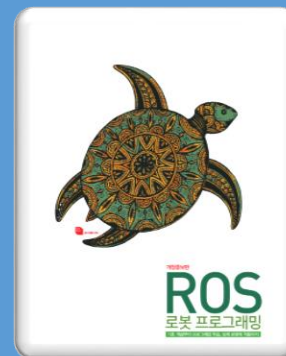


로봇, 센서, 모터

ROBOTIS

Open Source Team

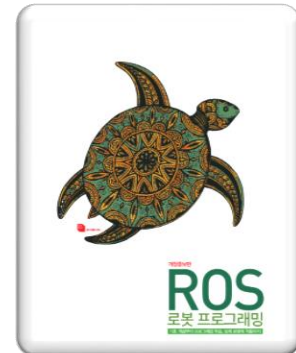
Yoonseok Pyo



교재
P. 204~243

Contents






























































































- I. 로봇 패키지
- II. 센서 패키지
- III. 카메라
- IV. 심도카메라
- V. 레이저 거리 센서
- VI. 모터 패키지
- VII. 다이나믹셀
- VIII. 공개 패키지 사용법



교재
P. 204~243

로봇 패키지

로봇 패키지 (<http://robots.ros.org/>)

| | | | | | | | | | | | |
|--|--|---|---|--|--|--|---|--|---|--|--|
|  210 Stanley Innovation V3 Segway |  220 Stanley Innovation V3 Segway |  223 Innok Heros |  224 Innok Heros |  Erle-Rover |  evarobot |  FANUC Robotics (ROS-Industrial) |  Festo Didactic Robotino |  Clearpath Robotics Grizzly |  Clearpath Robotics Husky |  Clearpath Robotics Jackal |  Clearpath Robotics Kingfisher |
|  420 Omni Stanley Innovation V3 Segway |  440LE Stanley Innovation V3 Segway |  440SE Stanley Innovation V3 Segway |  444 Innok Heros |  Fetch robotics: Fetch |  Fetch robotics: Freight |  Fraunhofer IPA Care-O-bot 3 |  Fraunhofer IPA Care-O-bot 4 |  Clearpath Robotics Ridgeback |  Cogniteam Hamster |  CoroWare Corobot |  Cyton-Gamma |
|  ABB Robotics (ROS-Industrial) |  Adept MobileRobots Pioneer family (P3DX, P3AT, ...) |  Blue Robotics BlueX |  Blue Robotics BlueSeeker Jr. |  GoTherel Robot |  i-Cart mini |  Ingenierie eRC |  Lego Mindstorms ADAS Development Vehicle |  Dataspeed Mobility Base |  Denso VS060 |  Dr. Robot Jaguar | |
|  Aldebaran Nao |  Allegro Hand SimLab |  Blue Robotics BlueX |  Blue Robotics BlueSeeker Jr. |  Intel Edison |  iRobot Roomba |  Kawada |  Kobuki |  Enova Robotics MiniLab |  Erle-Brain |  Erle-Brain 2 | |
|  Barrett Hand |  BipedRobin |  Bitcraze Crazyflie |  Blue Robotics BlueX |  Kinova JACO |  Kinova MICO |  Komodo |  Erle-Copter |  Erle-Copter Ubuntu Core special edition |  Erle-Hexacopter |  Erle-Plane | |
|  Lego NXT |  Lizi |  Maggie |  MecanumBot |  PAL Robotics REEM-C |  PAL Robotics TIAGo |  RazBot |  REEM |  ROS-Industrial |  Ros2Bot |  Shadow Hand |  Softbank Pepper |
|  Merlin miabotPro |  Milvus Robotics ATR |  Milvus Robotics MRP2 |  Milvus Robotics Robin |  Robonaut 2 |  RoboSavvy Self-balance platform |  Robotnik AGVS |  Robotnik GUARDIAN |  Tulip |  TurtleBot |  Universal Robots (ROS-Industrial) |  Videre Erratic |
|  Motoman, Yaskawa (ROS-Industrial) |  Nav2 |  Neobotix mp-500 |  Neobotix mpo-500 |  Robotnik RB-1 |  Robotnik RBCAR |  Robotnik SUMMIT XL |  Robotnik SUMMIT-X |  WheeledRobin |  Willow Garage PR2 |  Xaxxon Oculus Prime |  iRobot Roomba |

ROS
Enabled

센서 패키지

센서 패키지 (<http://wiki.ros.org/Sensors>)



센서 패키지의 종류

- **1D Range Finders**

- 저가의 로봇을 만들 때 사용할만한 적외선 방식의 직선거리 센서

- **2D Range Finders**

- 2차 평면상의 거리를 측정할 수 있는 센서로 주로 내비게이션에 많이 사용되는 센서

- **3D Sensors**

- Intel 사의 RealSense, Microsoft 사의 Kinect, ASUS 사의 Xtion과 같은 3차원 거리 측정에 사용되는 센서

- **Audio/Speech Recognition**

- 현재 음성인식 관련 부분은 매우 적지만, 지속해서 추가될 것으로 보임

- **Cameras**

- 물체인식, 얼굴인식, 문자판독 등에 많이 사용되는 카메라의 드라이버 및 각종 응용 패키지

- **Sensor Interfaces**

- USB 및 웹 프로토콜을 지원하는 센서는 매우 적음
- 아직까지도 많은 센서들은 마이크로프로세서에서 정보를 쉽게 얻을 수 있는 센서가 많음
- 이러한 센서는 마이크로프로세서의 UART 및 미니 PC 계열에서 ROS와의 연결을 지원함

실습 시간

'LRF, IMU, USB camera,
Depth camera, Robot Model
을 RViz를 통해 확인해보자.'

