

# 통신학회 단기강좌: 자율주행 핵심기술 SLAM

## 실습 3. SLAM 실습

민세웅, 김동현, 강인성  
Hanyang Univ.  
(sewoong@hanyang.ac.kr  
kissw@hanyang.ac.kr  
kangis@hanyang.ac.kr )

# Setup

- 실습환경

- Remote PC OS : Ubuntu 16.04 LTS ( Linux Mint 18.2)
- ROS : kinetic
- Turtlebot PC : Raspberry Pi 3 B+
- Turtlebot PC OS : Ubuntu MATE
- Turtlebot : Turtlebot3 burger

참조 : [http://emanual.robotis.com/docs/en/platform/turtlebot3/opencr\\_setup/#opencr-setup](http://emanual.robotis.com/docs/en/platform/turtlebot3/opencr_setup/#opencr-setup)

- ROS Packages 설치

- `$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers`
- `$ cd ~/catkin_ws/src/`
- `$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git`
- `$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git`
- `$ cd ~/catkin_ws && catkin_make`

# PC Setup

- 네트워크 설정

- ROS는 TurtleBot PC와 원격 PC 사이의 통신을 위해 IP 주소를 요구한다.
- Remote PC와 TurtleBot PC는 동일한 와이파이 라우터에 연결 되어야 한다.

	TurtleBot PC	Remote PC
ROS_MASTER_URI	<a href="http://IP_OF_REMOTE_PC:11311">http://IP_OF_REMOTE_PC:11311</a>	<a href="http://IP_OF_REMOTE_PC:11311">http://IP_OF_REMOTE_PC:11311</a>
ROS_HOSTNAME	IP_OF_TURTLEBOT	IP_OF_REMOTE_PC

# OpenCR Firmware Upload – Shell Script

- OpenCR Firmware upload를 하는 방법은 두가지가 있다.
  1. Shell Script (recommend)
  2. Arduino IDE (If you need to modify TurtleBot3's firmware)
- 실습에선 Shell Script를 사용하여 업로드 하였다.

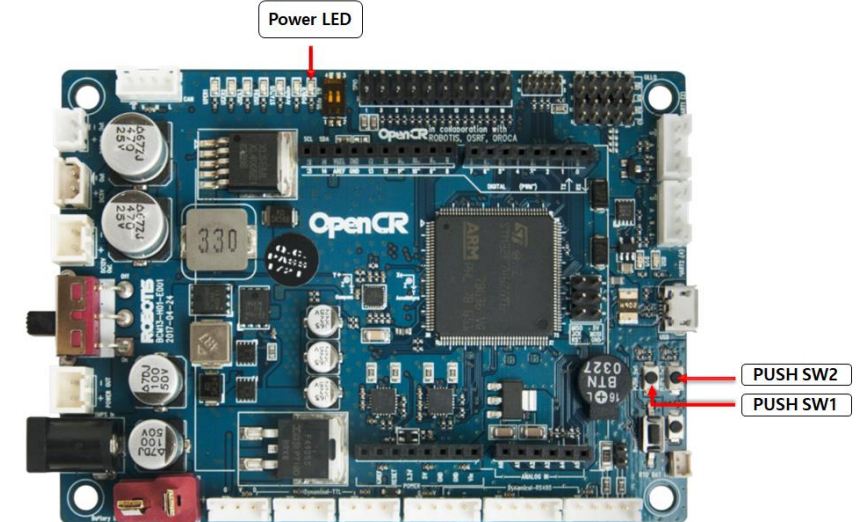
# OpenCR Firmware Upload – Shell Script

- OpenCR을 Remote PC에 연결한 뒤 사용하거나 Turtlebot PC(Raspberry Pi 3)에 연결한 뒤 다음 명령어를 실행한다.
- `$ export OPENCNCR_PORT=/dev/ttyACM0`
- `$ export OPENCNCR_MODEL=burger`
- `$ rm -rf ./opencnrc_update.tar.bz2`
- `$ wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencnrc_update.tar.bz2 && tar -xvf opencnrc_update.tar.bz2 && cd ./opencnrc_update && ./update.sh $OPENCNCR_PORT $OPENCNCR_MODEL.opencnrc && cd ..`

# OpenCR Firmware Upload – Shell Script

- 완료가 되면 jump\_to\_fw 텍스트가 출력된다.

