

실시간 이더넷 기반 스텝핑 모터 드라이브 개발

김진호, 하경재*
경남대학교 컴퓨터공학부 교수

Design of a Closed-Loop Stepping Motor Drive based on Real-Time Ethernet

Jin-Ho Kim, Kyung-Jae Ha*
Professor, Division of Computer Engineering, Kyungnam University

요약 본 논문에서는 다양한 실시간 이더넷 프로토콜에 쉽게 적용할 수 있는 폐루프 스텝핑 모터 드라이브를 개발하는 방법을 제안한다. 제안 방법은 다양한 유형의 실시간 이더넷 프로토콜에 공용으로 재사용할 수 있는 모터 드라이브 모듈과 각 실시간 이더넷 프로토콜마다 개발되는 통신 인터페이스 모듈로 나누어진다. 또한 계층적 구조를 사용하여 모션 제어 알고리즘을 다양한 실시간 이더넷 프로토콜과 모션 프로파일에 독립적으로 재사용할 수 있도록 하였다. 제안하는 방법을 검증하기 위해 EtherCAT 및 Mechatrolink III를 활용하여 폐루프 모터 드라이브를 개발하였으며, 실시간 이더넷 기반에서 정상적으로 제어되는 것을 확인하였다. 개발한 EtherCAT 및 Mechatrolink III의 실시간 통신 성능을 확인하기 위해 Cycle time을 측정하였으며, 그 결과 32개의 모터 드라이브를 연결한 경우 EtherCAT이 Mechatrolink III 보다 7.5배 정도 더 빠른 것을 확인하였다.

주제어 : 실시간 이더넷, 필드버스, 모터 드라이브, 스텝핑 모터, 이더캐트, 메카트로링크 3

Abstract This paper proposes the design of a closed-loop stepping motor drive for real-time Ethernet (RTE), which can be easily applied to a variety of RTE protocols. The proposed design is divided into a closed-loop step motor drive which can be reused for various types of RTE protocol and RTE module developed for each specific RTE protocol. It is based on a layered architecture so that the motion control algorithm can be easily reused independently of the RTE protocol and motion profile. To verify the proposed design, closed-loop motor drives based on EtherCAT and Mechatrolink III were developed and their performances were evaluated. Cycle time was measured to verify the real-time communication performance of the developed EtherCAT and Mechatrolink III based motor drive. As a result, the EtherCAT was 7.5 times faster than the Mechatrolink III when 32 motor drives were connected.

Key Words : Real-Time Ethernet, Fieldbus, Motor drive, Stepping motor, EtherCAT, Mechatrolink III

1. 서론

실시간 이더넷(RTE, Real-Time Ethernet)은 높은 대역폭, 시간 동기화, 실시간 데이터 전송 등 많은 장점을 제공한다 [1]. 따라서, 다양한 종류의 실시간 이더넷이 개발되었으며,

IEC(Internation Electrotechnical commission) 표준으로 등재되었다. 실시간 이더넷은 특히 고성능이 필요한 고급 자동화 장비나 제조용 로봇에 사용되어, 기존의 프로피버스(Profibus, process field bus), 디바이스넷(DeviceNet), 모드 버스(Modbus)와 같은 전통적인 필드버스를 대체하고 있다

*Corresponding Author : Kyung-Jae Ha(kjha@kyungnam.ac.kr)

[1,2]. 실시간 이더넷은 2020년도 이후 기존의 필드버스 보다 시장 점유율이 더 높아 질 것으로 예상된다[3].

스테핑 모터 및 서보 모터의 제어 방법의 차이[4]와 스텝핑 모터의 강한 정지 토크로 인해 스텝핑 모터 드라이브는 큰 힘이 필요 없는 단거리 고속 움직임이 필요한 디스펜서, 비전 검사기와 같은 시스템에서 서보 모터 대비 더 빠른 Settling time을 제공한다. 이와 같은 스텝핑 모터 드라이브의 장점으로 인해 현재 소형 모터 시장에서 스텝핑 모터 사용이 증가하고 있으나, 아직 실시간 이더넷을 기반으로 하는 스텝핑 모터 드라이브가 많지 않은 실정이다.

일반적으로, 전통적인 필드버스 기반의 스텝핑 모터 드라이브를 실시간 이더넷 기반의 시스템에 사용하기 위해서는 게이트웨이를 이용하여 서로 다른 통신을 통해 연결하고 있다. 그동안 다양한 게이트웨이 관련 연구[5-9]가 수행되었으나, 이들 연구에서 게이트웨이를 사용하는 방법은 게이트웨이 내에서의 메시지 전달 지연 문제와 필드버스의 낮은 통신 속도로 인해 높은 실시간 성능이 필요한 고성능 장비에 적용하기 어렵다는 문제점을 가지고 있다.