나눠 드린 센서들을

각자 PC의 RViz를 이용해 데이터를 확인해보세요.

오늘의 실습 준비물



센서 패키지 실습 #1 (USB Camera)

```
$ sudo apt-get install ros-kinetic-udev-camera
```

```
$ roslaunch uvc_camera uvc_camera_node
```

```
$ roslaunch uvc_camera uvc_camera_node _device:=/dev/video?
```

```
$ roslaunch image_view image_view image:=/image_raw
```

```
$ rqt_image_view image:=/image_raw
```

```
$ rviz
```

카메라가 2대 이상일 경우,
물음표 대신 사용하기 원하는
디바이스 번호를 입력 (특히, 노트북의 경우)

} 이미지 메시지를 보는 3가지 방법

* RViz의 Displays 옵션 변경

1) Fixed Frame 변경

Global Options > Fixed Frame = camera

2) 이미지 디스플레이 추가

rviz 좌측 하단의 Add 클릭한 후, Image 선택하여 추가한다.

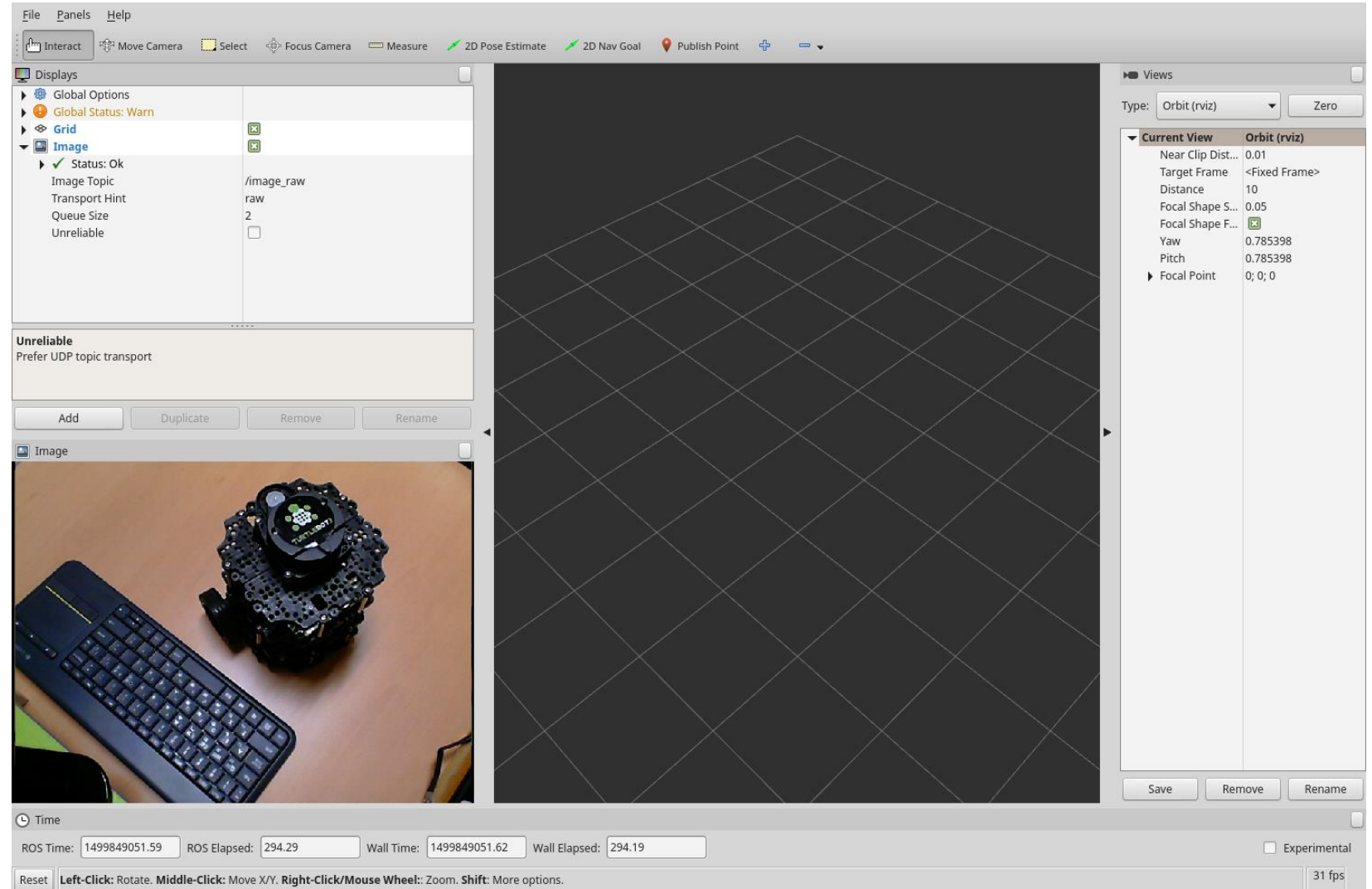
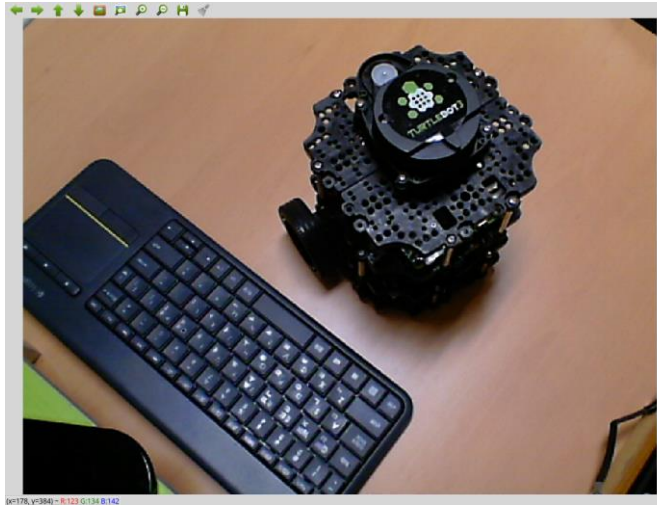
(Add > by display > rviz > Image)

