Journal of KIIT. Vol. 17, No. 1, pp. 39-44, Jan. 31, 2019. pISSN 1598-8619, eISSN 2093-7571 39
http://dx.doi.org/10.14801/jkiit.2019.17.1.39

멀티 코어 DSP 기반 EtherCAT 슬레이브 개발

박성문*, 최준영**

Development of EtherCAT Slave Based on Multi-Core DSP

Sung-Mun Park*, Joon-Young Choi**

본 논문은 BK21플러스, IT기반 융합산업 창의인력양성사업단에 의하여 지원되었고 부산대학교 기본연구지원사업(2년)에 의하여 연구되었음.

요 약

본 논문에서는 멀티 코어 DSP 기반 EtherCAT 슬레이브 구조를 설계하고 프로토콜 스택을 이식하여 EtherCAT 슬레이브를 개발한다. 개발한 EtherCAT 슬레이브는 EtherCAT 슬레이브 컨트롤러인 Beckhoff's ET1100 ASIC과 멀티 코어 DSP인 TI's TMS320F28379D로 구성된다. 기존 상업용 EtherCAT 슬레이브에서 사용하는 프로세서와 비교하여 빠른 처리 능력을 가진 DSP 코어를 채택하여 EtherCAT 슬레이브 스택의 실행 속도를 증가시킨다. 또한 EtherCAT 슬레이브 컨트롤러와 DSP 코어를 External Memory Interface를 통하여 연결함으로써 고속의 데이터 전송속도를 달성한다. 개발된 EtherCAT 슬레이브와 리눅스 기반의 IgH EtherCAT 마스터를 연결하여 EtherCAT 네트워크를 구성하고 다양한 실험을 수행하여 기존 EtherCAT 슬레이브와 비교하여 성능이 개선된 것을 검증한다.

Abstract

We develope an EtherCAT slave by designing an EtherCAT slave architecture based on multi-core DSP and porting the protocol stack. The developed EtherCAT slave consists of Beckhoff's ET1100 ASIC, an EtherCAT slave controller, and TI's TMS320F28379D SoC, a multi-core DSP. Adopting a DSP core with high-speed processing capability in comparison to processors used for existing commercial EtherCAT slaves, the execution speed is increased for the EtherCAT salve stack. Moreover, connecting the EtherCAT slave controller and the DSP core through External Memory Interface, the high-speed data transfer rate is achieved between the EtherCAT slave controller and the DSP core. We build an EtherCAT network consisting of the developed EtherCAT slave and the IgH EtherCAT master for Linux, and conduct various experiments. The experiment results verify that the performance of the developed slave is improved in comparison to existing EtherCAT slaves.

Keywords
industrial network, EtherCAT, multi-core DSP, EtherCAT slave

* 부산대학교 전기전자컴퓨터공학과
- ORCID: https://orcid.org/0000-0002-5946-8279
** 부산대학교 전자공학과 정교수(교신저자)
- ORCID: https://orcid.org/0000-0002-5160-3739

· Received: Oct. 04, 2018, Revised: Dec. 14, 2018, Accepted: Dec. 17, 2018
· Corresponding Author: Joon-Young Choi
Dept. of Electronic Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, 46241, Korea,
Tel.: +82-51-510-2490, Email: jyc@pusan.ac.kr

Ⅰ. 서 론

최근 자동화 분야에서 저비용, 높은 전송속도, 다양한 토폴로지 등의 장점으로 산업용 이더넷의 사용이 증가하고 있다[1][2]. 산업용 이더넷 프로토콜 중 하나인 EtherCAT은 표준 이더넷 프레임을 사용하지만 EtherCAT 슬레이브는 프레임이 통과하는 동안 각 슬레이브에서 처리해야 하는 데이터를 읽고 전송할 데이터를 프레임이 통과하는 동안 입력하는 'on-the-fly'방식을 사용한다. 결과적으로 EtherCAT은 프레임 처리에 별도의 지연이 발생하지 않고 오직 하드웨어 전달 지연만 발생하여 $100\mu s$ 보다 짧은 주기 시간으로 동작이 가능하다[3].

