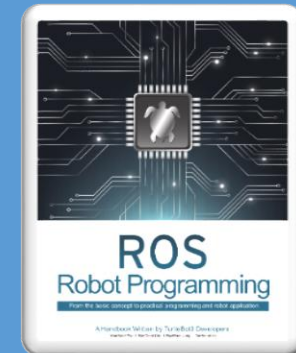


Robot, Sensor, Motor

ROBOTIS

KAIST



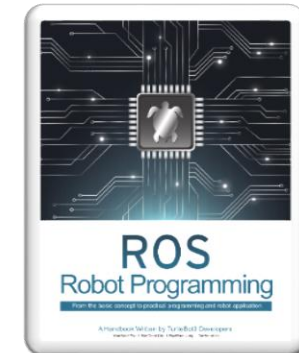
You Tube

Subscribe

Textbook
P. 194~231

Contents

- I. Robot Packages
- II. Sensor Packages
- III. Camera
- IV. Depth Camera
- V. Laser Distance Sensor
- VI. Motor Packages
- VII. Dynamixel
- VIII. How to use Open Packages



































































































You Tube

Subscribe

Textbook
P. 194~231

Robot Packages

Robot Package (<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.	 Blue Robotics BlueX	 GoTherel Robot	 i-Cart mini	 Ingenierie eRC	 Learner ADAS Development Vehicle	 Dataspeed Mobility Base	 Denso VS060	 Dr. Robot Jaguar
 Aldebaran Nao	 Allegro Hand SimLab	 Allegro Hand SimLab	 Allegro Hand SimLab	 Allegro Hand SimLab	 Intel Edison	 iRobot Roomba	 Kawada	 Kawada	 Enova Robotics MiniLab	 Erle-Brain	 Erle-Brain 2
 Barrett Hand	 BipedRobin	 Bitcraze Crazyflie	 Blue Robotics BlueROV	 Kinova JACO	 Kinova MICO	 Kobuki	 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

Sensor Packages

Sensor Package (<http://wiki.ros.org/Sensors>)



Type of sensor package

- **1D Range Finders**
 - Infrared linear distance sensor that can be used to make low-cost robots
- **2D Range Finders**
 - Sensors that can measure the distance on 2D plane, and is mainly used for navigation
- **3D Sensors**
 - Sensors used in 3D distance measurement such as Intel's RealSense, Microsoft's Kinect, ASUS's Xtion
- **Audio/Speech Recognition**
 - Currently, there are few voice recognition related parts, but it seems to be added continuously
- **Cameras**
 - Camera driver used for object recognition, face recognition, character recognition, etc. and various application packages
- **Sensor Interfaces**
 - Very few sensors support USB and web protocols
 - There are still many sensors that can acquire data from a microprocessor
 - These sensors can be used with UART in MCU, or ROS in mini PC.

Practice Time

'LRF, IMU, USB camera,
Depth camera, Robot Model
Let's check through Rviz'

Use the Rviz of your Pc to observe the data
from the sensors that you have received

