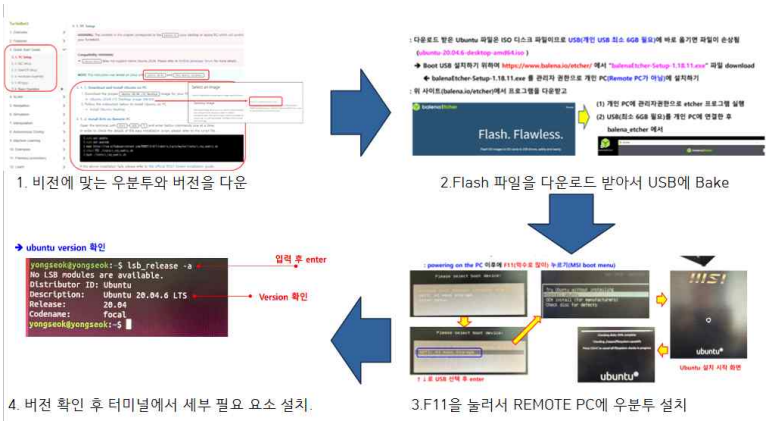
Reference site

1. source code : github.com의 ROBOTIS-GIT
=> https://github.com/ROBOTIS-GIT/turtlebot3
2. OpenCR (Control module for ROS) : github.com의 ROBOTIS-GIT
=> https://github.com/ROBOTIS-GIT/OpenCR
3. Robotis e-manual : https://www.robotis.com
=> https://emanual.robotis.com/docs/en/platform/turtlebot3/overview
소스코드 : https://github.com/robotpilot/ros-seminar
ST32F746 : http://www.arm.com/products/processors/cortex-a/cortex-a53-processor.php
라즈베리파이 : https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-4-model-b
패키지 다운로드 : http://repositories.ros.org/status_page/ros_noetic_default.html
ROS 사용법 및 패키지 설명 : http://wiki.ros.org
사용 버전
로봇 운영체제 : http://www.ros.org
RQT (실시간 모니터링)
- 노드와 그들 사이의 연결 정보 표시(rqt_graph)
- 인코더, 전압, 시간에 따라 변화하는 숫자를 플롯(rqt_plot)
- 데이터를 메시지 형태로 기록하고 재생(rqt_bag)
Rviz
- ROS에서 제공하는 3D visualization tool
ROS Master : Node와 Node 사이의 연결과 메시지 통신을 위한 네임 서버와 같은 역할
ROS Master가 실행이 되어야 노드 통신이 가능함 - 실행 명령어 : roscore
ROS Node (http://wiki.ros.org/Nodes) : ROS에서 실행되는 최소 단위의 프로세서를 의미. 즉 하나의 실행 가능한 프로그램이며, 하나의 목적에 하나의 node를 작성(각 프로그램 세분화)
센서 드라이브, 센서 데이터를 이용한 변환, 장애물 판단, 모터 구동, Encoder 입력, 내비게이션 등 세분화된 node
ROS Package (http://wiki.ros.org/Packages) : ROS 응용프로그램은 Package 단위로 개발
Package는 최소한 하나 이상의 node를 포함하거나 다른 Package의 node를 실행하기 위한 설정 파일을 포함
ROS Message (http://wiki.ros.org/Messages) : 노드 간의 데이터를 주고 받음. 메시지는 integer, floating point, Boolean 같은 변수 형태
ROS Topic (http://wiki.ros.org/Topics)
: publisher와 subscruber의 대화 형식임.
다수와 다수가 될 수 있고, 개인과 다수, 개인과 개인이 될 수 있음.
ROS Service (http://wiki.ros.org/Services) : ROS topic 통신 방식은 비동기 방식(필요에 따라 data 송수신) : ROS는 필요에 따라 동기 방식 사용 Service 명칭 - 요청이 있을 때 응답 하는 Service server, 요청하고 응답 받는 Service client로 구분 - Service는 토픽과는 달리 일회성 메시지 통신. 서비스의 요청과 응답이 완료되면 연결된 두 노드의 접속은 끊김
ROS Action (http://wiki.ros.org/actionlib) : 요청 처리 후 응답까지 오랜 시간 필요하고 중간 결과값이 필요한 경우에 사용되는 message 통신 방식 : 요청과 응답에 해당되는 goal과 result, 중간 결과값의 feedback 추가 - Action의 goal을 정하는 action client - goal에 맞춰 일을 수행하고 action result, feedback 전달하는 action server : 비동기 방식 양방향 메시지 통신
catkin (http://wiki.ros.org/catkin) : low-level build system macros and infrastructure for ROS. : ROS와 관련된 빌드, 패키지 관리, 패키지 간의 의존관계를 편리하게 사용하기 위함 : ROS의 빌드 시스템은 CMake(Cross Platform Make)를 이용, 패키지 폴더에 CMakeLists.txt 에 빌드 환경을 기술 ROS에서 CMake를 ROS에 맞게 수정하여, ROS에 특화된 catkin build syste
roscore (http://wiki.ros.org/roscore) : ROS master를 구동하는 명령어 : roscore는 같은 네트워크에서 하나만 구동 됨. - ROS를 구동하면 사용자가 정해놓은 ROS_MASTER_URI 변수에 기재된 URI 주소와 포트를 사용
rosrun (http://wiki.ros.org/rosbash#rosrun) : ROS의 기본 실행 명령어 : package에서 하나의 node를 실행하는데 사용 - 노드가 사용하는 URI 주소와 포트는 현재 노드가 실행 중인 컴퓨터에 저장된 ROS_HOSTNAME 환경 변수 값을 URI 주소로 사용
roslaunch (http://wiki.ros.org/roslaunch) : 여러 node를 실행하는데 사용. 명령어를 통해 하나 그 이상의 정해진 노드를 실행 : node를 실행할 때, package의 parameter나 node name 변경, node name space 설정, ROS_ROOT 및 ROS_PACKAGE_PATH 설정, 환경변수 변경 등 node 실행에 특화된 ROS 명령어 : roslaunch는 *.launch 파일을 사용하여 실행 node에 대한 설정을 해주며 XML(Extensible Markup Language)에 기반, XML 태그 형태의 다양한 옵션을 제공