3) 토픽 값 변경

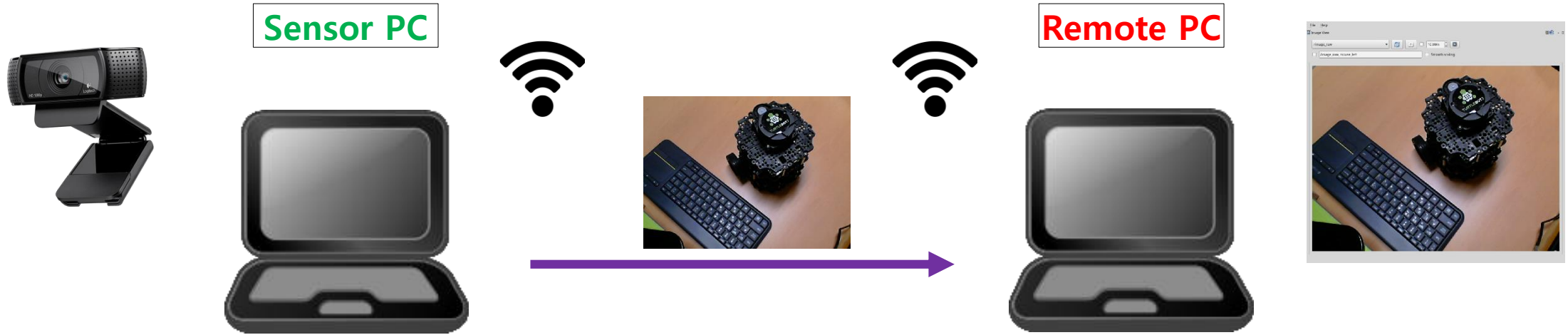
Image > Image Topic 의 값을 "/image_raw" 로 변경한다.



센서 패키지 실습 #1 (USB Camera)



센서 패키지 실습 #2 (원격으로 이미지 전송)



ROS_MASTER_URI = http://IP_OF_REMOTE_PC:11311

ROS_HOSTNAME = [IP_OF_SENSOR_PC](#)

ROS_MASTER_URI = http://IP_OF_REMOTE_PC:11311

ROS_HOSTNAME = [IP_OF_REMOTE_PC](#)

* ROS Master 를 Remote PC에서 구동했을 때의 예제

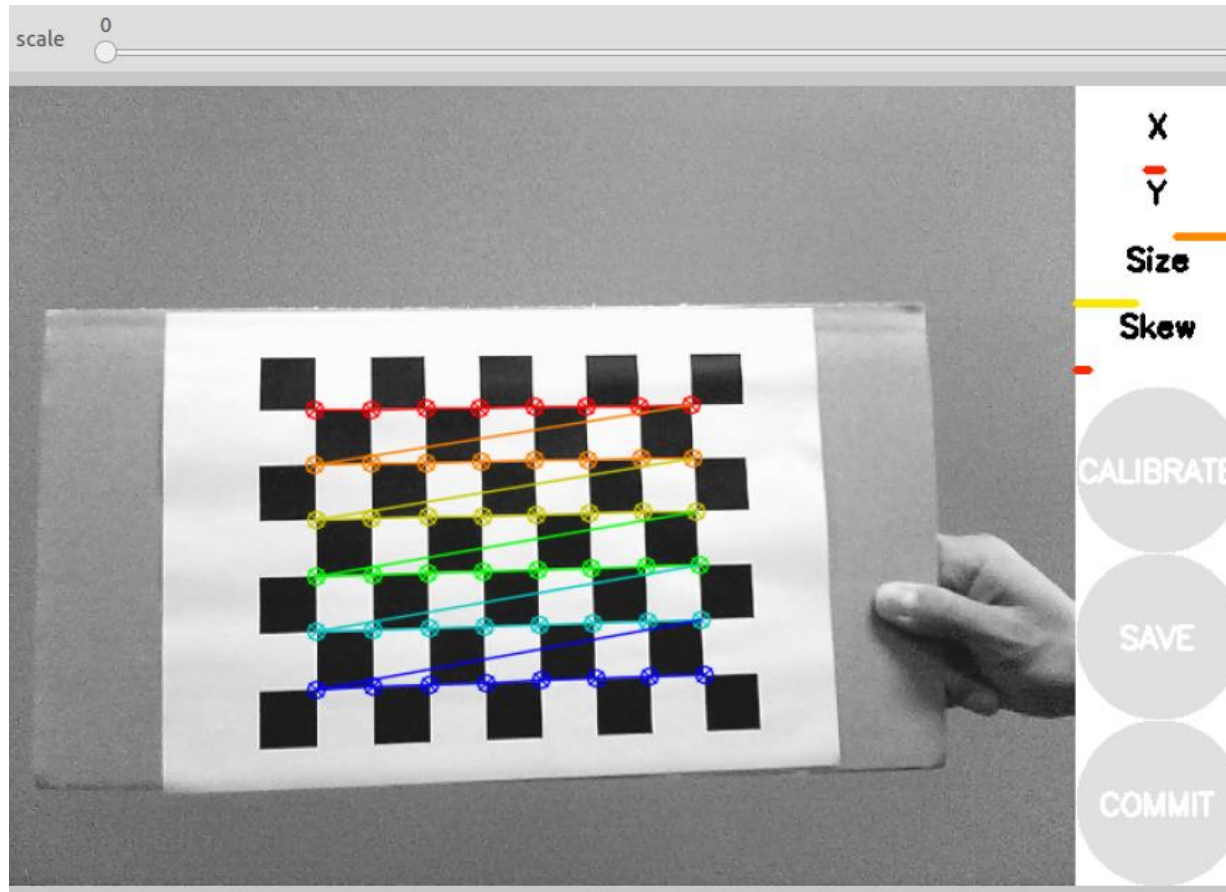
- 각 PC의 ~/.bashrc 파일 수정 (ROS_MASTER_URI 및 ROS_HOSTNAME)
- Remote PC에서 **roscore** 및 **rqt_image_view image:=/image_raw** 실행
- Sensor PC에서 **roslaunch uvc_camera uvc_camera_node** 실행