Today's Practice Material



Sensor Package Practice #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
```

If there are more than two cameras,
Enter the device number you want to use
instead of the question mark
(Especially, for notebooks)

Three ways to view image messages

* Change the display options of RViz

1) Change fixed frame

Global Options > Fixed Frame = camera

2) Add image display

Click 'Add' in the bottom left corner of Rviz, then select Image

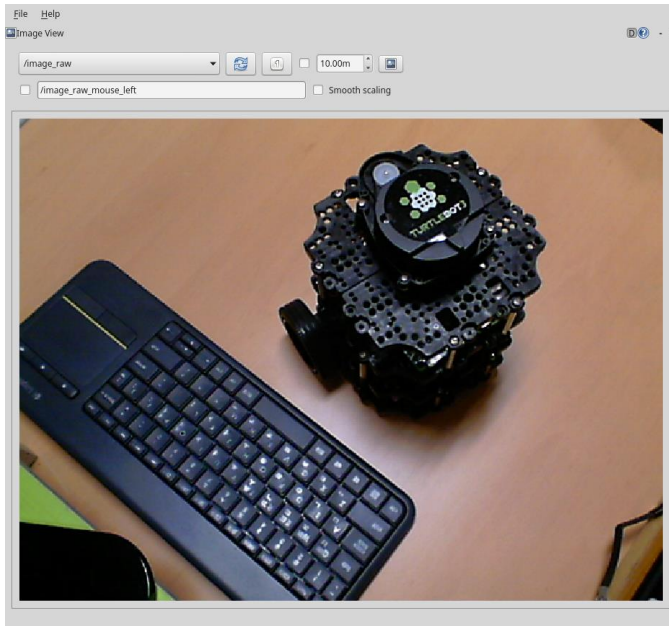
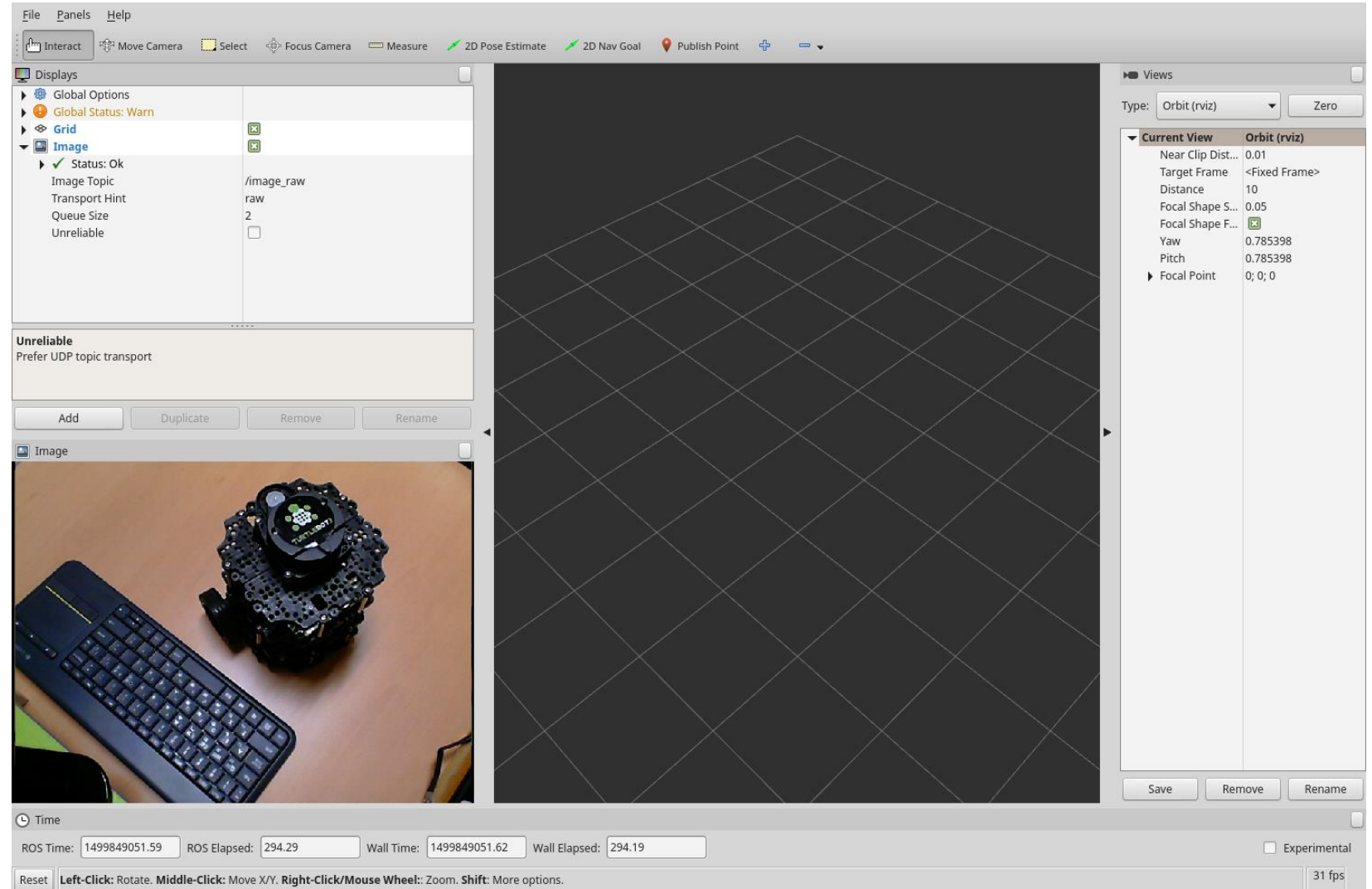
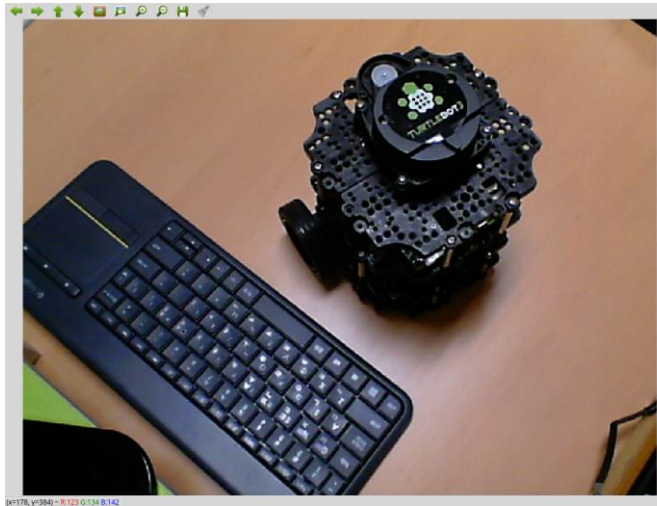
(Add > by display > Rviz > Image)

3) Change topic value

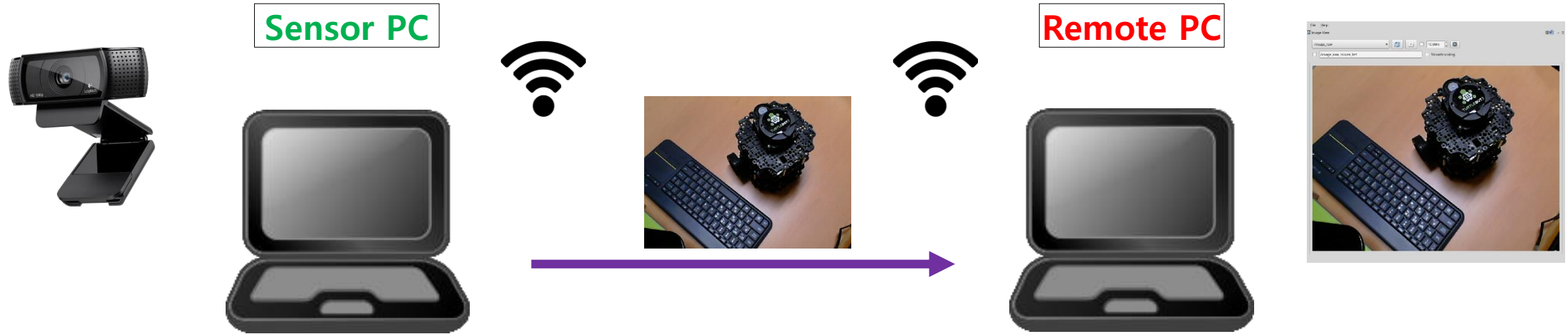
Change the value of 'Image > Image Topic' to "/image_raw"



Sensor Package Practice #1 (USB Camera)



Sensor Package Practice #2 (Transfer images remotely)