이를 해결하기 위해 다양한 실시간 이더넷 프로토콜 기반의 슬레이브 장치를 개발하는 다양한 연구[10-15]가 수행되었으나, 현재 국제 표준인 IEC 61158에만 10개 이상의 실시간 이더넷 프로토콜이 등재되어 있고, 이외의 국제 표준으로 등재되어 있지 않은 RTEXX와 같은 다수의 통신 프로토콜도 존재하고 있어 특정 실시간 이더넷 기반이 아닌 다양한 실시간 이더넷에 적용할 수 있는 스텝핑 모터 드라이브의 개발 연구가 필요하다.

본 논문에서는 앞서 설명한 문제를 해결하기 위해 다양한 실시간 이더넷에 적용할 수 있는 스텝핑 모터 드라이브를 개발할 수 있는 방법을 제안한다. 제안하는 개발 방법은 통신 프로토콜에 독립적인 페루프 스텝핑 모터 드라이브 모듈과 각 통신 프로토콜 별로 개발이 필요한 통신 인터페이스 모듈로 구성되며, 통신 인터페이스 모듈만 변경하여 다양한 실시간 이더넷 프로토콜을 쉽게 적용할 수 있도록 하는 것이다. 특히, 제안하는 방법은 계층적 구조(Layered Architecture)를 사용하여, 개발이 어려운 모션 제어 알고리즘을 다양한 실시간 이더넷용 모션 프로파일에 독립적으로 재사용할 수 있도록 하였다. 제안하는 방법을 검증하기 위해 EtherCAT(Ethernet for Control Automation Technology) 및 Mechatrolink III 기반의 페루프 스텝핑 모터 드라이브를 개발하였다. 개발된 모터 드라이브에 대해 X-Y 스테이지를 이용하여 정상 동작 여부를 확인하였고, 실

시간 통신 성능을 확인하기 위해 Cycle time을 측정하였다.

2. 관련 연구

실시간 이더넷은 IEEE 802.3 기반의 통신 프로토콜로서 기존의 CAN이나 RS-485 기반의 필드버스 프로토콜 대비 높은 대역폭과 고속의 통신 속도를 제공하기 위해 개발되었다. 실시간 이더넷은 TCP/IP 상단에서 동작하는 방식, 이더넷 상단에서 동작하는 방식, 이더넷을 수정하여 동작하는 방식 등 크게 세 가지 카테고리로 분류된다[16]. 본 논문에서 사용하는 EtherCAT 및 Mechatrolink III는 이더넷을 수정하여 동작하는 방식의 전용 IC를 사용하는 통신 프로토콜이다.

2.1 EtherCAT

EtherCAT은 독일 Beckhoff사에 의해 개발된 실시간 이더넷으로 Summation frame 기법, On-the-Fly 기법을 사용하며 현재 가장 빠른 실시간 이더넷으로 알려져 있다[16]. EtherCAT은 표준 이더넷의 물리 계층을 사용한다. 대부분의 주요 기능은 ET1100 또는 ET1200이라고 하는 전용 IC를 통해 구현된다. EtherCAT은 많이 사용되는 모터 드라이브, I/O 장치, 인버터 등을 위해 CANopen 프로파일을 사용한 표준 방식을 이용하여 특정 장치를 제어할 수 있는 환경을 제공한다. CANopen 프로파일은 데이터 디렉터리라고 하는 데이터 오브젝트의 모음과 각 오브젝트의 구동 방식을 정의한다. 오브젝트 디렉터리는 SDO(Service Data Object) 방식과, PDO(Process Data Object) 방식을 이용하여 접근할 수 있다. 일반적으로 실시간 통신을 위해 PDO가 사용되며, SDO는 비실시간 데이터로서 파라미터 설정 등의 용도로 활용된다. EtherCAT은 SERCOS 프로파일도 제공하지만 본 논문에서는 다루지 않는다.

2.2 Mechatrolink III

Mechatrolink III는 일본 야스가와 사에 의해 개발된 실시간 이더넷 프로토콜로 2014년에 IEC 표준으로 등재되었다. Mechatrolink III는 기존의 RS-485 기반으로 개발되어 있던 Mechatrolink II의 성능을 향상하기 위해 개발되었다. Mechatrolink III의 Cycle time은 동일한 수의 슬레이브를 동일한 배선의 길이로 연결한 경우 Mechatrolink II 보다 4배 정도 빠르다.

Mechatrolink III는 표준 이더넷 물리 계층을 사용하지

만, 커넥터를 산업용 미니 커넥터[17]를 표준으로 사용한다. 데이터 링크 계층은 JL-102 또는 JL-101과 같은 전용 IC를 이용하여 구현된다. Mechatrolink III는 모든 제조사에서 개발된 Mechatrolink III 기반 슬레이브 장치가 동일한 방식으로 사용될 수 있도록 명령 전송 표준인 프로파일을 제공하고 있으며, 널리 사용되는 장치인 I/O, 서보 모터, 스텝 모터, 인버터에 대한 프로파일을 제공한다.

