

# Polaris3d Kepler ROS Guide

---

Polaris3d Kepler를 ROS 환경에서 사용하는 유저 가이드 문서입니다. Kepler 데이터로 ROS 기반 어플리케이션을 개발할 때 사용할 수 있는 topic 목록과 설명 등을 제공합니다.

## 개요

### ROS Network Configuration

Kepler IP(Default): 192.168.1.20

Kepler IP(Default): 29721

### ROS topic 사용 예제

본 문서는 ROS 사용에 대한 이해가 있다는 전제 하에 작성되었습니다.  
ROS 사용이 익숙하지 않으시다면, 아래 가이드 문서를 참고해주세요.

#### - ROS Guide(Korean)

[<http://wiki.ros.org/ko/ROS/Tutorials>]

#### - ROS Guide(English)

[<http://wiki.ros.org/ROS/Tutorials>]

## ROS Publish Topic 분류

Kepler가 게시하는 topic들의 분류는 다음과 같습니다.

Type	Descriptions
<a href="#">Sensor</a>	Kepler가 제공하는 Sensor 데이터들입니다.
<a href="#">Estimation</a>	Kepler가 사용하는 센서들로부터 추정한 데이터들입니다.
<a href="#">Actuator</a>	Kepler에서 동작하는 Actuator 데이터들입니다.
<a href="#">Envrionment</a>	Kepler 상태에 대한 부가 정보입니다.

## ROS Subscriber Topic 분류

Kepler가 구독하는 topic들의 분류는 다음과 같습니다.

Type	Descriptions
<a href="#">Control</a>	로봇 수동 조작을 위해 구독하는 데이터들입니다.
<a href="#">Planning</a>	주어진 좌표 간 로컬 경로를 생성하기 위해 구독하는 데이터들입니다.

## ROS Topic 목록

Kepler가 게시 및 구독하는 topic들에 대한 정보입니다. rostopic list 명령어로 활성화된 topic 목록을 확인할 수 있습니다.

Topic	Message type	Descriptions
<a href="#">/ouster/scan_cloud</a>	<a href="#">sensor_msgs/PointCloud2</a>	Ouster Lidar의 scan 데이터에서 point들의 x,y,z 좌표를 게시합니다.
<a href="#">/realsense/depth_cloud</a>	<a href="#">sensor_msgs/PointCloud2</a>	RealSense Camera Depth Point들의 x,y,z 좌표를 게시합니다. 이 데이터는 480x270 해상도로 조절된 값입니다.
<a href="#">/imu</a>	<a href="#">sensor_msgs/Imu</a>	Imu 센서가 측정한 데이터를 게시합니다.
<a href="#">/sonar</a>	<a href="#">sensor_msgs/Range</a>	Sonar 센서가 측정한 거리(cm) 데이터를 게시합니다.
<a href="#">/odom</a>	<a href="#">nav_msgs/Odometry</a>	Lidar, Imu, Encoder로부터 추정된 odometry 데이터를 게시합니다.
<a href="#">/pose</a>	<a href="#">geometry_msgs/PoseStamped</a>	Lidar 데이터로부터 로봇의 현재 위치를 추정해 게시합니다.
<a href="#">/map</a>	<a href="#">sensor_msgs/PointCloud2</a>	현재 사용자가 선택한 지도 데이터를 불러와 게시합니다.
<a href="#">/local_path</a>	<a href="#">geometry_msgs/PoseArray</a>	local planning의 결과 경로를 게시합니다.
<a href="#">/command_status</a>	<a href="#">std_msgs/String</a>	로봇이 받은 명령 상태 정보를 게시합니다.
<a href="#">/cmd_vel</a>	<a href="#">geometry_msgs/Twist</a>	로봇 모터로부터 얻은 현재 cmd_velocity 데이터를 발행합니다.
<a href="#">/marker/feedback/position</a>	<a href="#">geometry_msgs/Point</a>	Marker 인식 결과 얻은 위치 정보를 x,y,z 좌표로 게시합니다.
<a href="#">/global_path</a>	<a href="#">nav_msgs/Path</a>	Local Planning을 수행하기 위한 전역 경로 좌표를 구독합니다.
<a href="#">/wheel/manual_ctrl</a>	<a href="#">geometry_msgs/Twist</a>	로봇 수동 조작 신호를 따르기 위해 cmd_velocity 데이터를 구독합니다.

