이더캣 통신시스템에서 분산 클럭 동기화에 관한 연구

문용선* · 트롱루안* · 이영필** · 차현록***

The Study on Distribution Clock Synchronization of EtherCAT Communication System

Yongseomn Moon*, Vo Trong Tuan Anh*, Youngpil Lee**, Hyunrok Cha***

요 약

본 논문에서는 현재 네트워크 기반 제어 시스템에서 사용이 되는 동기화 프로토콜 방식과 최신 산업용 이더넷 기술의 동기화 기술 구현에 사용되는 IEEE1588 동기화 프로토콜에 대한 전반적인 내용을 기술하였다. 또한 IEEE1588 시간 기반 동기화 기술을 사용하는 산업용 이더넷 기술 중의 하나인 이더캣 통신의 동기화 기술의 구현 및 실험을 수행하고, 실험 결과에 대한 평가, 문제점 개선 및 향후 계획에 대한 내용을 기술하였다.

ABSTRACT

In this paper, we describe a method for synchronization protocol method used in control system based on network and IEEE 1599 synchronization method which used for implementation of synchronization technology of advanced industrial Ethernet. We also implement and perform the experiment for synchronization technology of EtherCAT communication which is one of the industrial Ethernet technology used IEEE 1599 synchronization technology based on time. And we describe an evaluation for experiment result, improve the problem and future plan.

키워드

Network Control, Synchronization Protocol, EtherCAT Communication, IEEE1588

Ⅰ. 서 론

오늘날 IT 기술의 빠른 기술개발로 인하여 대부분의 기기들은 기존의 중앙 집중적인 동작 구조에서 탈피하여 고속의 네트워크 기반으로 하는 분산처리 구조 형태로 점차 변화되어 가고 있는 중이다. 이와 같은 네트워크 기반 분산 기기 및 시스템은 종래의 중앙 집중적인 구조에서 발생하는 과부하, 복잡한 배선, 진단 기능 부재 등의 문제를 해결함으로서 일반 사무 환경은 물론 다양한 산업 영역 등에 적용이 되고 있다. 그러나 산업 자동화 시스템 특히 산업용 모션 시스템에 이와 같은 네트워크 기반 제어 시스템이 도입되면서 네트워크상에 분산된 디바이스들의 동기화 문제가 새롭게 대두되었다. 네트워크 기반 제어 시스템의 경우 네트워크 거리에 따른 지연나 프로세스 처리 및 네트워크상에서 발생하는 지터 등의 요인으로 인하여 분산된 디바이스들이 동시 제어기 불가능하였기 때문에 다축의 정교한 모션이 필요한 산업 분야에서는 적

* 순천대학교 전자공학과/REDONE Technology
** REDONE Technology
*** 생산기술연구원
심사완료일자 : 2009. 11. 10
접수일자 : 2009. 10. 1

용이 어려운 문제가 발생하였기 때문이다.

이에 네트워크를 이용하여 분산된 디바이스들을 동기화 할 수 있는 다양한 방식 네트워크 기반 제어 시스템에 도입된 시점부터 현재까지 이르러 꾸준히 개발 되어왔다. 현재까지 개발 및 제안된 네트워크 기반 동기화 방식에는 메시지(Message) 기반 동기화, 주기적(Cyclic) 동기화, 시간(Time) 기반 동기화 방식 등이 있다.

기존의 필드버스 기반의 산업용 환경에서의 경우 메시지 기반 동기화 및 주기적 동기화 기술이 많이 사용되었어나 최근에는 산업용 이더넷 기반의 자동화 네트워크 시스템의 구조가 큰 각광을 가지면서 산업용 이더넷 많이 적용이 되는 분산 클럭을 기반으로 하는 IEEE1588 기반의 시 기반 동기화 기술이 자동화 시스템의 핵심 동기화 솔루션으로 적용되고 있다.