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](#)

* Example of running ROS Master on a remote PC

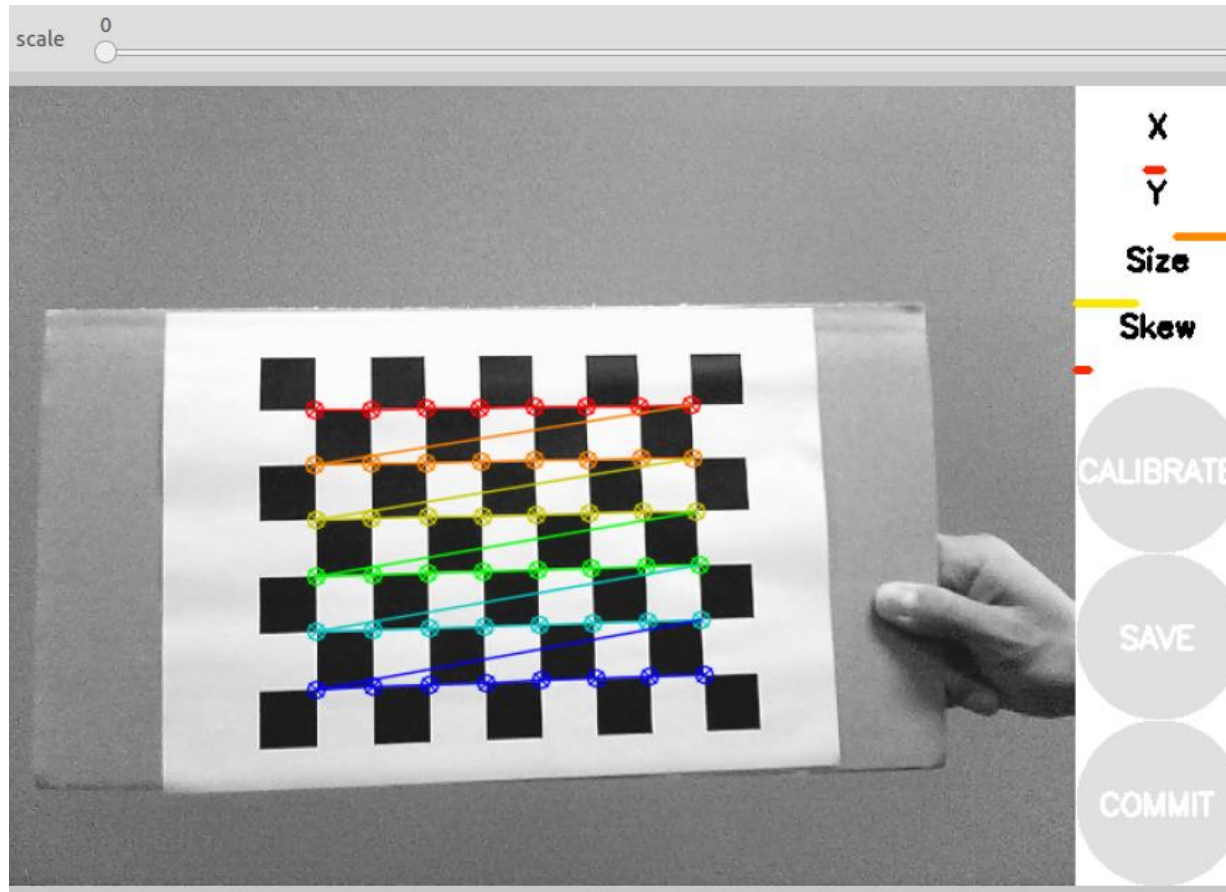
- Modify '`~/.bashrc`' for each PC (ROS_MASTER_URI and ROS_HOSTNAME)
- Run '`roscore`' & '`rqt_image_view image:=/image_raw`' on the remote PC
- Run '`roslaunch uvc_camera uvc_camera_node`' on the sensor PC

Sensor Package Practice #3 (Camera Calibration)

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

