

ROS launch



Intro to roslaunch

- Roslaunch
- roslaunch는 여러 개의 node를 실행시킬 수 있다.
(roslaunch를 여러 번 사용할 필요가 없다)
- roslaunch 실행에는 ros master가 필요 없으며, 없을 경우 자동으로 생성한다.
- roslaunch는 같은 네트워크 상에 있는 다른 PC에 있는 것을 실행하는 것도 가능하다.
- Parameter 서버에 Parameter를 등록함과 동시에 실행하는 것이 쉽다.
- roslaunch는 다른 roslaunch 파일을 재사용해서 생성할 수 있다.
- `$ roslaunch package_name file.launch` 명령어를 통해 사용할 수 있다.
- roslaunch 파일은 XML 문법을 따른다.

XML Syntax

- XML은 Extensible Markup Language의 줄임말이다.
- 문법은 HTML과 유사하며, 원하는 태그를 골라서 사용할 수 있다.
- <Hello>와 같은 형태로 여는 태그를 사용하며,
</Hello>와 같은 형태로 닫는 태그를 사용할 수 있다.
- 특정 데이터 전달 시에, 데이터 전달의 표준으로 사용하기 위해 개발되었다.

<?xml version="1.0" encoding="euc-kr" ?>

<수업 과목>

<수학>

<학생 명단>

<이름>Rachael</이름>

<학년>3</학년>

<나이>15</나이>

</학생 명단>

</수학>

</수업 과목>

How to use roslaunch

- roslaunch는 여러 node를 한 번에 실행하며, “거의 동시”에 실행이 된다.
- 거의 동시에 실행이므로, 실행 순서가 필요한 경우는 다른 방법이 필요
- 사용 방법은 XML과 동일하며, 내부의 규칙에 맞게 작성하면 된다.

- 가장 기본은 <launch> 태그를 이용하여, launch 파일임을 선언하는 것으로 부터 시작

```
<launch>
```

```
  <node ... />
```

```
</launch>
```

- 위와 같은 형태로 작성하여, launch태그 사이에 들어간 내용이 실행될 수 있도록 설정한다.

- launch파일을 실행하는 terminal에 적용되어 있는 terminal 변수를 사용하고 싶을 때는, \$(env variables) 형태로 사용할 수 있다.
- terminal 변수는 \$ export variables=50 를 입력하여 생성할 수 있다
- 확인의 경우 echo \$variables 를 통해 값 확인 가능하다.

```
wego/~/ export variables=50  
wego/~/ echo $variables  
50
```

```
<param name="foo" value="$(env HELLO)" />  
<param name="bar" value="$(optenv WORLD 50)" />  
<param name="fooo" value="$(optenv JESUS)" />
```

- `$(env 터미널변수)` 방식의 경우, 해당 터미널 변수가 없을 경우, 아예 실행이 되지 않는다.
- `$(optenv 터미널변수 default값)` 형태로 사용하면, 터미널변수가 있을 경우, 그 값이 사용되며, 없을 경우 default 값이 들어간다. 만약 default값이 없을 경우, 비어 있는 문자열이 들어가게 된다.

```
wego/~/ export variables=50  
wego/~/ echo $variables  
50
```

```
<param name="foo" value="$(env HELLO)" />  
<param name="bar" value="$(optenv WORLD 50)" />  
<param name="fooo" value="$(optenv JESUS)" />
```

- `$(find package_name)`의 형태를 사용할 시, 해당 Package의 경로를 반환해준다.
- ROS에서는 동일한 이름의 Node를 다수 생성하는 것이 불가능하며, 이 경우 먼저 생성되어 있던 Node가 죽게 되는 현상이 발생한다.
- 이 경우, name부분에 `$(anon name)` 형태를 사용하면, anonymous id를 사용할 수 있으며, 다수 실행이 가능해진다.

```
<arg name="this" default="two" />
```

```
<arg name="this" value="one" />
```

- 위와 같은 형태로, launch파일 내에서 사용할 수 있는 argument를 만들 수 있으며, default의 경우, 뒷부분 또는 launch파일 실행 시 변경이 가능하며, value로 되어있을 경우, 변경이 불가능하다.
- \$(arg this) 와 같은 형태로 호출해서 사용할 수 있다.