이에 본 논문에서는 IEEE1588 기반의 동기화 프로토콜을 사용하는 산업용 이더넷 통신 기술 중 최고 수준의 속도 및 동기화 성능을 보유하는 것으로 평가되어 지는 이더캣 통신 기술의 동기화에 대한 소개와 이러한 이더캣 통신을 통한 분산 클럭 동기화 기술에 대한 구현에 대한 내용을 기술하였다.

본 논문은 현재 네트워크 기반 제어 시스템에서 사용이 되는 동기화 프로토콜 방식과 최신 산업용 이더넷 기술의 동기화 기술 구현에 사용되는 IEEE1588 동기화 프롤토콜에 대한 전반적인 내용을 기술하였다. 또한 IEEE1588 시 기반 동기화 기술을 사용하는 산업용 이더넷 기술 중의 하나인 이더캣 통신의 동기화 기술의 구현 및 실험을 수행하였다.

II. 네트워크 기반 동기화 기술

2.1 네트워크 기반 동기화 기술

현재 산업 자동화 영역에는 네트워크 기반 동기화를 지원하는 다양한 네트워크 프로토콜들이 산재해 있다. 이러한 산업용 네트워크 프로토콜들은 또한 벤더 고유의 기술을 적용한 동기화 기술을 적용하고 있으므로 기술적인 분류는 매우 다양하다 할 수 있다. 그러나 세부적인 구현 기술에 대한 차이를 배제한 큰 범주에서의 기준을 적용하였을 경우 다음과 같이 크게 3가지 형태로 크게 분류할 수 있다.

2.1.1 메시지 기반 동기화 기술

초기 산업 자동화 영역에 사용되었던 저속의 필드버스(Filedbus) 네트워크들이 분산된 디바이스들 간의 동기화를 위해 사용되던 방식으로서 마스터(Master)와 슬레이브(Slave)로 구분된 네트워크 시스템 상에서 상위 마스터가 동기화를 위한 특정 메시지를 전송하며, 분산된 슬레이브들은 이러한 동기화 메시지를 수신한 후 슬레이브들 간의 동기화를 수행하는 방식을 따른다. 이러한 메시지 기반의 동기화 기술을 적용한 대표적인 네트워크 프로토콜에는 프로피버스(Profibus), 디바이스넷(DeviceNet) 등이 있다.

2.1.2 주기적 동기화 방식

주기적 동기화 방식은 마스터와 슬레이브 간에 고속의 주기적인 통신을 수행하는 네트워크 프로토콜들에 의해 주로 사용되는 동기화 방식으로서 고속의 연속적인 통신을 수행하는 과정 중에 상위 마스터가 하나의 주기적인 동기화 타이밍 계산 및 지령함으로서 슬레이브 디바이스들을 동기화하는 방식을 말한다. 주기적 동기화 방식을 사용하는 대표적인 네트워크 프로토콜에 서커스(SERCOS), 파워링크(PowerLink) 등이 있다.

2.1.3 시간 기반 동기화 방식

시간 기반 동기화 방식은 분산된 슬레이브 내에 정교하게 구현된 클럭에 의한 동기화를 방식으로서 즉, 서로 다른 클럭을 가지고 구동되는 슬레이브 디바이스들의 클럭을 동기화 알고리즘을 적용하여 하나로 일치시킴으로서 모든 디바이스들이 동일한 클럭을 기분으로 동기화되도록 하는 방식이다. 시간기반 동기화 방식을 사용하는 대표적인 프로토콜은 IEEE1588, GPS, SNTP 등이 있다.

2.2. IEEE1588 시간 기반 동기화 프로토콜

IEEE1588 시간기반 동기화 프로토콜은 네트워크를 기반으로 하는 계측 및 제어 시스템들의 정확한 클럭 동기화를 위한 방식으로서 Ethernet 네트워크 내의 어플리케이션 상에 적용하기에 최적화된 프로토콜이다.

이러한 IEEE1588 시간기반 동기화 기술은 정밀한 기준 클럭을 중심으로 서로 다른 구동 클럭을 가지는 분산된 디바이스들의 클럭을 보정(오프셋 및 지연 보정) 함으로서 시간을 동기화 하는 방식이다.