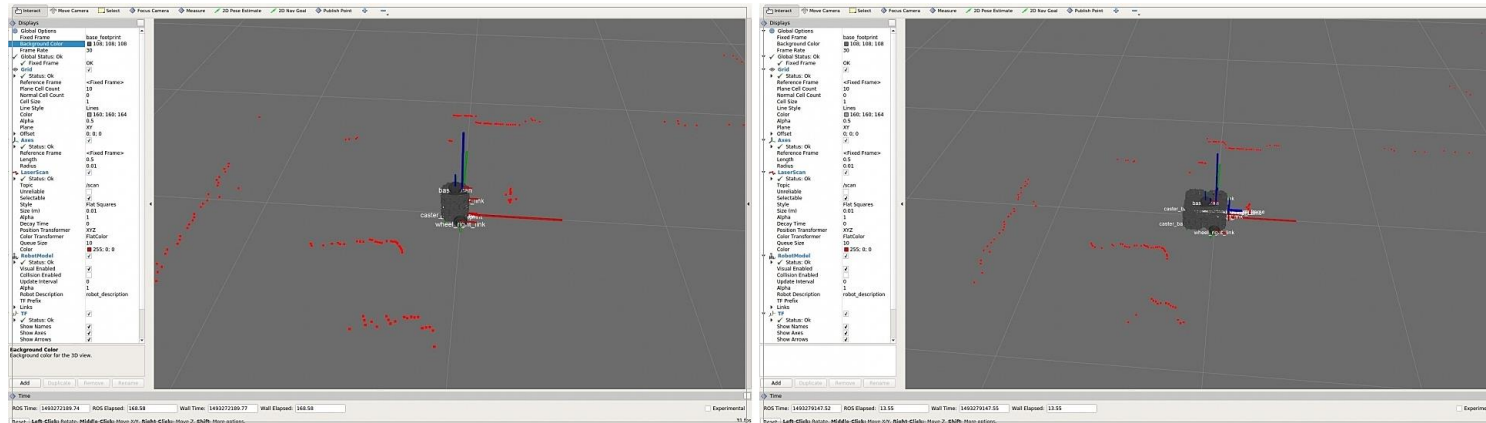
```
opencr_ld_main
[ ] file name      : burger.opencr
[ ] file size      : 172 KB
[ ] fw_name        : burger
[ ] fw_ver         : 1.0.17
[OK] Open port     : /dev/ttyACM0
[ ]
[ ] Board Name     : OpenCR R1.0
[ ] Board Ver      : 0x17020800
[ ] Board Rev      : 0x00000000
[OK] flash_erase   : 0.92s
[OK] flash_write   : 1.84s
[OK] CRC Check     : 11A1E12 11A1E12 , 0.005000 sec
[OK] Download
[OK] jump_to_fw
turtlebot@turtlebot:~$
```



- 업로드 후 OpenCR 보드의 스위치를 눌러 기본동작 테스트를 한다.
  - 배터리를 연결하면 LED에 빨간불이 켜진다.
  - SW1 을 누르면 로봇이 30cm 전진한다.
  - SW2 를 누르면 로봇이 180도 회전한다.

