
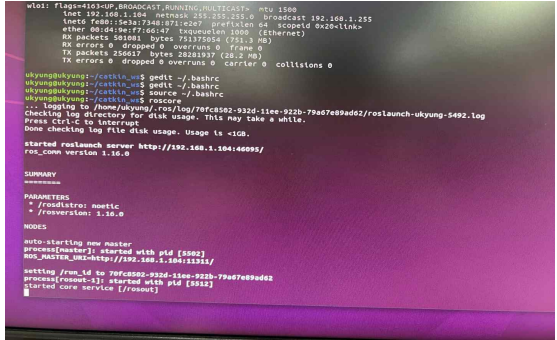

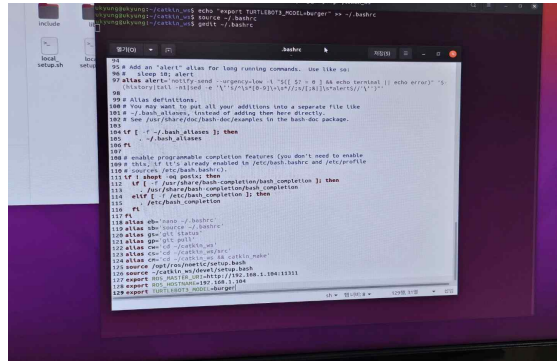

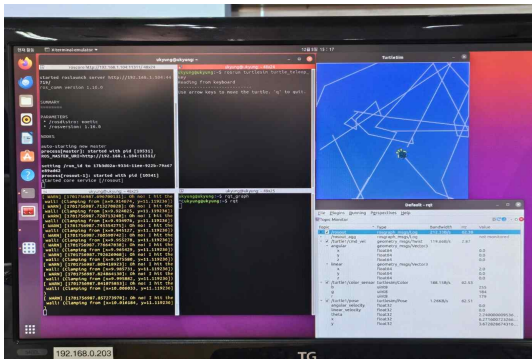

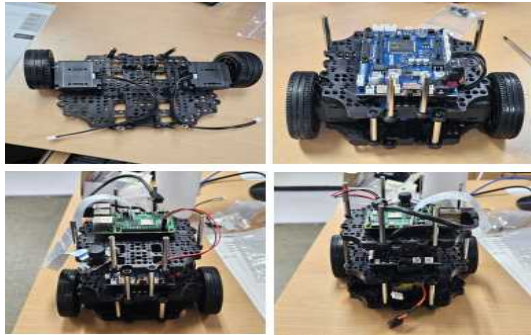
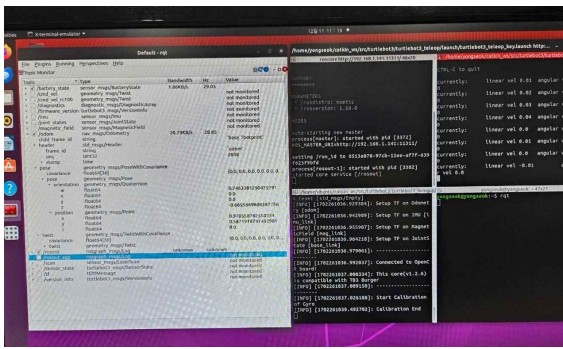
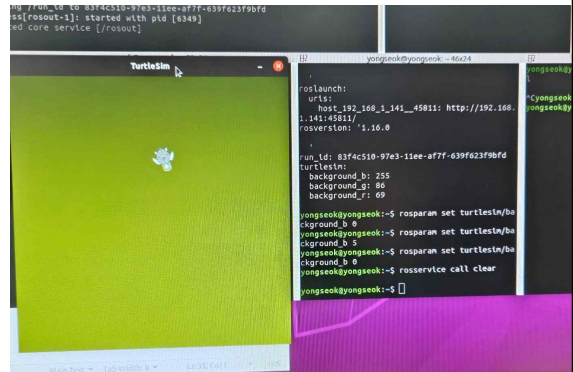


