

# 오픈소스 기반의 실시간 EtherCAT 제어 시스템의 구현

## Implementation of Real-time EtherCAT Control System based on Open Source

경윤진<sup>1</sup>, 최동일<sup>†</sup>  
Yunjin Kyung<sup>1</sup>, Dongil Choi<sup>†</sup>

**Abstract:** Real-time control communication network system is important for developing defense robots because it affects environmental interaction, performance, and safety. We propose a real-time control communication network using the Xenomai real-time operating system and the open-source EtherCAT master library, SOEM. EtherCAT is an Ethernet-based industrial communication method. It has low latency and many functions such as cable redundancy and distributed clock synchronization. We use Xenomai RTOS and Intel NUC to develop the system. Experimental tests demonstrate the Real-time EtherCAT master implementation, and communication with CiA301-based slave devices. The jitter measurement was conducted to validate the real-time performance of the system. The proposed system shows possibility for real-time robotics applications in various defense robots.

**Keywords:** EtherCAT, Real Time Control, Control Networks

### 1. 서론

4차 산업 혁명 시대에 접어들어 현대 사회에서 로봇은 다양한 산업 분야에 걸쳐 활용되고 있으며 그 범위를 국방 로봇 분야로 점차 확대해 가고 있다. 인구감소에 따른 병력손실을 막기 위한 일환으로 미래 국방 로봇의 개발은 국방력 유지 및 강화를 위한 필수 요소가 되었다.

국방 로봇을 개발하는데 있어서 실시간성은 중요한 요소이다. 실시간성은 로봇이 외부 환경과 상호작용하는 능력과도 연결되며, 더 나아가 로봇의 작업성능, 안전성, 신뢰성에 영향을 끼친다. 따라서 로봇 제어 시스템은 일정한 주기와 제어 명령을 전송하는 실시간 제어 통신 네트워크를 통해서 안정적으로 동작할 수 있어야 한다.

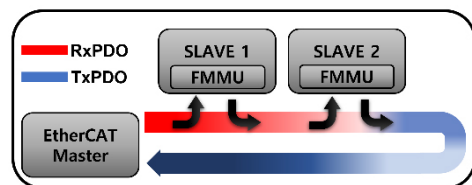
본 논문에서는 이러한 실시간 제어 통신 네트워크를 리눅스 기반의 실시간 운영체제 Xenomai와 오픈소스 EtherCAT

마스터 라이브러리인 SOEM (Simple Opensource EtherCAT Master)을 통해서 보다 신뢰성 높고 제어 성능이 뛰어난 실시간 제어 통신 네트워크 시스템을 제안한다.

### 2. EtherCAT 통신 네트워크

#### 2.1 EtherCAT

EtherCAT 통신은 Ethernet 통신 기반의 산업용 통신 방식이다. EtherCAT은 여러 산업용 Ethernet 통신 중 하나이다. [Fig. 1]과 같이 EtherCAT은 OSI 7계층 참조 모델의 물리층에 FMMU (Fieldbus Memory Management Unit)을 통해서 데이터 프레임이 각각의 슬레이브를 통과하며 각 슬레이브의 I/O 터미널에서 데이터를 직접 다루기 때문에 Latency가 거의 발생하지 않는다. 이러한 네트워크 모델이 다른 이더넷 기술과의 차별점을 가진다<sup>[1]</sup>.



[Fig. 1] Ethercat communication: FMMU data process<sup>[1,2]</sup>

Received : May. 26. 2023; Revised : Jul. 30. 2023; Accepted : Aug. 1. 2023

※ This research was supported by the Challengeable Future Defense Technology Research and Development Program through the Agency for Defense Development (ADD) funded by the Defense Acquisition Program Administration in 2021 (No.915018201)

1. Master Course, Mechanical Engineering, Myong Ji University, Yongin, Korea (yunjin544@gmail.com)

† Associate Professor, Corresponding author: Mechanical Engineering, Myong Ji University, Yongin, Korea (dongilc@mju.ac.kr)