# Bringup

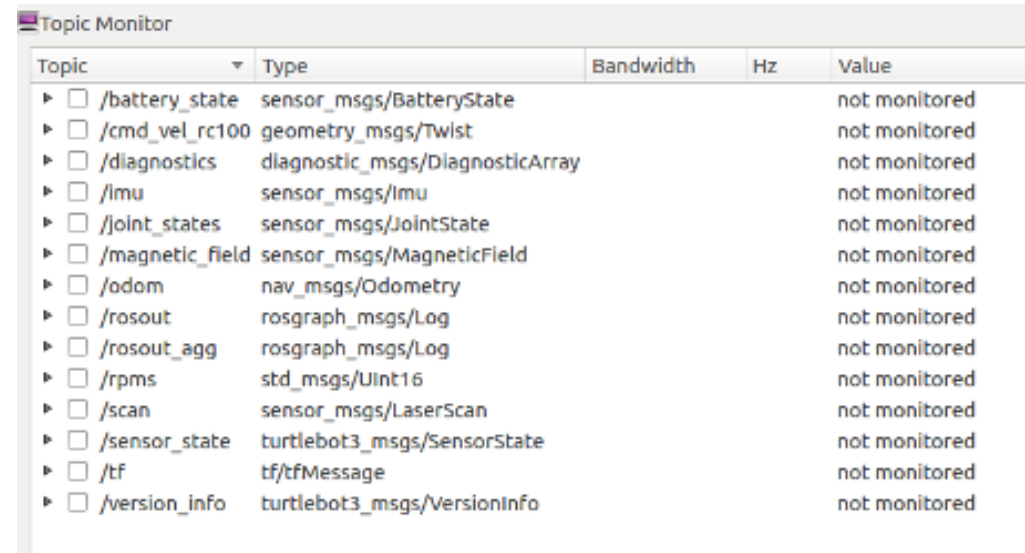
- [Remote PC] Rviz를 통해 시각적으로 확인하기
  - 다음 명령을 실행하기 전에 Turtlebot3의 모델 이름을 지정 해야 한다.
  - `$ export TURTLEBOT3_MODEL=${TB3_MODEL}`
  - `${TB3_MODEL}` 에 자신의 모델에 해당하는 모델명을 적으면 된다.
  - `$ roslaunch turtlebot3_bringup turtlebot3_remote.launch`
- 그 후 다음 명령어를 통해 rviz를 실행시킨다.
- `$ rosrn rviz rviz -d `rospack find turtlebot3_description`/rviz/model.rviz`





# Topic Monitor

- TurtleBot3 가 Publish 하는 Topic을 확인하는데 rqt 사용된다.
- Rqt는 ROS의 GUI 개발을 위한 Qt-based 프레임워크 이다.
- 터미널창에 다음과 같은 명령을 입력하여 rqt를 실행할 수 있다.
- [Remote PC]
  - \$ rqt
  - 만약 rqt를 실행해도 topic이 보이지 않는다면  
plugin -> Topics -> Topic Monitor를  
선택해준다.



Topic	Type	Bandwidth	Hz	Value
<input type="checkbox"/> /battery_state	sensor_msgs/BatteryState			not monitored
<input type="checkbox"/> /cmd_vel_rc100	geometry_msgs/Twist			not monitored
<input type="checkbox"/> /diagnostics	diagnostic_msgs/DiagnosticArray			not monitored
<input type="checkbox"/> /imu	sensor_msgs/Imu			not monitored
<input type="checkbox"/> /joint_states	sensor_msgs/JointState			not monitored
<input type="checkbox"/> /magnetic_field	sensor_msgs/MagneticField			not monitored
<input type="checkbox"/> /odom	nav_msgs/Odometry			not monitored
<input type="checkbox"/> /rosout	rosgraph_msgs/Log			not monitored
<input type="checkbox"/> /rosout_agg	rosgraph_msgs/Log			not monitored
<input type="checkbox"/> /rpms	std_msgs/UInt16			not monitored
<input type="checkbox"/> /scan	sensor_msgs/LaserScan			not monitored
<input type="checkbox"/> /sensor_state	turtlebot3_msgs/SensorState			not monitored
<input type="checkbox"/> /tf	tf/tfMessage			not monitored
<input type="checkbox"/> /version_info	turtlebot3_msgs/VersionInfo			not monitored

# Run SLAM Nodes

- [Remote PC] roscore 실행
  - `$ roscore`
- [TurtleBot]
  - `$ roslaunch turtlebot3_bringup turtlebot3_robot.launch`
- [Remote PC] Gmapping 알고리즘을 사용한 SLAM
  - `$ roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping`
- Turtlebot3는 Gmapping, Cartographer, Hector 등의 SLAM Method를 지원한다.
- `slam_methods:=${SLAM Method}` 를 통해 SLAM Method를 바꿀 수 있다.