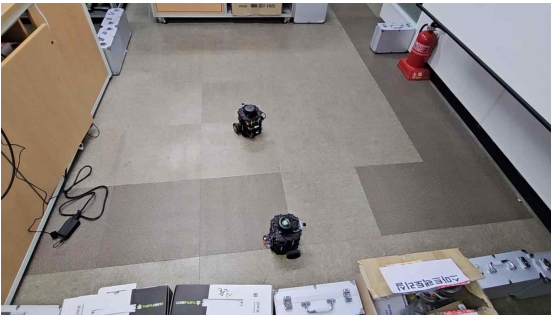
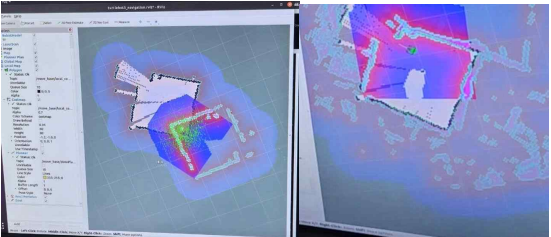
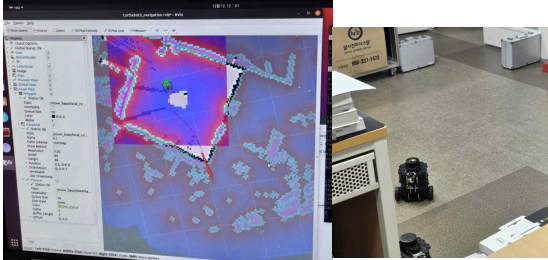
| | | | |
|----|---|----|---------------|
| 날짜 | 2023.12.04. | 조원 | 김유경, 최아영, 최영중 |
| 내용 | | | |
| 요약 | <p>● Turtlebot3 360° LiDAR, SBC(Raspberry PI 4), OpenCR(STM32F746 - ARM Cortex-M7 32bits)</p> <p>● ROS : 로봇 소프트웨어 플랫폼 하드웨어 추상화, 하위디바이스 제어, 로봇틱스 센싱, 인식, 위치 추정, 지도 작성, 내비게이션</p> <p>● ROS 구성 Client Libraries, 하드웨어 인터페이스, 데이터 통신을 위한 커뮤니케이션, Robotics Application Framework, Robotics Application, Simulation, Software Development Tool</p> <p>● ROS 특징 분산 프로세스, 패키지 단위 관리, 공개 repository, API 형태, 복수의 프로그래밍 언어 지원, 로봇에 대한 표준 메시지 정의, 로봇 기하학 라이브러리 제공, 내비게이션, Manipulation, Rviz, RQT etc</p> <p>● ROS 통신 topic, service, action, parameter</p> | | |
| 참조 | <p>● 중요 참조 URL</p> <p>1) Source code : github.com의 ROBOTIS-GIT https://github.com/ROBOTIS-GIT/turtlebot3</p> <p>(2) OpenCR (Control module for ROS) : github.com의 ROBOTIS-GIT https://github.com/ROBOTIS-GIT/OpenCR</p> <p>(3) Robotis e-manual : https://www.robotis.com https://emanual.robotis.com/docs/en/platform/turtlebot3/overview</p> <p>(4) documents download 방법 : https://github.com/robotpilot/ros-seminar</p> <p>● ROS 용어(http://wiki.ros.org/)</p> <p>ROS Master : 실행 명령어 roscore / Master 실행 시 각 Node 이름을 등록, 필요에 따라 정보 받기</p> <p>ROS Node : ROS에서 실행되는 최소 단위의 프로세스를 의미, 즉 하나의 실행 가능한 프로그램</p> <p>ROS Package : ROS 응용 프로그램은 Package 단위로 개발</p> <p>ROS Message : 노드 간의 데이터를 주고 받음</p> <p>ROS Topic : Publisher가 Subscriber에게 topic 전달 / 비동기 방식</p> <p>ROS Service : ROS는 필요에 따라 동기 방식</p> <p>ROS Action : 요청 처리 후 응답까지 시간이 필요, 중간 결과값이 필요한 경우에 사용되는 통신 방식</p> <p>ROS Parameter : default로 설정 값들이 지정되어, 필요에 따라 외부에서 읽거나 쓸 수 있음</p> <p>catkin : ROS 빌드 시스템은 CMake를 이용, 패키지 폴더에 CMakeLists.txt에 빌드 환경을 기술</p> <p>roscpp : ROS Groovy 이전 version에 적용된 build system</p> <p>roscpp : ROS master를 구동하는 명령어 / 같은 네트워크에서 하나만 구동 됨</p> <p>roslaunch : ROS의 기본 실행 명령어 / package에서 하나의 node를 실행하는데 사용</p> <p>roslaunch : 여러 node를 실행하는데 사용(*.launch 파일을 사용하여 실행)</p> <p>bag : ROS에서 주고받는 데이터를 저장하는 파일 포맷(확장자로 *.bag 사용)</p> <p>repository : 공개된 package가 저장된 URL 주소.</p> <p>graph : 실행 명령에는 rqt_graph과 roslaunch rqt_graph</p> <p>client library : roscpp, rospy, roslisp, rosjava, roslua, roscs, roseus, PhaROS, rosR 등</p> <p>CMakeLists.txt : ROS 빌드 시스템 catkin은 기본적으로 CMake 이용</p> | | |