센서 패키지 실습 #3 (카메라 캘리브레이션)

```
$ sudo apt-get install ros-kinetic-camera-calibration
```

```
$ rosruncamera uvc_camera uvc_camera_node
```

```
$ rosruncamera_calibration cameracalibrator.py --size 8x6 --square 0.024 image:=/image_raw camera:=/camera
```



센서 패키지 실습 #4 (Depth Camera)

```
$ sudo apt-get install ros-kinetic-openni2-camera ros-kinetic-openni2-launch (ASUS사의 Xtion의 경우)  
$ tar -xvf Sensor-Bin-Linux-x64-v5.1.0.41.tar.bz2 (*Xtion 구매시 CD안에 있음 또는 http://cafe.naver.com/openrt/6070)  
$ cd Sensor-Bin-Linux-x64-v5.1.0.41/  
$ sudo sh install.sh  
$ roslaunch openni2_launch openni2.launch
```

```
$ sudo apt-get install ros-kinetic-astra-camera ros-kinetic-astra-launch (ASTRA의 경우)  
$ wget https://raw.githubusercontent.com/tfoote/ros_astra_camera/master/orbbec-usb.rules  
$ wget https://raw.githubusercontent.com/tfoote/ros_astra_camera/master/install.sh  
$ sudo ./install.sh  
$ roslaunch astra_launch astra.launch
```

* RViz의 Displays 옵션 변경

1) Fixed Frame 변경

Global Options > Fixed Frame 을 "[camera_depth_frame](#)" 로 변경한다.

2) PointCloud2 추가 및 설정

rviz 좌측 하단의 Add 클릭한 후, [PointCloud2](#)를 선택하여 추가한다.

3) Topic 이름 및 세부 설정 변경



센서 패키지 실습 #4 (Depth Camera)

(RealSense의 경우)

```
$ sudo apt-get install ros-kinetic-librealsense ros-kinetic-realsense-camera  
$ roslaunch realsense_camera r200_nodelet_default.launch  
$ rosrn rviz rviz -d rviz/realsenseRvizConfiguration1.rviz
```

* RViz의 Displays 옵션 변경

1) Fixed Frame 변경

Global Options > Fixed Frame 을 "[camera_depth_frame](#)" 로 변경한다.