# Run SLAM Nodes

- TurtleBot3를 원격으로 조종하기 위해 새 터미널창을 열어준다.
- [Remote PC] Run Teleoperation Node
  - \$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

```
Control Your TurtleBot3!
```

```
-----  
Moving around:
```

```
      w  
a      s      d  
      x
```

```
w/x : increase/decrease linear velocity  
a/d : increase/decrease angular velocity  
space key, s : force stop
```

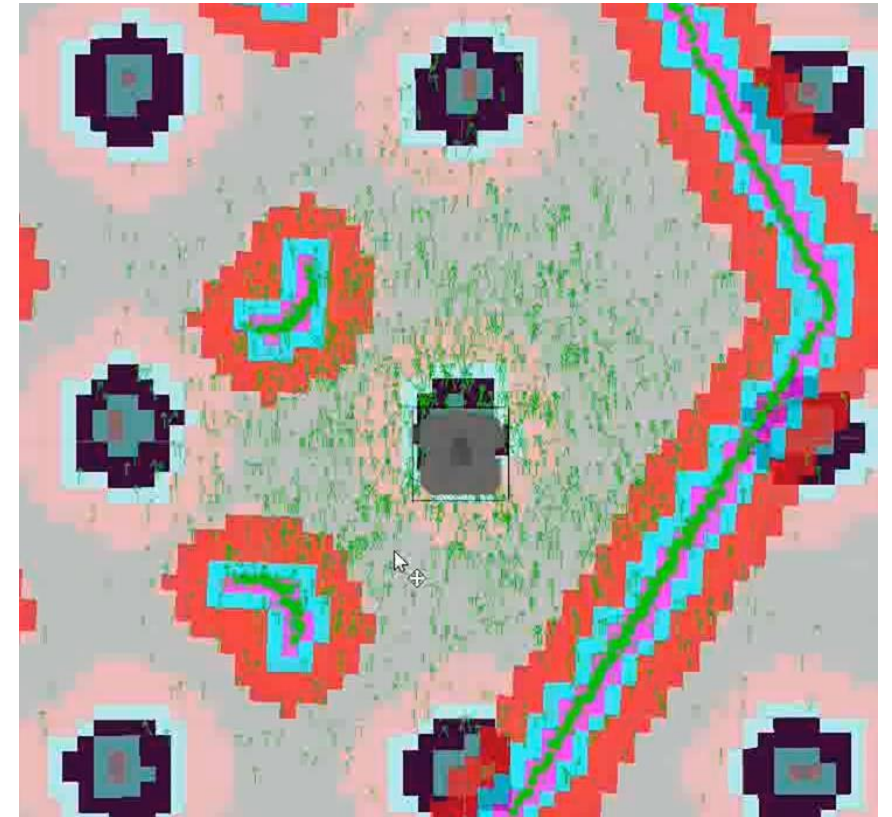
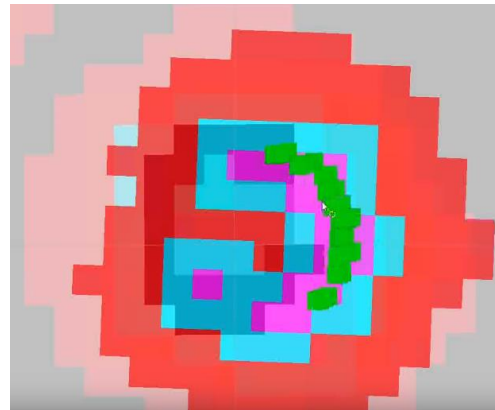
```
CTRL-C to quit
```

# Run Navigation Nodes

- [Remote PC] Run roscore
  - `$ roscore`
- [TurtleBot]
  - `$ roslaunch turtlebot3_bringup turtlebot3_robot.launch`
- [Remote PC] Launch the navigation file
  - `$ export TURTLEBOT3_MODEL=${TB3_MODEL}`
  - `$ roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/map.yaml`