2.3 관련 선행 연구

실시간 이더넷은 기존의 필드버스의 성능을 향상시키기 위해 개발되었으나, 기존의 필드버스를 사용하는 일부 장치들에 대해서는 다양한 게이트웨이 개발을 통해 실시간 이더넷 기반의 시스템에서 사용할 수 있도록 하는 연구[5-9]가 수행되었다. 특히, [5]에서는 다양한 필드버스와 연결할 수 있는 게이트웨이 구조를 제안하여, 다양한 실시간 이더넷과 연결할 수 있도록 하고 있으나, 게이트웨이 내에서 발생하는 지연으로 인해 고성능 장비에서 사용이 어렵다.

이외에 EtherCAT 기반의 모터 드라이브의 개발 방법에 관한 연구[10-15]도 수행되었으며, 특히 [10]에서는 통신 지연이 보장되는 모터 드라이브를 제안하여, 정밀한 모션 제어가 가능하다는 것을 제시하였다. 다만, 이와 같은 기존의 연구들은 EtherCAT 기반의 모터 드라이브 개발 방법을 제안하고 있어, 본 논문에서와 같이 다양한 실시간 이더넷 기반의 모터 드라이브의 개발 방법은 제시하고 있지 않다.

3. 제안 모터 드라이브 설계 방법

본 장에서는 제안하고자 하는 다양한 실시간 이더넷 적용 가능한 스텝핑 모터 드라이브 설계 방법을 설명한다.

3.1. 제안 모터 드라이브 아키텍처

제안하는 설계 방법은 Fig. 1과 같이 다양한 프로토콜에 적용 가능한 모터 드라이브 모듈과 실시간 이더넷 모듈로 구성된다. Fig. 1의 진한 회색 블록은 모터 드라이브의 DSP 모듈에서 수행되는 소프트웨어 기반으로 개발되며, 이외 블록은 하드웨어를 이용하여 개발된다. 페루프 모터 드라이브 모듈은 통신 프로토콜에 무관하게 설계되어 다양한 실시간 이더넷에 재사용할 수 있다. 실시간 이더넷 모듈은 실시간 이더넷 프로토콜별로 개발되는 하드웨어 모듈이며, 페루프 모터 드라이브 모듈과 커넥터를 이용하여 연결된다.

Fig. 2는 제안하는 스텝핑 모터 드라이브용 하드웨어의

구조도이다. 모터 드라이브 모듈은 TI사의 TMS320F2018을 기반으로 모션 제어 칩인 PCL6113를 사용하여 제어 명령에 대한 엔코더 입력을 실시간으로 계산하여 스텝핑 모터 제어에 필요한 제어 신호를 생성하며, 이를 통해 스텝핑 모터를 제어하기 위한 인버터 회로가 내장되어 있다. 실시간 이더넷 모듈은 통신 컨트롤러 IC 및 Ethernet PHY, 그리고 내부 회로와 외부 통신망을 물리적으로 분리하는 트랜스포머 회로로 구성된다.

제안하는 설계 방법에서 페루프 스텝핑 모터 드라이브 모듈은 다양한 실시간 이더넷에 재사용 가능하나, 탑재되는 SW는 실시간 이더넷에 맞게 변경되어야 한다. 따라서 제안하는 방법은 다양한 실시간 이더넷 적용 시 SW 변경을 최소화하기 위해 계층 구조를 사용하여 기존의 모션 제어 알고리즘을 재사용하며, 다음 3.2절에서 설명한다.

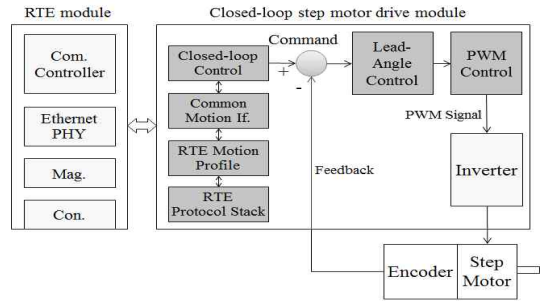


Fig. 1. Architecture of the proposed closed-loop stepping motor drive

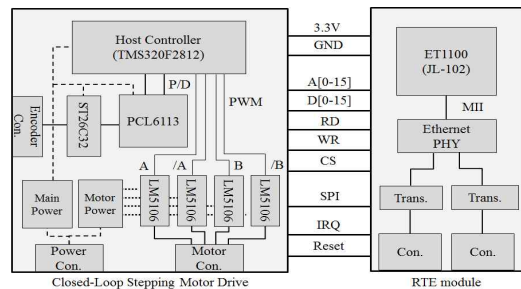


Fig. 2. Block diagram of the proposed closed-loop stepping motor drive

3.2. 소프트웨어 아키텍처

Fig. 3은 본 논문에서 제안하는 계층 구조 기반의 SW 아키텍처이다. 실시간 이더넷 기반 모터 드라이브는 모터 드라이브를 제어하는 제어 SW와 각 실시간 이더넷 프로토콜 별로 정의된 모션 프로파일, 그리고 실시간 이더넷 프로토

콜별로 통신 기능을 담당하는 통신 스택으로 구성된다. 본 논문에서 제안하는 설계 방법은 제어 SW가 다양한 실시간 이더넷에서 사용되는 제어 방법을 공용 인터페이스 형태로 제공하고 모션 프로파일 개발 시 공용 인터페이스를 활용함으로써 제어 SW를 실시간 이더넷에 독립적으로 재사용하고, 각 실시간 이더넷 프로토콜별로 제공되는 프로토콜 스택도 변경 없이 재사용할 수 있다.

