

ROS2 를 활용한 모터와 이더넷 통신 및 제어

김동준¹, 박성준¹, 위진혁¹, 도영수², 전재욱³

¹성균관대학교 기계공학부 학부생

²성균관대학교 정보통신대학 박사과정

³성균관대학교 정보통신대학 교수

jimmy98629@g.skku.edu, sj2park@skku.edu, wjins99@g.skku.edu

Motor Control via Ethernet Communication and ROS2

Dong-Jun Kim¹, Sung-Jun Park², Jin-Hyeok Wee³, Young-Soo Do⁴, Jae-Wook Jeon⁵

^{1, 2, 3}School of Mechanical Engineering, Sungkyunkwan University

^{4, 5}College of Information and Communication Engineering, Sungkyunkwan University

요 약

로봇 운영 체제인 Robot Operating System (ROS)은 다양한 로봇 프로젝트와 연구에서 광범위하게 활용되며 국제적인 활발한 커뮤니티가 형성되어 있다. 많은 로봇 기능들이 ROS 를 활용하여 개발되어왔고, 이 중에서도 Fastech 사의 Ezi-SERVO II PLUS E motor driver 를 ROS 환경에서 사용할 수 있도록 개발 작업이 진행되었다. ROS 는 Linux 기반이므로, 개발 환경으로는 초보자도 다루기 쉬우면서 비교적 저렴한 소형 컴퓨터인 Raspberry Pi 를 선택하였다. 또, Raspberry Pi 는 Linux 기반의 작은 컴퓨터로, 다양한 개별 프로젝트를 수행하기 위해 많은 사람들이 활용하고 있다. 이로 인해 Raspberry Pi 로 소규모 프로젝트를 진행하는 개발자들도 해당 모터 드라이버를 Raspberry Pi 와 ROS 를 통해 쉽게 사용할 수 있게 되었다.

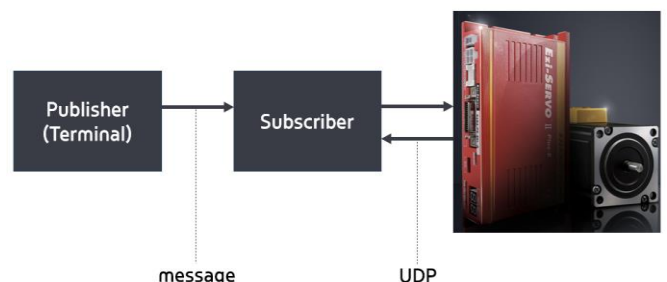
1. 서론

로봇과 모빌리티 분야에 익히 알려진 오픈소스 소프트웨어인, Robot Operating System(ROS)은 2006 년 ‘윌로우 개라지’ (Willow Garage)가 설립하였고, 수많은 로봇 기업이 만들어지고 지금의 로봇 기술이 발전할 수 있는 밑거름이 됐다[1]. 로보틱스 분야에서 ROS 의 여러 문제들이 해결된 ROS2 를 광범위하게 활용하는 추세이다. ROS2 에서 하드웨어와 상호작용하기 위해서는 소프트웨어 모듈인 ROS2 용 driver 가 필요하다. ROS2 를 통해 Fastech 사의 Ezi-SERVO II PLUS E motor driver 를 사용할 수 있도록 ROS2 용 driver 에 대한 연구를 진행하였다.

본 보고서는 ROS2 를 활용하여 Raspberry Pi 와 Ezi-SERVO II PLUS E motor driver 간의 Ethernet User Datagram Protocol(UDP) 통신을 구현한 연구 결과를 정리한 것이다. 이 연구의 주요 목표는 ROS2 에서 custom message 를 활용하여 Linux 환경의 Raspberry Pi 와 Motor Driver 와의 Ethernet UDP 통신을 성공적으로 구현하는 것이다. 개발 환경은 Raspberry Pi 4(RAM 8GB), Ubuntu 22.04 LTS 와 ROS2 Humble 이다.

2. 개발

Linux 환경에서 Motor Driver 와의 Ethernet UDP 통신을 구현하기 위해 ROS2 의 custom message 와 topic 를 활용하였다. ROS2 의 실행 단위인 node 들은 topic 을 통해 통신을 하는데, 이는 message 의 형식을 가진다[2]. 파스텍 회사의 모터 manual 을 참고하여 일부 기능들을 ROS2 topic 과 message 로 custom 구현하였고, 구현한 각 topic 들의 Subscriber 를 하나의 노드에 구현하였다. 각 subscriber 의 callback 함수에서는 해당 토픽으로부터 받은 메시지를 기반으로 Motor Driver 에 제어 명령을 전송하도록 하였다. Package 의 Blockdiagram 은 다음 그림 1 과 같다.

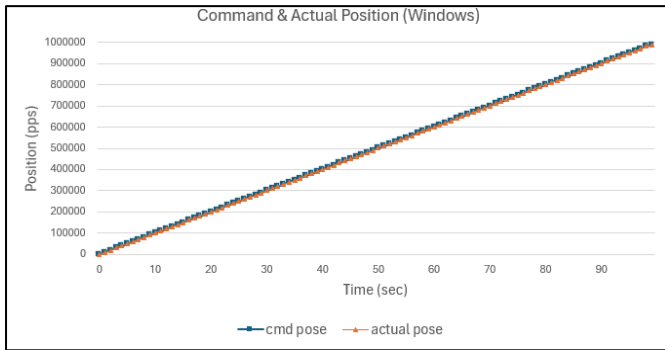


(그림 1) Block Diagram of the Package

기존 Fastech 회사에서 구현해 놓은 약 120 여개의 기능[3] 중에서 Motor 제어를 눈으로 확인하기 용이한 기능 10 개를, 우선적으로 Subscriber.cpp 파일에 구현하였다. 그림 1 의 Blockdiagram 처럼 구동이 되는 것을 확인 후, 기능 개선을 위해 Subscriber node 의 코드를 Class 로 모듈화하여 subscriber_class.cpp 파일에 구현하였다. 추가적으로, 모터 동작 중에 모터의 상태를 한눈에 살펴볼 수 있는 함수도 구현하였다.