```
$ rosrunc uvc_camera uvc_camera_node
```

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



Sensor Package Practice #4 (Depth Camera)

```
$ sudo apt-get install ros-kinetic-openni2-camera ros-kinetic-openni2-launch (In case of ASUS's Xtion)
$ tar -xvf Sensor-Bin-Linux-x64-v5.1.0.41.tar.bz2
$ 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 (In case of ASTRA)
$ 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
```

* Change the display options of RViz

1) Change fixed frame

Change 'Global Options > Fixed Frame' to "[camera_depth_frame](#)"

2) Add & configure PointCloud2

Click 'Add' at the bottom left of rviz, then select [PointCloud2](#)

3) Change topic name & detail settings



Sensor Package Practice #4 (Depth Camera)

(In case of RealSense)

```
$ 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
```

* Change the display options of RViz

1) Change fixed frame

Change 'Global Options > Fixed Frame' to "[camera_depth_frame](#)"

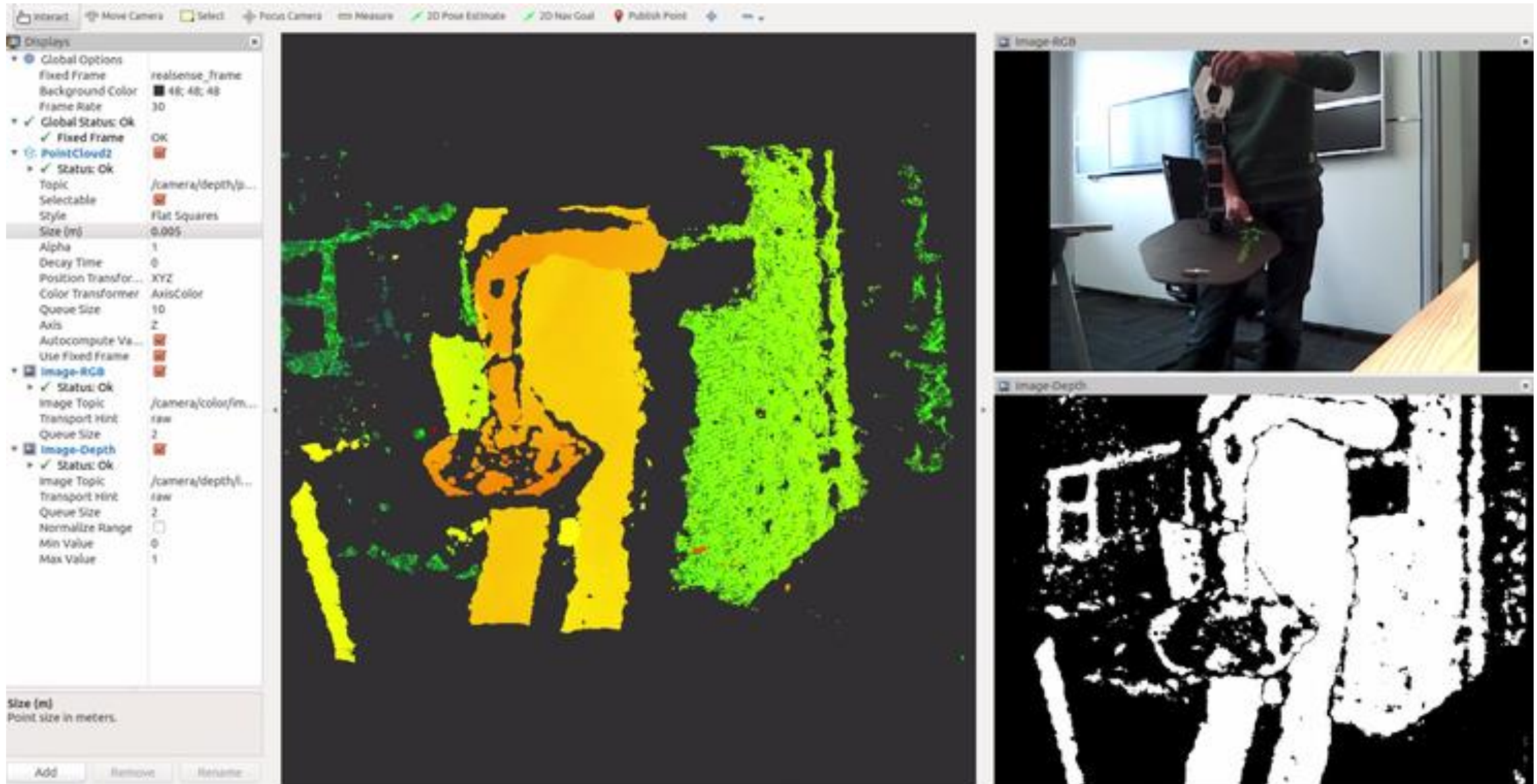
2) Add & configure PointCloud2

Click 'Add' at the bottom left of rviz, then select [PointCloud2](#)

3) Change topic name & detail settings



Sensor Package Practice #4 (Depth Camera)



Sensor Package Practice #5 (Stereo Camera)

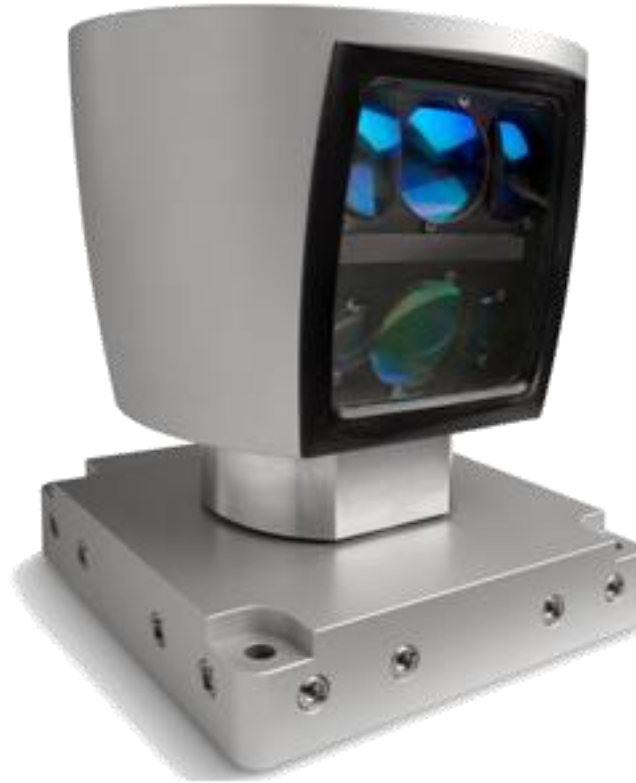
```
$ sudo apt-get install libv4l-dev libudev-dev ros-kinetic-rtabmap*  
$ cd ~/catkin_ws/src/  
$ svn export https://github.com/withrobot/oCam/trunk/Software/oCamS_ROS_Package/ocams  
$ cd ~/catkin_ws/ && catkin_make  
$ sudo gedit /etc/udev/rules.d/99-ttyacms.rules  
ATTRS{idVendor}=="04b4" ATTRS{idProduct}=="00f9",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