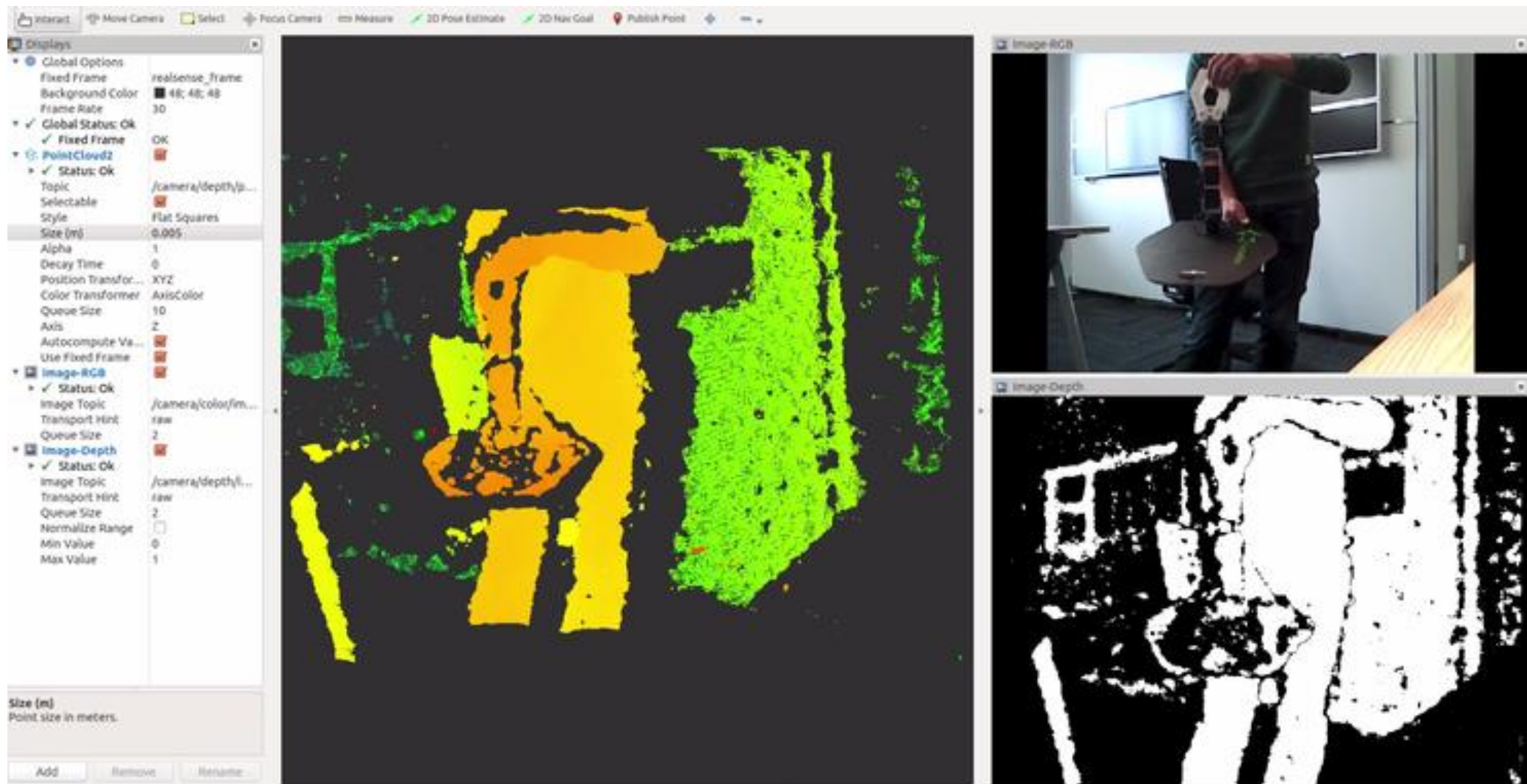
2) PointCloud2 추가 및 설정

rviz 좌측 하단의 Add 클릭한 후, [PointCloud2](#)를 선택하여 추가한다.

3) Topic 이름 및 세부 설정 변경



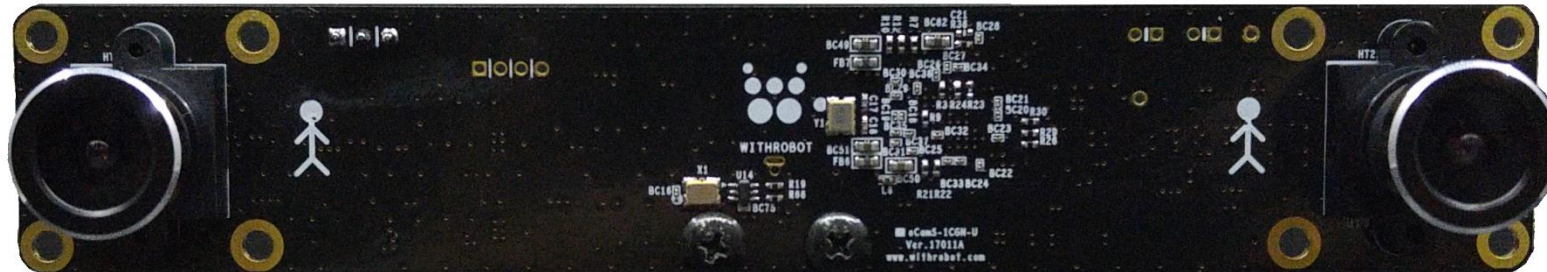
센서 패키지 실습 #4 (Depth Camera)



센서 패키지 실습 #5 (Stereo Camera)