그러나 더 높은 정밀도로 제어하기 위해 컴퓨터 수치 제어(Computerized Numerical Control, CNC)를 포함한 다축 모션 제어 분야나 멀티 레벨 컨버터와 같은 전력 제어 분야에서는 더 빠른 동작 주기 시간을 요구하고 있고, 기존 제어 알고리즘으로는 달성할 수 없는 높은 성능을 얻기 위해 복잡한 제어 알고리즘을 요구하는 응용분야가 증가하고 있다[4]-[6]. 그러나 이러한 요구와는 달리 기존에 널리 사용되고 있는 산업용 이더넷 장치들은 싱글 코어 기반으로 제작이 되어 있기 때문에 고속의 데이터 전송속도를 유지하면서 복잡한 제어 알고리즘 수행하기에 한계가 있다[7].

이러한 문제점들을 해결하기 위하여 본 논문에서는 두 개 이상의 독립적인 CPU 코어가 존재하는 멀티 코어 DSP를 채택하여 EtherCAT 통신을 위한 전용 DSP 코어를 할당하여 고속 통신을 수행하고 복잡한 제어 어플리케이션은 다른 코어에 할당하여 제어 알고리즘을 수행하여 EtherCAT 통신과 제어 알고리즘이 서로 부하를 주지 않는 병렬 처리가 가능한 EtherCAT 슬레이브를 개발한다. 또한 EtherCAT 전용 DSP 코어와 EtherCAT 슬레이브 컨트롤러(EtherCAT Slave Controller, ESC) 사이에서 고속의 데이터 전송을 위해 EMIF(External Memory Interface)로 연결하는 구조를 설계하고 EtherCAT 슬레이브 스택이 TMS320F28379D 프로세서에서 동작할 수 있도록 기존 스택을 수정하고 이식한다. 개발한 EtherCAT 슬레이브의 성능은 리눅스 기반 EtherCAT 마스터와 기존 상업용 EtherCAT 슬레이브를 연결하여 EtherCAT 네트워크를 구성하고 비교 실험을 통해 가능한 최소 주기를 측정하여 성능이 개선됨을 검증한다.

본 연구의 목적에 따라 2장에서는 EtherCAT 프로토콜에 대해 설명하고, 3장에서는 EtherCAT 슬레이브 구현 방법에 대해 설명한다. 4장에서는 제안한 방법으로 구현된 EtherCAT 슬레이브의 성능을 확인하고, 5장에서 결론을 제시한다.

Ⅱ. EtherCAT 프로토콜 개요

EtherCAT은 산업용 이더넷 프로토콜 중 하나로 다량의 데이터를 고속으로 전송할 수 있고 정확한 동기화 성능으로 모션 제어와 같은 산업용 응용분야에서 많이 사용되고 있다[1]. EtherCAT은 마스터/슬레이브의 구조로, 일반적으로 그림 1과 같이 하나의 마스터와 여러 개의 슬레이브로 네트워크를 구성한다. EtherCAT 통신을 위해 마스터는 일반적인 이더넷 컨트롤러 카드로 구현이 가능한 반면, 슬레이브는 ESC라는 특정 하드웨어가 요구된다.