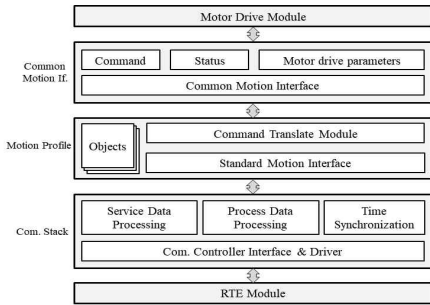


Fig. 3. Block diagram of interface between motor drive and RTE module.

실시간 이더넷의 프로토콜 스택은 어플리케이션에 독립적이면서 공통으로 제공되기 때문에 일반적으로 프로토콜 협회 또는 제조사에 의해 제공된다. 프로토콜 스택의 세부 개발 방법 및 SW 구조는 통신 프로토콜마다 상이하지만, 일반적으로 통신 IC와 인터페이스를 위한 통신 컨트롤러 인터페이스 및 디바이스 드라이버 모듈, 시간 동기화 모듈, 고속의 주기적 데이터 송수신을 위한 프로세스 데이터 처리 모듈, 이벤트성 비실시간 데이터 전송을 위한 서비스 데이터 처리 모듈 등으로 구성된다. 대다수의 통신 프로토콜은 기 제공되는 것을 재사용할 수 있으나, MCU(Micro Controller Unit) 등과 같은 호스트 컨트롤러와 통신 IC 간의 인터페이스 회로 및 디바이스 드라이버는 사용하는 환경에 맞게 개발해야 한다.

표준 모션 프로파일은 모터드라이브를 제어할 수 있는 표준 인터페이스로 IEC 61800-7 또는 국제 표준이 아닌 자체 표준 사양으로 정의되어 있다. 프로토콜 스택은 모든 어플리케이션에서 공용 사양으로 협회 또는 제조사에 의해 제공되는 것에 비해, 모션 프로파일은 각 모터 드라이브마다 제어 SW가 다를 수 있기 때문에, 제어 명령의 사양만 정의되어 있을 뿐 세부 SW는 모터 드라이브 개발사에서 직접 개발하여야 한다.

모션 프로파일의 세부 사양은 프로토콜별로 다르지만 크게 모션 제어에 필요한 정보를 저장하는 객체와 표준 인터

페이스로 구성된다. 표준 인터페이스는 제어 명령과 명령에 사용되는 객체의 집합으로 구성되며, 네트워크로 수신한 데이터를 어떤 객체에 저장하고 어떻게 모터를 제어할 지에 대해 정의한다. 모션 프로파일의 객체와 표준 인터페이스는 모션 프로파일마다 다르지만 속도 제어, 위치 제어, 토크 제어와 같이 모션 제어에서 널리 사용되는 범용의 제어 방식은 모든 모션 프로파일에서 유사한 형태로 제공된다.

일반적으로 모션 프로파일 모듈은 각 실시간 이더넷 프로토콜이 사용하는 모션 프로파일에 맞게 수신한 제어 명령대로 모터 드라이브를 제어하는 SW이다. 모션 프로파일은 앞서 설명한 바와 같이 기본적인 제어 방법은 모두 동일하지만 프로토콜 개발사에 따라 세부 정의가 조금씩 달라 모션 프로파일은 각 프로토콜별로 별도로 개발되고 있는 실정이다. 본 논문에서는 이와 같은 문제를 개선하기 위해 대다수의 모션 프로파일에서 사용되는 공통적인 기능은 공용 인터페이스로 정의하고 모션 프로파일에서는 기존 통신 표준 형태로 명령을 수신한 후 이를 해석하여 공용 인터페이스로 변환하여 호출하는 역할을 수행한다. 이와 같은 방식을 이용하여 개발된 모터 드라이브의 제어 SW는 대다수의 실시간 이더넷에서 별도의 제어 SW의 변경 없이 재사용이 가능하다.

Table 1은 제안하는 공용 모션 인터페이스로서 대다수의 모션 프로파일에서 사용되는 범용적인 모션 제어 기능에 대해 실시간 이더넷 프로토콜 및 모션 프로파일에 독립적인 형태로 인터페이스를 제공한다. Table 2는 제안하는 공용 인터페이스에서 사용되는 파라미터로서 대다수의 실시간 이더넷에서 지원되는 항목으로 구성하였다. 본 논문은 스텝 모터를 대상으로 인터페이스를 정의하고 실제 개발하였으나, 제안하는 아키텍처를 활용하여 다양한 서보 드라이브, I/O, 인버터 등의 장치에도 적용할 수 있을 것으로 예상된다.

