

리눅스 임베디드 시스템상에서의 모션 컨트롤 시스템 구현

신민호¹, 성우진¹, 김시연², 오성빈³, 전재욱⁴

¹성균관대학교 전자전기공학부 학부생

²성균관대학교 기계공학부 학부생

³성균관대학교 전자전기컴퓨터공학과 박사과정

⁴성균관대학교 전자전기컴퓨터공학과 교수

smh9800@g.skku.edu, dnwls2855@g.skku.edu, siyeon1251@g.skku.edu

An Implementation of Motion Control System for Linux Embedded Systems

Min-Ho Shin¹, Woo-Jin Sung¹, Si-Yeon Kim², Sung Bhin Oh¹, Jae Wook Jeon¹

¹Dept. of Electronic and Electrical Engineering, Sungkyunkwan University

²Dept. of Mechanical Engineering, Sungkyunkwan University

요 약

FASTECH 社의 Ezi-Servo Plus-E 제품군은 Windows 기반의 모션 라이브러리와 GUI 프로그램을 이용하여 모션 컨트롤을 수행한다. 이를 Linux기반의 GUI 프로그램과 모션 라이브러리를 작성하여 Raspberry Pi 환경에 적용하였다. Windows와 Linux 기반의 두 프로그램으로 각각 모터를 구동시키고 모터의 위치값을 비교하여 150 미만의 작은 차이를 확인하였다. 이 연구를 통해 Linux 임베디드 시스템으로 모션 컨트롤 시스템을 구축하려는 잠재 고객의 수요를 맞출 수 있을 것으로 기대한다.

1. 서론

Windows는 PC 운영체제 점유율 대부분을 차지하지만, Linux도 다양한 용도로 증가하는 추세이다. Windows와 Linux는 서로 다른 개발 환경과 기능에 대한 다른 라이브러리 및 API를 제공하기도 한다. 따라서 기존에 컴파일된 프로그램(.exe, .desktop, .dll 등)은 다른 환경에서 바로 실행할 수 없다.

FASTECH 社의 Ezi-SERVO Plus-E 제품군은 Ethernet을 통해 최대 254축을 구동할 수 있는 스테핑 모터 제어 시스템으로 Windows기반 프로그램을 제공한다. 따라서 Linux 환경에서 사용하지 못했다.

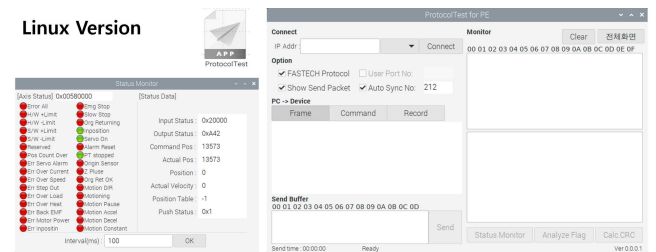
본 논문에서는 공통 소프트웨어 API 및 하드웨어 인터페이스를 제공하는 SBC (Single-board Computer)인 Raspberry Pi 환경에서 모션 컨트롤 시스템을 제어하는 Linux 기반 프로그램을 설계 및 구현한다. 그리고 기존 Windows 기반 프로그램과 성능을 비교 분석한다.

2. 프로그램의 개발

FASTECH 社에서는 Windows 기반의 모션 라이브러리(DLL)와 여러 GUI 프로그램을 제공한다.[2]

이 중에서 테스트와 모니터링을 할 수 있는 ProtocolTest 프로그램과 모션 라이브러리 일부를

Raspberry Pi OS를 설치한 Raspberry Pi 4 Model B 8GB, Raspberry Pi 400 모델에서 구현하였다.



(그림 1) Linux 환경으로 구현한 프로그램

GUI 구현을 위해서 GTK(GIMP Toolkit)와 Glade 개발환경을 활용해 C언어로 프로그램을 구현하였다. 구현한 프로그램을 사용하여 Ethernet 통신(UDP/TCP)을 통해 FASTECH社의 모터 드라이버 제어 신호를 주고 받을 수 있다.

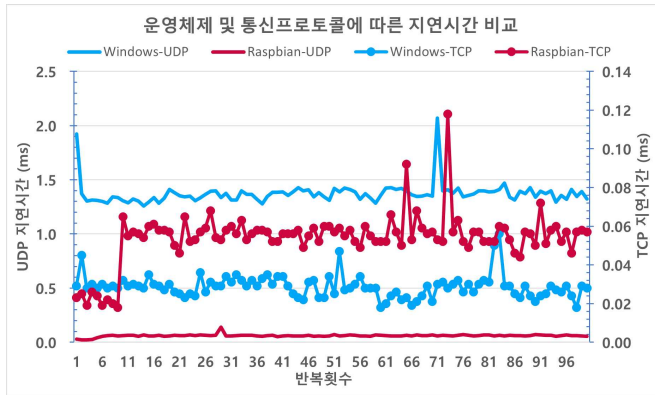
개발된 프로그램과 테스트를 위해 사용한 소스 코드는 첨부된 github repo 링크를 참고할 수 있다.[1]

2.1 시스템 분석

FASTECH 社 모터 드라이버와의 UDP/TCP 통신 지연시간 차이를 Windows 환경과 Raspberry Pi 환경에서 확인하였다.

Windows 상에서는 기본 제공되는 "winsock2.h"와

Linux 상에서는 “arpa/inet.h”의 헤더를 사용하여 C 언어로 FASTECH 社 통신기능 매뉴얼[2]에 따라 현재 정보를 모두 받아오는 Frame(TYPE : 0x43)을 전송하고 수신될 때까지의 시간을 기록하였다.