| 날짜 | 2023.12.05. | 조원 | 김유경, 최아영, 최영중 |
|----|--|----|--|
| 내용 | | | |
| 요약 | <p>● 목표 ROS 개발 환경 구축</p> <p>●과정 개인 PC에서 미니 PC(Remote PC)에 Ubuntu와 ROS 설치</p> <p>(1) Ubuntu 20.04 LTS Desktop image(64-bit)</p> <p>(2) Boot USB Tool Download</p> <p>(3) UCB에 Ubuntu 20.04 LTS 설치</p> | | |
| 참조 |  | |  |
| | Remote PC에 Keyboard, mouse, monitor 연결 후 전원 ON -> Ubuntu 설치 확인 / TP - Link 준비 | | roscore 명령어로 실행 noetic 버전 확인 |
| |  | |  |
| | IP 주소 확인 /원격 데스크톱의 기본 포트 3389 확인 | | ROS Turtlebot 이름 변경 환경변수 추가 |
| |  | |  |
| | Window PC에서 mstsc 실행 / 유무선 통신 네트워크가 달라 연결 거부 | | terminator → roscore → node → key → rgt_graph |

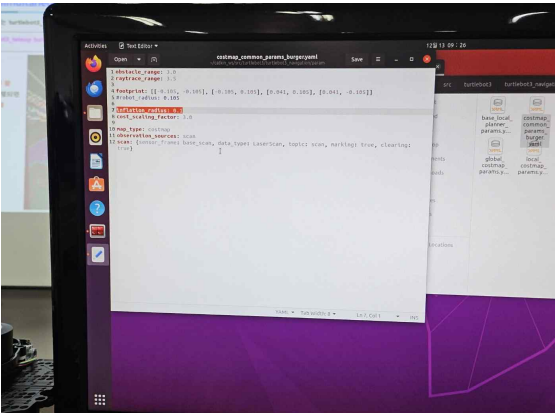
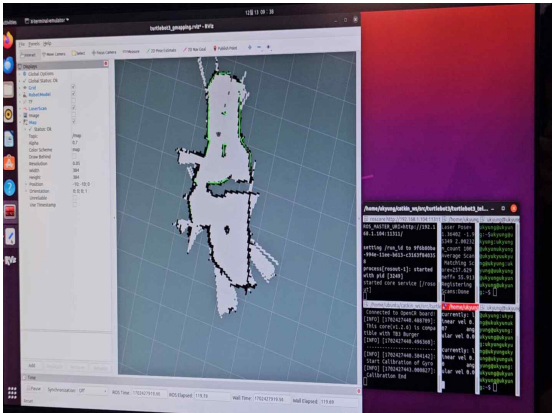
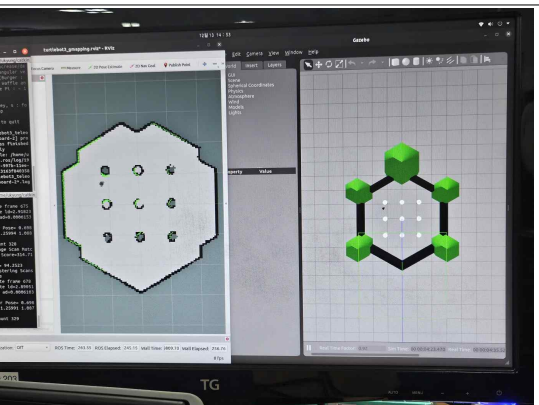
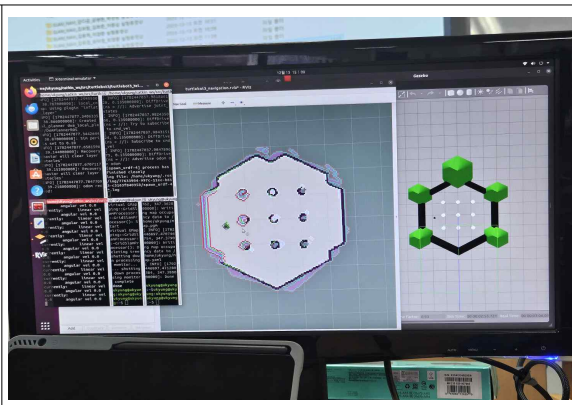
| | | | |
|----|--|--|---------------|
| 날짜 | 2023.12.06. | 조원 | 김유경, 최아영, 최영중 |
| 내용 | | | |
| 요약 | <p>● 목표 ROS 개발환경 구축 - TurtleBot3 burger 조립하기</p> <p>●과정 (1) TurtleBot3 burger 조립하기 (2) MicroSD에 Raspberry Pi OS(Raspbian OS), ROS 굽기 (3) Install ROS SBC(Single Board Computer) - Raspberry Pi 4B 작업하기</p> | | |
| 참조 |  |  | |
| | TurtleBot3 burger 조립 재료 준비 | 1층~3층 조립 | |
| |  |  | |
| | LiDAR까지 조립 완료 | MicroSD에 Raspberry Pi OS(Raspbian OS), ROS 굽기 | |
| |  |  | |
| | Raspberry Pi 4B에 tb3_rpi4+_noetic_20220224.img가 구워진 MicroSD 카드 삽입 후 전원 ON -> 설정한 Ubuntu4 IP 주소 확인 | 원격 제어 연결 / 연결된 IP 확인 | |