Table 1. Proposed common motion interface

Interface	Description
SetParameter	Write the specific value of the parameter
GetParameter	Read the specific value of the parameter
ServoEnable	Servo on/off
AlarmReset	Reset the servo alarm
MoveStop	Stop the motor with deceleration
EmergencyStop	Stop the motor without deceleration
MoveOrigin	Move the motor to origin position
MovePosition	Move the motor to specific position
MoveToLimit	Move the motor to the positive or negative limit position
MoveVelocity	Move the motor using a specific velocity
PositionOverride	Change the target position while the motor is running
VelocityOverride	Change the target velocity while the motor is running

Table 2. Proposed common parameters

Parameter	Unit	Description
Target position	pulse	Target position that the drive should move to
Target velocity	rpm	Target velocity that the drive should move to after ramp up
Acceleration	s ²	Configured acceleration for ramp up operation
Deceleration	s ²	Configured deceleration for ramp down operation
Actual position	pulse	Actual position of the motor measured by an encoder
Actual velocity	rpm	Actual velocity of the motor measured by an encoder
Actual acceleration	s ²	Actual acceleration of the motor measured by an encoder
Actual deceleration	s ²	Actual deceleration of the motor measured by an encoder
AccelCurve	-	Configuration of the acceleration and deceleration curve (S or linear curve)
Command	-	Command that the drive should execute
Status	-	Status of the drive including alarms

4. 구현 및 실험 결과

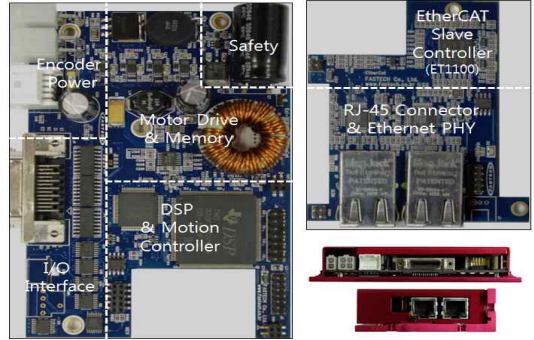
본 장에서는 앞서 설명한 실시간 이더넷 프로토콜에 독립적인 모터 드라이브 설계 방법을 활용하여 실제 EtherCAT 및 Mechatrolink III 기반 페루프 스텝핑 모터 드라이브를 개발하고, 그 결과를 검증한다.

4.1. 프로토타입 모터 드라이브 구현

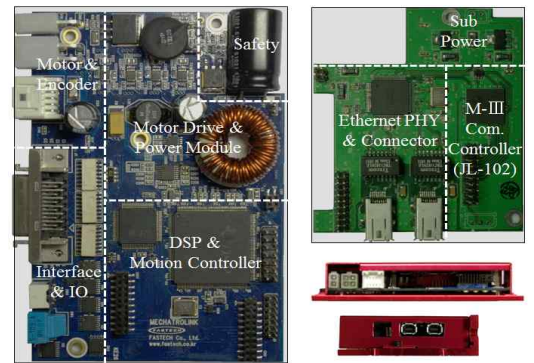
제안하는 설계 방법의 검증을 위해 EtherCAT 및 Mechatrolink III 기반의 스텝핑 모터 드라이브를 개발하였다. Fig. 4의 (a)는 개발된 EtherCAT 기반 페루프 스텝핑 모터 드라이브이며, Fig. 4의 (b)는 개발된 Mechatrolink III 기반 페루프 스텝핑 모터 드라이브이다. 개발된 두 가지 모터 드라이브 모듈이 약간 상이한 것은 PCB 레이아웃 시 반영되는 통신 커넥터의 크기가 다르기 때문이다. 즉, Mechatrolink III는 산업용 Mini 커넥터를 사용하고, EtherCAT은 이더넷 표준 커넥터로 크기가 큰 RJ45 커넥터를 사용하기 때문이다.

모터 드라이브를 평가하기 위해 EtherCAT 및 Mechatrolink III의 상용 마스터 프로그램을 이용하였으며, 본 논문에서 제안 설계한 모터 드라이브를 정상적으로 제어할 수 있었다. 개발된 모터 드라이브의 성능을 확인하기 위해 두 개의 모터 드라이브를 이용하여 Fig. 5와 같이 X-Y 스테이지를 통해 마스터 상에서 간단한 제어 알고리즘을 이용하

여 원을 그리는 시험을 진행하였다. 결과적으로 EtherCAT은 TwinCAT 이라는 표준 마스터 프로그램 기반에서 성공적으로 원호 제어가 가능하였고, Mechatrolink III 역시 마스터 샘플 코드 상에서 원호 제어가 가능하였다.



(a) EtherCAT based stepping motor drive



(b) Mechatrolink III based stepping motor drive

Fig. 4. Developed closed-loop stepping motor drive

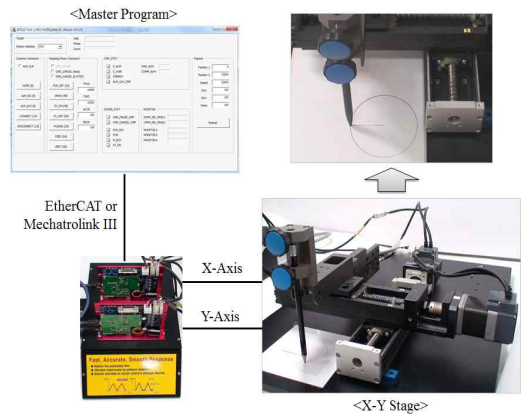


Fig. 5. Experiment environment for the developed motor drive.