[Table 1] EtherCAT Master Library

|                   |                           |                           |
|-------------------|---------------------------|---------------------------|
|                   | SOEM                      | IgH                       |
| Publisher         | RT-LAB                    | EtherLAB                  |
| OS                | Linux, Windows, MAC, MCU  | Linux                     |
| Supported RT Core | Xenomai, RTAI, RT-Preempt | Xenomai, RTAI, RT-Preempt |

또한 EtherCAT Redundancy(케이블 이중화), Distributed clock(분산 클럭) 기능을 지원한다. 케이블 이중화는 두개의 Ethernet 케이블을 링 토폴로지 연결하게 되며, 두개의 케이블 중 한 개의 케이블이 손실되더라도 라인 토폴로지로 통신 시스템을 전환 및 유지하여 통신의 신뢰성을 높이고 이상이 발생할 슬레이브를 추정할 수 있도록 한다. 분산 클럭 기능은 각 슬레이브 마다의 작은 지연마저 보정하여 같은 스텝의 제어 명령이 동시에 실행될 수 있게 한다.

### 2.2 EtherCAT Master Library

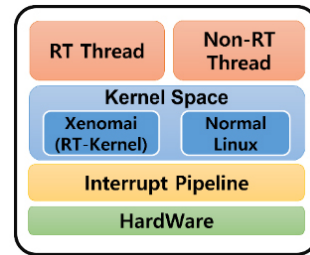
EtherCAT 마스터를 개발하기 위한 오픈소스 프로젝트는 크게 SOEM과 IgH 프로젝트가 있다. 각 라이브러리 프로젝트 발행 기관과 개발 환경은 [Table 1]과 같다.

SOEM에 경우 유저 레벨에서 동작하는 반면, IgH의 경우 커널 레벨에서 동작하기에 성능 자체는 우수한 점을 가지고 있다. 하지만 IgH는 지원하는 랜카드가 제한적이고 개발 환경 구성이 복잡한 단점이 존재하며, 프로그램의 동작이 라이브러리보다 소프트웨어가 가까운 성격을 띄고 있다. 반면 SOEM은 EtherCAT 라이브러리로서 다양한 플랫폼에 이식이 용이하며 실시간 EtherCAT 구현을 선택적으로 개발할 수 있는 장점을 가지고 있다. 본 논문에서는 추후 여러 개발 환경과 플랫폼에 대한 로봇 시스템 개발을 지원하기 위해서 호환성이 높은 SOEM을 사용하였다<sup>[3-6]</sup>.

## 3. 실시간 EtherCAT 마스터 스택

### 3.1 실시간 운영체제(RTOS) : Xenomai

정밀한 시간 정확성을 요구하는 프로그램을 작성하는 기법을 실시간 프로그래밍이라 한다. 일반적인 운영체제는 사용자로 하여금 정밀한 시간 정확도 및 정밀도를 제공하지 않는다. 따라서 실시간 프로그래밍을 위해서는 실시간 운영체제를 별도 운영할 수 있어야 한다. 리눅스에서는 이를 위해서 오픈소스로 개발된 실시간 운영체제 패치 프로젝트들이 존재한다. Xenomai, RTAI, PREEMPT\_RT가 있다. 본 논문에서는 실시



[Fig. 2] Structure of linux patched xenomai

간 Ethernet 드라이버 프로젝트인 RT-Net을 활용하기 위해서 Xenomai를 이용하여 실시간 패치를 진행하였다<sup>[4]</sup>. [Fig. 2]와 같이 Xenomai 패치는 기존 리눅스 커널 공간에 실시간 프로그램들을 관리하는 커널이 추가되어 공존하는 형태로 패치가 진행된다.

패치시에 RT-Net 옵션을 활성화하여 실시간 네트워크 드라이버가 설치될 수 있도록 해야 하며, RT-Net 드라이버는 지원되는 네트워크 카드가 별도로 존재한다.

