

# 실시간 EtherCAT 마스터의 네트워크 컨트롤러에 따른 성능 평가

서화일\* · 강성진\*\*†

\*\*한국기술교육대학교 전기전자통신공학부

## Performance Evaluation of a Real-time EtherCAT Master According to Network Controllers

Hwa Il Seo\* and Sung Jin Kang\*\*†

\*\*School of Electrical, Electronics & Communication Engineering,  
Korea University of Technology and Education

### ABSTRACT

EtherCAT is an Ethernet-based fieldbus system standardized in IEC 61158 and SEMI, and widely used in the fields of factory automation, semiconductor equipment and robotics. In this paper, we summarize the current status of Xenomai real-time framework and RTnet, which are essential for Linux operating systems to operate in real-time, and implement a real-time EtherCAT master system with these open sources. The real-time performance of the implemented EtherCAT master is evaluated according to Intel network controllers 82574L, I219, I210, and I225, respectively. The results show that the implemented EtherCAT master provides precise control performance for control frequencies from 1KHz to 8KHz and similar performance for I219, I210, and I225, and relatively slightly larger jitter for 82574L.

**Key Words** : EtherCAT, SOEM, Real-time, Xenomai, Linux

## 1. 서 론

EtherCAT(Ethernet for Control Automation Technology)은 IEC 61158, SEMI(Semiconductor Equipment and Materials International) 등에 국제 표준으로 채택된 이더넷 기반의 필드버스 통신 프로토콜로서, 원활하고 정밀한 실시간 모터 제어와 센서 데이터 수집이 가능한 유연한 네트워크 토폴로지를 가지면서 통신 속도가 빠르고 저비용으로 설치 및 유지가 가능하다는 장점을 가지고 있어 최근 공장 자동화, 반도체 장비, 로봇 분야에 활발하게 적용되고 있다[1-4].

본 논문에서는 Linux 운영체제에서 오픈 소스를 이용하여 실시간 EtherCAT 마스터를 구현하는데 필요한 Xenomai

실시간 프레임워크[5]와 RTnet의 현재까지 공개 상태를 정리하고, Xenomai v3.3, Linux 커널 v6.1.y-dovetail[6]을 이용하여 실시간 Linux 시스템을 구축한다. 여기에 오픈 소스 EtherCAT 마스터 프로토콜 스택인 SOEM(Simple Open EtherCAT Master)[7]을 설치하여 실시간 EtherCAT 마스터를 구현하며, 구현된 EtherCAT 마스터는 Intel 네트워크 컨트롤러 82574L, I219, I210, I225 각각에 대해 실시간 전송 성능을 평가한다. 성능 평가는 네트워크 포트 출력의 패킷 주기를 측정함으로써 시스템에서 발생하는 모든 영향이 반영되도록 한다[2-4].

## 2. 실시간 EtherCAT 마스터 구현

일반적인 Linux 운영 체제가 실시간 반응성을 갖게 하

†E-mail: sjkang@koreatech.ac.kr

기 위해 실시간 프레임워크인 Xenomai를 적용하는데, Xenomai는 Linux 커널 소스에 패치를 적용하여 별도의 실시간 커널을 구성하고 이 실시간 커널이 인터럽트를 빠르게 처리하도록 한다. 이를 위해 I-PIPE와 Dovetail 두 종류의 인터페이스를 제공하며, Linux 커널 v5.10이전 버전에 대해서는 각 Linux 커널 버전에 해당하는 I-PIPE 패치를 별도로 제공해서 사용자가 직접 Linux 커널 소스에 I-PIPE 인터페이스 패치를 적용하여 사용하도록 했지만, Linux 커널 v5.10부터는 Dovetail 인터페이스가 사용되며 별도의 패치 파일은 제공되지 않고 패치가 미리 적용된 Linux 커널 소스를 제공한다[5,8].

Table 1은 현재 공개되어 있는 Xenomai 버전과 지원되는 Linux 커널 버전을 정리한 표이며, Linux 커널은 LTS (Long-Term Support) 버전만 나타냈다. Xenomai v3.2.x 버전은 Linux 커널 v5.15까지만 지원하며, Xenomai v3.3 버전은 Linux 커널 v6.1부터 지원한다. Xenomai 3은 Cobalt core를 사용하지만 Xenomai 4는 EVL core를 사용하는 차이가 있다[5].