(In case of oCam-Stereo)



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

Sensor Package Practice #6 (LDS)



Sensor Package Practice #6 (LDS)

```
$ cs (In case of LDS)
$ 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
```

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

```
$ sudo apt-get install ros-kinetic-urg-node (In case of HOKUYO)
$ sudo chmod a+rw /dev/ttyACM0
$ rosrn urg_node urg_node
```

* Change the display options of RViz

- 1) Change fixed frame: Global Options > **Fixed Frame = laser**
- 2) Add & configure Axes: Click 'Add' at the bottom left of rviz, then add **Axes** (Change 'Length' & 'Radius' is option)
- 3) Add & configure LaserScan: Click 'Add' at the bottom left of rviz, then add **LaserScan**
(**Topic** designation is required, 'Color Transformer', 'Color', etc. are options)

Sensor Package Practice #6 (LDS)

The screenshot displays the RViz (Robot Visualization) interface with a 2D **TopDownOrtho** view. The main visualization area shows a laser scan (LDS) as a series of red points forming a grid-like pattern. A green line and a red line are also visible, originating from a blue point.

Views: TopDownOrtho

Fixed Frame : laser

Axis Length : 1

Axis Radius : 0.1

Topic name : /scan

Color Conversion Criteria : FlatColor

Color : 255;0;0 (Red)

Displays

- Global Options
 - Fixed Frame: laser
 - Background Color: 48; 48; 48
 - Frame Rate: 30
- 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
 - Radius: 0.1
- LaserScan
 - Status: Ok
 - Topic: /scan
 - Unreliable: ☐
 - Selectable: ☒
 - Style: Flat Squares
 - Size (m): 0.05
 - Alpha: 1
 - Decay Time: 0
 - Position Transformer: XYZ
 - Color Transformer: FlatColor
 - Queue Size: 10
 - Color: 255; 0; 0

Views

Type: TopDownOrtho (rviz) Zero

Current View TopDownOrtho (rviz)

Near Clip Dist...	0.01
Target Frame	<Fixed Frame>
Scale	192.139
Angle	-0.514998
X	-0.989068
Y	1.62129

LaserScan

Displays the data from a sensor_msgs::LaserScan message as points in the world, drawn as points, billboards, or cubes.

[More Information.](#)

Add Duplicate Remove Rename

Time

ROS Time: 1499957187.23 ROS Elapsed: 219.22 Wall Time: 1499957187.27 Wall Elapsed: 219.12

