

# Universal Robots ROS Driver

ROS Melodic

김은수

# 참고

- <https://www.universal-robots.com/ko/>
- [https://github.com/UniversalRobots/Universal\\_Robots\\_ROS\\_Driver](https://github.com/UniversalRobots/Universal_Robots_ROS_Driver)
- [https://github.com/UniversalRobots/Universal\\_Robots\\_ROS\\_Driver/blob/master/ur\\_robot\\_driver/resources/externalcontrol-1.0.2.urcap](https://github.com/UniversalRobots/Universal_Robots_ROS_Driver/blob/master/ur_robot_driver/resources/externalcontrol-1.0.2.urcap)
- [https://github.com/UniversalRobots/Universal\\_Robots\\_ROS\\_Driver/blob/master/ur\\_robot\\_driver/doc/install\\_urcap\\_cb3.md](https://github.com/UniversalRobots/Universal_Robots_ROS_Driver/blob/master/ur_robot_driver/doc/install_urcap_cb3.md)



**UNIVERSAL ROBOTS**

**::: ROS.org**

# 목차

---

- **Development Environment**
- **Universal Robots ROS Driver**
  - Feature
  - Contents
  - Install
- **ur\_robot\_driver에 대한 UR 로봇 설정**
- **UR3 Control Using the ROS Driver**

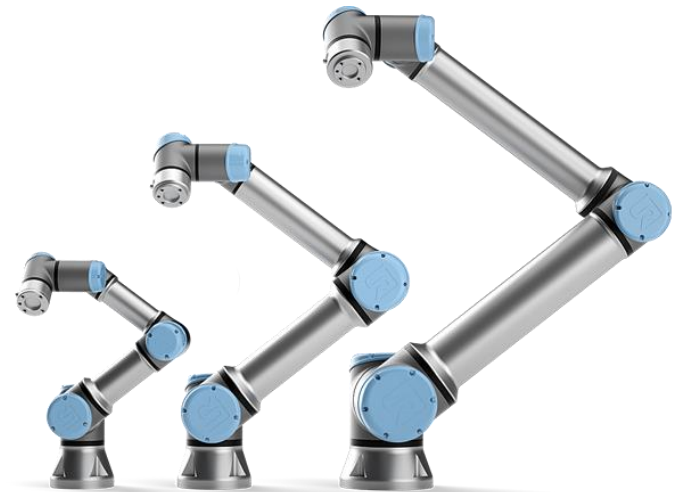
# Development Environment

- OS
  - Ubuntu 18.04
- ROS
  - ROS Melodic
- UR3
  - CB3 소프트웨어 버전  $\geq 3.7$



# Universal Robots ROS Driver

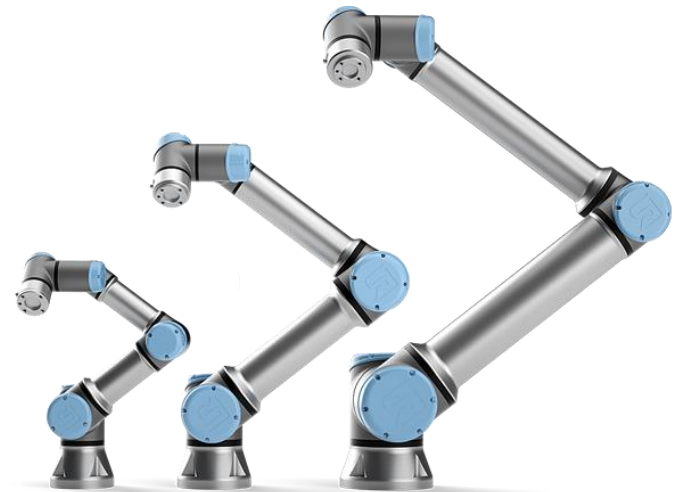
- Universal Robots ROS Driver
  - 모든 사용자들에게 UR 로봇과 ROS 간에 안정적이고 지속 가능한 인터페이스를 제공



# Universal Robots ROS Driver

## ■ 특징

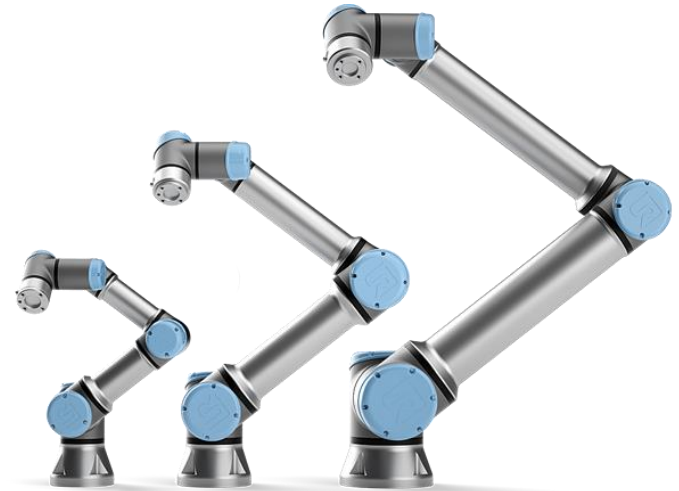
- 모든 CB3 (소프트웨어 버전  $\geq 3.7$ ) 및 e-Series (소프트웨어  $\geq 5.1$ ) 로봇에서 작동
- URCaps 시스템을 사용하면 ROS 측에서 전송된 제어 명령을 처리하는 프로그램이 로봇에서 실행
- 티치 펜던트(TF)에서보다 복잡한 UR 프로그램의 일부로 ROS 구성 요소를 사용할 수 있음