### 3.2 Implementation of EtherCAT Master

본 논문에서 EtherCAT 마스터를 개발 하기위해서 사용된 Master PC는 Intel사의 NUC 9 Pro Kit Mini PC이다. 해당 PC 사양은 [Table 2]와 같다<sup>[5]</sup>.

기존에 설치된 네트워크 인터페이스 카드 중 i210-AT은 RT-Net을 지원한다. 다만 추후 확장성을 위해서 4 Port 모두 RT-Net이 지원되는 Intel I340-T4 카드 별도로 구성하였다.

개발에 사용된 소프트웨어의 버전은 [Table 3]과 같다.

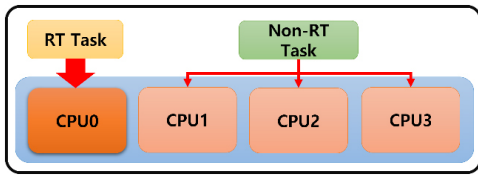
본 논문에서는 위와 같은 환경에서 개발을 진행하였으며, GUI 프로세스에 의한 Latency를 방지하기 위해서 리눅스 OS

[Table 2] Master PC Specification

|                                     |   |
|-------------------------------------|---|
| CPU                                 | Intel Core i7-9850H   |
| Network Interface Card (On board)   | Intel® Ethernet Connection i219-LM Intel® Ethernet Connection i210-AT |
| Network Interface Card (PCI-e Slot) | Intel® Ethernet Connection 210-AT (Intel I340-T4 4Port)               |
| Memory/GB                           | 16  |

[Table 3] Version of development tool

|                      |                        |
|----------------------|------------------------|
| OS                   | Ubuntu 20.04 Sever LTS |
| Linux Kernel         | 5.4.180                |
| RT-Core              | Xenomai 3.2.1          |
| SOEM                 | 1.4.0                  |
| Build Tool           | CMake, Ninja           |
| Programming Language | C++                    |



[Fig. 3] CPU isolation

인 Ubuntu 20.04 Server LTS에서 진행하였다. 프로그래밍 언어를 C++로 선정하며 마스터 프로그램을 작성하였다.

다른 프로세스에 의한 예기치 못한 부하로 인한 Latency를 방지 위해서, [Fig. 3]과 같이 리눅스의 코어 분리 기능을 이용하여 EtherCAT 마스터 프로세스는 분리된 CPU 코어에 실행되도록 작성하였다. 또한 통신 프로세스와 로봇 제어 프로세스간 통신을 위해 통신 방법 중 하나인 Shared Memory를 이용하였다.

### 4. 실험 환경

#### 4.1 CPU-isolation Latency Test

실시간 패치가 된 리눅스에서 Xenomai가 제공하는 Latency test와 Cyclic test를 실행하여 CPU-isolation의 효과를 확인해 보기 위해서 CPU-isolation을 진행한 경우와 진행하지 않은 경우에 대해서 비교 실험을 계획하였다.

실험은 Stress라는 CPU 부하 프로그램을 실행한 상태에서 Latency test 와 Cyclic test를 실행하여 각 코어가 활성화되는 양상을 비교하였다.

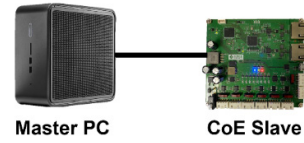
#### 4.2 EtherCAT Master Test

구성한 EtherCAT 마스터의 성능을 검증하기 위해서 아래와 같은 실험 방법을 진행하였다.

마스터 PC에서 Slave 장치에게 명령을 1 kHz의 전송 주기를 통해서 MCU에 연결된 LED를 켜고 끄를 반복하는 Toggle 제어 명령을 전송하여 이를 산업용 오실로스코프의 Jitter 분석 기능을 이용하여 측정 및 비교하였다. 이는 SPI통신 및 MCU의 제어 주기에 따른 지연이 포함될 수 있으므로 상대적인 지표를 얻기 위한 실험을 진행하였다. 구성 실험 환경에 대한 개략도는 [Fig. 4]와 같다.