4.3. Cycle time 측정

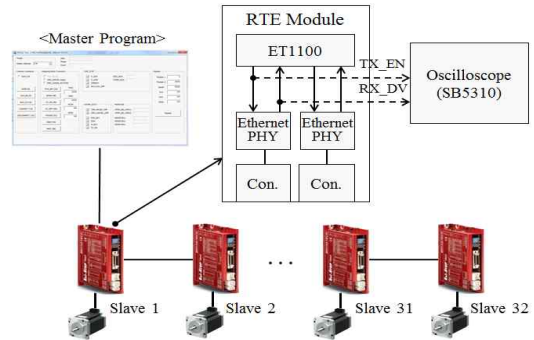
Cycle time은 마스터가 제어 명령을 송신하고 모든 슬레이브의 응답 메시지를 수신하는데 걸리는 시간이다. 즉, Cycle time은 자동화 장비 및 산업용 로봇에서 마스터가 모든 장치의 정보를 읽고 제어 명령을 내릴 수 있는 가장 짧은 시간으로, Cycle time이 짧을수록 고속의 정밀 제어가 가능해 지기 때문에 산업용 네트워크에서 Cycle time은 가장 중요한 성능 지표 중 하나이다. Cycle time은 기존의 저속의 CAN 또는 RS-485 기반의 필드버스에서 실시간 이더넷으로 시장의 트렌드가 변화하고 있는 가장 큰 이슈 중 하나이다.

Cycle time은 연결된 슬레이브 장치의 개수와 전송하는 데이터의 길이에 비례하여 증가한다. 데이터의 길이는 사용하는 모션 프로파일에 따라 고정되기 때문에 실질적으로는 슬레이브 개수에 따라 변경된다. 마스터 소프트웨어의 시간 오차를 최소화하기 위해 Cycle time은 하드웨어로 측정할 수 있는 가장 가까운 지점을 오실로스코프로 측정하였다. 통신 배선의 길이는 슬레이브 개수 및 데이터 길이에 비해 매우 짧은 시간이므로 본 논문에서는 무시한다.

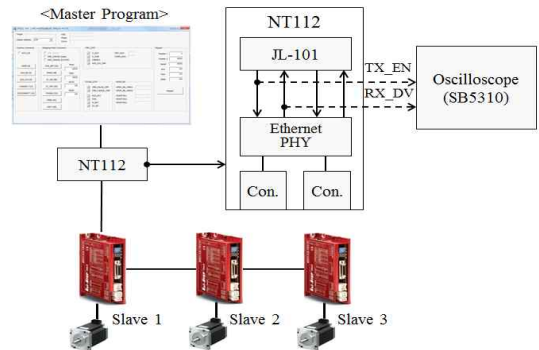
Fig. 6의 (a)는 EtherCAT 기반의 모터 드라이브 32개를 연결하여 Cycle time을 측정할 때 사용한 실험 환경을 보여준다. 마스터와 32개의 모터 드라이브는 데이터 체인 토폴로지를 이용하여 연결하였으며, 각 모터 드라이브는 0.2m 케이블로 연결하였다. 마스터가 전송하는 EtherCAT 프레임의 데이터 크기는 CiA 402 프로파일에 따라 각 모터 드라이브 별로 8byte를 사용한다. EtherCAT의 경우 PC SW로는 마스터에서 HW 신호 측정이 어려워 첫 번째 슬레이브 장치의 Ethernet PHY 칩과 통신 컨트롤러 칩인 ET1100 사이의 TX_EN 신호와 RX_DV 신호를 측정하였다. TX_EN 및 RX_DV 신호는 이더넷 컨트롤러와 Ethernet PHY를 연결하는 표준 MII(Media Independent Interface) 신호 중 하나로, 해당 슬레이브의 통신 채널로 이더넷 메시지를 전송할 때 TX_EN 신호가 Enable 되고, 이더넷 메시지가 수신될 때 RX_DV 신호가 Enable 된다.

Fig. 6의 (b)는 Mechatrolink III 기반 모터드라이브 3개를 연결한 실험 환경이다. EtherCAT과 다르게 Mechatrolink III는 Cycle time 시뮬레이터를 제공하고 있어, 3개의 실제 모터드라이브를 연결하여 Cycle time을 실측하고 시뮬레이션 결과와 같은지 확인하였다. 결과적으로 실측한 Cycle time과 시뮬레이션 결과는 동일하여 EtherCAT과 같이 32개 슬레이브를 개발하지 않고 시뮬레

이션 결과를 제시하였다. 실험을 위해 각 모터 드라이브는 1미터의 선으로 연결하였으며, 각 제어기는 Mechatrolink III 표준 스텝핑 모터 드라이브 프로파일에 따라 48byte의 메시지를 사용한다. Mechatrolink III는 PC의 마스터 인터페이스 카드 내의 Ethernet PHY와 통신 컨트롤러 NT112 사이의 TX_EN 및 RX_DV 신호를 측정하였다.