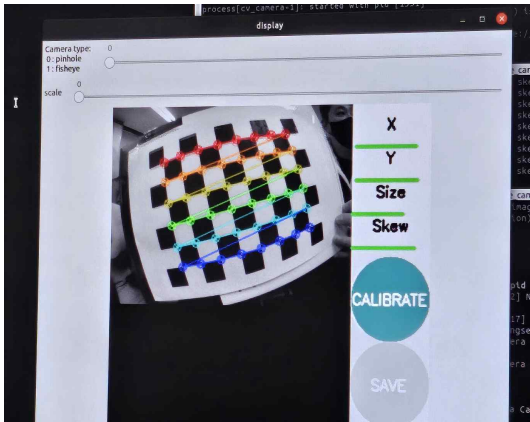

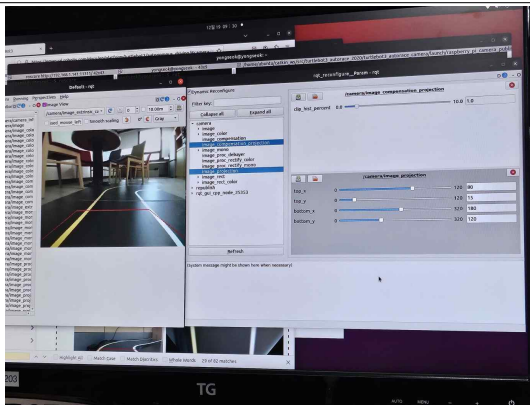
| | | | |
|----|-------------|----|---------------|
| 날짜 | 2023.12.11. | 조원 | 김유경, 최아영, 최영중 |
|----|-------------|----|---------------|

| 내용 | | | |
|----|---|--|---|
| 요약 | <p>● 목표</p> <p>ROS 개발환경 구축 - OpenCV 설정, turtlesim과 turtlebot을 사용하여 ROS 명령어 사용</p> <p>● ROS 명령어</p> <ul style="list-style-type: none"> - ROS shell 명령어 <p>roscd: ROS 패키지의 디렉터리로 이동 rosls: ROS 패키지의 파일 목록 확인</p> <ul style="list-style-type: none"> - ROS 실행 명령어 <p>roscore: master(ROS 네임 서비스) + rosout(로그 기록) + parameter server(파라미터 관리) roslaunch: 노드를 여러 개 실행 rosclean: ROS 로그 파일을 검사하거나 삭제</p> <ul style="list-style-type: none"> - ROS 정보 명령어 <p>rostopic: 토픽 정보 확인 rosservice: 서비스 정보 확인 roswatch: 노드 정보 확인 rostopic: 파라미터 정보 확인, 수정 rostopic: 메시지 기록 rostopic: 메시지 정보 확인 rostopic: 서비스 정보 확인 rostopic: ROS release version 확인</p> <ul style="list-style-type: none"> - ROS catkin 명령어 <p>catkin_create_pkg: 패키지 자동생성 catkin_init_workspace: 폴더 초기화</p> <p>* ros turtlesim 이용하여 node, topic, service 실습하기 위한 사전단계</p> <p>[new terminal] roscore [new terminal] roslaunch turtlesim turtlesim_node [new terminal] roslaunch turtlesim turtle_teleop_key [new terminal] rqt_graph</p> | | |
| | 참조 |   | <p>rqt를 이용하여 선속도와 각속도 좌표를 확인</p> <p>turtlesim 노드의 배경색 관련 파라미터 blue = 0으로 설정</p> |