최근들어 산업자동화 영역에서의 버스 시스템의 구조가 종래의 저속의 필드 버스 구조에서 속도, 성능 및 통합 적인 측면에서 최고의 성능을 발휘하는 산업용 이더넷으로 변경되면서부터 대부분의 산업용 이더넷 기반의 제어 네트워크에서 IEEE1588 동기화 알고리즘을 도입하고 있는 추세이다[1].

표 1. 시간기반 동기화 프로토콜 비교
Table 1. Comparison of time base synchronization protocol

Item	SNTP	GPS	IEEE1588
응용 분야	wide area	wide area	A few subnets
통신	internet	Satellite	LAN
정밀도	<ms	<us	<us
관리	configured	n/a	Self-orginized
특별 하드웨어	No	Receiver	with or without

그림 1, 2에 IEEE1588 동기화 프로토콜의 클럭 오프셋 및 지연을 보정하는 구조를 나타내었다.

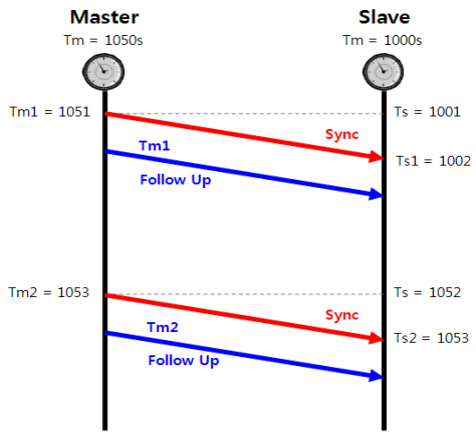

그림 1. IEEE1588 클럭 오프셋 보정 구조
Fig.1 The correction structure of IEEE 1588 clock offset

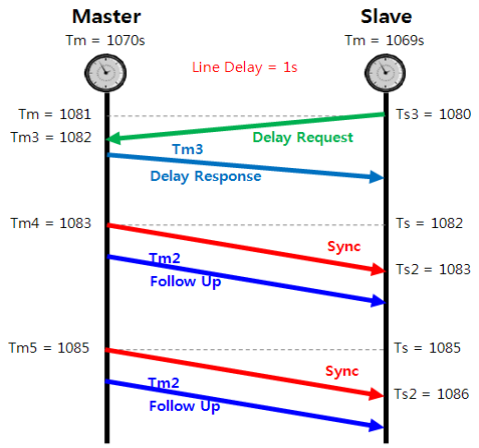

그림 2. IEEE1588 클럭 지연 보정 구조
Fig.2 The correction structure of IEEE 1588 clock delay

III. 이더캣 분산 클럭 동기화 기술

산업용 이더넷 통신 기술 중에 하나인 이더캣 통신 네트워크는 IEEE1588 시간 기반 동기화 기술을 적용하여 디바이스들 상에 기 구현된 클럭을 일치시킴으로서 분산된 디바이스들을 동기화 한다. 즉, 세부적인 동기화 구조는 다를 수 있지만 전반적인 동기화 알고리즘의 구조(오프셋 및 지연 보정 등)는 IEEE1588 구조와 동일하다는 것이다. 이더캣 동기화 기술 구현에 대한 세부적인 절차는 다음과 같다.

3.1. 네트워크 토폴로지 분석

이더캣 마스터는 분산된 디바이스들의 클럭을 동기화 하기 위하여 최초 네트워크 상에 연결된 디바이스들의 수 및 링크의 구성을 체크함으로서 동기화를 위한 네트워크 상태 및 토폴로지를 파악한다. 이는 각각의 분산된 디바이스 상에 구현된 포트 상태 레지스터의 정보를 체크함으로서 구현이 가능하다. 그림 3은 4개의 이더캣 네트워크상에 연결된 4개의 디바이스를 기준으로 네트워크 상태를 체크하는 개념적인 구조를 나타낸다[2][3][4].

