Building Docker image

```
$ ./build_erc.sh
```

Running container

```
$ docker run -it --name=ur_erc --env="DISPLAY" --env="QT_X11_NO_MITSHM=1"
--volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" --volume="/dev:/dev" --privileged ur3_erc
```

Enable X server host:

```
$ xhost +local:root
```

or if you're concerned about security:

```
$ xhost +local:‘docker inspect --format=’{{ .Config.Hostname }}’ <NAME>‘
```

Then, you can run command in running container:

```
$ docker exec -it ur_erc /bin/bash
```

Now you can run commands below in docker container.

UR3 simulation with MoveIt! and gripper

Simulation in Gazebo:

```
$ rosrun ur_gazebo ur3_erc.launch
```

MoveIt! Planner:

```
$ roslaunch ur3_moveit_config ur3_moveit_planning_execution.launch sim:=true limited:=true
```

RViz GUI:

```
$ roslaunch ur3_moveit_config moveit_rviz.launch config:=true
```

Change in Displays bookmark *Global Options/Fixed Frame* to ‘base_link’.

UR3 simulation with MoveIt! joystick teleop

Simulation in Gazebo:

```
$ rosrun ur_gazebo ur3_erc_workcell.launch
```

MoveIt! Planner:

```
$ roslaunch ur3_moveit_config ur3_moveit_planning_execution.launch sim:=true limited:=true
```

Joystick control:

```
$ roslaunch ur3_moveit_config joystick_control.launch
```

RViz GUI:

```
$ roslaunch ur3_moveit_config moveit_rviz.launch config:=true
```

In *MotionPlanning/Planning* bookmark enable *Allow External Comm.*.

Joystick Command Mappings

Command	PS3 Controller	Xbox Controller	Arctic Controller
+-x/y	left analog stick	left analog stick	left analog stick
+-z	L2/R2	LT/RT	L2/R2
+-yaw	L1/R1	LB/RB	L1/R1
+-roll	left/right	left/right	left/right
+-pitch	up/down	up/down	up/down
change planning group	select/start	Y/A	9/10
change end effector	triangle/cross	back/start	1/3
plan	square	X	4
execute	circle	B	2

Gripper with real robot

Launch ROS Driver for UR3

```
$ roslaunch ur_modern_driver ur3_bringup.launch robot_ip:=<ROBOT_IP>
```

Run gripper node:

```
$ rosrun ur_rg2_gripper gripper_erc_node.py
```

Now you can run one of predefined commands:

```
$ rostopic pub /gripper/command std_msgs/String 'close'
$ rostopic pub /gripper/command std_msgs/String 'open'
$ rostopic pub /gripper/command std_msgs/String 'semi_close'
$ rostopic pub /gripper/command std_msgs/String 'semi_open'
```