Topic	Message type	Descriptions
<a href="#">/lift/manual_ctrl</a>	<a href="#">std_msgs/Float32</a>	수동 Lift 조작 신호를 따르기 위한 거리(cm) 데이터를 구독합니다.
<a href="#">/linear_guide/manual_ctrl</a>	<a href="#">std_msgs/Float32</a>	수동 Linear Guide 조작 신호를 따르기 위한 거리(cm) 데이터를 구독합니다.
<a href="#">/suction/manual_ctrl</a>	<a href="#">std_msgs/Float32</a>	수동 Suction 조작 신호를 따르기 위한 흡입 강도 데이터를 구독합니다.

## Sensor

Kepler가 사용 중인 Sensor 데이터들은 크게 Lidar, Realsense Camera, Sonar, Imu 등이 있습니다. 센서 종료는 Kepler 버전에 따라 달라질 수 있습니다.

### Lidar (/ouster/scan\_cloud)

*Ouster Lidar에서 얻은 scan 데이터의 x,y,z 좌표 정보들을 sensor\_msgs/PointCloud2 타입으로 게시합니다. Kepler 기준 좌표축으로 변환한 데이터입니다.*

#### Example

```
$ rostopic echo /ouster/scan_cloud

#print example
header:
  seq: 1616
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "map"
height: 1
width: 2214
fields:
  -
    name: "x"
    offset: 0
    datatype: 7
    count: 1
  -
    name: "y"
    offset: 4
    datatype: 7
    count: 1
  -
    name: "z"
    offset: 8
```

```

    datatype: 7
    count: 1
    is_bigendian: False
    point_step: 12
    row_step: 26568
    data: [16, 123, 129, ... 0, 342, 174]
    is_dense: False

```

## RealSense Camera (/realsense/depth\_cloud)

*Realsense Camera에서 얻은 point cloud 데이터의 x,y,z 좌표 정보들을 sensor\_msgs/PointCloud2 타입으로 게시합니다. Kepler 기준 좌표축으로 변환한 데이터입니다.*

### Example

```

$ rostopic echo /realsense/depth_cloud

#print example
header:
  seq: 1616
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "map"
height: 1
width: 5400
fields:
  -
    name: "x"
    offset: 0
    datatype: 7
    count: 1
  -
    name: "y"
    offset: 4
    datatype: 7
    count: 1
  -
    name: "z"
    offset: 8
    datatype: 7
    count: 1
is_bigendian: False
point_step: 12
row_step: 26568
data: [56, 223, 13, ... 0, 0, 0]
is_dense: False

```

## Imu (/imu)

*Imu에서 얻은 데이터를 sensor\_msgs/Imu 타입으로 게시합니다. Kepler 기준 좌표축으로 변환한 데이터입니다.*

#### Example

```
$ rostopic echo /imu

#print example
header:
  seq: 4243
  stamp:
    secs: 527
    nsecs: 116716464
  frame_id: "map"
orientation:
  x: 0.0
  y: 0.0
  z: 0.0
  w: 0.0
orientation_covariance: [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0]
angular_velocity:
  x: -0.000956347677857
  y: -0.000509549863636
  z: -0.00109129492193
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.00591037096456
  y: -0.231394946575
  z: 9.99276256561
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

#### Sonar (/sonar)

*Sonar에서 얻은 거리 데이터를 sensor\_msgs/Range 타입으로 게시합니다. 단위는 cm입니다.*

#### Example

```
blarblar...
```

---

## Estimation

---

Kepler는 Sensor 데이터들로부터 추정된 유용한 데이터들을 제공합니다. 각 정보는 Odometry, Current Position 등입니다.

## Odometry (/odom)

Kepler가 사용하는 각종 센서들로부터 추정한 로봇의 odometry 정보입니다. *nav\_msgs/Odometry* 타입으로 게시합니다.

### Example

```
$ rostopic echo /odom

#print example
header:
  seq: 99
  stamp:
    secs: 0
    nsecs: 0
  frame_id: 'map'
child_frame_id: 'map'
pose:
  pose:
    position:
      x: 0.0
      y: 0.0
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: 0.0
      w: 0.0
  covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
twist:
  twist:
    linear:
      x: 0.0
      y: 0.0
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: 0.0
  covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

## Current Position (/pose)

전역 지도에서 *Kepler*가 추정한 로봇의 현재 위치 정보입니다. *geometry\_msgs/PoseStamped* 타입으로 게시합니다.

#### Example

```
$ rostopic echo /pose

#print example
header:
  seq: 41
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "map"
pose:
  position:
    x: 0.0
    y: 0.0
    z: 0.0
  orientation:
    x: -0.0121852047741
    y: -0.00152712513227
    z: -0.00302079459652
    w: 0.999920010567
```

#### map (/map)

사용자가 선택한 전역 지도 정보입니다. *sensor\_msgs/PointCloud2* 타입으로 게시합니다.