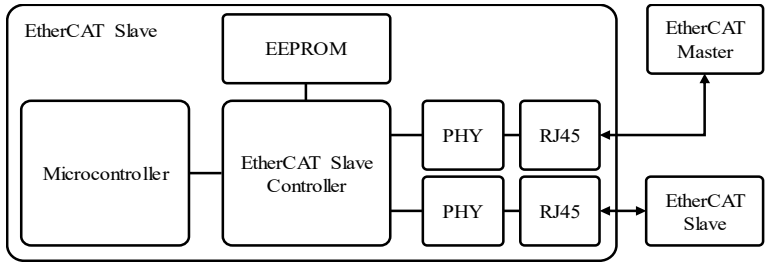


그림 1. EtherCAT 네트워크 구성도
Fig. 1. EtherCAT network topology

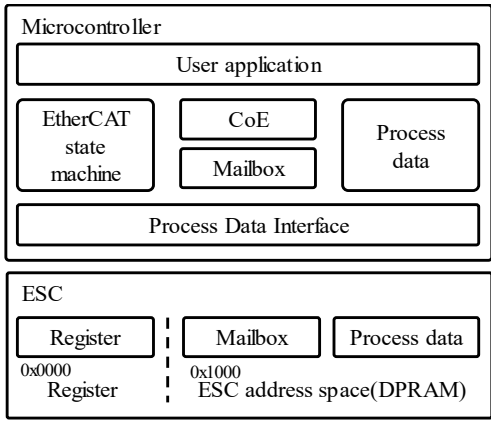


그림 2. EtherCAT 슬레이브 구조
Fig. 2. EtherCAT slave architecture

ESC는 EtherCAT 프레임을 처리하고 다음 슬레이브로 전달하는 역할을 하며, FPGA나 ASIC의 형태로 구현되어 있거나 프로세서 안에 내장된 서브 프로세서에 펌웨어 형태로 구현된 형태가 있다[8]. 슬레이브 하드웨어는 ESC, 두 채널의 이더넷 PHY와 RJ45 커넥터, ESC와 연결되어 마스터로부터 받은 데이터에 대한 처리를 하는 마이크로컨트롤러, 슬레이브에 대한 설정이 저장된 EEPROM으로 구성된다.

그림 2는 EtherCAT 슬레이브의 구조를 나타낸다. EtherCAT 슬레이브는 ESC와 마이크로컨트롤러 부분으로 나눌 수 있는데, ESC는 설정을 위한 레지스터가 있고, 비주기 데이터 저장을 위한 메일박스 영역과 주기 데이터 저장을 위한 Process data 영역이 DPRAM에 할당되어 있다. 마이크로컨트롤러 부분은 ESC로부터 데이터를 교환하기 위한 Process Data Interface(PDI)와 EtherCAT 상태 머신, CANopen over EtherCAT(CoE) 프로파일, 비주기 데이터 처리를 위한 메일박스, 주기 데이터 처리를 위한 Process data, 사용자 어플리케이션 부분으로 나뉘며, 이를 EtherCAT 슬레이브 스택이라고 한다[9].

III. 멀티 코어 DSP 기반 EtherCAT 슬레이브 구현

본 장에서는 EtherCAT 슬레이브의 구현을 위해 멀티 코어 DSP를 기반으로 한 EtherCAT 슬레이브의 구조 설계와 EtherCAT 슬레이브 스택의 이식 방법을 제안한다.

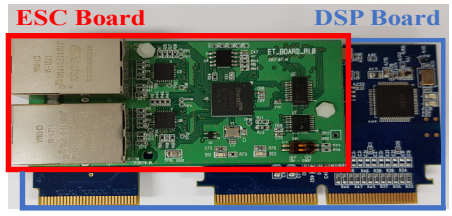


그림 3. 개발한 EtherCAT 슬레이브 하드웨어
Fig. 3. Developed EtherCAT slave hardware

멀티 코어 DSP 기반 EtherCAT 슬레이브 하드웨어는 그림 3과 같이 ESC 보드와 DSP 보드로 구성된다. ESC 보드는 Beckhoff 사의 ET1100 ASIC과 RJ45, PHY, EEPROM으로 구성되어 있으며, TI 사에서 제공하는 참조 설계도를 수정하여 제작하였다[10]. DSP 보드는 TMS320F28379D 프로세서를 사용하는 TI 사의 상용 보드인 F28379D controlCARD를 이용하고, DSP 보드와 ESC 보드는 고속 데이터 전송을 위해 EMIF로 연결되어 있다.