| | | | |
|----|---|--|---------------|
| 날짜 | 2023.12.12 | 조원 | 김유경, 최아영, 최영중 |
| 내용 | | | |
| 요약 | <p>● 목표</p> <p>ROS turtlebot으로 SLAM과 Navigation 실습</p> <p>● SLAM & Navigation</p> <p>LIDAR를 이용, 내 위치를 실시간으로 파악해서 장애물 맵을 그리는 것</p> <p>LIDAR는 빛을 통하여 감지 및 측정한다</p> <p>Map 완성 시 map.pgm 파일과 map.yaml 파일으로 저장한다.</p> <p>● 추측 항법(dead reckoning)</p> <p>양 바퀴 축의 회전 값을 이용(주행기록계 odometer)</p> <p>이동 거리와 회전 값을 계산, 위치 측정</p> <p>※ 실습 시 주의할 점</p> <p>배터리 충전을 60% 이상으로 유지한다.</p> <p>Navigation과 SLAM은 동시에 동작시키지 않는다.</p> | | |
| | | | |
| 참조 |  | | |
| | key와 Rviz를 이용하여 외곽을 한 바퀴 돌면서 Map을 구현 | rqt graph를 실행하여 node간 통신 data 확인 | |
| |  | | |
| | SLAM으로 그린 Map을 불러와서 Navigation Map과 일치시킨다. |  | |
| | | 2D Nav Goal 버튼을 사용하여 목표하는 곳으로 Navigation 동작 | |

| | | | |
|----|------------|----|---------------|
| 날짜 | 2023.12.13 | 조원 | 김유경, 최아영, 최영중 |
|----|------------|----|---------------|

| 내용 | |
|----|---|
| 요약 | <ul style="list-style-type: none"> ● 목표 ROS turtlebot으로 SLAM과 Navigation 실습 SLAM 이론 학습 ● SLAM & Navigation 실습 팽창 영역(inflation radius) 값 변경 - 1에서 0.1로 작게 설정 후 실행 ● SLAM 위치 추정 알고리즘 <ul style="list-style-type: none"> - 추측 항법(dead reckoning) - Kalman filter: Noise가 포함된 선형 시스템에서 object의 상태를 추적하는 재귀 필터 - Particle Filter: 많은 측정을 통해 가능성 높은 입자를 특징함. 비선형 시스템에서 유용하게 사용 ● Gazebo Simulation <ul style="list-style-type: none"> - Gazebo 는 3차원 시뮬레이터로 물리 엔진과 그래픽 효과 사용 - Real turtlebot3 사용이 아닌 가상의 turtlebot을 이용하여 맵핑 또는 네비게이션 사용할 수 있다 |
| 참조 | <div>   </div> <p>inflation radius 값 0.1로 변경 -> SLAM으로 그린 map이 더 정교해짐</p> <div>   </div> <p>Gazebo를 이용한 터틀봇 시뮬레이션 1 (key)</p> <p>Gazebo를 이용한 터틀봇 시뮬레이션 2 (Navigate)</p> |

