

최종적으로 해야하는 순서

1. 리눅스 세팅

- 로스 코어, 토픽 에코 on
- 로스 브릿지(서버 연동)
- Gazebo on
- 네비게이션 on

2. 모스키토(MQTT), 모비우스(Mobius),Thyme on

- app.js open 후 데이터 전송 여부 확인
- simple.html open
- 목표값 입력, 디바이스 색깔 선택, 디바이스 카메라 퀄리티 선택 버튼 입력

3. 코드 simple.html 수정부분 설명

- 목표값 입력, 디바이스 색깔 선택, 디바이스 카메라 퀄리티
- html파트
- publish 버튼을 통해 적용시키는 코드
- listener코드
- rqt_graph 첨부

최종 결과물

- simple.html 부트 스트랩 변경화면 소개

virtualBox -> ubuntu 설치

1. Ros 설치
2. Ros Melodic 설치
3. Turtlebot3 설치 (melodic기반)

<https://emanual.robotis.com/docs/en/platform/turtlebot3/navigation/#ros-1-navigation>

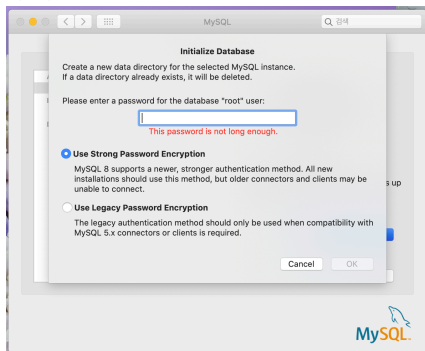
네비게이션에서 시뮬레이터까지 설치 동작 확인

Mobius 설치 / nCube 설치

https://github.com/IoTKETI/Mobius/wiki/Mobius_v2.0.0_KR_Windows

맥의 경우는 안내가 없기 때문에 brew를 사용해서 모비우스를 동작해야 함.

MySQL은 워크벤치로 동작시키지만 이 때 리거시로 체크하고 설정해주어야 함.



모스키토를 동작시킬 때 윈도우와 다르기 때문에 아래를 숙지하고 동작시켜야 하며 비서비스로 실행해야 실행 로그를 볼 수 있음. 또한 백그라운드에서 동작하기 때문에 터미널을 켜고 해도 프로세스가 종료되지 않음. 강제로 kill을 해주어야 함.

모스키토 설치

```
brew install mosquito
```

모스키토 서비스 실행

```
brew services start mosquito
```

모스키토 서비스 중지

```
brew services stop mosquito
```

비서비스 실행

```
/usr/local/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
```

```
lemonheim~$ /usr/local/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
1515670129: mosquitto version 1.4.14 (build date 2017-10-22 16:34:39+0100) starting
1515670129: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
1515670129: Opening ipv6 listen socket on port 1883.
1515670129: Opening ipv4 listen socket on port 1883.
```

메세지 구독

```
mosquitto_sub -h [주소] -p [포트] -t [주제, 방제목]
```

```
/usr/local/opt/mosquitto/bin/mosquitto_sub -h 127.0.0.1 -p 1883 -t topic
```

```
mosquitto_sub -h localhost -p 1883 -t nuns
```

메시지 발행(퍼블리싱)

```
mosquitto_pub -h [주소] -p [포트] -t [주제, 방제목] -m [메세지]
```

```
/usr/local/opt/mosquitto/bin/mosquitto_pub -h 127.0.0.1 -p 1883 -t topic -m "test message"  
mosquitto_pub -h localhost -p 1883 -t nuns -m "hello"
```

```
lsof -i:1883  
kill -9
```

```
control \
```

```
64 113
```

또한 nCube time을 써야하고 최근 것으로 다운받으면 충돌이나 누락되는 것이 있기 때문에 깃허브에 올라와있는 것으로 다운받고 실행해서 로그 찍었을 때 3105가 포트로 찍여야 함.

Rosbirde 설치 / 세팅

리눅스에 있는 ros와 윈도우에 있는 모비우스가 서로 연결되어야 하는데 이때 사용하는 것이 rosbridge이다.

<https://noel-embedded.tistory.com/742>

특히 여기서는 포트를 할당해주는 포트포워딩, 방화벽 인증을 해주어야 한다.

현재 사용하고 있는 인터넷 공유기의 포트를 서로 할당해야한다. 때문에 iptime을 사용하거나 skt 공유기를 사용하는 경우에 따라 다르게 포트포워딩을 해주어야 한다.

