

[파이참(PyCharm) 설치]

google – [pycharm](#)(Community-기타버전) - 2020.3 월 버전 - 2020.3.5 – Linux(tar.gz)

압축풀기

[터미널] (ctrl+alt+T) // 폴더에서 마우스 오른쪽버튼 - 터미널에서 열기

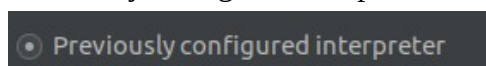
```
cd ~/다운로드/pycharm-community-2020.3.5/bin  
./pycharm.sh
```

NewProject 클릭

(파이썬 의존성때문에 오류나는 경우가 있음..)

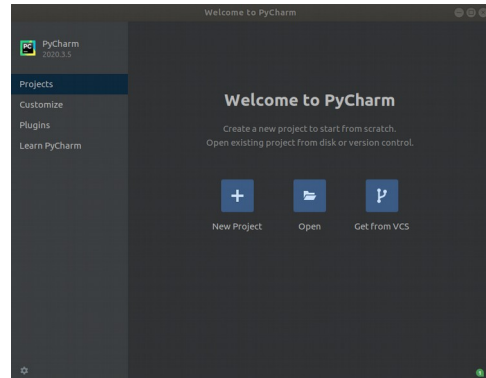
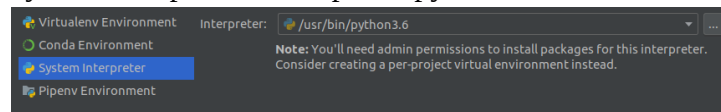
(로컬에 있는

Previously configured interpreter 클릭



Interpreter 에서 ... 누르고

System Interpreter - Interpreter python3.6



(File – Settings – Project: pythonProject – Python Interpreter 에서 설정해도 됨)

메뉴바에서 Tools – Create Command line Launcher - “charm” 입력

// 터미널에서 실행명령어?

Tools – Create Desktop Entry – 체크표시

// 아이콘 생성

Run 해서 정상작동하는지 확인

File – Settings – Project~ - Python Interpreter - 톱니바퀴 – Show All - 아래 폴더모양 클릭



아래 + 모양 클릭 - “/opt/ros/dashing/lib/python3.6/site-packages” - OK 버튼 클릭

아래 + 모양 클릭 - 위에 집모양 클릭 🏠

“-/home/phil/colcon_ws/install/open_manipulator_msgs/lib/python3.6/site-packages” - OK 클릭

[터미널]

```
cs # cd colcon_ws/src
```

```
ros2 pkg create pick_and_place --build-type ament_python --dependencies rclpy std_msgs geometry_msgs sensor_msgs open_manipulator_msgs
```

```
# 한줄로 붙여서 입력 # 패키지생성
```

--dependencies : 의존성 정보들 ← 매니플레이터를 돌려서 값들을 확인해보면 됨,,?

집어서 들어올려서 목적지로 이동?

1. Initial → Home position
2. Pick 위치로 이동
3. Gripper 열기 - 닫기
4. Place 위치로 이동
5. Gripper Open(열기)
6. 1.으로 돌아감

<필요한 것> : pick/place 위치정보(Joint 위치, 3 차원 위치-Gripper 기준), gripper 제어명령어, Manipulator 이동명령어(topic,service)

[터미널]

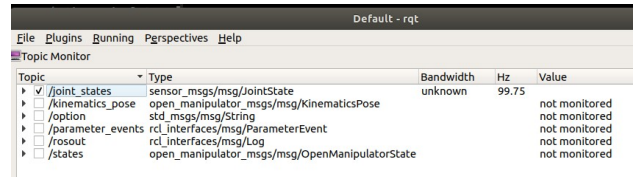
```
ros2 launch open_manipulator_x_controller open_manipulator_x_controller.launch.py
```

[새 터미널]

