

ROS_3일차_보고서_20기_인턴_2025402040_컴퓨터정보공학부_노기문

과제2: hw1,3한번에 띄우기

사이트 요약

1. Launch-Main (“Launch”)

Launch 파일은 여러 노드(executable)를 동시에 시작하고 설정할 수 있게 해 줌.

주요 역할들:

Creating a launch file — 시스템 구성 → 노드, 인자, 리매핑(remappings), 네임스페이스(namespace) 등을 정의하는 파일 작성.

Launching & monitoring multiple nodes — 여러 노드를 띄우고 상태 추적.

Using substitutions — 반복적 사용 가능하게 파라미터나 경로 같은 값 치환 가능하게 함.

Using event handlers — 실행 중 이벤트 감지해서 동적 변화 대응 가능.

Managing large projects — 프로젝트가 커질 경우 launch 파일 구조, arguments, 기본 설정(defaults), remapping, 네임스페이스, config 파일(YAML, RViz 설정 등) 관리에 관한 규칙과 모범 예제 제공됨.

2. Creating Launch Files

Launch 파일 형식: XML, YAML, Python 중 하나 선택 가능.

예: turtlesim 패키지를 사용해서

두 개의 turtlesim_node 노드 실행 (다른 namespace)

mimic 노드 실행해서 한 노드의 pose를 받아 다른 노드를 조종함 (topic remapping)

구성 요소:

node 태그 또는 action 으로 패키지(pkg), 실행파일(exec or executable), 이름(name), namespace, remap 등이 들어감.

Python 버전은 generate_launch_description() 함수로 LaunchDescription 객체 반환하고 Node 액션 사용.

실행 방식:

launch 파일 직접 실행 (2 launch <file>)

패키지 내의 launch 파일로 실행 (2 launch <패키지명> <launch파일명>)

3. Launch-System (“Integrating launch files into 2 packages”)

목표: standalone launch 파일을 만든 뒤, 이를 2 패키지 내부에 포함시키고 배포 가능하게 구조 만들기.

절차:

새 패키지 만들기 (Ros2 pkg create)

패키지 안에 launch/ 디렉토리 생성하고 그곳에 launch 파일 위치시킴.

Python 패키지라면 setup.py의 data_files에 launch 디렉토리 포함시키기.

C++ 패키지라면 CMakeLists.txt에 install 규칙 추가해서 share/<프로젝트명>/launch로 복사 설치되게 함.

빌드(colcon build)하고 소스 환경 활성화(source). 그 후 2 launch <패키지명> <launch파일>로 실행 가능.

먼저 런치 파일의 두 파일인 hw_01_04_bringup.xml과 hw_all.launch.py를 살펴 보겠습니다.

첫 번째 hw_01_04_bringup.xml은 사이트에 나온대로

1 다른 패키지(hw_01)의 launch 파일을 불러와 실행. 여기서는 hw01_turtlesim.launch.py.

2 hw_04 패키지의 hw_04_node 실행.

name="robot_arm": 노드 이름 지정, output="screen": 로그를 터미널에 출력.

<param from=...>: YAML 설정 파일(arm_params.yaml)을 파라미터로 로드.

즉, hw_01의 turtlesim 관련 launch를 불러오고, hw_04의 로봇팔 노드를 파라미터와 함께 실행합니다. 하지만

python방식을 적용하여 더욱더 효율적으로 사용을 하였습니다. => 이부분을 AI 사용하였습니다

그리하여

ament_index_python.get_package_share_directory: 패키지의 share 경로 가져오기. 여기서 arm_params.yaml 경로를 얻고

LaunchDescription: 실행할 액션(Node 등)들을 리스트로 담은 뒤 Node 실행합니다.

turtlesim 패키지의 turtlesim_node 실행, 이름 sim.

hw_01 패키지의 hw_01_node 실행, 이름 drawer, 로그 출력.

hw_04 패키지의 hw_04_node 실행, 이름 robot_arm, YAML 파라미터 로드, 로그 출력.

즉, turtlesim 시뮬레이터 + hw_01 노드 + hw_04 로봇팔 노드를 동시에 실행하도록 정하여 런치 파일을 제작했습니다.

보고서를 끝까지 다 못쫓습니다. 죄송합니;다.