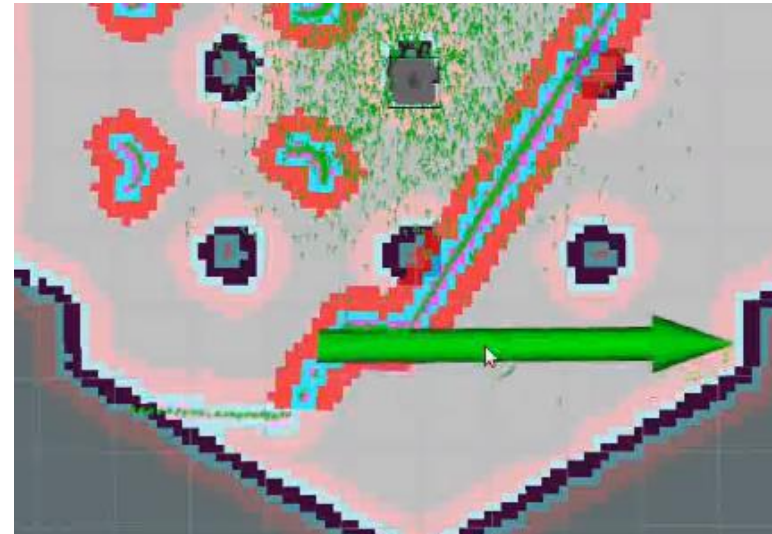
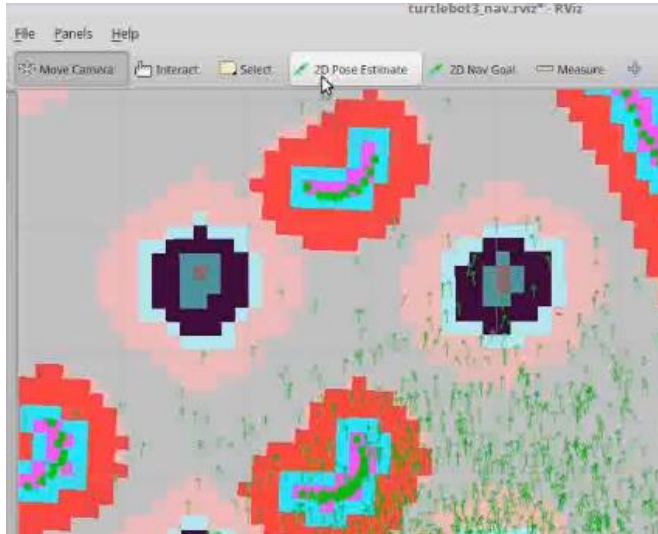
# Run Navigation Nodes

- SLAM Node을 통해 지도를 작성한 뒤 Navigation을 위해 TurtleBot이 지도상에서 어디에 위치해 있는지 초기 위치를 알려줘야 한다.
- 녹색 화살표 : 예측된 로봇의 위치
- 녹색 화살표의 방향 : 예측된 로봇의 방향
- 초록색 점 : 센서 값
- 분홍색 영역 : 장애물로 추정되는 부분
- 빨간색 영역 : 로봇 중점이 이 영역을 넘었을 때 충돌할 위험이 있음.
- 초기 위치를 모르기 때문에 화살표가 랜덤하게 뿌려져 있다