# Universal Robots ROS Driver

## ■ 패키지 내용

- `controller_stopper` : 로봇의 상태에 따라 컨트롤러를 중지했다가 다시 시작하는 작은 외부 도구. 로봇이 ROS에서 보낸 명령을 받아들이지 않는 상태일 때 유용
- `ur_calibration` : 로봇의 교정 정보를 추출 및 변환
- `ur_controllers` : 속도 확장 인식 컨트롤러와 함께 이 드라이버에 도입된 컨트롤러
- `ur_robot_driver` : 실제 드라이버 패키지



# Universal Robots ROS Driver

## ■ 설치

```
# source global ros
$ source /opt/ros/melodic/setup.bash

# create a catkin workspace
$ mkdir -p catkin_ws/src && cd catkin_ws

# clone the driver
$ git clone https://github.com/UniversalRobots/Universal_Robots_ROS_Driver src/Universal_Robots_ROS_Driver

# clone fork of the description. This is currently necessary, until the changes are merged upstream.
$ git clone -b calibration_devel https://github.com/fmauch/universal_robot.git src/fmauch_universal_robot

# install dependencies
$ sudo apt update -qq
$ rosdep update
$ rosdep install --from-paths src --ignore-src -y

# build the workspace
$ catkin_make

# activate the workspace (ie: source it)
$ source devel/setup.bash
```

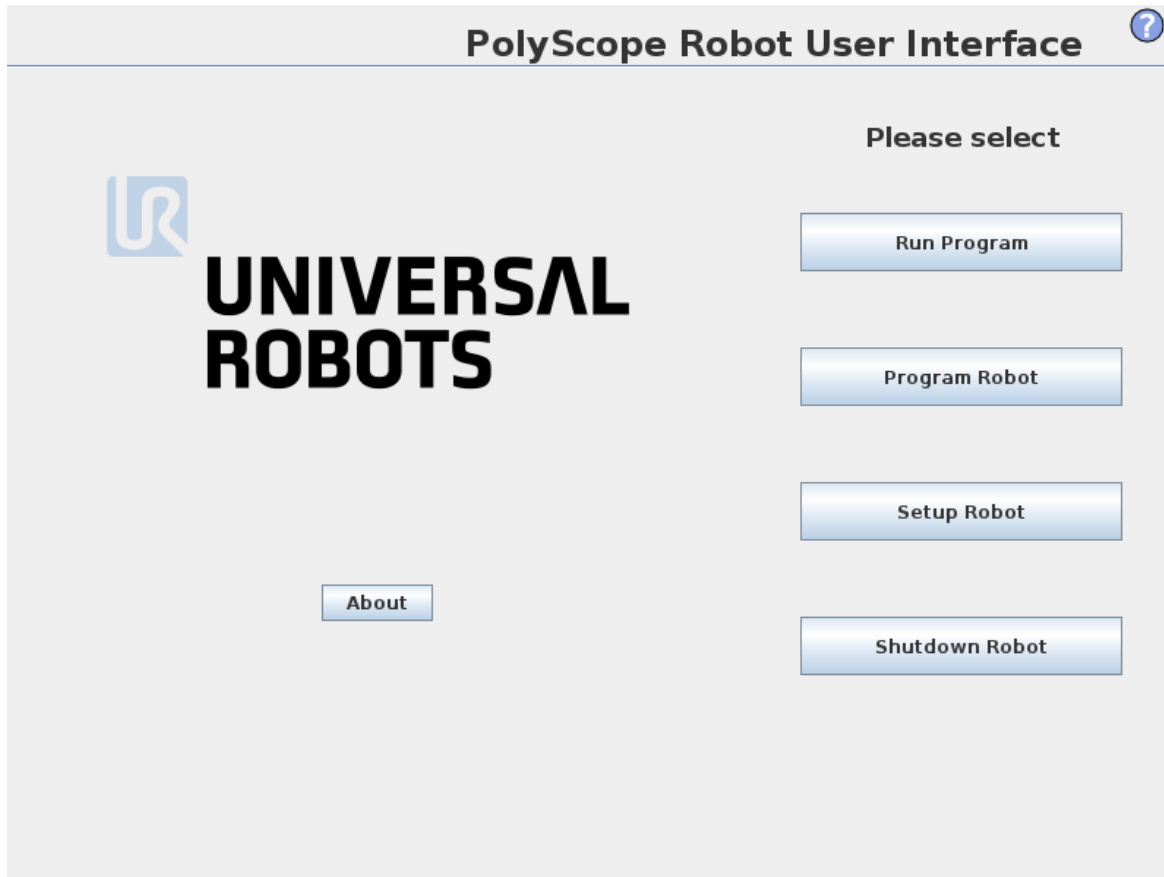


# Universal Robots ROS Driver

- ur\_robot\_driver에 대한 UR 로봇 설정
  - ur\_robot\_driver를 사용하려면 아래 externalcontrol-1.0.2.urcap 설치
  - [https://github.com/UniversalRobots/Universal\\_Robots\\_ROS\\_Driver/blob/master/ur\\_robot\\_driver/resources/externalcontrol-1.0.2.urcap](https://github.com/UniversalRobots/Universal_Robots_ROS_Driver/blob/master/ur_robot_driver/resources/externalcontrol-1.0.2.urcap)
  - USB에 urcap 파일을 담고, TP에 연결