#### Example

```
$ rostopic echo /map

#print example
header:
  seq: 1616
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "map"
height: 1
width: 2214
fields:
  -
    name: "x"
    offset: 0
    datatype: 7
    count: 1
  -
```

```

    name: "y"
    offset: 4
    datatype: 7
    count: 1
  -
    name: "z"
    offset: 8
    datatype: 7
    count: 1
  is_bigendian: False
  point_step: 12
  row_step: 26568
  data: [456, 344, 229, ... 323, 222, 113]
  is_dense: False

```

## Local path (/local\_path)

*Kepler가 현재 지도를 바탕으로 path planning해 구한 local planning의 결과를 경로로 나타낸 정보입니다. geometry\_msgs/PoseArray 타입으로 게시합니다.*

### Example

```

$ rostopic echo /local_path

#print example
header:
  seq: 194
  stamp:
    secs: 0
    nsecs: 0
  frame_id: ''
poses:
  -
    position:
      x: -0.0619981922209
      y: -0.211999684572
      z: 0.0
    orientation:
      x: -37.4119987488
      y: 5.63800048828
      z: -0.687827169895
      w: 0.72587454319
  -
  ...
  -
    position:
      x: -0.0619981922209
      y: -0.211999684572
      z: 0.0
    orientation:
      x: -37.4119987488

```



```
y: 5.63800048828
z: 0.930579006672
w: -0.366091132164
```

---

## Actuator

---

Kepler Actuator들은 `cmd_velocity`, Marker 인식 feedback 정보들을 제공합니다.

### Command Velocity (/cmd\_vel)

로봇 모터로부터 얻은 선속도와 각속도 정보입니다. *geometry\_msgs/Twist* 타입으로 게시합니다.

#### Example

```
$ rostopic echo /cmd_vel

#print example
linear:
  x: 0.300000011921
  y: -0.089333333075
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -0.0
```

### Marker Feedback (/marker/feedback/position)

Marker 인식 결과 얻은 feedback 데이터들 중 marker position에 대한 정보입니다. *geometry\_msgs/Point* 타입으로 게시합니다.

#### Example

```
$ rostopic echo /marker/feedback/position

#print example
x: 0.2
y: 2.2
z: 0.0
```

---

## Environment

---

Kepler의 명령 입력 상태 등을 제공합니다.

### Command Status (/command\_status)

각종 명령 입력 상태에 대한 정보를 게시합니다.

#### Example

```
data: "Status: AUTO_DRIVING"
```

---

## Control

---

사용자가 수동 조작으로 로봇 및 각종 Acuator를 제어할 수 있도록 관련 데이터를 구독합니다.

### Manual Wheel Control (/wheel/manual\_ctrl)

로봇 수동 조작을 위한 선속도, 각속도 값을 구독합니다. 메세지 타입은 *geometry\_msgs/Twist*입니다.

#### Example

```
blarblar...
```

### Manual Lift Control (/lift/manual\_ctrl)

Lift 수동 조작을 위한 이동 거리 값을 구독합니다. 단위는 *cm*이며, 메세지 타입은 *std\_msgs/Float32*입니다.

#### Example

```
blarblar...
```

### Manual Linear Guide Control (/linear\_guide/manual\_ctrl)

*Linear Guide* 수동 조작을 위한 이동 거리 값을 구독합니다. 단위는 *cm*이며, 메세지 타입은 *std\_msgs/Float32*입니다.

#### Example

```
blarblar...
```

### Manual Suction Control (/suction/manual\_ctrl)

*Suction* 수동 조작을 위한 흡입 강도 값을 구독합니다. 메세지 타입은 *std\_msgs/Float32*입니다.

#### Example

```
blarblar...
```

## Planning

사용자가 수동 조작으로 로봇 및 각종 Acuator를 제어할 수 있도록 관련 데이터를 구독합니다.

### Global Path (/global\_path)

로봇 전역 경로 데이터를 구독합니다. 구독한 전역 경로로부터 자율 주행 경로를 로컬 경로로 생성합니다. 데이터 타입은 *nav\_msgs/Path*입니다.

#### Example

```
$ rostopic pub /global_path nav_msgs/Path '{header: {stamp: now, frame_id:
"map"},
  poses: [header: {stamp: now, frame_id: "map"},
           pose : {position: {x: 0.3, y: 0.1, z: 0}, orientation : {x: 0.1,
y: 0, z: 0, w: 0.2}}
         ,
           header: {stamp: now, frame_id: "map"},
           pose : {position: {x: 0.3, y: 0.1, z: 0}, orientation : {x: 0.1,
y: 12, z: 0, w: 0.2}}
         ]
}'
```