EtherCAT 슬레이브 소프트웨어 구현을 위해 [9]에서 제공하는 Beckhoff 사의 EtherCAT 슬레이브 스택 코드를 사용하는데 제공되는 EtherCAT 슬레이브 스택 코드는 사용자 어플리케이션과 PDI를 제외한 EtherCAT 상태 머신, CoE, 메일박스와 Process data 부분이 구현되어 있다. 그러나 이 코드는 일반적으로 정의하고 있는 것과 같이 char형 변수의 크기를 8비트로 정의하여 사용하고 있는 반면, 본 논문에서 사용하는 멀티 코어 DSP인 TMS320F28379D 프로세서는 char형 변수를 16비트로 정의한다. 이로 인해 TMS320F28379D 프로세서는 char형 변수와 int형 변수가 모두 16비트이고 이때 int형 변수에 대한 sizeof 연산의 결과는 2가 아닌 1이 된다. 따라서 EtherCAT 슬레이브 스택 코드에서 정의하는 1바이트가 8비트, int 형의 크기가 2인 것과는 달라 char 배열로 선언된 문자열을 읽어올 때 전체 문자를 읽어오지 못하고, int 배열로 선언된 변수들의 크기가 절반만 인식이 되어 전체를 다 읽지 못하는 경우가 발생한다.

TMS320F28379D 프로세서에 EtherCAT 슬레이브 스택 코드를 이식하기 위해 TMS320F28379D 프로세서에서 동작할 수 있도록 스택 코드의 수정이 요구된다. int나 long으로 선언된 변수들의 크기를 읽는 함수에 2배를 한 값을 반환한다. char 배열로 선

언된 문자는 EtherCAT 프레임으로 보내기 위해 int 배열로 변환을 하는데, 이때 char나 int 둘 다 16 비트이므로 하나의 int 변수 안에 하나의 char 변수만 포함이 된다. 그러나 프레임을 받은 마스터에서는 이를 두 개의 char로 나눠서 저장을 하여 첫 번째 글자와 NULL 문자를 저장하게 되고 다음 문자는 읽지 못하는 경우가 발생한다. 따라서 char 배열을 int 배열로 변환할 때 int 배열 하나당 char 변수 2개가 포함되도록 재배열하는 함수가 필요하다. 일반적으로 'ab'라는 문자열은 아스키 코드로 '0x61', '0x62'라고 선언이 되어 int 배열로 변환이 되면 '0x6261'로 변환이 되는데 TMS320F28379D 프로세서에서는 '0x0061', 0x0062'라고 선언이 되어 int 배열로 변환이 될 때 '0x0061' 0x0062'로 변환이 된다. 따라서 문자열의 재배열을 통해 '0x6261'이 되도록 재배열한 후 int 형으로 변환하여 EtherCAT 프레임으로 보내면 마스터에서는 NULL 문자를 인식하지 않고 입력한 문자인 'ab'를 인식할 수 있다.

또한 ESC와 DSP 코어를 연결하는 PDI로 EMIF를 채택함으로써 고속의 데이터 전송 속도를 달성한다. EtherCAT 슬레이브 스택 코드에는 PDI는 구현되어 있지 않기 때문에 TMS320F28379D 프로세서에서 수행할 수 있도록 EMIF를 구현하여 ESC의 메모리 영역에 데이터를 읽거나 쓰는 함수를 구현한다.

수정한 EtherCAT 슬레이브 스택 코드와 PDI는 TMS320F28379D 프로세서의 코어 중에서 전용 코어를 할당하여 실행하고 다른 코어는 제어 알고리즘 등 다른 어플리케이션을 실행하는 역할을 수행하여 각 코어는 서로의 연산 부하에 영향을 미치지 않는다. EtherCAT 전용 코어의 사용자 어플리케이션에는 제어 알고리즘이 아닌 두 코어 사이에 존재하는 공유 메모리를 통해 다른 코어와 데이터를 주고받을 수 있는 코드가 추가가 되어 제어 알고리즘과 EtherCAT 통신 처리를 분리한다.