# Run Navigation Nodes

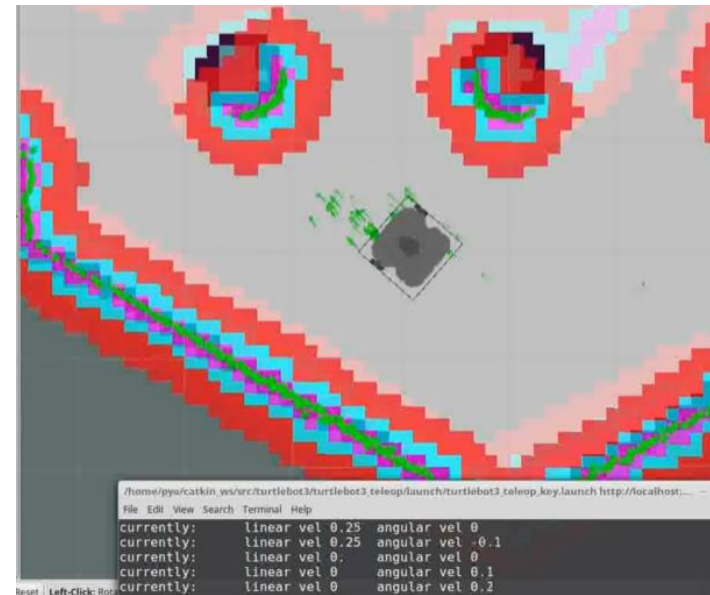
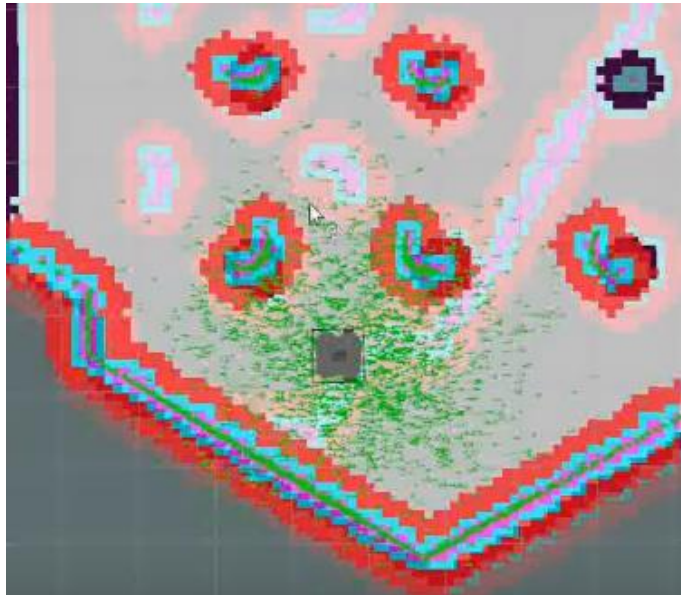
- 초기 위치를 잡아주지 않아도 동작을 하지만 현재 위치를 추정하기 까  
지 시간이 오래 걸림!
- 따라서 rviz에서 다음과 같은 방법으로 초기 위치를 설정해준다.
- [Remote PC] Estimate Initial Pose
  1. 2D Pose Estimate 버튼을 클릭한다.
  2. 현재 터틀봇이 있는 위치와 방향을 선택해준다.



# Run Navigation Nodes

- [Remote PC] Estimate Initial Pose

3. 초기 위치가 바뀌고 바뀐 위치를 기준으로 화살표들이 랜덤하게 뿌려져 있다.
4. `$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`
5. 위 명령어를 실행시키고 로봇을 회전시키거나 앞뒤로 살짝 움직여 준다.
6. 이 때 센서 값을 받아오면서 기존의 맵과 매칭하여 가장 그럴싸한 위치를 추정한다.