rqt

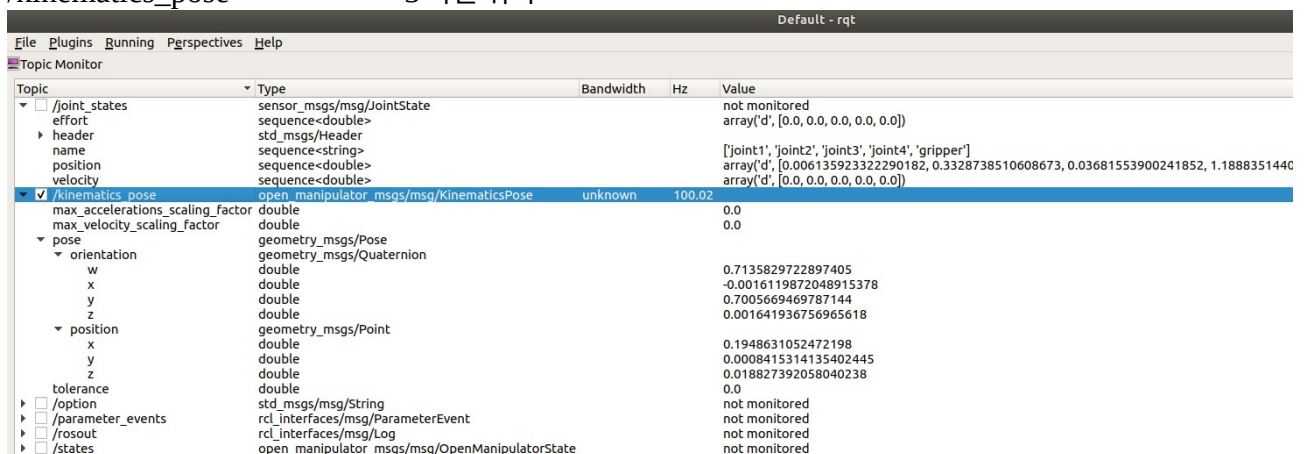
Plugins - Topics - Topic Monitor

/joint_states 체크표시, 더블클릭
position, name 확인가능



Topic	Type	Bandwidth	Hz	Value
✓ /joint_states	sensor_msgs/msg/JointState	unknown	99.75	
▸ /kinematics_pose	open_manipulator_msgs/msg/KinematicsPose			not monitored
▸ /option	std_msgs/msg/String			not monitored
▸ /parameter_events	rcl_interfaces/msg/ParameterEvent			not monitored
▸ /rosout	rcl_interfaces/msg/Log			not monitored
▸ /states	open_manipulator_msgs/msg/OpenManipulatorState			not monitored

/kinematics_pose - 3 차원 위치



Topic	Type	Bandwidth	Hz	Value
▸ /joint_states	sensor_msgs/msg/JointState			not monitored
▸ effort	sequence<double>			array('d', [0.0, 0.0, 0.0, 0.0, 0.0])
▸ header	std_msgs/Header			
▸ name	sequence<string>			['joint1', 'joint2', 'joint3', 'joint4', 'gripper']
▸ position	sequence<double>			array('d', [0.006135923322290182, 0.3328738510608673, 0.03681553900241852, 1.1888351440])
▸ velocity	sequence<double>			array('d', [0.0, 0.0, 0.0, 0.0, 0.0])
✓ /kinematics_pose	open_manipulator_msgs/msg/KinematicsPose	unknown	100.02	
▸ max_accelerations_scaling_factor	double			0.0
▸ max_velocity_scaling_factor	double			0.0
▸ pose	geometry_msgs/Pose			
▸ orientation	geometry_msgs/Quaternion			
▸ w	double			0.7135829722897405
▸ x	double			-0.0016119872048915378
▸ y	double			0.7005669469787144
▸ z	double			0.001641936756965618
▸ position	geometry_msgs/Point			
▸ x	double			0.1948631052472198
▸ y	double			0.0008415314135402445
▸ z	double			0.018827392058040238
▸ tolerance	double			0.0
▸ /option	std_msgs/msg/String			not monitored
▸ /parameter_events	rcl_interfaces/msg/ParameterEvent			not monitored
▸ /rosout	rcl_interfaces/msg/Log			not monitored
▸ /states	open_manipulator_msgs/msg/OpenManipulatorState			not monitored

rqt 종료

파이썬실행

File – Open – /home/phil/colcon_ws/src/pick_and_place – OK

File – Settings – Project~ - Python Interpreter - Python 3.6 – OK – OK

pick_and_place 폴더에서

마우스오른쪽버튼 – New - Python 파일 – [main.py](#) , [get_kinematics.py](#) (파이썬 파일 만들기)

[[get_kinematics.py](#)]

.. 코드작성 ..

`ros2 topic echo kinematics_pose`

numpy array 와 list 차이점 → 넘파이 넘기면 값이 복사되서 들어감. 리스트는 얕은복사?(둘다 값이 바뀔)

→ 리스트에서 값 복사할 때 `copy()` 사용

[[main.py](#)]

.. 코드작성 ..

[[setup.py](#)] 추가수정

```
entry_points={
    'console_scripts': [
        'pick_and_place = pick_and_place.main:main'
    ],
}
```

[터미널]

```
cb # 빌드
ros2 run pick_and_place pick_and_place ctrl+C
```

pick 위치이동, Gripper – joint 이용

Place 위치이동 – kinematics 이용

joint_control.py 파이썬파일 만들기

[[joint_control.py](#)]

.. 코드 작성 ..

rqt 에서 plugins – service - service caller 타입 확인가능

`ros2 service list`

[터미널]

```
cb # 빌드
ros2 run pick_and_place pick_and_place
```

<예제파일 다운로드>

[터미널]

colcon_ws/src 에서 # cs

git clone https://github.com/jungsuyun/open_manipulator_x_tutorial.git

google – notion

client.call	동기식
client.call_async	비동기식