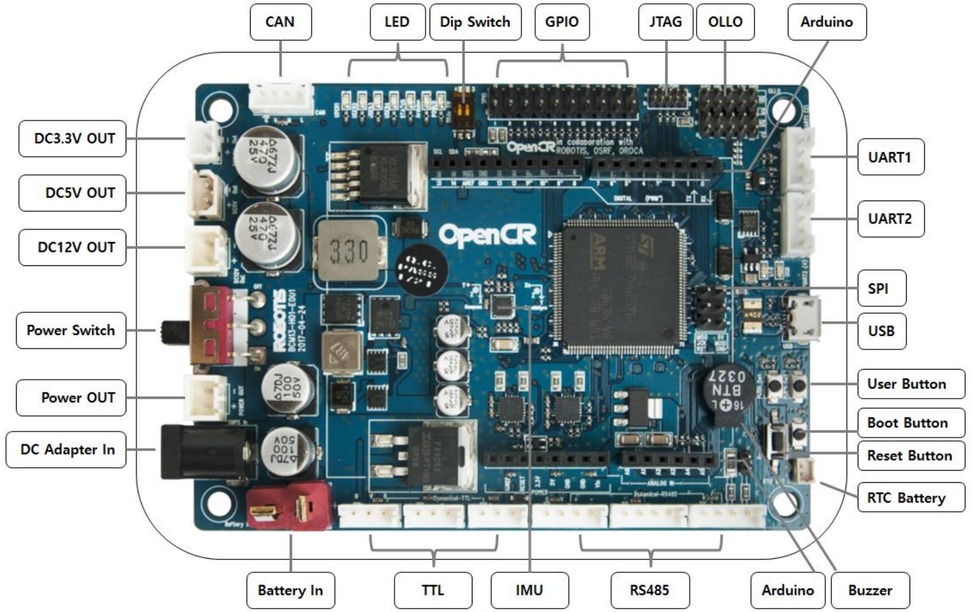
—UART 같은 경우 115200bps와 같은 속도로는 메시지의 개수가 많아지면 응답 및 처리속도가 느려짐

01. EMBEDDED SYSTEM

- OpenCR (Open-source Control Module for ROS)

- ROS를 지원하는 임베디드 보드이며 터틀봇3의 메인 제어기
- 오픈소스 H/W, S/W
- rosserial의 제약사항을 극복하기 위한 구성

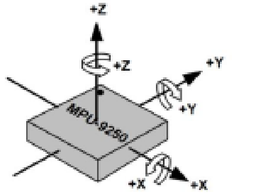
—32-bit ARM Cortex-M7 with FPU (216MHz, 462D MIPS)
—1MB 플래시 메모리
—320KB SRAM
—Float64 지원
—UART가 아닌 USB 패킷 전송 사용



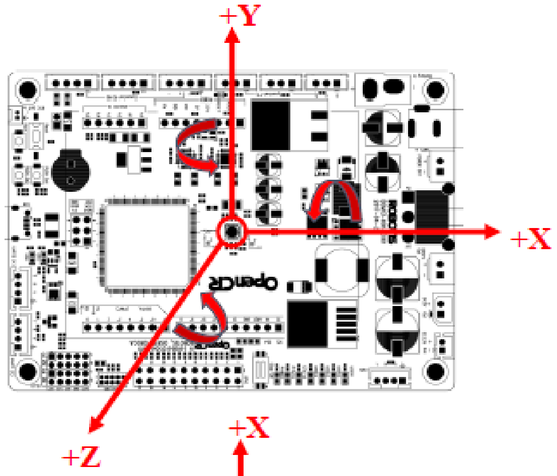
01. EMBEDDED SYSTEM

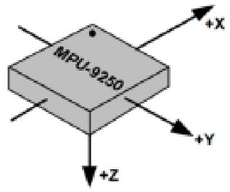
- OpenCR (Open-source Control Module for ROS)