[Fig. 4] Environment of ethercat master test



[Fig. 5] Environment of CoE master test

#### 4.3 CoE (CANopen over EtherCAT) Master 개발

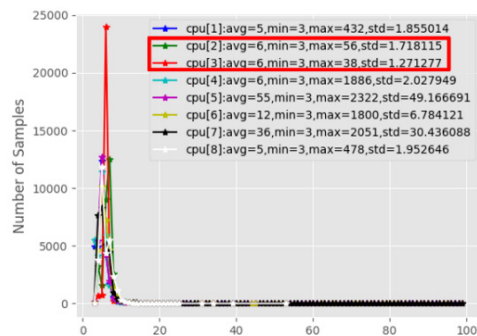
구성한 EtherCAT 마스터의 모터드라이브 및 I/O 모듈과 같은 상용 제어기와 통신을 위해서 CiA301 기반의 테스트 프로그램을 작성하여 테스트를 진행하였다. 슬레이브 장치는 [Fig. 5]와 같이 EtherCAT 통신과 다른 통신 방법간 연결을 해주는 EtherCAT 브릿지 보드를 통해서 진행하였다. 테스트는 전송데이터를 PDO 프로토콜을 통해서 매핑된 데이터를 전송하면 응답으로 1이 더해진 데이터를 반환하는 것을 확인함으로써 CoE 프로토콜의 구현을 확인하였다.

### 5. 실험 결과

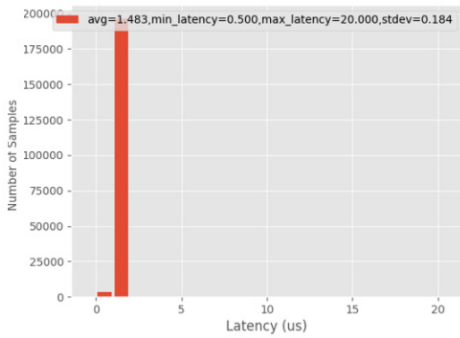
[Fig. 6]과 같이 Cyclictest를 진행한 결과 CPU-isolation을 진행하였을 때, Normal 부트와 비교한 결과 해당 실험에서 분리된 코어 2,3에 대해서 Max latency가 감소한 것을 확인할 수 있다. 분리된 코어만의 Latency 변화 양상을 더욱 면밀히 보기 위해서 Latency test를 해당 코어 할당하여 비교했다.

[Fig. 7], [Fig. 8]과 같이 Latency test를 진행한 결과 Normal Boot에서 측정된 최대 Latency가 20us에서 CPU-isolation을 진행한 경우에는 16us로 감소한 것을 확인할 수 있었다.

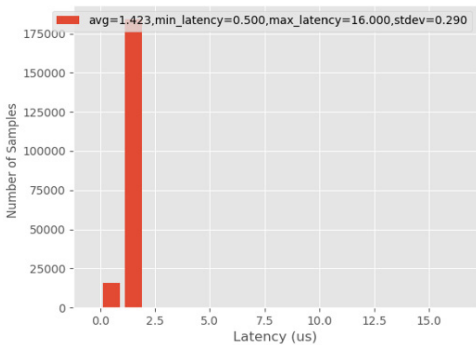
[Fig. 9], [Fig. 10]에는 구성한 EtherCAT 마스터의 모터드라이브 및 I/O 모듈과 같은 상용 제어기와 통신을 위해서 CiA301 기반의 테스트 프로그램을 작성하여 테스트를 진행하였다. 슬레이브 장치는 EtherCAT 통신과 다른 통신 방법간 연결을 해주는 EtherCAT 브릿지 보드를 통해서 진행하였다. 테스트는 전송데이터를 PDO 프로토콜을 통해서 매핑된 데이터를 전송하면 응답으로 1이 더해진 데이터를 반환하는 것을 확인함으