(그림 2) 통신 지연시간 비교

TCP에서 UDP보다 두 운영체제 반응속도가 더 빠른 특성을 보이므로 FASTECH 社의 모터 드라이버와 통신의 특성으로 보인다. 그리고 UDP의 경우 Raspbian OS 환경에서 더 빠르고 예서는 Windows 환경에서 더 빠른 결과를 얻을 수 있었다. 이 정도의 미묘한 오차는 제어 신호를 전달하였을 때 큰 성능의 차이를 보이지 않을 것으로 보인다. 개발한 프로그램으로 Raspbian Pi를 제어기로 사용했을 때도 기능에 문제가 없었다.

3. 프로그램 검증

| Position Table | | | | | | | | | | |
|----------------|-----|----------|---------|----------|-------|-------|-----------|------------|-------------|--------------|
| No. | CMD | Position | Low Spd | High Spd | Accel | Decel | Wait Time | Continuous | Check Inpos | JP Table No. |
| 0 | | | | | | | | | | |
| 1 | 3 | 89636 | 1 | 10000 | 100 | 100 | 1000 | | | 2 |
| 2 | 3 | 0 | 1 | 10000 | 100 | 100 | 1000 | | | 3 |
| 3 | 3 | 25000 | 1 | 50000 | 100 | 100 | 1000 | | | 4 |
| 4 | 3 | 0 | 1 | 50000 | 100 | 100 | 1000 | | | 5 |
| 5 | 3 | 2500 | 1 | 100000 | 100 | 100 | 1000 | | | 6 |
| 6 | 3 | 5000 | 1 | 100000 | 100 | 100 | 1000 | | | 7 |
| 7 | 3 | 7500 | 1 | 100000 | 100 | 100 | 1000 | | | 8 |
| 8 | 3 | 10000 | 1 | 100000 | 100 | 100 | 1000 | | | 9 |
| 9 | 3 | 12500 | 1 | 100000 | 100 | 100 | 1000 | | | 10 |
| 10 | 3 | 15000 | 1 | 100000 | 100 | 100 | 1000 | | | 11 |
| 11 | 3 | 17500 | 1 | 100000 | 100 | 100 | 1000 | | | 12 |
| 12 | 3 | 20000 | 1 | 100000 | 100 | 100 | 1000 | | | 13 |
| 13 | 3 | 22500 | 1 | 100000 | 100 | 100 | 1000 | | | 14 |
| 14 | 3 | 25000 | 1 | 100000 | 100 | 100 | 1000 | | | 15 |

(그림 3) Position Table, 이하 ‘PT’라고 한다

구현한 프로그램을 검증하기 위해 FASTECH 社의 Ezi-SERVOII Plus-E 모션 컨트롤 시스템을 사용하였다. Raspbian Pi OS 환경의 프로그램을 같은 PT를 이용해 (그림 3)에 제시된 parameter에 따라 모터를 구동하였다.

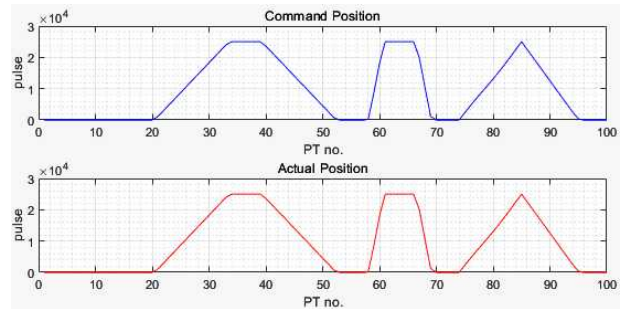
그리고 MATLAB에서 C언어 프로그램과 연동하여 모터의 위치 데이터를 받고 그래프로 도식화했다.

3.1 결과

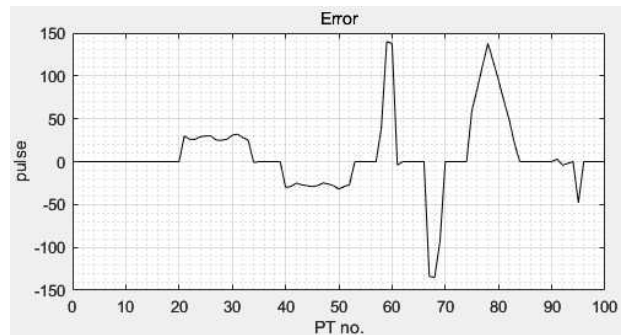
(그림 4)은 PT를 한번 동작하였을 때 모터의 위치값(0 ~ 25000)을 각각 나타낸 것이며 파란 선은 명

령의 위치값, 빨간 선은 Raspbian Pi OS 환경에서 구동시켰을 때의 위치값을 나타낸 것이다.

(그림 5)는 모터를 구동시켜보았을 때의 모터 위치의 차이값을 나타낸 것이다. 위치값은 최대 25000이다. 이때 차이는 150 이내의 미소한 차이를 보이며 기존 프로그램의 오차와 비슷하게 제어가 되었다.



(그림 4) 모터의 명령 위치값과 실제 위치값



(그림 5) 위치값 차이

4. 결론

본 논문에서는 Raspbian Pi가 Ethernet 통신을 통한 모션 컨트롤 제어기로 사용하는 데에 대한 적합성을 검증하였고, Linux 기반의 모션 컨트롤 제어 프로그램을 구현하였다. 이를 검증하기 위해 PT를 사용하여 제어해보고 위치값을 확인하였다. 위치값은 최대 25000일 때, 차이는 150 이내로 큰 오차가 존재하지 않음을 확인하였다.

ACKNOWLEDGMENT

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

참고문헌

- [1] SKKUFastech. “https://github.com/SKKUFastech/Co-op” Github, 11 Aug 2023,
- [2] 파스텍.(2021). 210129_Manual_Ezi-SERVOII Plus-E 통신 기능편_KOR, 부천