| | | | |
|----|---|----|--|
| 날짜 | 2023.12.18 | 조원 | 김유경, 최아영, 최영중 |
| 내용 | | | |
| 요약 | <ul style="list-style-type: none"> ● 목표 ROS turtlebot으로 line tracing 실습 ● RPi Camera(g) - Fisheye lens 모든 버전의 Pi에 지원 시야각 160도 ● 실습 과정 install Autrace Packages 내부 경통 보정(Intrinsic Camera Calibration) 외부 경통 보정(Extrinsic Camera Calibration) ● 카메라 calibration <ul style="list-style-type: none"> - 영상 촬영할 때, 영상 품질은 카메라의 위치 및 방향 등 외부적(extrinsic)요인에 의해 결정되나, 실제 이미지는 사용된 렌즈, 렌즈와 이미지 센서와의 거리, 렌즈와 이미지 센서가 이루는 각 등 카메라 내부(intrinsic) 기구 조립 요소에 의해 영향받음 - 내부 경통 보정(Intrinsic Camera Calibration): 초점거리, 주점 등 카메라 렌즈의 광학적 요소를 보정 - 외부 경통 보정(Extrinsic Camera Calibration): 카메라의 설치 높이, 방향 등 카메라와 외부공간과의 기하학적 관계에 의한 카메라 보정 | | |
| 참조 |  | |  |
| | Intrinsic Camera Calibration 를 위해 checkboard 사용 | | image_projection |
| |  | |  |
| | rqt_reconfigure 값 조정 | | rqt 3개 image 실행 배치 화면 |

| | | | |
|----|--|----|---------------|
| 날짜 | 2023.12.19 | 조원 | 김유경, 최아영, 최영중 |
| 내용 | | | |
| 요약 | <p>● 목표</p> <p>ROS turtlebot으로 line tracing 실습</p> | | |
| 참조 | <div data-bbox="296 360 1430 822" data-label="Image"> <p>The screenshot shows the ROS rqt monitor interface. On the left, a list of topics is displayed, including /battery/state, /camera/camera_info, /camera/image, /camera/image_color, /camera/image_color/compressed, /camera/image_color/compressed/parameter_descriptions, /camera/image_color/compressed/parameter_updates, /camera/image_color/compressedDepth, /camera/image_color/compressedDepth/parameter_descriptions, /camera/image_color/compressedDepth/parameter_updates, /camera/image_color/theora, /camera/image_color/theora/parameter_descriptions, /camera/image_color/theora/parameter_updates, /camera/image_compensated, /camera/image_mono, /camera/image_mono/compressed, /camera/image_mono/compressed/parameter_descriptions, /camera/image_mono/compressed/parameter_updates, /camera/image_mono/compressedDepth, /camera/image_mono/compressedDepth/parameter_descriptions, /camera/image_mono/compressedDepth/parameter_updates, /camera/image_mono/theora, /camera/image_mono/theora/parameter_descriptions, /camera/image_mono/theora/parameter_updates, /camera/image_proc_debayer/parameter_descriptions, /camera/image_proc_debayer/parameter_updates, /camera/image_proc_rectify_color/parameter_descriptions, /camera/image_proc_rectify_color/parameter_updates, /camera/image_proc_rectify_mono/parameter_descriptions, /camera/image_proc_rectify_mono/parameter_updates, /camera/image_projected_compensated, /camera/image_projected/compressed, /camera/image_rect, /camera/image_rect_color, /camera/image_rect_color/compressed, /camera/image_rect_color/compressed/parameter_descriptions, and /camera/image_rect_color/compressed/parameter_updates. On the right, two camera feeds are shown: /detect/image_lane/compressed and /camera/image_projected_compensated. The /detect/image_lane/compressed feed shows a green line on a dark background, and the /camera/image_projected_compensated feed shows a yellow line on a dark background.</p> </div> <p>Turtlebot3 주행 - rqt 모니터링</p> <div data-bbox="296 925 834 1397" data-label="Image"> <p>A photograph of a Turtlebot3 robot on a black track with yellow and white lines. A red fire extinguisher is visible on the left side of the track.</p> </div> <div data-bbox="869 918 1417 1397" data-label="Image"> <p>A photograph of a Turtlebot3 robot on a black track with yellow and white lines. A red fire extinguisher is visible on the left side of the track.</p> </div> <div data-bbox="296 1429 834 1960" data-label="Image"> <p>A photograph of a Turtlebot3 robot on a black track with yellow and white lines. A red fire extinguisher is visible on the right side of the track.</p> </div> <div data-bbox="855 1429 1422 1960" data-label="Image"> <p>A photograph of a Turtlebot3 robot on a black track with yellow and white lines. A red fire extinguisher is visible on the right side of the track.</p> </div> <p>Turtlebot3 주행 실습</p> | | |