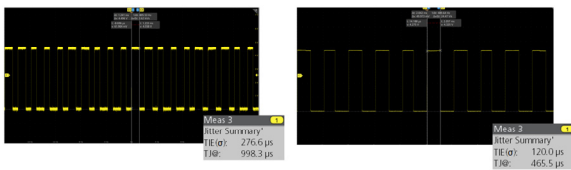
[Fig. 6] Result of CPU-isolation cyclictest



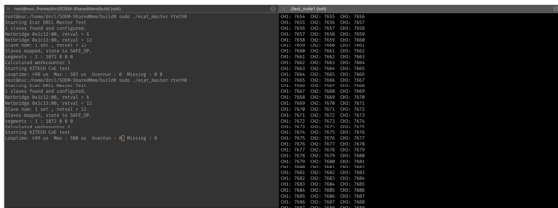
[Fig. 7] Result of normal linux latency test



[Fig. 8] Result of CPU-isolation latency test



[Fig. 9] Result of ethercat master test



[Fig. 10] Result of CoE master test

로써 CoE 프로토콜의 구현을 확인하였다.

작성한 EtherCAT 마스터를 기반으로 CiA301 기반의 슬레이브와 PDO 통신이 정상적으로 이뤄지는 것을 확인했으며, EtherCAT 프로세스가 원하는 목표 주기를 가지고 구동하는 것을 확인했다.


## 6. 결 론

Xenomai와 Stress 패키지를 이용하여 실시간성을 검증하였으며, Xenomai API를 통한 실시가 리눅스 패치를 통해서 EtherCAT 마스터를 구현하였다.

실제 제어까지 성능 평가를 위해서 산업용 오실로스코프를 통해 Toggle 제어 명령 실험 진행하였으며 결과를 통해서 실시간 EtherCAT 마스터가 구현됨을 확인하였다. 또한 산업용 표준 CANopen 프로토콜 중 CiA301 기반의 CoE 슬레이브와 정상적인 통신을 확인함으로써, 상용 모터 제어(CiA402) 및 I/O 모듈 (CiA401)을 정상적으로 동작이 가능함을 확인하였다.

## References

- [1] EtherCAT Technology Group, *EtherCAT Overview & ETG Introduction*, [Online], <https://www.ethercat.org>, Accessed: Aug. 10, 2023.
- [2] MotionControl, *Understanding Industrial Network EtherCAT*, [Online], <https://www.motioncontrol.co.kr>, Accessed: Aug. 10, 2023.
- [3] S. Y. Kim and E. C. Shin, "Performance Verification of Opensource-based EtherCAT Master by Linux Platform," *Journal of the Korean Society of Precision Engineering Conference*, [Online], vol. 2017, no. 5, pp. 149-150, May, 2017, [Online], <http://www.riss.kr/link?id=A103271379>.
- [4] IgH, *EtherLab*, [Online], <https://gitlab.com/groups/etherlab.org/>, Accessed: Aug. 16, 2023.
- [5] T.-Y. Son and S.-R. Rim, "An Experimental Performance Evaluation with Xenomai for WSN," *Journal of the Korea Academia-Industrial cooperation Society*, vol. 18, no. 1, pp. 709-714, Jan., 2017, DOI: 10.5762/KAIS.2017.18.1.709.
- [6] S. J. Kang, "A Study on Implementation of Real-time EtherCAT Master," *Journal of the Semiconductor & Display Technology*, vol. 20, no. 2, pp. 131-136, Jun., 2021, [Online], <https://korea-science.kr/article/JAKO202121137920799.page>.



**경 윤 진**  
 2022 명지대학교 기계공학과(학사)  
 2022~현재 명지대학교 기계공학과 석사과정

관심분야: 제어 네트워크, 로봇 제어, 임베디드 시스템



**최 동 일**  
 2005 한국과학기술원 기계공학과(학사)  
 2007 한국과학기술원 기계공학과(석사)  
 2012 한국과학기술원 기계공학과(박사)  
 2018~현재 명지대학교 기계공학과 교수

관심분야: 국방로봇, 족형로봇, 로봇 제어, 임베디드 시스템, 이동로봇