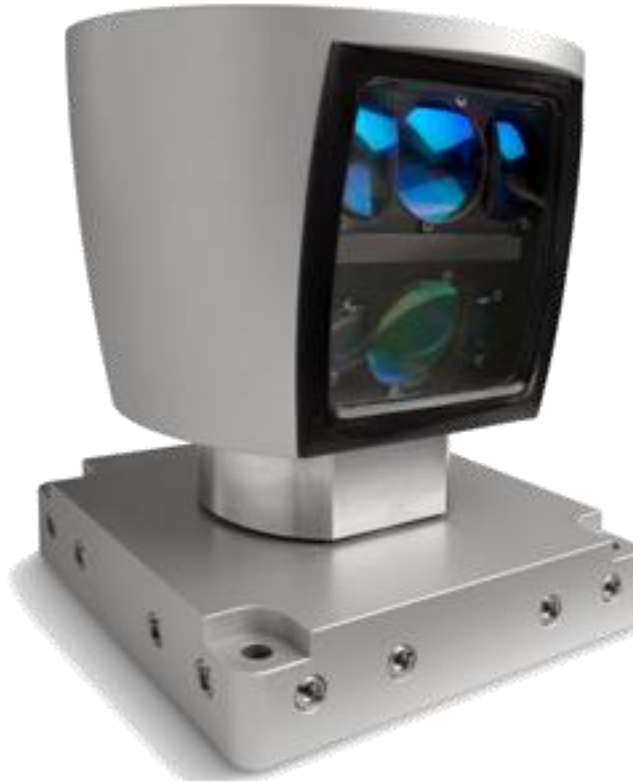
```
$ sudo apt-get install libv4l-dev libudev-dev ros-kinetic-rtabmap*  
$ cd ~/catkin_ws/src/  
$ svn export https://github.com/withrobot/oCam/trunk/Software/oCamS_ROS_Package/ocams  
$ cd ~/catkin_ws/ && catkin_make  
$ sudo gedit /etc/udev/rules.d/99-ttyacms.rules  
ATTRS{idVendor}=="04b4" ATTRS{idProduct}=="00f9", MODE="0666", ENV{ID_MM_DEVICE_IGNORE}="1"  
ATTRS{idVendor}=="04b4" ATTRS{idProduct}=="00f8", MODE="0666", ENV{ID_MM_DEVICE_IGNORE}="1"  
$ sudo udevadm control --reload-rules  
$ roslaunch ocams pointcloud.launch
```

(oCam-Stereo의 경우)



<https://github.com/withrobot/oCam/tree/master/Products/oCamS-1CGN-U>

센서 패키지 실습 #6 (LDS)



센서 패키지 실습 #6 (LDS)

```
$ cs
$ git clone https://github.com/ROBOTIS-GIT/hls_lfcd_lds_driver.git
$ cd
$ sudo chmod a+rw /dev/ttyUSB0
$ roslaunch hls_lfcd_lds_driver view_hlds_laser.launch
```

(LDS의 경우)

```
$ cs
$ git clone https://github.com/robopeak/rplidar_ros.git
$ cd
$ sudo chmod a+rw /dev/ttyUSB0
$ roslaunch rplidar_ros rplidar.launch
```

(RPLiDAR의 경우)

```
$ sudo apt-get install ros-kinetic-urg-node
$ sudo chmod a+rw /dev/ttyACM0
$ rosrn urg_node urg_node
```

(HOKUYO의 경우)

* RViz의 Displays 옵션 변경

- 1) Fixed Frame 변경: Global Options > Fixed Frame = laser
- 2) Axes 추가 및 설정: rviz 좌측 하단의 Add 클릭한 후, Axes 선택하여 추가한다. (Length 및 Radius 변경은 옵션)
- 3) LaserScan 추가 및 설정: rviz 좌측 하단의 Add 클릭한 후, LaserScan 선택하여 추가한다.
(Topic 지정은 필수, Color Transformer, Color 등은 옵션)

센서 패키지 실습 #6 (LDS)

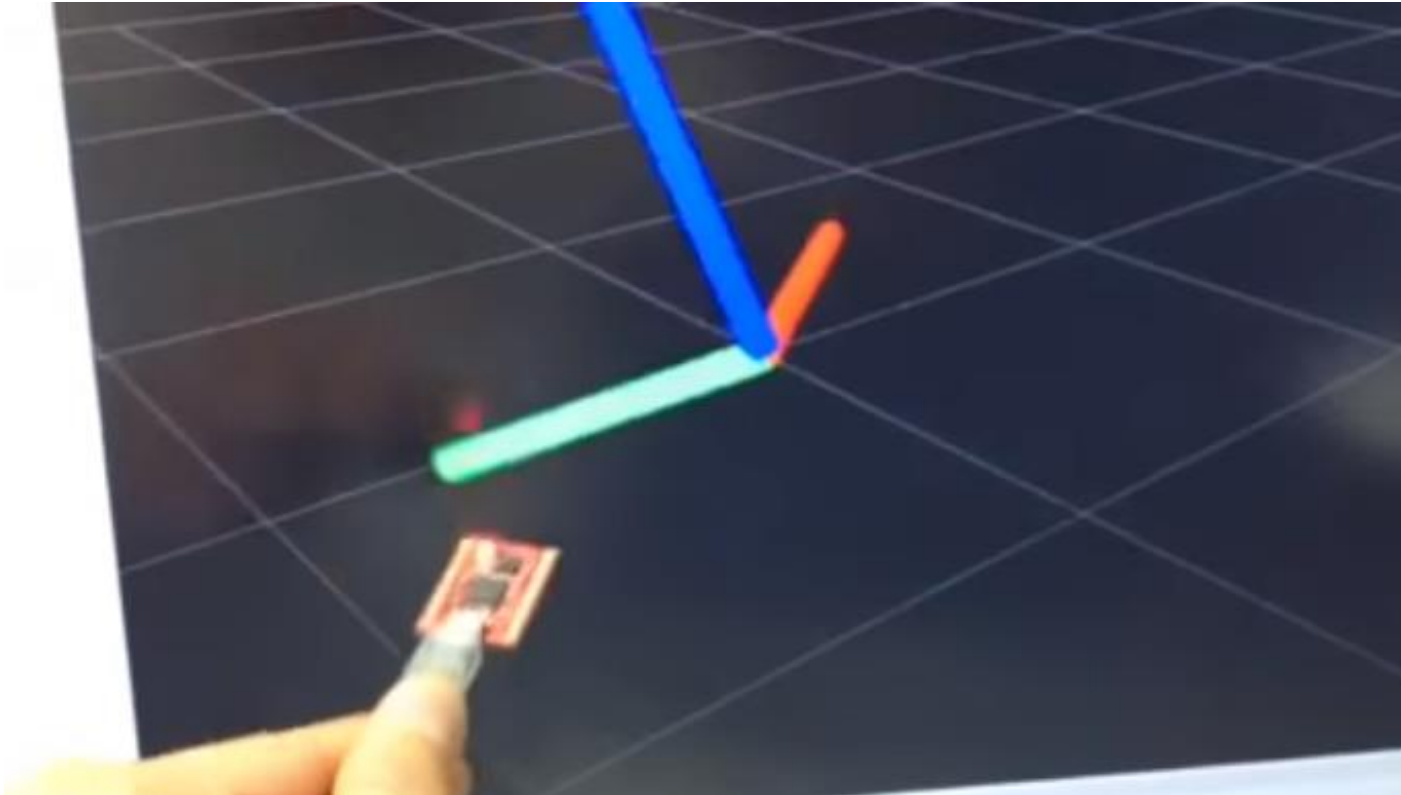
The screenshot displays the ROS2 GUI with the following components and annotations:

- Displays Panel (Left):**
 - Global Options:** Fixed Frame: laser (Annotated: 고정프레임 : laser)
 - Grid:** Status: Ok, Reference Frame: <Fixed Frame>, Plane Cell Count: 50, Normal Cell Count: 0, Cell Size: 1, Line Style: Lines, Color: 160; 160; 164, Alpha: 0.5, Plane: XY, Offset: 0; 0; 0
 - Axes:** Status: Ok, Reference Frame: <Fixed Frame>, Length: 1 (Annotated: 축의 길이 : 1), Radius: 0.1 (Annotated: 축의 반경 : 0.1)
 - LaserScan:** Status: Ok, Topic: /scan (Annotated: 토픽명 : /scan), Unreliable: ☐, Selectable: ☒, Style: Flat Squares, Size (m): 0.05, Alpha: 1, Decay Time: 0, Position Transformer: XYZ, Color Transformer: FlatColor (Annotated: 색상변환기준 : FlatColor), Queue Size: 10, Color: 255; 0; 0 (Annotated: 색상 : 255;0;0 (빨강))
- Views Panel (Right):** Type: TopDownOrtho (rviz) (Annotated: Views:TopDownOrtho)
- 3D Visualization (Center):** A top-down orthographic view showing a laser scan as red points and a green line representing the scan's path.
- Time Panel (Bottom):** ROS Time: 1499957187.23, ROS Elapsed: 219.22, Wall Time: 1499957187.27, Wall Elapsed: 219.12

센서 패키지 실습 #7 (IMU)

```
$ cs  
$ git clone https://github.com/robotpilot/myahrs_driver.git  
$ cd  
$ sudo chmod a+rw /dev/ttyACM0  
$ roslaunch myahrs_driver myahrs_driver.launch
```

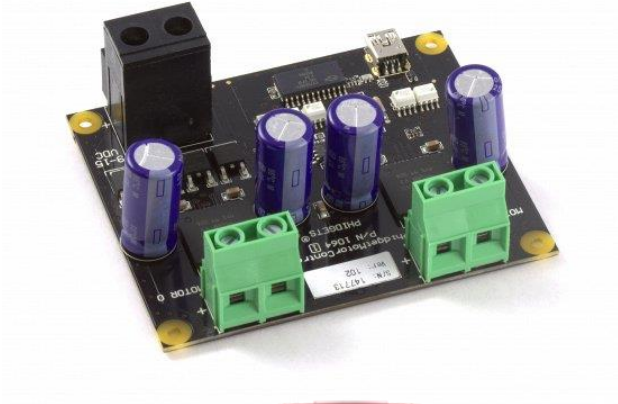
(withrobot사의 myAHRS+)