- 기본 장착 센서
  - Gyroscope 3Axis
  - Accelerometer 3Axis
  - Magnetometer 3Axis
- 통신지원
  - USB, SPI, I2C
  - TTL, RS485, CAN

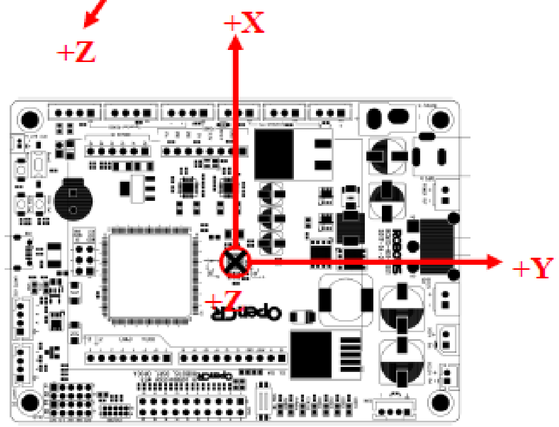


Gyroscope, Accelerometer





Magnetometer

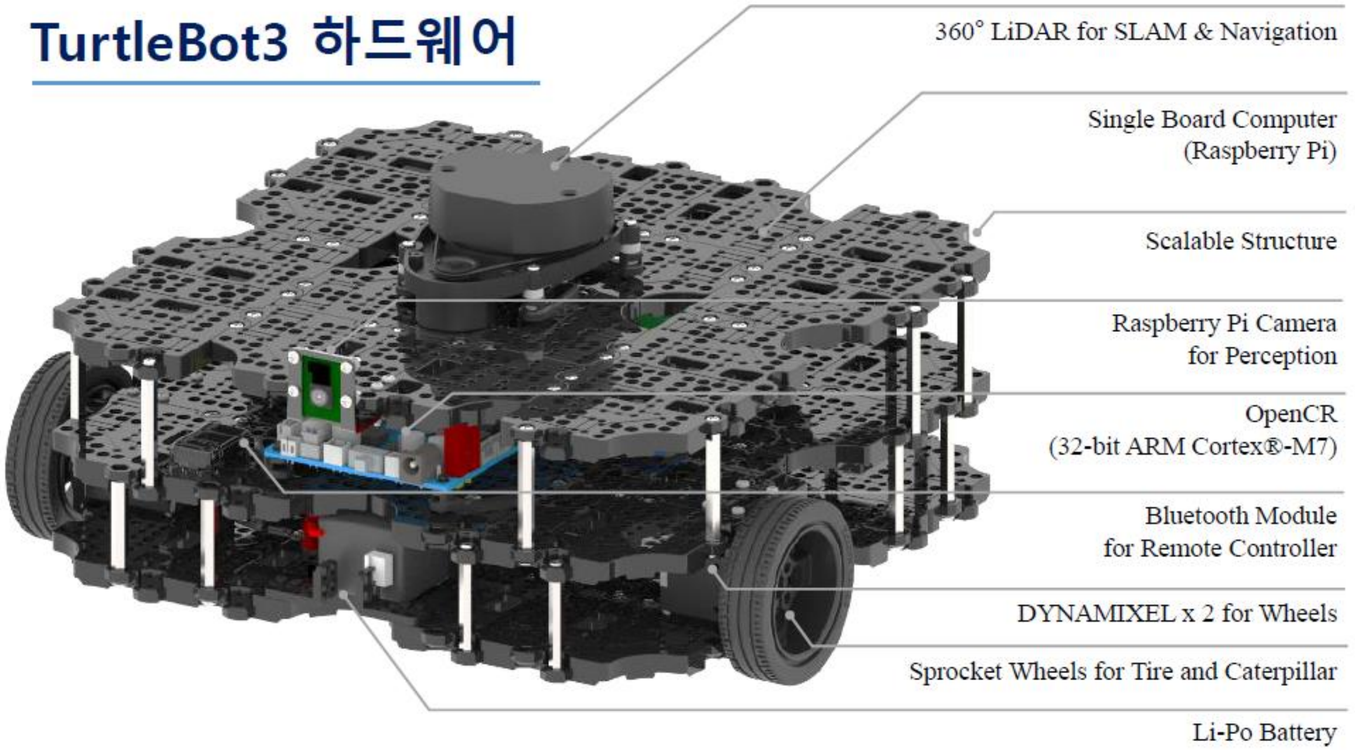


02. TURTLEBOT3

– Turtulebot3

- ROS 공식 로봇 플랫폼
- 전 세계 수 많은 연구소, 학교, DIY 에서 사용 중
- SLAM, Navigation, Gazebo, Rviz 서포트
- 참고사이트

—http://wiki.ros.org/Robots/TurtleBot
—http://turtlebot3.robotis.com



02. TURTLEBOT3

- Turtulebot3

https://youtu.be/9OC3J53RUsk

02. TURTLEBOT3

Turtlebot3 공식 사이트 참조