Ⅳ. 성능 시험

4.1 성능 시험 환경 구축

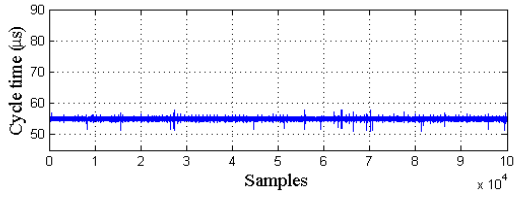
본 논문에서 개발한 EtherCAT 슬레이브의 성능 시험을 위해, [11]에서 제공하는 리눅스 기반 오픈 소스 EtherCAT 마스터인 IgH EtherCAT 마스터와 슬레이브를 연결하여 성능 검증 시험을 수행하였다. EtherCAT 마스터 PC는 Intel Core I5-2500 Processor, 4 GB DDR3 RAM과 Intel 82574L Gigabit Ethernet Controller(E1000E 이더넷 드라이버)로 구성되어 있고, 리눅스 커널은 RTAI 패치가 된 3.4.9 버전을 사용하였다.

IgH EtherCAT 마스터는 일반적인 이더넷 드라이버를 그대로 사용하면서 마스터의 동작을 할 수 있도록 마스터와 이더넷 드라이버 사이에 Generic 드라이버가 포함되어 동작하는 방식과, 일반적인 이더넷 드라이버를 수정하여 마스터 전용 이더넷 드라이버로 동작하는 방식이 있다. 본 논문에서는 Generic 드라이버와 E1000E 전용 드라이버를 모두 사용하여 실험한다.

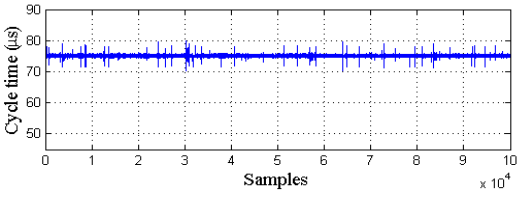
마스터에서 제어 명령을 전송하는 통신 주기가 짧을수록 제어 명령에 따른 모션의 정밀도가 향상되기 때문에 산업용 네트워크 시스템에서 중요한 성능 검증 지표 중 하나는 최소 통신 주기이다[12]. 따라서 EtherCAT 마스터 PC와 개발한 슬레이브, TI 사의 TMDSICE3359, Infineon 사의 XMC4800 슬레이브 보드를 한 대씩 연결하여 각각의 슬레이브가 달성 가능한 최소 통신 주기를 측정하여 슬레이브의 성능을 비교 검증한다. 마스터는 16 비트 데이터를 2개 보내고 슬레이브는 2개의 데이터를 받아 더한 값을 다시 마스터로 보낸다. 이 때 마스터가 데이터를 보내는 주기를 5μs씩 감소시키는데 데이터 처리 및 EtherCAT 통신에서 오류가 발생하지 않을 때까지 감소시키고 오류가 없는 최소 주기일 때 마스터에서 데이터를 보내는 주기를 측정한다.

4.2 시험 결과

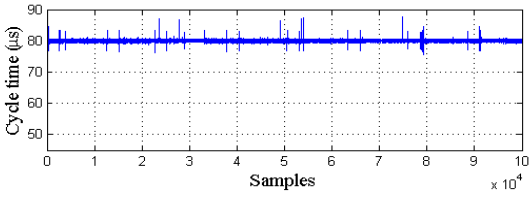
그림 4는 E1000E 전용 드라이버를 사용하였을 때 측정된 최소 주기이다. 개발한 EtherCAT 슬레이브의 경우 최소 주기가 55μs이고, Infineon 사의 슬레이브는 75μs, TI 사의 슬레이브는 80μs이다. E1000E 전용 드라이버를 사용하였을 때 개발한 슬레이브의 최소 주기가 다른 상업용 슬레이브보다 빠름을 알 수 있다.