일자	2023.12.05	보고자	조현미, 송시경, 이송원
주제	ROS 개발 환경 구축		

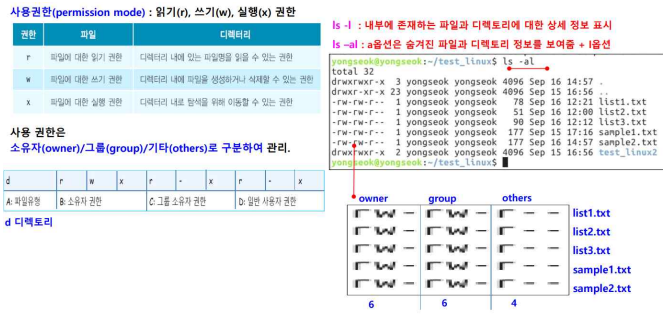
활동 1 개발 환경 설치 Process



1. 비전에 맞는 우분투와 버전을 다운
2.Flash 파일을 다운로드 받아서 USB에 Bake
3.F11을 눌러서 REMOTE PC에 우분투 설치
4. 버전 확인 후 터미널에서 세부 필요 요소 설치.

5. 시작 전 업데이트 및 업그레이드 실시.
6. ROS를 설치전 그 전 패키지 전부 삭제 필수.
7. emanual site에서 버전을 찾고 위 표기 순서대로 진행
8. 빌드 경로 ws변경 및 빌드 진행. 후 get ~/.bashrc에서 IP변경
9. roscore 명령어로 마스터가 실행 되는지 확인 후 마무리

주요 명령어



∴ 폴더의 사용 권한을 나타내는 2진수. ex)$ chmod 777 sample1.txt <= 이 형식으로 변형 가능.