모터 패키지

모터 패키지 <http://wiki.ros.org/Motor%20Controller%20Drivers>

- PhidgetMotorControl HC
- Roboteq AX2550 Motor Controller
- ROBOTIS Dynamixel



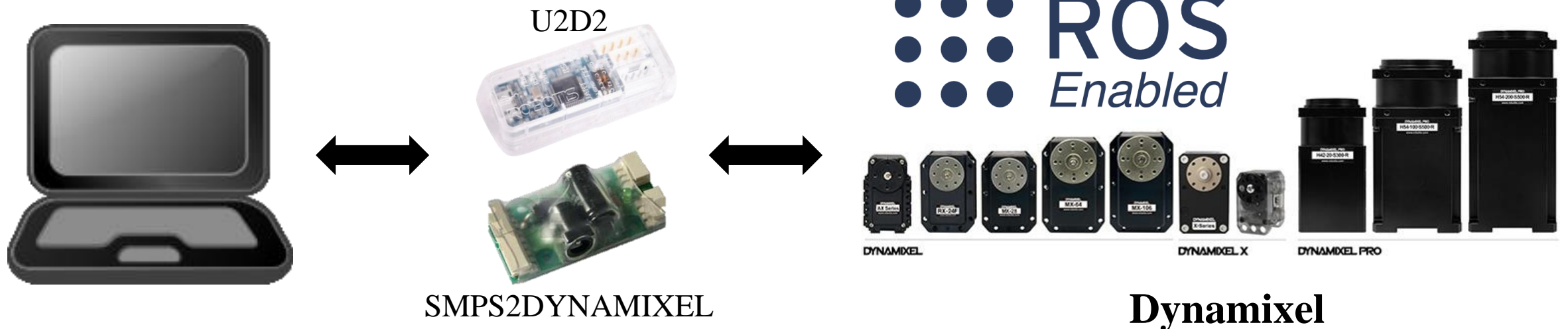
 **ROS**
Enabled



ROS 패키지를 이용하여 다이나믹셀 제어하기

- **DynamixelSDK** (http://wiki.ros.org/dynamixel_sdk)
 - 3대 OS지원 (Linux, Windows, MacOS)
 - C, C++, C#, Python, Java, MATLAB, LabVIEW 등의 프로그래밍 언어 지원
 - ROS 지원
- **dynamixel_workbench** (http://wiki.ros.org/dynamixel_workbench)
 - ROS에서 사용하기 쉽도록 다양한 예제 제공
 - ROS 대응 GUI 툴 제공

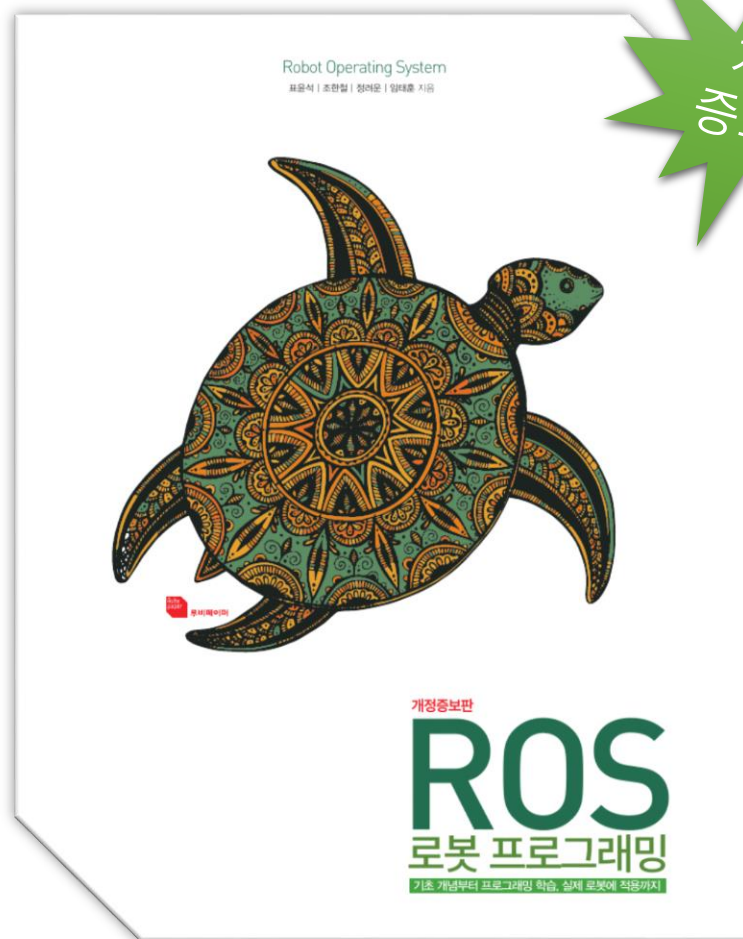
DYNAMIXEL SDK



질문 대환영!

* 온라인 상의 질문이라면
오로카 및 로열모를 이용해주세요!

여기서! 광고 하나 나가요~



✓ Direct Link

국내 유일! 최초! ROS 참고서!
ROS 공식 플랫폼 **TurtleBot3** 개발팀이
직접 저술한 바이블급 ROS 책

여기서! 광고 둘 나가요~

TURTLEBOT3

인공지능(AI) 연구의 시작, ROS 교육용 공식 로봇 플랫폼

터틀봇3는 ROS기반의 저가형 모바일 로봇으로
교육, 연구, 제품개발, 취미 등 다양한 분야에서 활용 할 수 있습니다.

✓ Direct Link



• Collaboration with  open robotics  intel

여기서! 광고 셋 나가요~



- ✓ • 오로카
- www.oroqa.org
- 오픈 로보틱스 지향
- 풀뿌리 로봇공학의 저변 활성화
- 공개 강좌, 세미나, 프로젝트 진행

- ✓ • 로봇공학을 위한 열린 모임 (KOS-ROBOT)
- www.facebook.com/groups/KoreanRobotics
- 로봇공학 통합 커뮤니티 지향
- 일반인과 전문가가 어울러지는 한마당
- 로봇공학 소식 공유
- 연구자 간의 협력

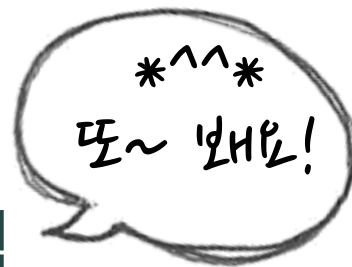
혼자 하기에는 답답하시다고요?

커뮤니티에서 함께 해요~

끝.

표윤석

Yoonseok Pyo
pyo@robotis.com
www.robotpilot.net



www.facebook.com/yoonseok.pyo