(a) 개발한 슬레이브
(a) Developed slave

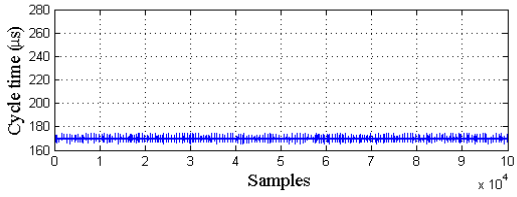


(b) Infineon 슬레이브
(b) Infineon slave

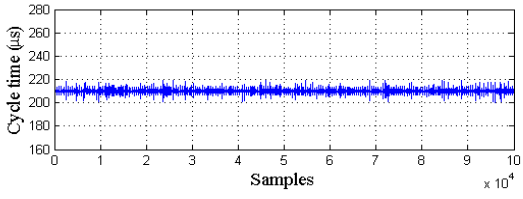


(c) TI 슬레이브
(c) TI slave

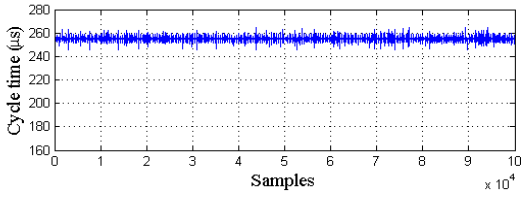
그림 4. E1000E 드라이버 사용 시 측정된 최소 주기
Fig. 4. Measured minimum cycle times when using the E1000E driver



(a) 개발한 슬레이브
(a) Developed slave



(b) Infineon 슬레이브
(b) Infineon slave



(c) TI 슬레이브
(c) TI slave

그림 5. Generic 드라이버 사용 시 측정된 최소 주기
Fig. 5. Measured minimum cycle times when using the Generic driver

그림 5는 Generic 드라이버를 사용하였을 때 측정된 최소 주기이다. 개발한 EtherCAT 슬레이브의 경우 최소 주기가 170μs이고, Infineon사의 슬레이브는 210μs, TI사의 슬레이브는 255μs이다. Generic 드라이버를 사용하였을 때도 개발한 슬레이브가 다른 슬레이브에 비해 최소 주기가 빠름을 확인 할 수 있다. Generic 드라이버를 사용할 경우 기존의 이더넷 드라이버와 EtherCAT 마스터사이에 Generic 드라이버가 포함되는 형태이므로 전용 드라이버에 비해 상대적으로 속도가 느림을 확인할 수 있다.

기존 상용 슬레이브와 비교하여 최소 주기 성능이 향상된 원인을 다음과 같이 분석할 수 있다. TI사 슬레이브에서 사용하는 ARM Cortex-A8, Infineon사 슬레이브에서 사용하는 ARM Cortex-M4와 비교하여 빠른 처리 능력을 가진 TMS320F28379D 프로세서를 채택하였고, EtherCAT 슬레이브 스택의 처리를 위하여 전용 코어를 할당하여 펌웨어 형태로 스택을 구현하였으며, DSP와 ESC 사이의 인터페이스로 고속 데이터 전송이 가능한 EMIF를 채택하여 구현하였기 때문에 성능이 향상된 것으로 분석할 수 있다.

V. 결 론

본 논문에서는 멀티 코어 DSP 기반 EtherCAT 슬레이브의 구조를 설계하고 EtherCAT 슬레이브의 스택을 이식하여 EtherCAT 슬레이브를 개발하였다. 개발된 슬레이브 하드웨어는 ESC로 사용되는 ET1100 ASIC과 멀티 코어 DSP로 사용되는 TMS320F28379D 프로세서로 구성된다. 멀티 코어 DSP인 TMS320F28379D 프로세서에 맞게 EtherCAT 슬레이브 스택 코드를 EtherCAT 처리를 위한 전용 코어에 이식하였고, TMS320F28379D와 ESC의 고속의 데이터 전송을 위해 PDI로 EMIF를 채택하여 구현하였다. 개발된 EtherCAT 슬레이브는 리눅스 기반 EtherCAT 마스터와 연결하여 가능한 최소 주기

를 측정하고 다른 슬레이브와의 비교 실험을 통해 성능을 검증한다.