Iptime을 사용하는 경우는 위 링크처럼 진행하고 skt의 경우는

<https://m.blog.naver.com/PostView.nhn?blogId=dlrhkd369&logNo=221195715732&proxyReferer=https:%2F%2Fwww.google.com%2F>

이 링크처럼 게이트웨이 192.168.25.1 로 들어가서 할당해주면 된다.

```
Roslaunch rosbridge_server rosbridge_websocket.launch
```

```
Route get default >> 맥에서 현재 맥주소 찾기
```

```
-바꾸기 ---
```

```
Sudo apt-get install ros-melodic-rosbridge_server  
Sudo apt-get install ros-melodic-rosbridge-suite
```

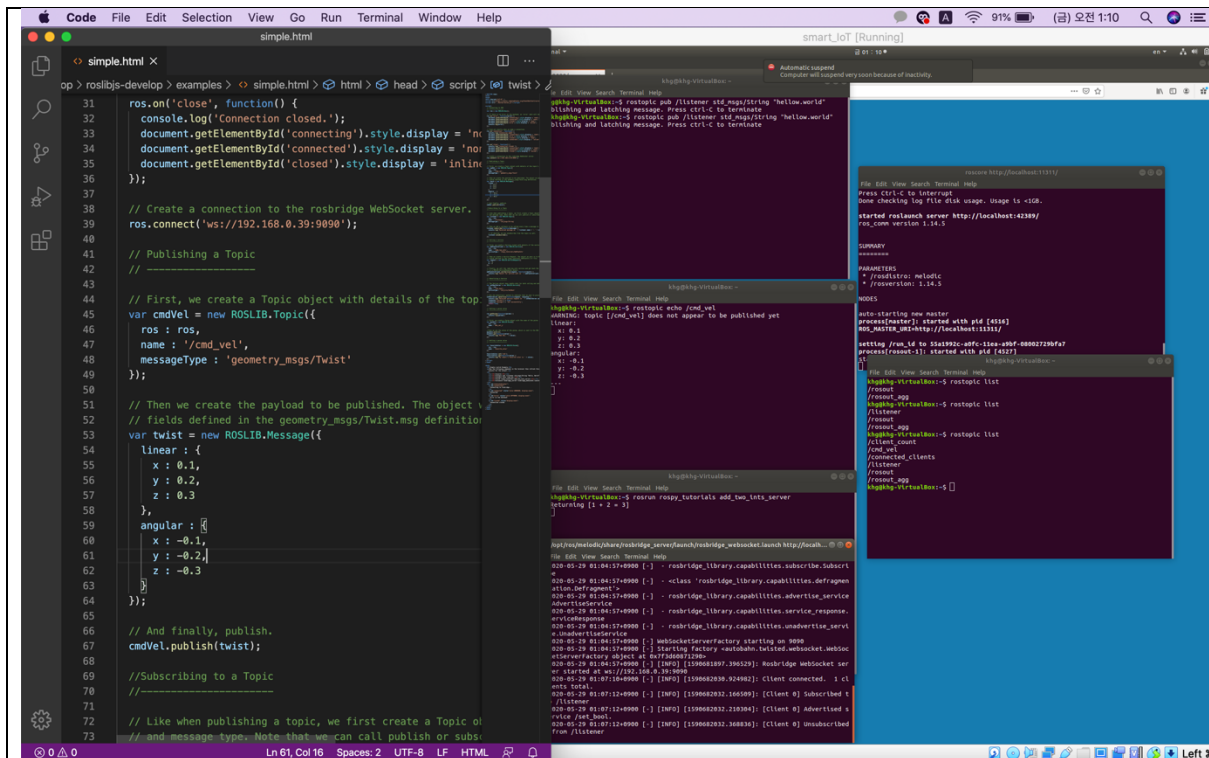
```
Rocs rosbridge_server
```

```
Cd launch/
```

```
Ls
```

```
Sudo gedit rosbridge_websocket.launch
```

```
->바꾸기
```



Roslibjs -> test.js 에서 바꾸기

- Launch Gazebo

```
$ export TURTLEBOT3_MODEL=waffle_pi
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

- Launch SLAM

```
$ export TURTLEBOT3_MODEL=waffle_pi
$ roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping
```

- Remotely Control TurtleBot3

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

- Save the Map

```
$ rosrn map_server map_saver -f ~/map
```

이건 가제보

Virtual Navigation Execution Procedure

Terminate all applications that were executed during the virtual SLAM practice and execute related packages in appear on the previously generated map. After setting the initial position of the robot on the map, set the destination figure below. The initial position only needs to be set once.

- Execute Gazebo