(a) Experiment environment for EtherCAT



(b) Experiment environment for Mechatrolink III

Fig. 6. Experiment environment to measure cycle time

Fig. 7은 상기 설명한 실험 환경에서 오실로스코프를 이용하여 측정한 결과이다. EtherCAT의 경우 32개의 슬레이브를 연결한 경우 Cycle time은 최소 약 77us, 최대 약 215us이다. Mechatrolink III의 경우 3개의 슬레이브를 연결한 경우 Cycle time이 최대 약 65us이며, 이는 시뮬레이션 결과와 동일하다. EtherCAT와의 비교 평가를 위해 시뮬레이터를 이용하여 32개의 슬레이브를 연결한 경우 1.62ms정도로 EtherCAT 대비 매우 큰 Cycle time을 가진다. 이와 같은 성능 차이는 EtherCAT이 Summation frame 기법 및 On-the-Fly 기법을 사용하여 하나의

Ethernet Frame에 32개 슬레이브의 송/수신 데이터를 모두 처리하는 데 비해 Mechatrolink III는 48byte 데이터를 각 슬레이브마다 송수신하므로 더 많은 시간이 소요된다. 결론적으로 32개 스텝핑 모터 드라이브를 데이터 체인 토폴로지로 연결한 경우 EtherCAT이 7.5배 정도 더 빠르며, 고속의 Cycle time이 필요한 경우 EtherCAT이 더 적합하다.

Mechatrolink III에 대한 Cycle time은 본 논문에서의 실험 결과와 Mechatrolink 협회에서 제공하는 시뮬레이션 결과와 동일함을 확인하였다. EtherCAT의 Cycle time에 대해서는 기존의 Cycle time에 대하여 이론적으로 분석한 연구[18]와 비교하였다. 기존의 연구에서 각 36 Byte를 통신하는 32개의 슬레이브를 연결한 경우 최소 133 us가 소요된다는 결과와 비교할 때, 본 연구에서 각 8 Byte 씩 32개 슬레이브를 연결한 경우 측정된 최소 Cycle time이 77us로서 데이터 사이즈의 차이를 고려할 때 유사한 수준으로 판단된다. 다만, 본 연구에서 측정된 최대 Cycle time 215us는 기존 연구에서 제시한 최소 Cycle time과 큰 차이를 보여 이에 대한 추가 분석이 필요하다.

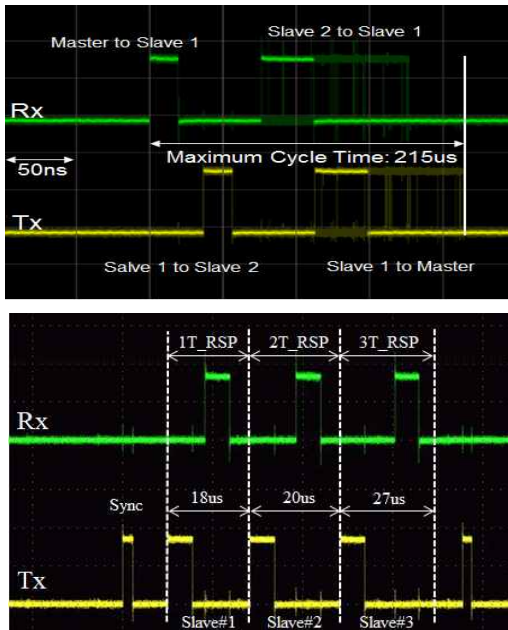


Fig. 7. Measured signals for EtherCAT and Mechatrolink III

5. 결론

본 논문에서는 다양한 실시간 이더넷 프로토콜에 쉽게 적용할 수 있는 페루프 스텝핑 모터 드라이브를 개발하는 방법을 제안하고, 실제 EtherCAT 및 Mechatrolink III를 적용하여 구현하였으며 제안하는 방법이 실제 모터 드라이브 개발에 유용함을 증명하였다. 개발된 모터 드라이브를 이용하여 간단한 원호 운동을 통해 동기 제어가 가능한지 확인하였으며, 실제 모터 드라이브를 이용하여 Cycle time을 측정하였다. 측정 결과 32개의 모터 드라이브를 연결한 경우 EtherCAT이 Mechatrolink III 보다 7.5배 정도 빠른 성능을 가진다는 것을 확인하였다. 다만, 본 연구에서는 Cycle time에 대한 실제 측정 결과만을 제시하여, 향후 다양한 실시간 이더넷에 대한 Cycle time의 이론적 분석 방법에 대한 연구가 필요하며, Cycle time 이외에 시간 동기화, 결함 발생 시 복구 시간 등 실시간 이더넷에서 중요한 다양한 성능에 대한 비교 분석에 관한 연구가 필요하다.