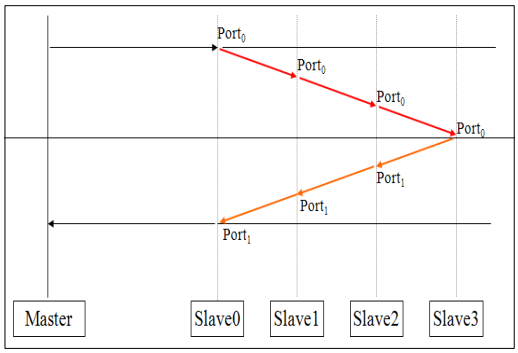

그림 3. 디바이스 수 및 네트워크 링크 상태 체크
Fig. 3 The number of device and state check of network link

그림 4는 체크된 네트워크 상태를 기반으로 파악된 마스터와 분산된 디바이스들과의 네트워크 토폴로지 구조를 나타낸다.

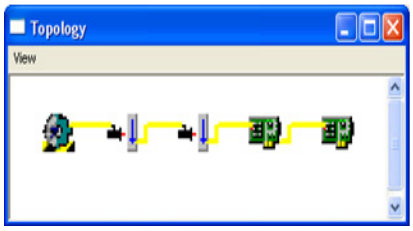

그림 4. 링크 상태를 통한 네트워크 토폴로지 계산 (라인 토폴로지)
Fig. 4 The calculation of network topology(line topology) through the link state

3.2. 오프셋 보정 (Offset Compensation)

네트워크 토포롤지의 분석이 완료되면 분산된 디바이스들 상에 가동 중인 로컬 클럭을 각각 수신함으로서 디바이스들 간의 클럭 오프셋을 계산한 후 계산된 클럭 오프셋 값을 각 디바이스들에 전송함으로서 클럭을 보정하는 과정을 수행한다.

클럭 오프셋에 대한 기본 계산식은 (1)과 같다.

$$Offset = Send\ Time - Received\ Time \quad \text{.....(1)}$$

클럭 오프셋 계산을 위하여 이더캣 마스터는 이더캣 디바이스 내에 구현된 포트 스탬프 레지스터를 가동하며 포트 스탬프 레지스터에 스탬핑된 로컬 클럭의 정보를 수신함으로서 각각의 디바이스들에 대한 클럭 및 오프셋의 계산이 가능하게 된다. 그림 5는 4개의 디바이스에 대한 클럭 오프셋 계산 과정에 대한 개념적인 구조를 나타낸다[2][3][4].

3.3. 지연 보정 (Delay Compensation)

클럭 오프셋 보정을 통하여 분산된 디바이스들의 각 클럭은 라인 지연만큼의 오차를 가지며 보정이 이루어진다. 라인 지연의 보정은 다음과 같은 절차로 이루어진다.

3.3.1. 포트 지연 계산

라인 지연의 계산을 위하여 최초 분산 된 디바이스들의 각 입출력 포트 상의 지연를 측정한다. 표준 이더캣 디바이스들은 입출력 포트 상에 메시지 수신 및 피드백 시간을 측정하는 포트 스탬프 레지스터를 가지고 있으므로 다음 식 (2)와 같이 포트 지연의 계산이 가능하다.

$$dt_i = T_{i1} - T_{i0} \quad \text{.....(2)}$$

그림 5. 클럭 동기화 모드에 따른 클럭 오프셋 계산 과정
Fig. 5 The compute process of clock offset by clock synchronization mode

4개의 디바이스를 기준으로 하였을 겨우 디바이스들의 포트 지연 계산에 대한 개념적인 구조는 그림 6와 같다.

