* 포맷하는 방법  
  로봇과 함께 동봉된 내용물을 보면 usb가 있다. 해당 usb에 clonezilla를 이용한 포맷파일이 들어있으니 로봇을 잘못 수정한 경우 해당 usb로 포맷을 진행한다.
* Remote PC와 통신하는 방법  
  Remote PC와 통신할 때 설정을 어떻게 하느냐가 중요하다.   
  먼저 Remote PC의 /etc/hosts 파일의 경우 uri를 ‘192.168.0.200 rb1-180131AB’ 로 적어준다.  
  그리고 .bashrc 파일의 경우 Remote PC에서export ROS\_MASTER\_URI=http://192.168.0.200:11311,   
  export ROS\_HOSTNAME=192.168.0.157(본인 Remote PC의 IP)로 적어주도록 한다.
* Remote PC로 원격조종하는 방법  
  rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py 를 이용하여 원격조종이 가능하다. 하지만 teleop\_twist\_keyboard.py의 퍼블리셔의 토픽을 rb1-base의 토픽과 같게 수정해주어야 한다. 즉, ‘cmd\_vel’을 ‘/rb1\_base/pad\_teleop/cmd\_vel’로 수정하여야 한다. 어느 토픽으로 보내야하는지는 rqt\_graph를 통해 확인 가능하다.
* Gazebo & Rviz를 이용해 localization & navigation하는 법  
  roslaunch rb1\_base\_sim\_bringup rb1\_base\_complete.launch move\_base\_robot\_a:=true amcl\_and\_mapserver\_a:=true  
  기본적으로는 위 명령어를 실행하면 된다.  
  이후 2D Pose Estimate를 이용해 지도에서 로봇이 있다고 예상되는 지점을 정해줌으로써, 로봇과 맵의 좌표를 일치시켜주는 작업을 한 뒤, 2D Nav Goal을 이용해 로봇을 운동 시킨다.
* Git에서 받은 파일에 문제가 있는 경우  
  로봇에 존재하는 common파일, robotnik\_sensor 파일을 가져와 catkin\_make 해준다.
* 실제로 SLAM 하는 법  
  roslaunch rb1\_base\_navigation rb1\_base\_gmapping.launch 실행 후  
  rviz를 켜서 맵핑과정을 관측한 뒤 , 맵이 완성되면  
  rosrun map\_server map\_saver -f map\_name 을 통해 맵을 해당 경로에 저장한다.
* Map 저장이 안되는 경우  
  map\_server 패키지의 map\_saver 노드는 /map 토픽의 데이터를 저장하게 된다. 하지만 rb1\_base\_gmapping에서 발행하는 토픽은 /rb1\_base/map이기 때문에, topic tools 패키지를 이용하여 relay를 만들어 /rb1\_base/map 토픽을 /map 토픽으로 발행되도록 만든다. 즉, rb1\_base\_gmapping.launch 파일의 마지막 줄에   
  <node pkg=”topic\_tools” type=”relay” name=”map\_relay” args=”/rb1\_base/map /map”/>  
  를 추가해주면 된다.
* 실제로 Navigation 하는 법  
  roslaunch rb1\_base\_navigation move\_base\_amcl.launch 를 실행한 뒤,   
  rviz를 실행해 2D pose estimation으로 위치를 추측해주고 수동조작으로 로봇을 조작하면서 주변 환경에 따른 자신의 위치를 더 정확하게 추측해준 뒤 2D navigation을 이용해 로봇의 navigation 작업을 수행해준다.
* Navigation Map 변경 방법  
  move\_base\_amcl.launch 파일을 보면 소스의 초반부에 맵을 호출하는 경로가 초기화 되어 있다. 이 부분을 수정해주면 된다.
* Multi\_point\_nav.py파일 사용 방법  
  navigation과 rviz를 실행한 후 터미널 창에 python multi\_point\_nav.py를 실행하고 2D pose estimation을 이용하여 초기 위치를 지정해주면 multi\_point\_nav.py 파일 내에 설정되어 있는 location에 따라 로봇이 이동을 진행하게 된다.
* Multi\_point\_nav.py에 Elevator Service기능 넣는 방법  
  from robotnik\_msgs.srv import SetElevator, SetElevatorRequest, SetElevatorResponse 임포트 후   
  rospy.wait\_for\_service(‘/rb1\_base/robotnik\_base\_control/set\_elevator’)  
  srv = rospy.ServiceProxy(‘/rb1\_base/robotnik\_base\_control/set\_elevator’,SetElevator)  
  req = SetElevator()  
  req.action = 1 or -1  
  srv(req)

본 매뉴얼은 한국산업기술대학교 메카트로닉스공학과 소유의 RB1-BASE AGV 로봇의 설치 및 작동 간에 참고 및 유의점에 대하여 2019년 2월 졸업생인 메카트로닉스공학과 12학번 김수호가 작성하였으며, 수정 및 배포는 자유이며 수정부분만 명시하도록 한다.

김수호

H.P : 010-2339-3774

E-MAIL : suho0515@gmail.com