http://turtlebot3.robotis.com

- Turtulebot3 개발환경 (소프트웨어)

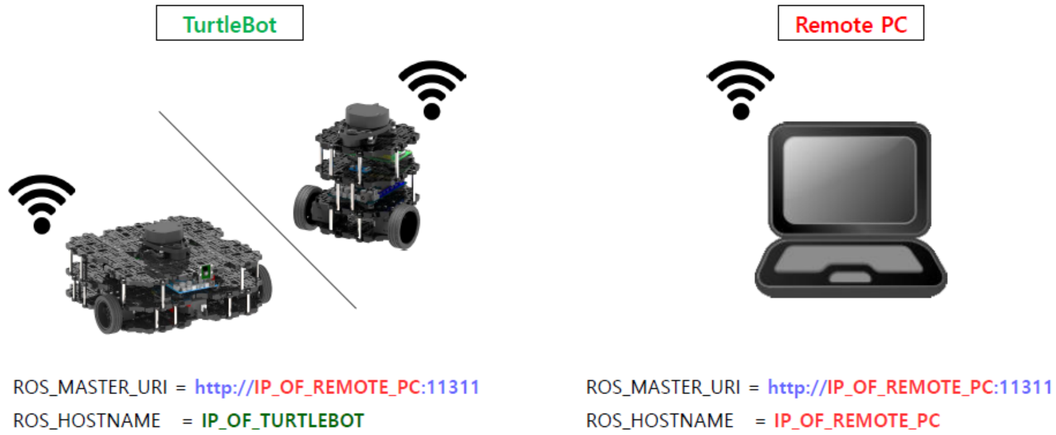
• 기본 설치 패키지 (SLAM, Navigation, Gazebo 관련)

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-t
wist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimag
e-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-
rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-am
cl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro r
os-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gma
pping ros-kinetic-navigation
```