# Universal Robots ROS Driver

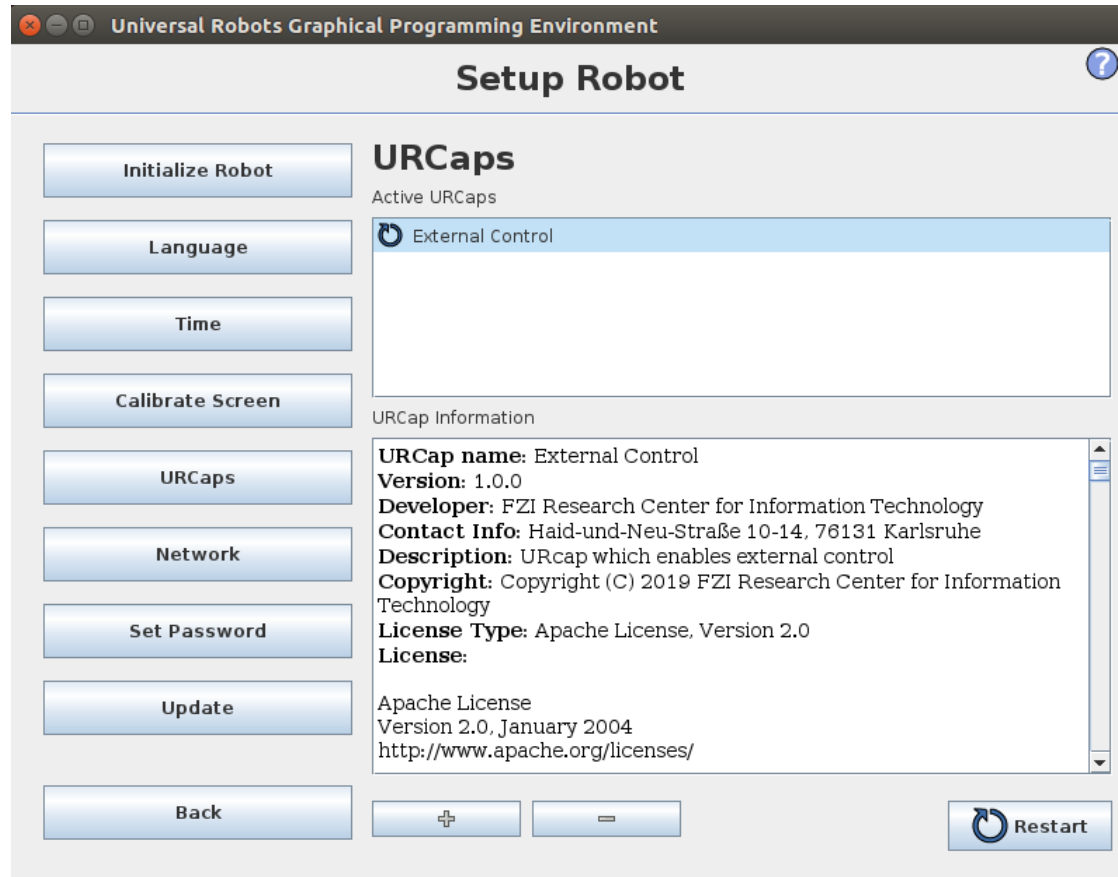
- ur\_robot\_driver에 대한 UR 로봇 설정



- 시작 화면 - 로봇 설정 - URCaps

# Universal Robots ROS Driver

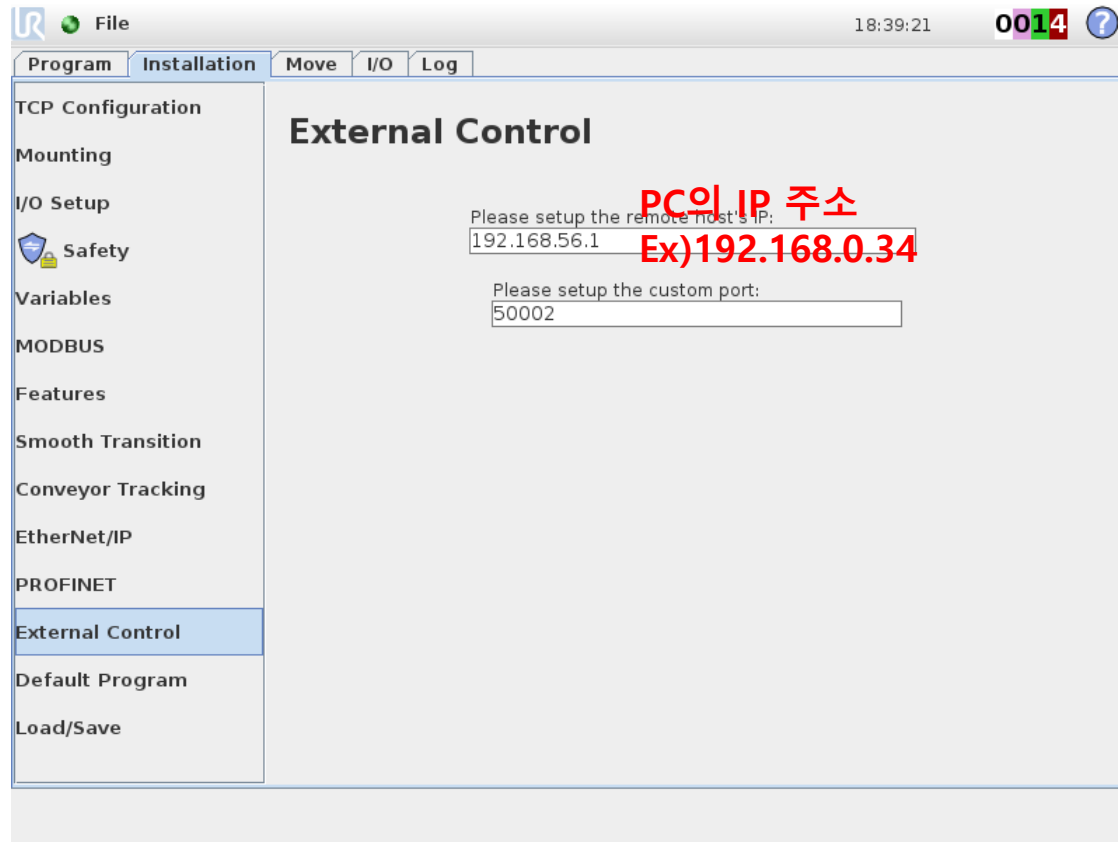
## ■ ur\_robot\_driver에 대한 UR 로봇 설정



- 하단의 + 부호 클릭  