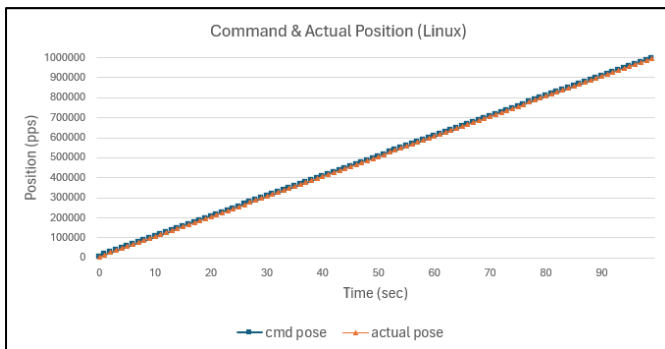
3. 검증

Linux 상에서 만든 프로그램을 검증하고자 motor 상태를 출력하는 함수를 채택하였다. Windows 와 Linux, 각 Operating System(OS)에서 motor 를 등속도로 100 바퀴를 돌아가게끔 한 후 1 초마다 motor 의 상태(모터의 실제 위치, 이상 위치)를 Comma Separated Value(CSV) 파일에 출력하여 motor 의 실제 위치와 이상 위치의 차이를 그래프로 시각화하여 관찰하였다.



(그림 2) Graph of Command and Actual Position (Windows)

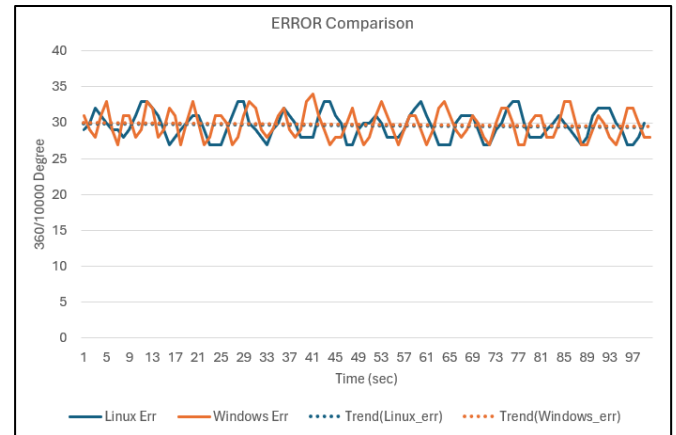
그림 2 는 Windows 상에서 기존 회사의 프로그램을 통해 Motor Driver 와 통신하여, Motor 를 구동 후 Motor 의 상태를 가져오는 함수를 호출한 결과를 나타낸다. 시작과 구동 과정에서 작은 오차는 존재하지만 100 바퀴 회전 후 오차가 0 으로 줄어드는 것을 볼 수 있다.



(그림 3) Graph of Command and Actual Position (Linux)

그림 3 은 Linux 상에서 ROS2 의 publisher 와 subscriber 를 통해 Motor Driver 에 명령을 전달하고, Motor Driver 를 통해 Motor 를 구동한 결과를 나타낸다.

Windows 와 마찬가지로 시작과 구동 과정에서 작은 오차는 존재하지만 100 바퀴 회전 후 오차가 0 으로 줄어드는 것을 볼 수 있다.



(그림 4) Error Comparison

그림 4 는 Windows 환경에서 motor driver 와 통신한 결과와 Linux 환경에서 motor driver 와 통신한 결과, 각 OS 에서 실제 위치와 이상 위치의 차를 그래프로 나타내고 각 결과의 추세선을 나타낸다. 그 결과, 두 추세선이 거의 일치하는 것을 확인할 수 있다.

4. 결론

본 연구에서는 ROS2 를 활용하여 Raspberry Pi 와 Motor Driver 와의 Ethernet UDP 통신을 성공적으로 구현함으로써, 다양한 응용 분야에서 활용 가능한 플랫폼을 구축하였다. 이를 통해 ROS2 및 하드웨어 간의 통신 및 제어에 대한 더 나은 방법을 모색하고, 안정성과 신뢰성을 높이는 데 기여하고자 한다. 앞으로의 연구에서는 연구 결과를 더 발전시켜 보다 정교한 시스템을 구축하고 다양한 응용 분야에서의 활용 가능성을 탐구할 계획이다.

ACKNOWLEDGEMENT

이 논문은 정부(교육부-산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임(P0022098, 2023 년 미래형자동차 기술융합 혁신인재양성사업)

참고문헌

- [1] 김동원, “로봇 운영체제 ‘ROS’의 진화, 농사짓는 ‘항공 모함’을 만든다”, <디지털조선일보>, 2023 년 8 월 4 일자.
- [2] 표윤석, 임태훈, "ROS2 로 시작하는 로봇 프로그래밍", 부천시, 루비페이퍼, 2023
- [3] ㈜ 파스텍, "Ezi-SERVO II Plus-E 사용자 매뉴얼 통신 기능편", 부천시, ㈜ 파스텍, 2021