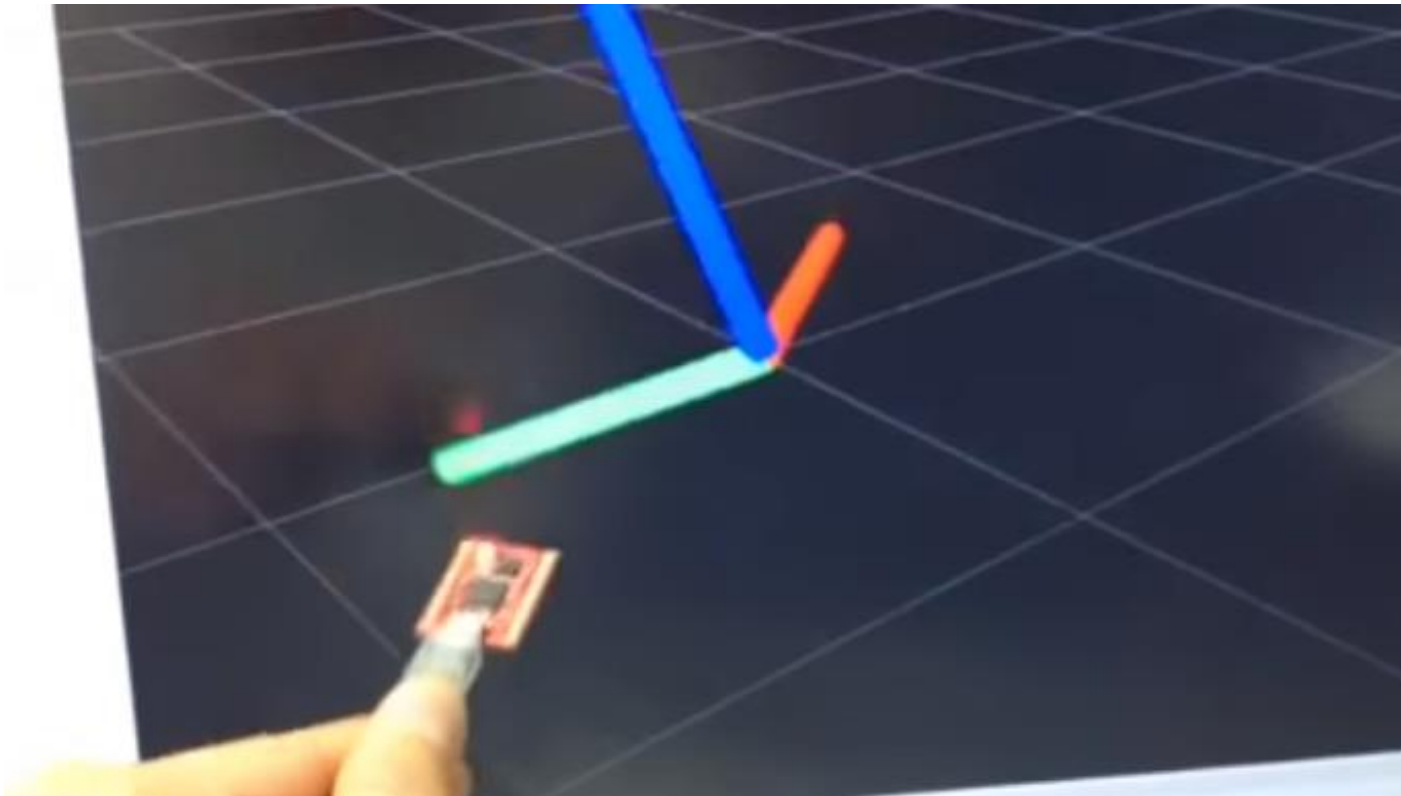
Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Zoom. Shift: More options.