```
$ export TURTLEBOT3_MODEL=waffle_pi
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

- Execute Navigation

```
$ export TURTLEBOT3_MODEL=waffle_pi
$ roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/map.yaml
```

참고) Ros 명령어

Ros 명령어

cat 내용보기

find 찾기 find ./ -name 'setup.bash'

find ./ -name '*.jpg'

rm -rf 삭제

rm -r

roscd 패키지의 위치를 알고 싶으면 cd 를 사용하면 됨. -> 로스만을 위해 만들어짐

gedit 파일을 열어줌! 새창으로다가

rostopic list 로스가 받아오는 센서들을 볼 수 있음

rqt_graph 어떻게 움직이는지 그래프를 볼 수 있음

rospack + tab 로스에서 관리하는 패키지들을 나열함 그중에서 find 를 가장 많이 씀. rospack find turtlebot3_gazebo

그 다음은 depends1 인데 이건 의존성을 확인함 rospack depends1 my_first_pkg

catkin에서 패키지 파일들을 저장하는게 src

catkin에 새로운 패키지를 만들려면 src 에서 만들어야겠지? -> catkin_create_pkg my_first_pkg std_msgs rospy roscpp

여기서 std~ 는 의존성을 나타냄

```
lsgx@lsgx-VirtualBox:~$ rosnode list
/rosout
lsgx@lsgx-VirtualBox:~$ rosnode info /rosout
-----
Node [/rosout]
Publications:
* /rosout_agg [roscpp_msgs/Log]
Subscriptions:
* /rosout [unknown type]
Services:
* /rosout/get_loggers
* /rosout/set_logger_level
contacting node http://localhost:44745/ ...
Pid: 5093

lsgx@lsgx-VirtualBox:~$ rosnode info /turtle1
-----
Node [/turtle1]
Publications:
* /turtle1_color_sensor [std_msgs/Color]
* /turtle1_pose [std_msgs/Pose]
Subscriptions:
* /turtle1/cmd_vel [std_msgs/Float64]
Services:
* /clear
* /kill
* /reset
* /spawn
* /turtle1/set_pen
* /turtle1/teleport_absolute
* /turtle1/teleport_relative
* /turtle1/get_loggers
* /turtle1/set_logger_level
contacting node http://localhost:44575/ ...
Pid: 5302
Connections:
* topic: /rosout
* to: /rosout
* direction: outbound (36399 - 127.0.0.1:59740) [26]
* transport: TCPROS
```

connections는 depends 녀석들..

```
lsgx@lsgx-VirtualBox:~$ roslaunch turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle. 'q' to quit.
█

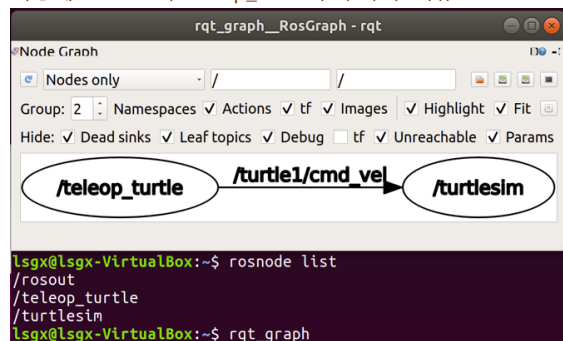
contacting node http://localhost:44575/ ...
Pid: 5302
Connections:
* topic: /rosout
  * to: /rosout
  * direction: outbound (36399 - 127.0.0.1:59740) [26]
  * transport: TCPROS
* topic: /turtle1/cmd_vel
  * to: /teleop_turtle (http://localhost:41685/)
  * direction: inbound (43536 - localhost:47019) [28]
  * transport: TCPROS
```

turtlesim에 키보드를 사용한다고 명령어 치고나서 다시 `roslaunch info /turtlesim` 하면 새로운 호스트가 추가가 됨.

더불어 사용하고 있는 `roslaunch list` 는

```
lsgx@lsgx-VirtualBox:~$ roslaunch list
/roslaunch
/teleop_turtle
/turtlesim
```

이렇게 turtlesim 과 teleop_turtle이 추가가 되쥬



이런 꼴로 동작하게

```
lsgx@lsgx-VirtualBox:~$ rostopic list
/roslaunch
/roslaunch_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
```

사용하는 센서인 topic list