REFERENCES

- [1] D. Jansen & H. Buttner. (2004). Real-time ethernet the EtherCAT solution. *Computing & Control Engineering Journal*, 15(1), 16-21. DOI : 10.1049/cce:20040104
- [2] J. D. Decotignie. (2005). Ethernet-Based Real-Time and Industrial Communications. *Proceedings of the IEEE*, 93(6), 1102-1117. DOI : 10.1109/JPROC.2005.849721
- [3] T. Moore. (2013). *Industrial Ethernet Growing but Fieldbus Remains Dominant*. <http://www.automation.com/automation-news/article/industrial-ethernet-growing-but-fieldbus-remains-dominant>
- [4] AMCI, *Stepper vs. Servo*, <http://www.amci.com/tutorials/tutorials-stepper-vs-servo.asp>
- [5] J. Jasperneite & J. Feld. (2005). PROFINET: an integration platform for heterogeneous industrial communication systems. *IEEE Intl. Conf. on ETFA*, 815-822.
- [6] R. T. Qu & R. LIM. (2003). An architecture for the interconnecting heterogeneous fieldbuses in industrial automation. *IEEE Intl. Symp. on CIRA*, 396-400.
- [7] X. Xie, D. Deng & X. Deng. (2011). Design of

- Embedded Gateway Software Framework for Heterogeneous Networks Interconnection. *Intl. Conf. on ICEOE*, 306-309.
- [8] S. S. Yoon, J. C. Lee, S. J. Cho & J. W. Jeon. (2010). Gateway between high-performance Fieldbus and serial communication. *Intl. Conf. of ICCAS*, 971-974.
- [9] J. H. Kim, D. T. Tuan, B. S. Yeom & J. W. Jeon. (2013). Design of a Seamless Gateway for Mechatrolink III. *IEEE Intl. Conf. on ICIT*, 25-28.
- [10] K. Kim, M. Sung & H. W. Jin. (2012). Design and Implementation of a Delay-Guaranteed Motor Drive for Precision Motion Control. *IEEE trans. on Industrial Informatics*, 8(2), 351-365.
DOI : 10.1109/TII.2011.2166774
- [11] M. Sung, K. Kim, H. W. Jin & T. Kim. (2011). An EtherCAT-based Motor Drive for High Precision Motion Systems. *IEEE Intl. Conf. on INDIN*, 163-168.
- [12] C. Kang, Y. Pang, C. Ma & C. Li. (2011). Design of EtherCAT Slave Module. *Int. Conf. of Mechatronics and Automation*, 1600-1604.
- [13] J. K. Lee, Y. H. Song, S. Lee, K. C. Lee & Y. J. Lee. (2011). Implementation of Multi-axis Smart Driver System via EtherCAT Network based on IEC61800 standard. *Int. Conf. on ICCAS*, 1871-1874.
- [14] J. H. Park, S. Lee, K. C. Lee & Y. J. Lee. (2010). Implementation of IEC61800 based EtherCAT Slave Module for Real-time Multi-axis Smart Driver System. *Intl. Conf. on ICCAS*, 682-685.
- [15] G. Cheng & X. Wang. (2012). Design of Tri-axial Motion Servo Control Systems Based on EtherCAT. *IEEE Intl. Conf. on ICIST*, 51-54.
- [16] M. Felsler. (2005). Real-Time Ethernet-Industry Prospective. *Proceedings of the IEEE*, 93(6), 1118-1129.
DOI : 10.1109/JPROC.2005.849720
- [17] *Industrial Mini I/O - Field Installable*.
<http://www.te.com/products/mini-io>
- [18] G. Prytz. (2008). A performance analysis of EtherCAT and PROFINET IRT. *IEEE Intl. conf. on ETFA*, 408-415.

김진호(Jin-Ho Kim)

[정회원]



- 2007년 2월 : 성균관대학교 컴퓨터공학(공학사)
- 2009년 2월 : 성균관대학교 전기전자 컴퓨터공학부 (공학석사)
- 2015년 2월 : 성균관대학교 전기전자 컴퓨터공학부 (공학박사)

- 2015년 4월 ~ 2019년 2월 : 현대자동차 책임연구원
- 2019년 3월 ~ 현재 : 경남대학교 컴퓨터공학부 조교수
- 관심분야 : 차량 임베디드시스템 및 실시간 네트워크
- E-Mail : kimjh@kyungnam.ac.kr

하경재(Kyung-Jae Ha)

[정회원]



- 1980년 2월 : 성균관대학교 전기공학과 (공학사)
- 1982년 2월 : 성균관대학교 컴퓨터공학 (공학석사)
- 1989. 2월 : 성균관대학교 컴퓨터공학 (공학박사)

- 1980년 9월 ~ 1984년 3월 : 산업연구원 연구원
- 1997년 1월 ~ 1998년 1월 : 미국 웨인주립대 visiting scholar
- 1984년 3월 ~ 현재 : 경남대학교 컴퓨터공학부 교수
- 관심분야 : 컴퓨터 구조
- E-Mail : kjha@kyungnam.ac.kr