

ROS 2 기반 로봇암 제어 및 통신 시스템 구축

(개인 프로젝트)

1. 프로젝트 개요

본 프로젝트는 Ubuntu Linux 환경에서 ROS 2(Robot Operating System 2)를 활용하여 로봇암을 제어하고, 상태 데이터를 실시간으로 송수신하는 비동기 분산 통신 시스템을 구축하는 것을 목표로 진행했습니다.

단순한 제어 코드 구현을 넘어,
소프트웨어가 3D 공간상의 물리적 하드웨어(로봇암)를 어떻게 인식하고 제어하는지에 대한 **기구학적 원리 이해**와,
ROS 2의 **노드 기반 메시징 아키텍처 설계**에 중점을 두었습니다.

프로젝트 전 과정(워크스페이스 구축, 패키지 생성, Python 스크립트 작성, 의존성 오류 해결 및 디버깅)은 개인 주도로 수행하였습니다.

2. 문제 정의

하드웨어와 소프트웨어가 연동되는 시스템에서는 다음과 같은 기술적 과제가 존재합니다.

- 다수의 센서 및 액추에이터 데이터를 지연 없이 동시 처리해야 함
- 물리적 3D 좌표계를 수학적 연산(행렬, 벡터 등)으로 변환해야 함
- 패키지 의존성, 빌드 설정, 통신 환경 구성 오류가 빈번하게 발생함

핵심 문제

“ROS 2의 분산 통신 환경에서 로봇암 제어를 위한 안정적인 Publish/Subscribe 파이프라인

인을 어떻게 설계하고, 실행 오류를 체계적으로 디버깅할 것인가?”

3. 개발 환경 및 제어 대상

구분	내용
운영체제	Ubuntu (Linux 환경)
프레임워크	ROS 2
개발 언어	Python
타겟 시스템	다관절 로봇암 (3D 좌표 기반 제어)
개발 패키지	my_ros2_assignment

4. 해결 접근 방식

4-1. 통신 아키텍처 설계

- 시스템을 기능 단위별 독립 노드(Node)로 분리
- 목표 관절 각도를 전송하는 **Publisher 노드** 구현
- 로봇 상태 데이터를 수신 및 모니터링하는 **Subscriber 노드** 구현
- Topic 기반 메시지 흐름 설계 및 통신 구조 명확화

이를 통해 ROS 2의 분산 메시징 구조를 실제 환경에 적용했습니다.

4-2. 제어 로직 구현

- Python 기반 로봇 제어 스크립트 작성
- 3D 공간 좌표 변환 로직 구현
- 기구학 개념을 적용하여 목표 위치까지 관절 각도 계산
- 일정 주기로 제어 명령을 송신하도록 Rate 제어 적용

5. 문제 발생 및 해결 과정

문제 1. 패키지 가시성 및 실행 오류

문제 상황

ros2 run 명령어 실행 시, 생성한 my_ros2_assignment 패키지 또는 실행 파일을 인식하지 못하는 오류 발생.

해결 과정

- setup.py 및 package.xml 내 Entry Points 및 의존성 선언 점검
- 실행 파일 권한 부여 (chmod +x)
- 워크스페이스 루트에서 colcon build 재빌드
- source install/setup.bash로 환경 변수 재적용

결과

OS가 패키지를 정상 인식하고 노드 실행 성공.

문제 2. 노드 간 메시지 통신 확인 불가

문제 상황

Publisher는 실행되었으나, 로봇 동작이 없거나 상태 데이터가 수신되지 않는 통신 병목 발생.

해결 과정

- ros2 node list로 실행 노드 확인
- ros2 topic list로 토픽 생성 여부 점검
- ros2 topic echo로 메시지 흐름 실시간 확인
- 메시지 타입 불일치 및 Publish 주기(Rate) 문제 발견 후 수정

결과

자연 없는 양방향 통신 구조 확보 및 정상 제어 성공.

6. 주요 성과

- ROS 2 기반 Publish/Subscribe 아키텍처에 대한 실질적 이해 확보
- 3D 좌표 제어 및 하드웨어 연동 소프트웨어 개발 경험 축적
- Linux CLI 기반 빌드·환경설정·디버깅 역량 강화
- 분산 통신 시스템에서의 오류 추적 및 문제 해결 능력 향상

👉 향후 스마트팩토리, 머신비전, 로봇 제어 시스템 개발 시 발생하는 통신 및 환경 설정 이슈를 해결할 수 있는 기반 역량 확보

7. 사용 기술 스택

- **OS:** Ubuntu Linux
- **Language:** Python
- **Framework:** ROS 2
- **Core Concepts:**
Node, Topic, Message, Publish/Subscribe, Kinematics, Colcon Build, CLI Debugging

8. GitHub Repository

https://github.com/rjaekawpxm1-netizen/portfolio/tree/main/Projects/ros2_ws/src/my_ros2_assignment