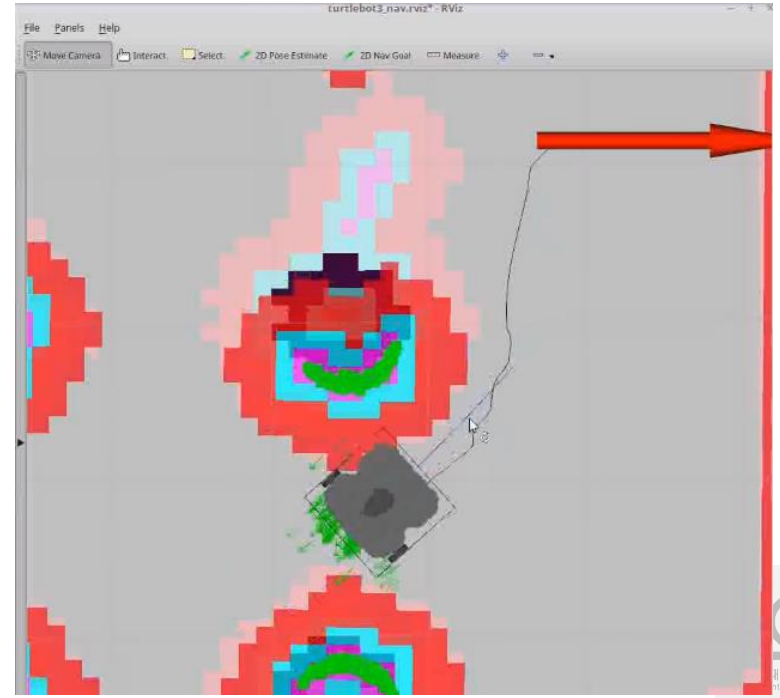
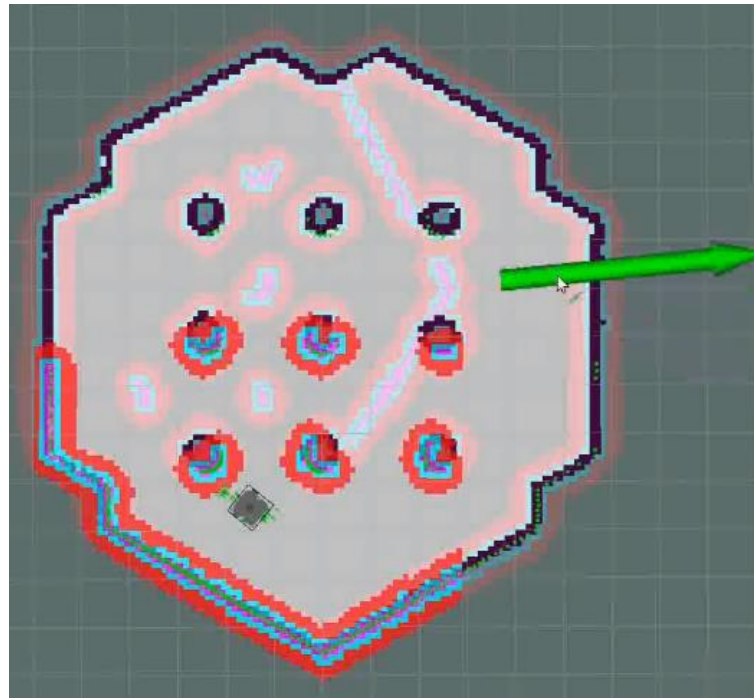
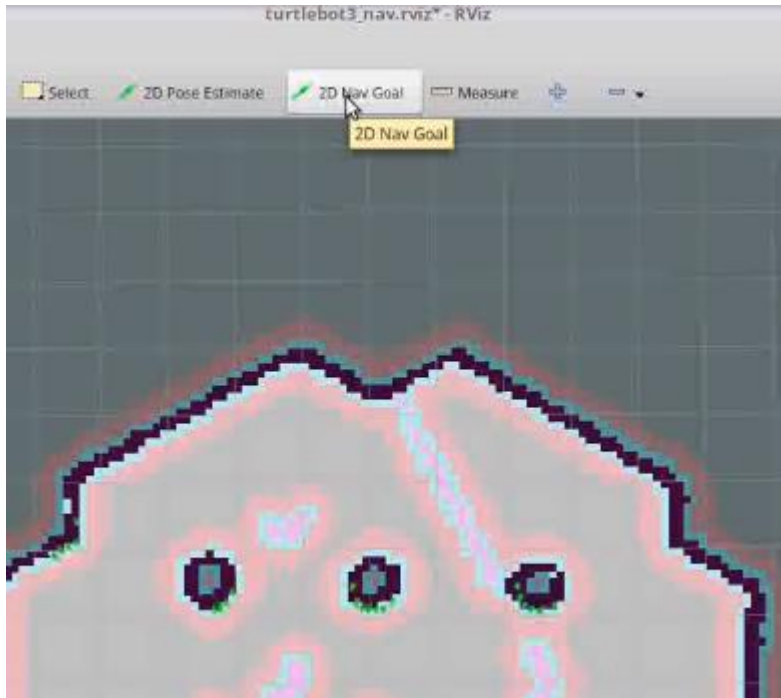


# Run Navigation Nodes

- [Remote PC]

주의 : Navigation을 하기 전 turtlebot3\_teleop\_key node를 종료해준다.

1. 2D Nav Goal 버튼을 클릭한다.
2. TurtleBot을 이동시킬 위치와 도착 시 바라보게 할 방향을 설정해준다.
3. Rviz에서 나온 경로를 따라 TurtleBot이 이동함을 확인 할 수 있다.





감사합니다.