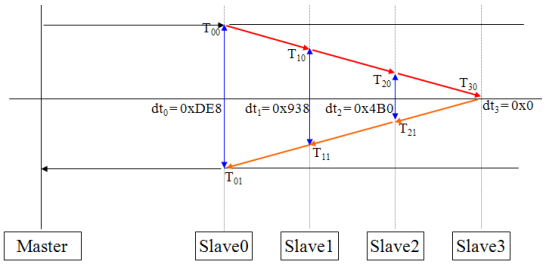

그림 6. 포트 지연 계산
Fig. 6 The compute port delay

3.3.2. 포트 지터 보상

입출력 클럭 스탬핑 시 발생하는 지터를 보상하고 포트 지연 계산 과정의 정밀도를 향상시키기 위하여 포트 지연 계산 과정은 수~수십 차례 수행한 후 평균을 취하는 방식을 적용한다.

$$dt_i = \frac{\sum_{k=1}^{n}(T_{i1_k} - T_{i0_k})}{n} \quad \text{.....(3)}$$

3.3.3. 라인 지연 계산

최종적으로 계산된 포트 각 디바이스들의 포트 지연 정보를 기반으로 실제 네트워크 상에 발생하는 라인 지연를 계산하고 각 디바이스들에 전송하는 과정을 수행한다. 라인 지연의 계산은 식(3) 과 같이 각 디바이스들 간의 포트 지연의 차를 구하면 된다.

$$ld_i = (dt_0 - dt_i)/2 \quad \text{.....(4)}$$

4개의 디바이스를 기준으로 하였을 경우 각 디바이스 간의 라인 지연 계산에 대한 개념적인 구조는 그림 7과 같다.

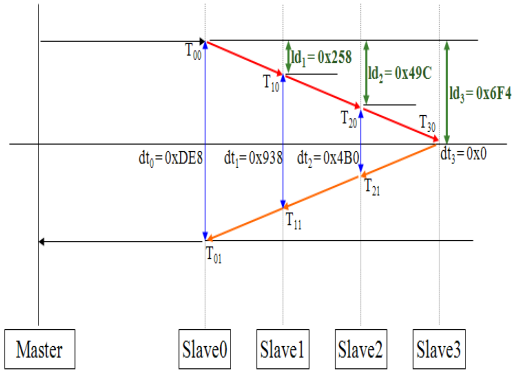

그림 7. 라인 지연 계산
Fig. 7 The compute line delay

3.4. 클럭 분산 및 동기화 (Drift Compensation)

클럭 오프셋 및 라인 지연 보정 과정을 통하여 클럭 동기화를 위한 상위 마스터의 처리과정은 완료되었다. 마지막 과정으로서 상위 제어기는 기준 클럭(Reference Clock)으로 사용되는 첫 번째 노드의 클럭을 네트워크를 통하여 주기적으로 분산 시켜 각 슬레이브 디바이스 상에 기 구현된 DC 엔진을 가동 시켜주면 된다.

기준 클럭을 수신 받은 각 클럭은 DC 제어 엔진을 가동하며 식(5)에 의해 계산된 시간 차($\triangle t$)에 따라 로컬 클럭을 가/감속시켜 클럭을 동기화 시킨다. (Drift Compensation)

$$\triangle t = (t_{localtime} + t_{offset} + t_{propagationdelay}) - t_{receivedtime} \quad \text{.....(5)}$$

상위 마스터로부터 기준클럭을 수신받은 각각의 디바이스들은 이와 같은 DC 제어를 통하여 모두 동일한 클럭으로 정렬하게 되며 동기화 완료 후 싱크(Sync) 신호 및 동기화 인터럽트 신호를 발생함으로서 다바이스들의 동기화가 이루어지게 된다. [2][3][4]

Ⅳ. 이더캣 동기화 시스템 구현 및 실험

본 연구에서는 이더캣 동기화 기술의 구현을 위하여 이더캣 기반의 서보 드라이버 4축과 소프트 리얼 타임 커널을 지원하는 INTime 이더캣 마스터를 구현하여

동기화 실험을 수행하였다. 세부적인 시스템 구조는 그림 8과 같다.