◦ifconfig 네트워크 설정 확인 (유무선) /◦iwconfig 무선랜 설정들을 확인/
◦CTRL + ALT + T 터미널 창 열기 /◦CTRL + Shift + Q 터미널 창 종료/
◦sudo 명령어(super_user do) root 권한이 필요할 경우 /◦cd ~ 체널 변경/
◦ls 내부 폴더 확인/◦mkdir 폴더 제작/◦rmdir 폴더 삭제/◦sudo rm 파일 삭제/
◦roscore 마스터 활성화/◦Ctrl + shift + O 수평분할/◦Ctrl + Shift + E 수직분할/
◦Ctrl + shift + W 현재 활성화 된 창 닫기/◦$ rqt_graph 현재 통신 상태확인 가능/
◦bashrc 상태를 한번에 실행/◦sh 한번에 진행
◦gedit ~/.bashrc 입력 값 확인.
ctrl + c 터미널 창 진행 상황 종료

일자	2023.12.06	보고자	조현미, 송시경, 이송원
주제	Tutlebot 조립 및 환경 구축		

활동 1 tutlebot assemble Process

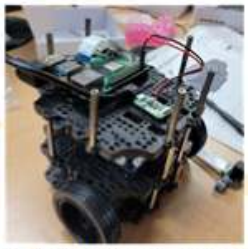
1.보드 조립
TIP : 나사 부분이 평평하게

2.바퀴 조립
TIP : 바퀴 부분이 앞부분

3.양 옆 조립 1,2번 좌우 구별 중요 케이블 정리.

4.OPEN CR보드 조립
TIP : 케이블 정리 중요.

5.OPEN CR보드 와 바퀴보드 조립



6.라즈베리 파이 4 조립 및 케이블 정리 필수.

7.라이다 조립 및 나머지 부분 조립 후 카메라가 지상을 보게 숙여서 고정.

활동 2 tutlebot 환경 구축 Process

1.Robotis에서 라즈베리파이와 ROS혼용되어 있는 버전 다운 후 SD카드에 bake.

2.와이파이 제어 및 할당

3.SD카드를tutlebot에 삽입 후 원격조정 설정.

4.마스터와 호스트 아이피 설정.

5.같은 IP를 피하기 위한 호스트 네임 변경

6.tutlebot과 remote pc 연결

7.IP확인 및 라즈베리파이와 remote pc id 수정

8.환경변수 확인 및 업데이트 및 업그레이드.