=> 연결된 USB의 urcap 파일 표시됨
- 파일을 선택하고 열기, Restart

# Universal Robots ROS Driver

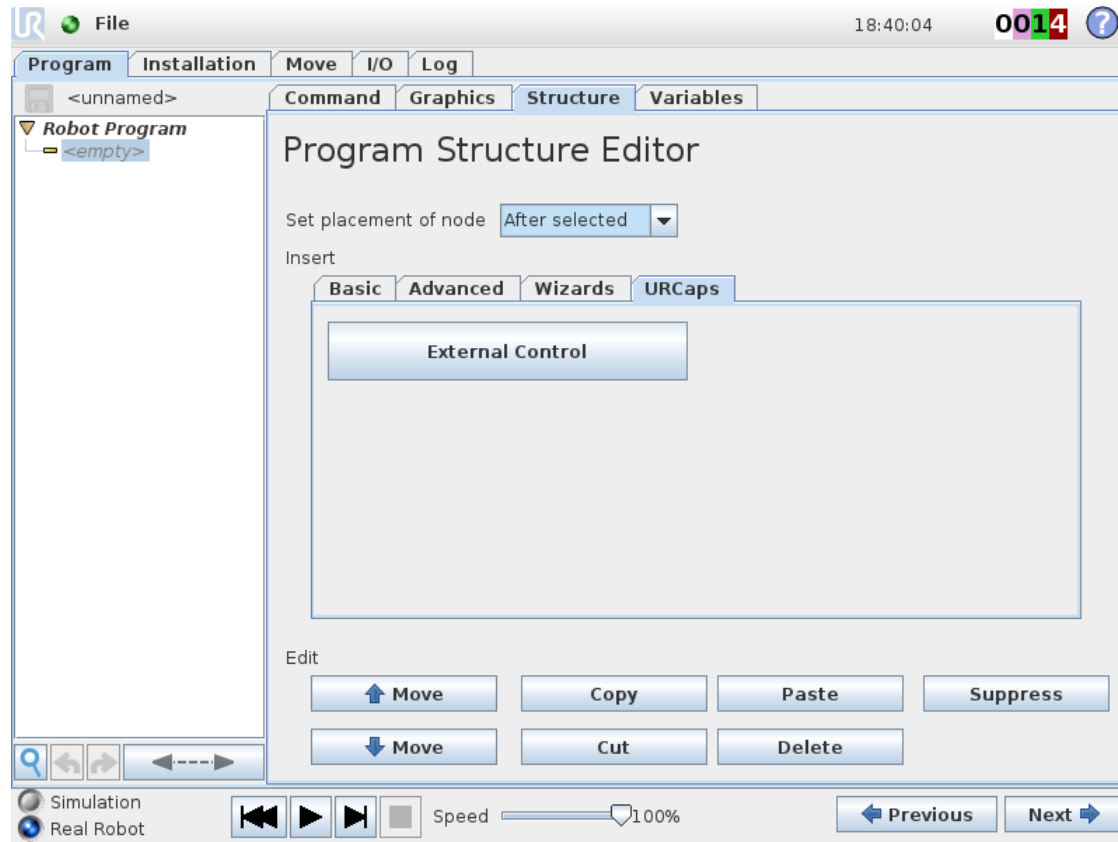
## ■ ur\_robot\_driver에 대한 UR 로봇 설정



- 재부팅 후 설치 섹션에서 External Control 클릭
- ROS 드라이버를 실행할 외부 PC의 IP 주소를 설정
- custom port는 그대로 유지