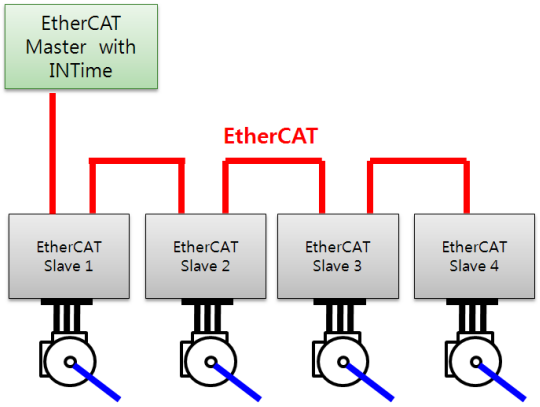

그림 8. 이더캣 동기화 테스트용 시스템 구조
Fig. 8 The system structure for test of EtherCAT synchronization

구현된 이더캣 동기화 시스템의 기본적인 사양은 표 2와 같다.

표 2. 이더캣 동기화 시스템 사양
Table 2 The specification of EtherCAT synchronization system

구분	사양	
이더캣 마스터	운용환경	Windows XP with INTime
	리얼타임	Soft Real-Time
	사이클타임	200us
	이더캣스택	EtherCAT COE Master with DC
	노드 수	4
이더캣 슬레이브	노드 간 거리	5m
	모터	MAXSON EC Motor(100W)
	타입	EtherCAT Bldc Driver
	통신속도	100Mbps
	이더캣스택	EtherCAT COE Slave with DC

이더캣 마스터 구현시 DC 동기화 처리의 개선을 위하여 DC 동기화 스택을 소프트 리얼타임 커널을 지원하는 인타임 환경 내에서 구현을 하였다. 그림 9는 실제 개발된 이더캣 마스터의 구조를 나타낸다[5][6][7].

이더캣 동기화 실험을 위하여 구현한 이더캣 네트워크 시스템을 통하여 본 연구를 통하여 구현한 이더캣 동기화 알고리즘을 동작성능에 대한 테스트를 수행하였으며 동기화 결과의 확인은 이더캣 디바이스 상에 구현된 시스템 클럭을 수신 및 디스플레이 함으로서 실제 동기화가 이루어졌는지 판단하는 방식을 적용하였으며 실험결과는 그림 10과 같다.

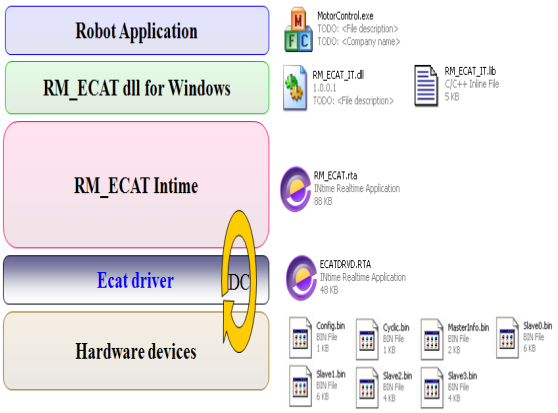

그림 9. 개발된 INTime 이더캣 마스터 아키텍처
Fig. 9 Developed INTime EtherCAT master architecture

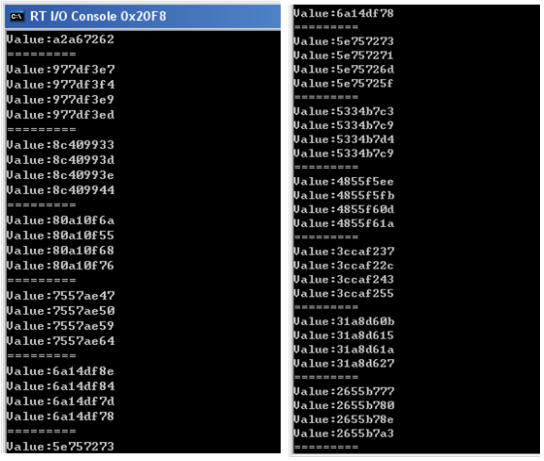

그림 10. 동기화된 이더캣 시스템 클럭
Fig. 10 Synchronized EtherCAT system clock

실험은 30회 가량 수행하였으며 그림 10의 실험 결과를 통하여 알 수 있듯이 분산된 이더캣 디바이스들의 시스템 클럭이 -10ns ~ +10ns 오차 범위를 가지고 동기화 되고 있음을 알 수 있었다.