```
$ cd ~/catkin_ws/src/
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
$ cd ~/catkin_ws&& catkin_make
```

02. TURTLEBOT3

– Turtulebot3 개발환경 (네트워크)



- **Laptop(노트북) 네트워크 설정**

1. Turtlebot3와 같은 wifi로 설정 (mechatronics_1 로 와이파이 설정)
2. Terminal에 ‘ifconfig’를 입력하여 ip addres를 획득 (inet address ~~)
3. Terminal에 ‘cd ~’를 입력하여 home으로 이동 (~$ 확인)
4. Terminal에 ‘sudo nano .bashrc’를 입력 (nano: 문서편집 기능)
   (.bashrc : Terminal이 켜질 때 작동될 내용들)
5. 맨 밑으로 이동하여 (alt + /) 다음과 같이 입력 후 저장 (ctrl + x)
   export ROS_MASTER_URI=http://IP_OF_REMOTE_PC:11311 (2의 IP 주소)
   export ROS_HOSTNAME = IP_OF_REMOTE_PC (2의 IP 주소)

6. $ source ~/.bashrc
   (.bashrc에 갱신된 내용을 실행시킴)

02. TURTLEBOT3

- Turtulebot3 개발환경 (네트워크) ➔ 2인 1조로 진행 (1인씩 실습)

터틀봇 위 라벨에 작성되어 있음

- **Turtlebot3 네트워크 설정**

1. Laptop에서 ssh pi@IP_OF_TURTLEBOT 입력 (Turtlebot3의 라즈베리파이에 원격 접속)
2. Password 입력 : turtlebot (입력값이 안보임)
3. Terminal에 'ifconfig'를 입력하여 ip address를 획득
4. Terminal에 'cd ~'를 입력하여 home으로 이동
5. Terminal에 'sudo nano .bashrc'를 입력
   (.bashrc : Terminal이 켜질 때 작동될 내용들)
6. 맨 밑으로 이동하여 (alt + /) 다음과 같이 입력 후 저장 (ctrl + x)
   export ROS_MASTER_URI=http://IP_OF_REMOTE_PC:11311 (노트북의 IP 주소)
   export ROS_HOSTNAME = IP_OF_TURTLEBOT
7. $ source ~/.bashrc
   (.bashrc에 갱신된 내용을 실행시킴)

03. REMOTE CONTROL

– Turtulebot3 원격 제어 [학생]