# Universal Robots ROS Driver

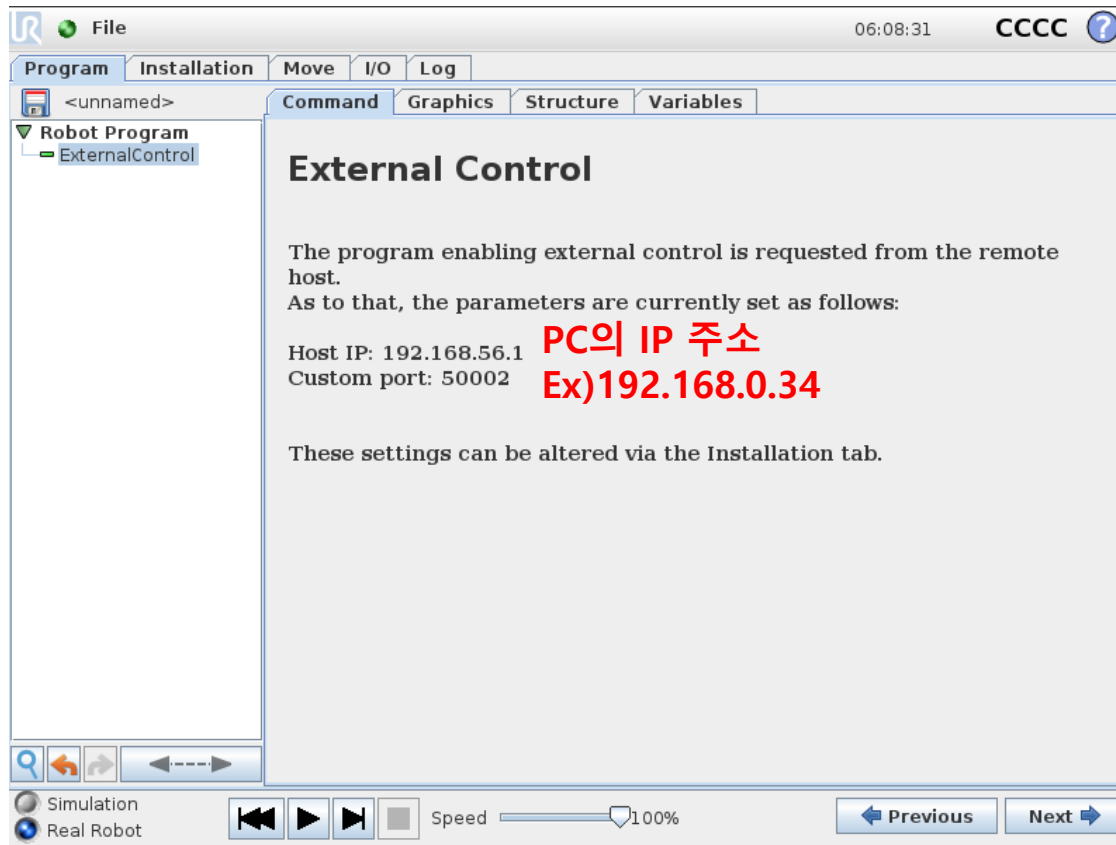
- ur\_robot\_driver에 대한 UR 로봇 설정



- 새 URCap을 사용하려면 새 프로그램 작성하고 구조-URCaps 들어가서 External Control 노드를 프로그램 트리에 삽입

# Universal Robots ROS Driver

## ■ ur\_robot\_driver에 대한 UR 로봇 설정



- 명령 탭을 다시 클릭하면 설치 내에 입력한 설정이 표시됨
- 확인한 다음 프로그램 저장

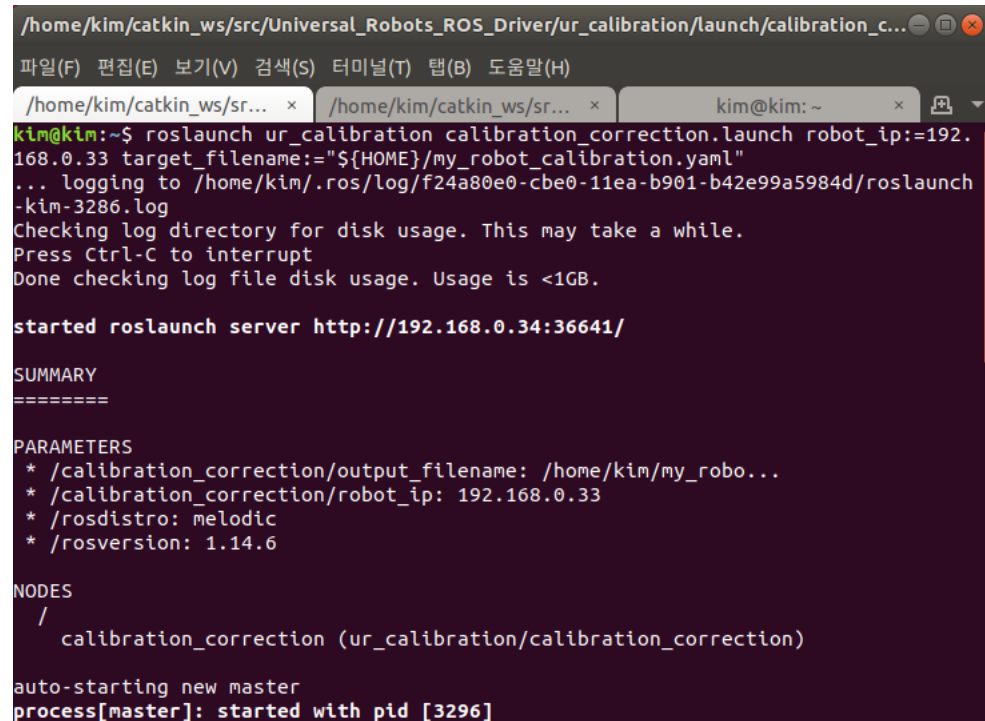
# Universal Robots ROS Driver

## ■ 교정 정보 추출

```
$ roslaunch ur_calibration calibration_correction.launch \
  robot_ip:= <robot_ip> target_filename:="${HOME}/my_robot_calibration.yaml"
```

UR3의 IP 주소

Ex) 192.168.0.33



```
/home/kim/catkin_ws/src/Universal_Robots_ROS_Driver/ur_calibration/launch/calibration_c...
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 탭(B) 도움말(H)
/home/kim/catkin_ws/sr... x /home/kim/catkin_ws/sr... x kim@kim: ~ x
kim@kim:~$ roslaunch ur_calibration calibration_correction.launch robot_ip:=192.168.0.33 target_filename:="${HOME}/my_robot_calibration.yaml"
... logging to /home/kim/.ros/log/f24a80e0-cbe0-11ea-b901-b42e99a5984d/roslaunch-kim-3286.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.0.34:36641/

SUMMARY
=====

PARAMETERS
* /calibration_correction/output_filename: /home/kim/my_robo...
* /calibration_correction/robot_ip: 192.168.0.33
* /rostdistro: melodic
* /rosversion: 1.14.6

NODES
/
  calibration_correction (ur_calibration/calibration_correction)

auto-starting new master
process[master]: started with pid [3296]
```