Ⅴ. 결론 및 향후 과제

본 논문에서는 네트워크 기반 동기화 기술에 대한 전반적인 소개와 최신 산업용 이더넷 기술 중의 하나인 이더캣 통신을 이용한 분산된 디바이스들의 클럭 동기화 기술을 구현하는 방법에 대한 내용을 기술하였다. 또한 이더캣 동기화 기술의 성능평가를 위하여 소프트 리얼타임 환경을 지원하는 INTime 용 이더캣 마스터와 4축의 이더캣 BLDC 서보 드라이버를 각각 자체적으로 구현하였다. 실험 결과 이더캣 동기화에 대한 성능은 -10ns ~ +10ns 오차를 가지는 동기화 성능을 실현하였으며 이는 기존의 동기화 네트워크의 기준 성능인 1us 이내를 훨씬 상회하는 결과이므로 비교적 우수한 동기화가 구현되었음을 알 수 있었다.

본 논문에서 적용한 이더캣 시스템은 노드간의 거리가 5m 이며 노드의 수가 4축으로서 아주 소규모 시스템이라 할 수 있어 차후에는 노드간의 거리의 장거리화와 다량의 노드 시스템에 이더캣 동기화 기술을 적용한 후 그 성능평가를 실시하는 것이 과제로 남는다.

감사의 글

본 논문은 스마트친환경 가전사업에 의해 지원에 의해 수행되었음.

참고 문헌

[1] INS, "IEEE1588-Precise Time Synchronization as the Basis for Real Time Applications in Automation" 2004.
[2] EtherCAT Technology Group, "EtherCAT Specification V1.0", 2004.
[3] Beckhoff, "EtherCAT Slave Controller (ESC10/20 Hardware Data Sheet)", 2005.
[4] EtherCAT Technology Group, "EtherCAT Communicatin", 2007.
[5] EtherCAT Technology Gourpu, "IEC 61800-7 ETG Implementation Guideline for ther CiA402 Drive Profile", 2007.
[6] TenAsys, "INtime Software User Guid", 2003
[7] TenAsys, "INtime Real-time for Windows Introduction", 2005.

저자 소개

문용선(Yong-seon Moon)

1983년 2월: 조선대학교 전자공학과 졸업(공학사)
1989년 2월: 조선대학교 대학원 전자공학과 졸업(공학박사)
1992년 - 현재 순천대학교 정보통신공학부 교수
/레드원테크놀러지(주) 기술이사
※관심분야 : 산업통신망 및 로봇, 실시간 모션 제어

트롱루안

(Vo Trong Tuan Anh)

2006년 9월: Vietnam National University, University of Natural Sciences - Information Technology
2008년 2월 : 순천대학교 전자공학과(공학석사)
※관심분야 : 로봇 제어, 모터 제어, 산업통신망

이영필(Young-pil Lee)

2006년 2월: 순천대학교 전자공학과 (공학사)
2008년 2월: 순천대학교 전자공학과 (공학석사)
2008년 ~ 현재 : 레드원테크놀러지(주) 연구원
※관심분야 : 로봇 제어, 모터 제어, 산업통신망

차현록 (Hyunrok Cha)

전남대학교 전기공학과 대학원(공학석사)
동경공업대학교 물리정보시스템(공학박사, 졸업예정)
2000년 ~ 2004년: 삼성광주전자 선임연구원
2004년 ~ 현재: 한국생산기술연구원 선임연구원
※관심분야 : 엑츄에이터 설계, 초음파응용