- 모든 사용 가능한 태그에는 if와 unless에 해당하는 태그가 있으며, if=value 형태로 사용하며, "1" 또는 "True"일 경우, 동작하고 "0" 또는 "False"일 경우 동작하지 않게 한다.
- unless의 경우 if와 반대로 동작한다.

```
<param if="True" name="hello" value="world" />
```

- 위와 같은 형태로 사용할 수 있다.

- <node> 태그를 사용하여, 가장 기초적인 node를 실행할 수 있다.
- 만약 `roslaunch hello_pkg hello` 라는 이름의 node를 실행하는 것을 launch파일로 작성한다면

<launch>

```
<node name='hello_world' pkg="hello_pkg" type="hello" />
```

</launch>

- name의 경우 `roslaunch list`에서 확인할 수 있는 이름을 지정해주면 된다.
- pkg의 경우 실제 실행하려는 node가 있는 패키지의 이름을 넘겨주면 된다.
- type의 경우 실제 실행하려는 node의 이름을 넘겨준다. (cpp일 경우 확장자가 없으며, 파이썬인 경우 .py 확장자를 가진다.)

- `<include>` 태그를 활용하면, 다른 XML파일 또는 launch파일을 포함시킬 수 있다.
- `file="$(find pkg)/path/filename.xml"`과 같은 속성을 추가하면, 다른 xml파일을 추가할 수 있다.
- `ns="foo"`와 같은 속성을 추가하여, namespace를 추가할 수 있다.
- `clear_parms="true|false"`의 속성을 추가하여, 실행 시, 해당 private parameter를 지울 수 있다.
- `pass_all_args="true|false"`의 속성을 추가하여, 현재 launch파일에서 생성한 argument를 포함된 자식 launch파일에도 전달해서 실행할 수 있다.

- `<remap>` 태그를 이용하여, node 실행 시 생성되는 Topic의 이름을 변경할 수 있다.
- `<remap from="original_name" to="new_name" />`형태로 사용하며,
해당 node 내부에 있는 `original_name`으로 된 topic을
`new_name`으로 변경하여 실행한다.
- Subscribe 또는 Publish하는 Topic의 이름을 변경할 수 있다.

- `<param>` 태그를 사용하여, 실행할 때, parameter server에 등록할 parameter를 정할 수 있다.

```
<param name="today" value="20210419" type="int"/>
```

- `type`은 타입이 확실할 경우에 명시하며, 애매할 경우 생략 가능하다.
- `name`으로 parameter server에 등록이 되며, `value` 값으로 등록이 된다.
- launch파일 실행 후, `rosparam list`,
`rosparam get /today` 와 같은 명령어로 확인이 가능하다.
- parameter를 등록 시에, `value` 부분에서
`$(eval 2.* 3.1415 * arg('radius'))` 와 같은 형태로 계산된 결과를 입력하는 것도 가능하다.

- <rosparam> 태그를 이용하여, yaml파일을 불러오거나, 저장할 수 있다.

```
<rosparam command="load" file="$(find rosparam)/example.yaml" />
```

- command 속성은 load, dump, delete 세 가지가 있으며, 불러오기, 저장하기, 지우기 세가지이다.
- file은 불러오기 또는 저장하기를 할 때, 경로를 지정해주면 된다.



본사(기술연구소 및 사무실) : 16914 경기도 용인시 기흥구 구성로 357(청덕동) 용인테크노밸리 B동 513호

기술연구소(서울) : 04799 서울특별시 성동구 성수동2가 280-13, 삼환디지털벤처타워 401호

대표전화 : 031 – 229 – 3553

팩스 : 031 – 229 – 3554

제품문의: go.sales@wego-robotics.com

기술문의: go.support@wego-robotics.com