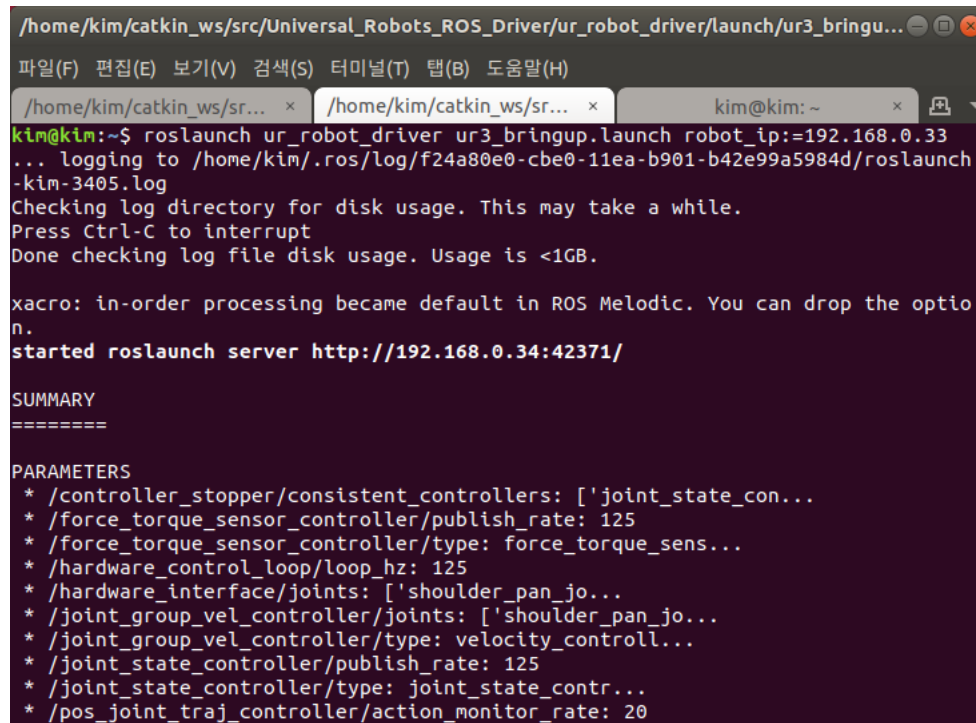
[PC]

# Universal Robots ROS Driver

## ■ 빠른 시작

```
$ roslaunch ur_robot_driver <robot_type>_bringup.launch robot_ip:=192.168.0.33
```

ur3



```
/home/kim/catkin_ws/src/Universal_Robots_ROS_Driver/ur_robot_driver/launch/ur3_bringu...
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 탭(B) 도움말(H)
/home/kim/catkin_ws/sr... x /home/kim/catkin_ws/sr... x kim@kim: ~ x
kim@kim:~$ roslaunch ur_robot_driver ur3_bringup.launch robot_ip:=192.168.0.33
... logging to /home/kim/.ros/log/f24a80e0-cbe0-11ea-b901-b42e99a5984d/roslaunch
-kim-3405.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

xacro: in-order processing became default in ROS Melodic. You can drop the optio
n.
started roslaunch server http://192.168.0.34:42371/

SUMMARY
=====

PARAMETERS
* /controller_stopper/consistent_controllers: ['joint_state_con...
* /force_torque_sensor_controller/publish_rate: 125
* /force_torque_sensor_controller/type: force_torque_sens...
* /hardware_control_loop/loop_hz: 125
* /hardware_interface/joints: ['shoulder_pan_jo...
* /joint_group_vel_controller/joints: ['shoulder_pan_jo...
* /joint_group_vel_controller/type: velocity_controll...
* /joint_state_controller/publish_rate: 125
* /joint_state_controller/type: joint_state_contr...
* /pos_joint_traj_controller/action_monitor_rate: 20
```

[PC]



# Universal Robots ROS Driver

- TP에서 ExternalControl 삽입된 프로그램 실행

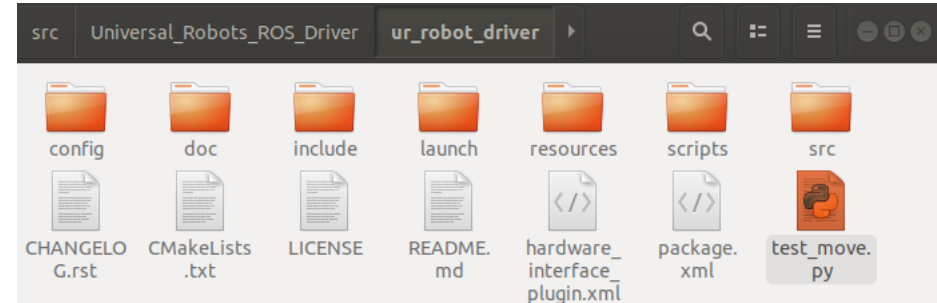
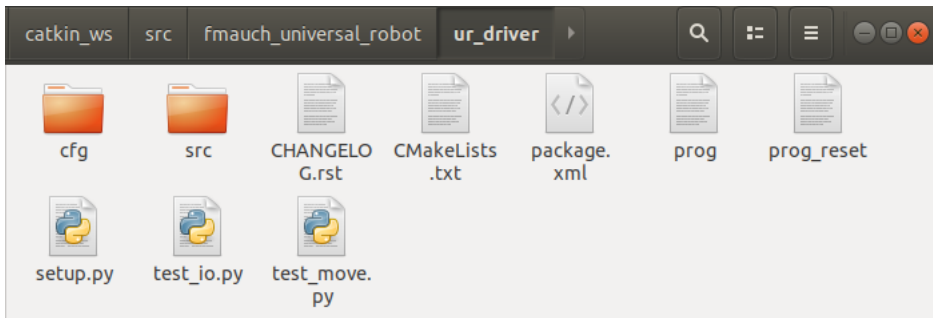