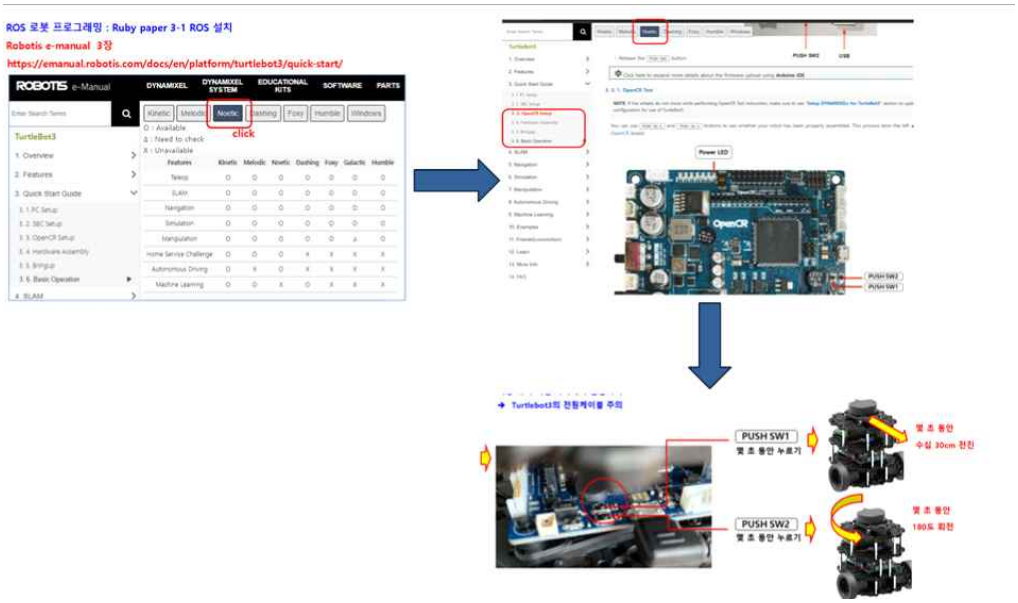
9.추가 업데이트 및 수정사항 적용

철칙

1. 고유의 IP 숙지.
2. tutle bot 원격 제어 종료 시에 sudo shutdown now 명령어로 종료
3. 모든 부품이며, 와이파이를 제어하고 정하는 역량 키워야함.
4. + - 단자를 잘 확인하여 결합하고 케이블 정리 필수. 또한 SD카드 맞는 부분에 삽입

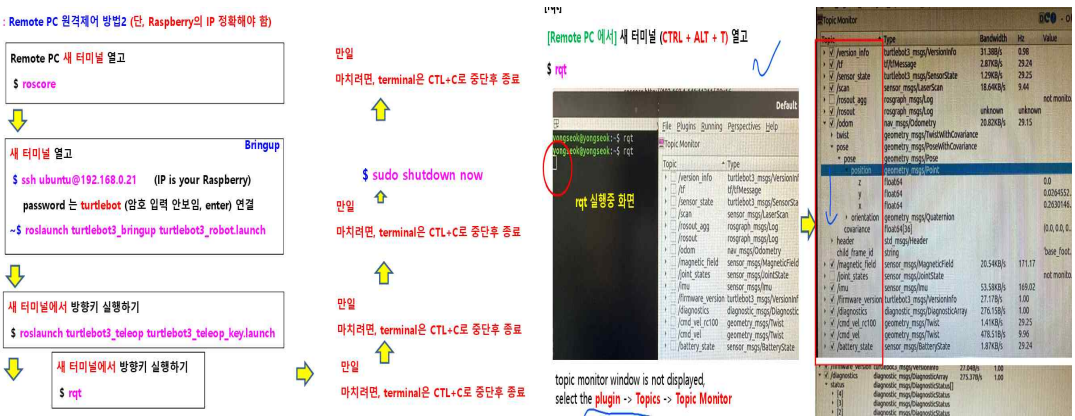
일자	2023.12.11	보고자	조현미, 송시경, 이송원
주제	Turtle bot Open CR 구축 및 ROS 통신 명령어 조작		

활동 1 OPEN CR 구축



turtle bot에서 스위치로 필수 확인!

활동 2 turtle bot 원격 제어

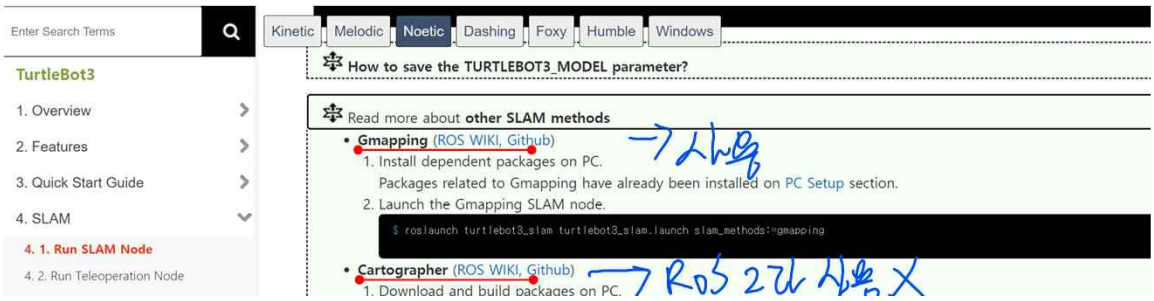


철칙

1. rosnode list를 필수로 확인
2. rqt graph 활성화하여 경로 확인.
3. rostopic echo /topic_이름 사용하여 실시간 통신 확인.
4. 모든 작업이 끝날 시에는 Ctrl + C
5. 마지막은 sudo shutdown now 으로 마무리.