Experimental 31 fps

Sensor Package Practice #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
```

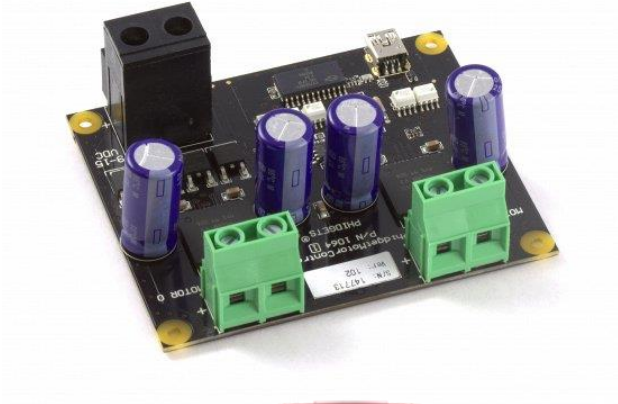
(In case of withrobot's myAHRS+)



Motor Packages

Motor Package (<http://wiki.ros.org/Motor%20Controller%20Drivers>)

- PhidgetMotorControl HC
- Roboteq AX2550 Motor Controller
- ROBOTIS Dynamixel



 **ROS**
Enabled

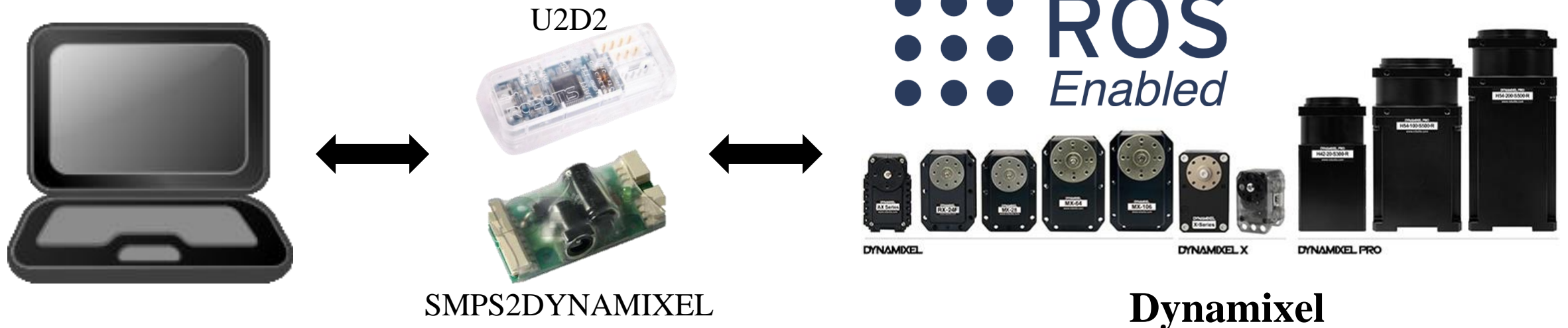


Contorolling Dynamixel with ROS Package

- **DynamixelSDK** (http://wiki.ros.org/dynamixel_sdk)
 - Support 3 representative OS (Linux, Windows, MacOS)
 - Support programming language such as C, C++, C#, Python, Java, MATLAB, LabVIEW, etc.
 - Support ROS

DYNAMIXEL SDK

- **dynamixel_workbench** (http://wiki.ros.org/dynamixel_workbench)
 - Provide a variety of examples for ease of use in ROS
 - Provide GUI tool for ROS



Question Time!

Advertisement #1



Free

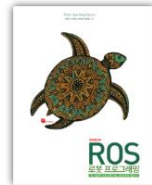


Download link



Language:

English, chinese, Japanese, Korean



“ROS Robot Programming”

A Handbook is written by TurtleBot3 Developers

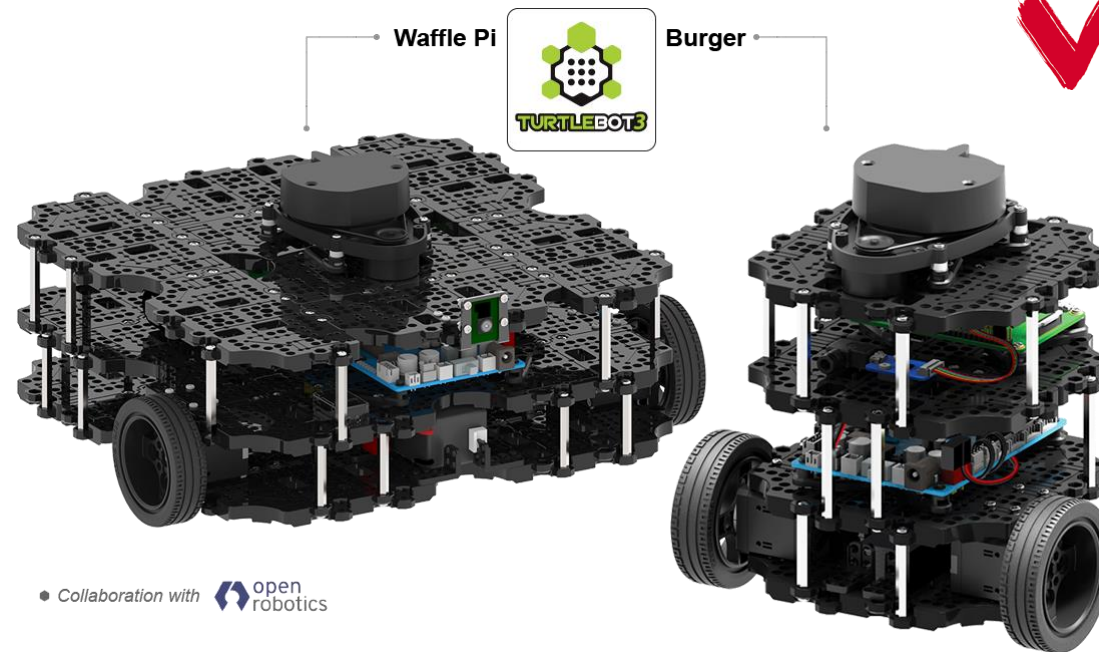
Advertisement #2

TURTLEBOT3

AI Research Starts Here
ROS Official Platform

TurtleBot3 is a new generation mobile robot that's modular, compact and customizable. Let's explore ROS and create exciting applications for education, research and product development.

✓ [Direct Link](#)



Advertisement #3



www.robotsource.org

The 'RobotSource' community is the space for people making robots.

We hope to be a community where we can share knowledge about robots, share robot development information and experiences, help each other and collaborate together. Through this community, we want to realize open robotics without distinguishing between students, universities, research institutes and companies.

Join us in the Robot community ~

END.