- 실행 후, 터미널에 위와 같이 뜨면 성공

```
[ INFO] [1595399285.009571277]: Robot requested program  
[ INFO] [1595399285.009740909]: Sent program to robot  
[ INFO] [1595399285.107951536]: Robot ready to receive control commands.
```

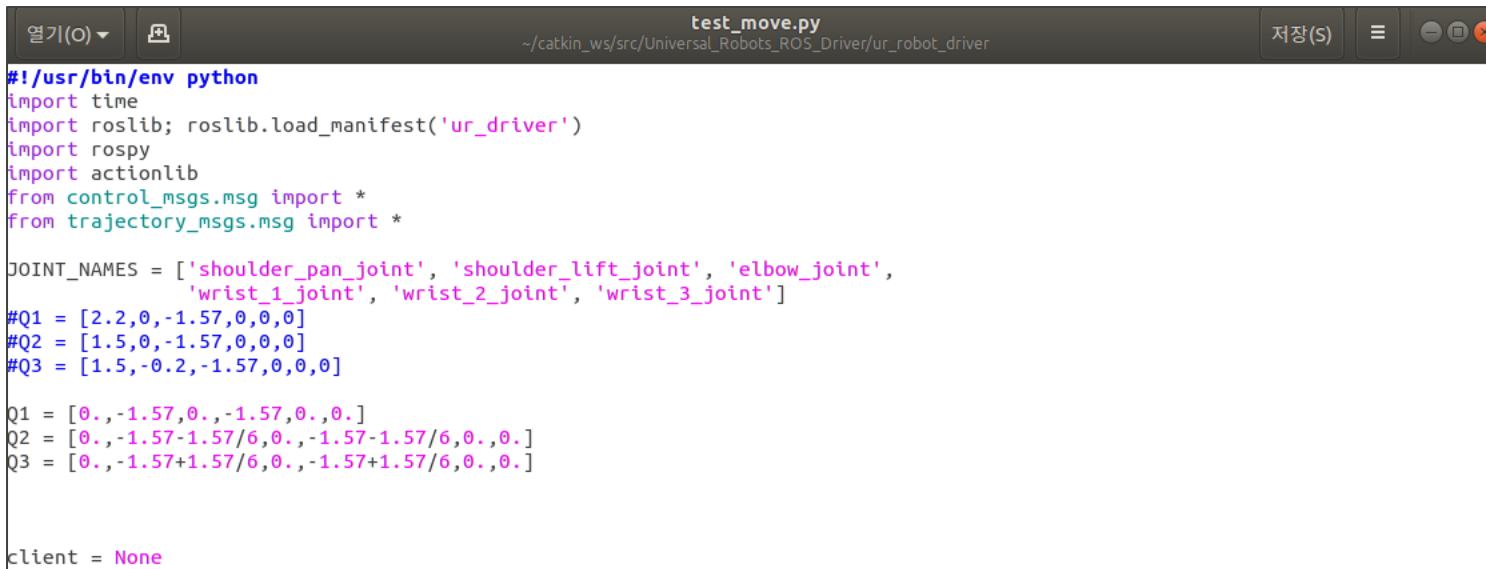
# Universal Robots ROS Driver

- fmauch\_universal\_robot/ur\_driver/test\_move.py를  
Universal\_Robots\_ROS\_Driver/ur\_robot\_driver로 이동



# Universal Robots ROS Driver

- test\_move.py 텍스트편집기 열기
- 안전을 위해 Q1, Q2, Q3 값 수정



```
#!/usr/bin/env python
import time
import roslib; roslib.load_manifest('ur_driver')
import rospy
import actionlib
from control_msgs.msg import *
from trajectory_msgs.msg import *

JOINT_NAMES = ['shoulder_pan_joint', 'shoulder_lift_joint', 'elbow_joint',
               'wrist_1_joint', 'wrist_2_joint', 'wrist_3_joint']
#Q1 = [2.2,0,-1.57,0,0,0]
#Q2 = [1.5,0,-1.57,0,0,0]
#Q3 = [1.5,-0.2,-1.57,0,0,0]

Q1 = [0.,-1.57,0.,-1.57,0.,0.]
Q2 = [0.,-1.57-1.57/6,0.,-1.57-1.57/6,0.,0.]
Q3 = [0.,-1.57+1.57/6,0.,-1.57+1.57/6,0.,0.]

client = None
```

- /follow\_joint\_trajectory를 /scaled\_pos\_joint\_traj\_controller/follow\_joint\_trajectory로 수정

```
try:
    rospy.init_node("test_move", anonymous=True, disable_signals=True)
    client = actionlib.SimpleActionClient('/scaled_pos_joint_traj_controller/follow_joint_trajectory',
```