시험 결과를 통하여 개발된 EtherCAT 슬레이브가 전용 드라이버를 사용 시 55μs, Generic 드라이버를 사용 시 170μs로 다른 슬레이브와 비교하여 가능한 최소 주기가 더 빠름을 확인하였다. 성능 개선의 원인은 처리 속도가 빠른 DSP 코어를 전용으로 사용하고 ESC와 DSP 사이의 고속 데이터 전송을 위한 EMIF를 채택하고 슬레이브 스택을 펌웨어 형태로 구현한 것으로 분석된다.

References

[1] C. J. An, H. C. Yi, H. W. Kim, S. M. Park, and J. Y. Choi, "Improving packet loss rate of EtherCAT master dependent on hardware performance", Journal of KIIT, Vol. 13, No. 4, pp. 77-83, Apr. 2015.

[2] J. H. Lee, H. W. Kim, and J. Y. Choi, "Design and Implementation of embedded system hardware using RM57L843 microprocessor based on Cortex-R", Journal of KIIT, Vol. 16, No. 4, pp. 57-65, Apr. 2018.

[3] S. M. Park, H. Kim, H. W. Kim, C. N. Cho, and J. Y. Choi, "Synchronization improvement of distributed clocks in EtherCAT networks", IEEE Communications Letters, Vol. 21, No. 6, pp. 1277-1280, Jun. 2017.

[4] D. Orfanus, R. Indergaard, G. Prytz, and T. Wien, "EtherCAT-based platform for distributed control in high-performance industrial applications", IEEE 18th Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1-8, Sep. 2013.

[5] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "On the Accuracy of the Distributed Clock Mechanism in EtherCAT", Proceedings of the IEEE International Workshop Factory Communication Systems, pp. 43-52, May 2010.

[6] Z. Fang and Y. Fu, "A networked embedded real-time controller for complex control systems", Control and Decision Conference (CCDC), pp. 3210-3215, Dec. 2011.

[7] T. Maruyama and Y. Tsutomu, "Spatial-temporal communication redundancy for high performance EtherCAT master", IEEE 22nd Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1-6, Sep. 2017.

[8] M. Soni, "EtherCAT on Sitara processors", White paper, Texas Instruments, Apr. 2016.

[9] "Application note ET9300", Beckhoff, Nov. 2017.

[10] "EtherCAT interface for high performance C2000 MCU", Texas Instruments, Nov. 2016.

[11] IgH EtherCAT Master for Linux, http://www.etherlab.org. [accessed: Oct. 01, 2018]

[12] I. Kim and T. Kim, "Guaranteeing isochronous control of networked motion control systems using phase offset adjustment", Sensors, Vol. 15, No. 6 pp. 13945-13965, Jun. 2015.

저자소개

박 성 문 (Sung-Mun Park)

2014년 : 부산대학교 전자전기공학부(공학사)
2016년 : 부산대학교 전자전기컴퓨터공학과(공학석사)
2016년 3월 ~ 현재 : 부산대학교 전기전자컴퓨터공학과 박사과정
관심분야 : 임베디드 시스템, 제어 시스템

최 준 영 (Joon-Young Choi)

1994년 : 포항공과대학교 전자전기공학과(공학사)
1996년 : 포항공과대학교 전자전기공학과(공학석사)
2002년 : 포항공과대학교 전자전기공학과(공학박사)
2005년 3월 ~ 현재 : 부산대학교 전자공학과 교수
관심분야 : 임베디드 시스템, 제어 시스템