- Remote PC에서 roscore 구동하여 ROS Master 실행

—$ roscore

- 원격접속 명령어(ssh)를 통해 turtlebot으로 접속함. (앞장에서 이미 실행함)

—$ ssh pi@IP_OF_TURTLEBOT

- Turtlebot3 PC에서 turtlebot3_robot.launch 런치 파일 실행
  : turtlebot에 메시지가 전달되면 작동할 준비가 되도록 함.

—$ roslaunch turtlebot3_bringup turtlebot3_robot.launch --screen

- Remote PC에서 turtlebot3_teleop_key.launch 런치 파일을 실행하여 키보드로 조작

—$ export TURTLEBOT3_MODEL=waffle_pi

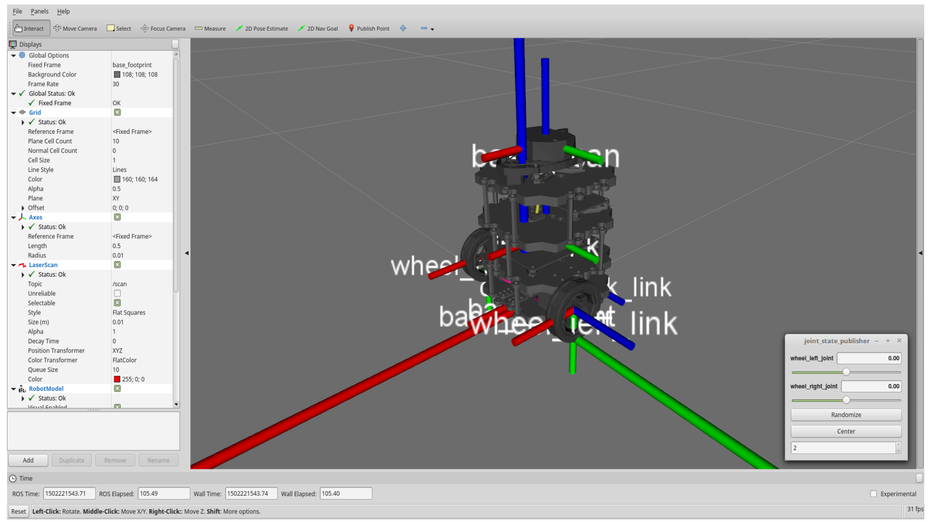
—$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch --screen

03. REMOTE CONTROL

– Turtulebot3 시각화 [학생]

- Remote PC에서 **Rviz** 를 실행하여 시각화

—$ export TURTLEBOT3_MODEL=waffle_pi
—$ roslaunch turtlebot3_bringup turtlebot3_model.launch



◆ Rviz 디스플레이 메뉴에서 Fixed frame: “odom”으로 선택

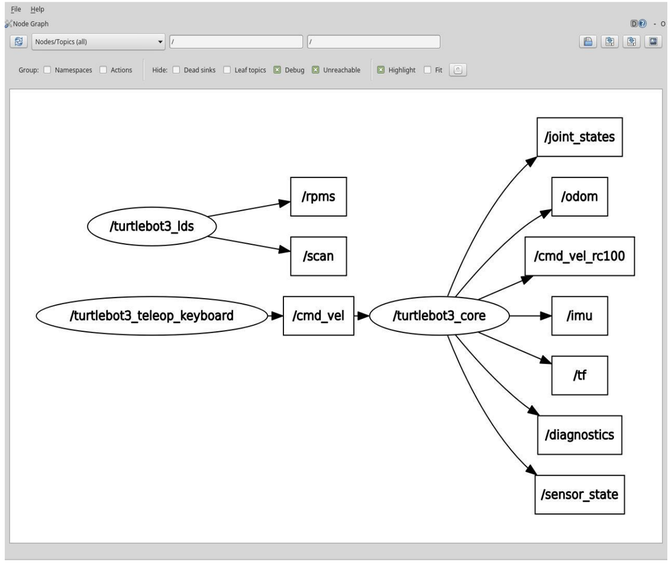
03. REMOTE CONTROL

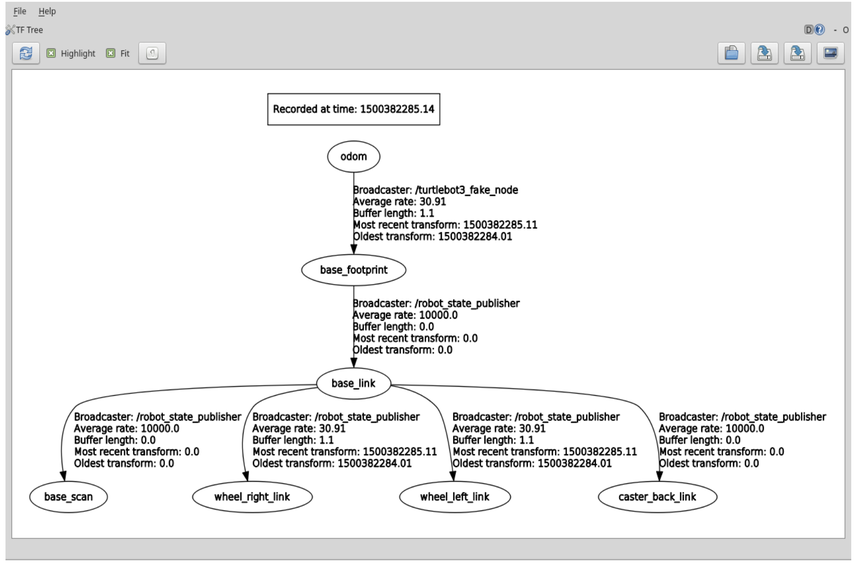
- Turtulebot3 Topic / TF [학생]

• Remote PC에서 rqt를 실행하여 Node Graph와 TF를 확인

—$ rqt_graph

—$ rqt　　(Plugins > Visualization > TF Tree)





THANK YOU

SOURCES

[1] ROBOTIS ROS SEMINAR, Pyo yunseok , 2018.1.4

[2] Programming Robots with ROS, Morgan Quigley / Brian Gerkey / William D. Smart, 2017.3.31

[3] ROS용어 정리, Monster's wastebasket, http://jungmonster.tistory.com/154

[4] roswiki(rosserial), http://wiki.ros.org/rosserial

[5] turtlebot3, http://turtlebot3.robotis.com