# Universal Robots ROS Driver

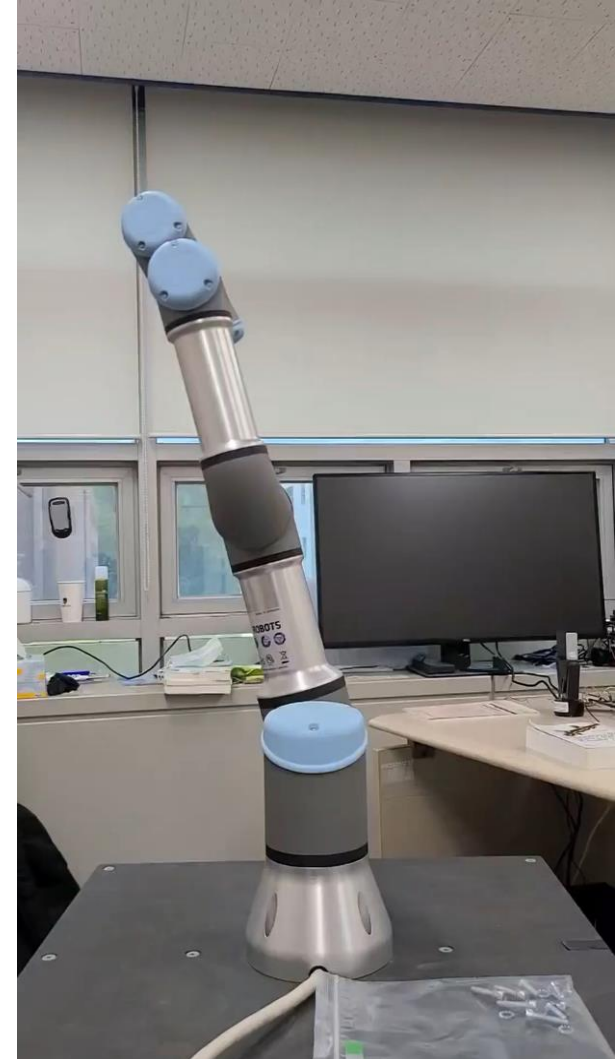
- test\_move.py 저장 후 실행

```
$ rosrun ur_robot_driver test_move.py
```

```
kim@kim:~$ rosrun ur_robot_driver test_move.py  
Waiting for server...  
Connected to server  
█
```

- ROS Driver를 통하여 ROS에서 UR3 Control 성공!!

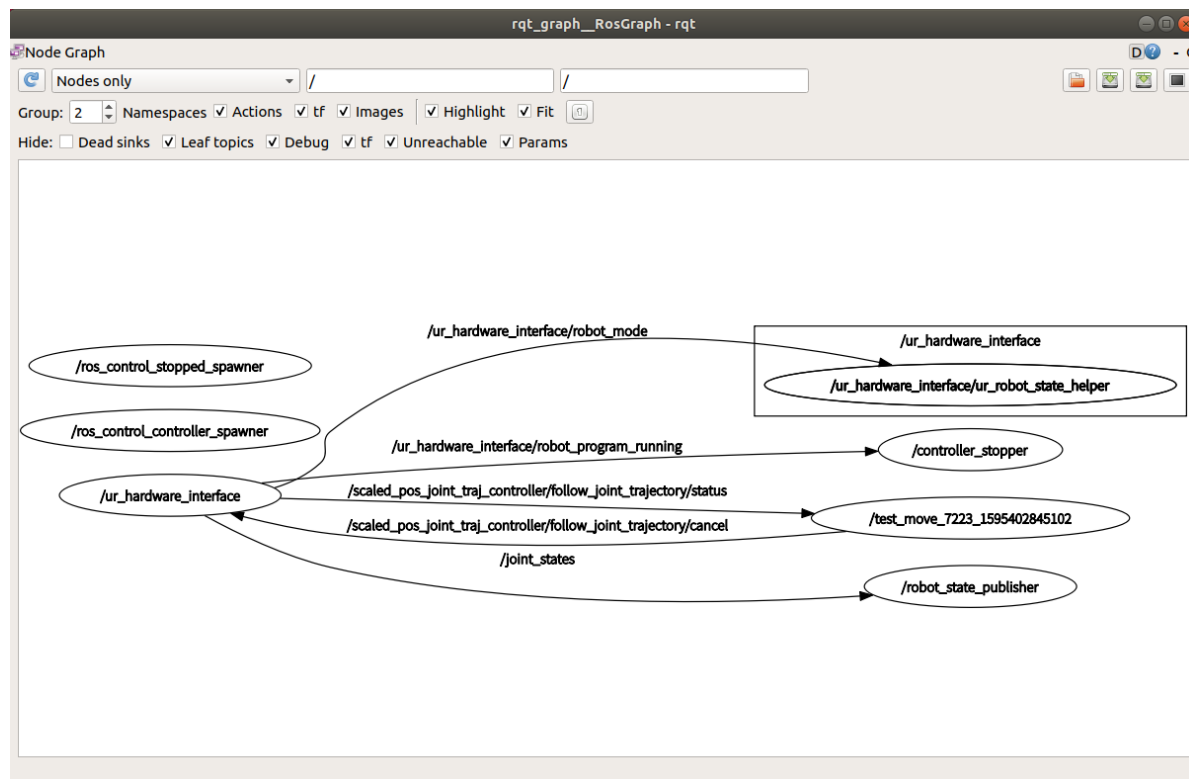
[PC]



# Universal Robots ROS Driver

- rqt\_graph

```
$ rqt_graph
```



[PC]

- THANK YOU