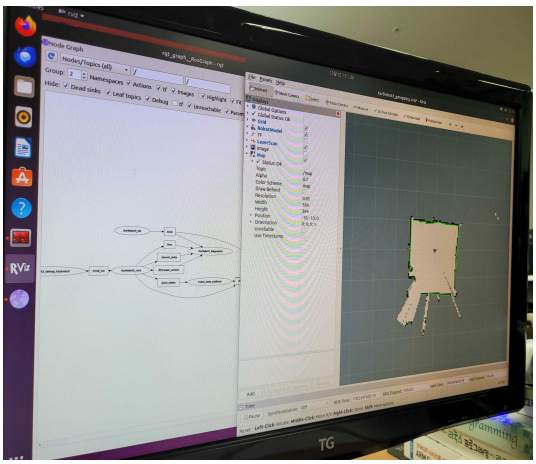
일자	2023.12.12	보고자	조현미, 송시경, 이송원
주제	SLAM 과 Navigation을 활용한 Turtle bot 조작		

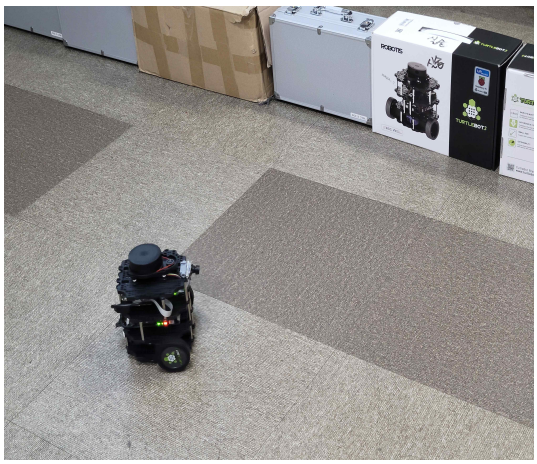
G-mapping 알고리즘 설치



gmapping은 라이다, 레이더 센서 데이터를 기반으로 하는 확률론적인 방법을 사용하여 로봇의 위치 및 환경 맵을 추정하며, 터틀봇은 자체적으로 이동하면서 센서 데이터를 수집하고, gmapping 알고리즘이 이 정보를 활용하여 로봇의 현재 위치를 추정하고 환경의 맵을 동시에 작성함.

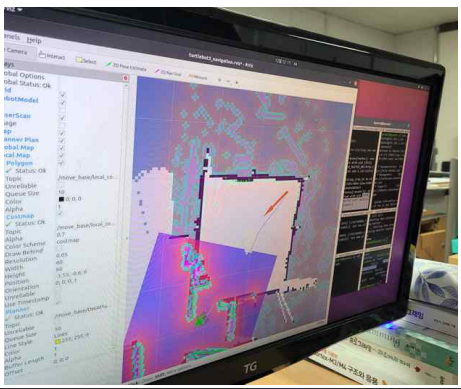
SLAM 을 활용한 map 생성

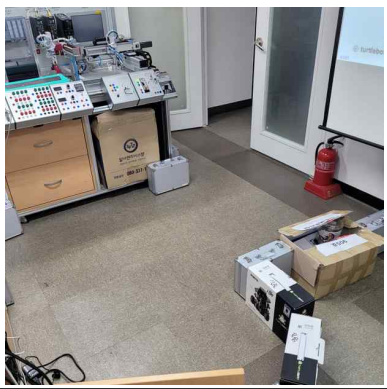




라이다의 측정 범위(3m 이내)를 넘어서지 않는 경기장을 제작하여 map 생성 후 map을 save한 후 확인.

navigation을 활용한 자율주행





Rviz 화면에서 map 일치를 시켜 준 뒤 2D Nav goal을 활용하여 자율 주행 테스트 진행.