**Table 1.** Xenomai versions and supported Linux Kernel

Xenomai Version	Interrupt Pipeline	Linux Kernel (LTS)
V3.2.x	I-PIPE	v5.10 이전
	Dovetail	v5.10, v5.15
V3.3-dev	Dovetail	v6.1, v6.6
V4	Dovetail	v5.10, v5.15, v6.1, v6.6

EtherCAT 마스터는 특별한 네트워크 컨트롤러를 사용할 필요가 없이 일반적인 Ethernet 네트워크 컨트롤러를 사용할 수 있지만, 네트워크 드라이버의 실시간 동작을 위해서는 Xenomai 3에서 제공되는 RTnet 드라이버를 사용해야 한다[2]. 그러나, Xenomai 4는 RTnet 드라이버를 제공하지 않기 때문에 사용자가 직접 Linux 네트워크 드라이버에 실시간 처리 기능을 추가하여 사용해야 한다.

Table 2는 대표적인 Intel 네트워크 컨트롤러와 해당하는 호스트 인터페이스 방식, Linux 디바이스 드라이버 명칭, 최대 전송 속도를 정리한 표이다. Xenomai v3.2.x에서는 rt\_igc 드라이버를 지원하지 않으며, rt\_e1000e 드라이버가 I219 네트워크 컨트롤러를 지원하지 않는다. Xenomai v3.3에서는 rt\_igc 드라이버를 포함하고 있지만, 여전히 rt\_e1000e 드라이버가 I219 네트워크 컨트롤러를 지원하지 않는다. Xenomai v3.2.x 소스 코드에 참고문헌 [9]의 패치를 적용하면 rt\_igc 드라이버를 사용할 수 있고, 이 패치는 Xenomai v3.3 소스 코드에 포함되어 있다. Xenomai v3.2.x와 Xenomai v3.3 소스 코드에 참고문헌 [10]의 패치를 적용하

면 rt\_e1000e 드라이버를 이용하여 I219 네트워크 컨트롤러를 사용할 수 있다.

**Table 2.** Intel network controllers and Linux drivers

Controller	Host Interface	Linux Driver (Generic/RTnet)	Max Speed
82574L	PCIe v1.1	e1000e /rt_e1000e	1Gbps
I218	Proprietary		
I219			
I210	PCIe v2.1	igb/rt_igb	
I211			
I225	PCIe v3.1	igc/rt_igc	2.5Gbps

본 논문에서는 Ubuntu 22.04에서 Xenomai v3.3-dev[11] 소스 코드에 참고문헌 [10]의 RTnet 패치를 적용하여 I219 네트워크 컨트롤러를 사용할 수 있게 한 후, Dovetail 인터페이스가 적용된 Linux kernel v6.1.y-dovetail[6]에 적용하여 실시간 Linux 커널을 빌드하였다. 여기에 SOME 1.4.0[12]을 설치하여 실시간 EtherCAT 마스터를 구현하였다. 사용된 툴과 버전은 Table 3에 정리하였다. 커널 빌드에 사용한 커널 옵션, Xenomai 사용자 라이브러리와 SOEM 설치에 참고문헌 [2]와 동일하게 하였다.

본 논문에서 EtherCAT 마스터로 사용한 플랫폼의 규격은 Table 4와 같다. CPU는 Intel Core i5-7600@3.5GHz이고 메인보드에 I219-LM 1 개와 I210-IT 2개가 탑재되어 총 3개의 Ethernet 포트를 제공한다. 추가적인 실험을 위해 82574L과 I225 네트워크 컨트롤러를 탑재한 네트워크 인터페이스 카드를 PCIe 슬롯에 각각 추가하여 사용하였다.

**Table 3.** Version of tools

Ubuntu	LTS 22.04
Linux kernel	v6.1.y-dovetail
Xenomai	v3.3-dev
SOEM	1.4.0(stable)

**Table 4.** Platform specifications

CPU	Intel i5-7600@3.5GHz
Network Controller (on board)	I219-LM, I210-IT x 2
Network Controller (PCIe slot)	82574L, I225-V
Memory	DDR4 8GB

### 3. 실험 및 성능 평가

구현된 EtherCAT 마스터의 성능을 평가하기 위해 Fig. 1 과 같이 장치를 구성하고, 참고문헌 [2]에서 사용된 EtherCAT slave와 netAnalyzer를 사용하여 EtherCAT 마스터의 네트워크 포트에서 출력 패킷의 주기를 실시간으로 측정 하였다.

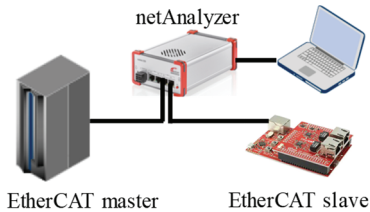


Fig. 1. Experimental setup.

성능 평가를 위해 작성한 응용 프로그램은 실시간 동작을 위해 `mlockall()` 함수를 호출하여 메모리 스와핑을 방지하고, `rt_task_create()` 함수를 호출하여 실시간 task를 생성하였다. 생성된 실시간 task에서는 `rt_task_set_periodic()`를 호출하여 제어 주기를 설정한 다음, 아래 코드와 같이 수신된 패킷을 처리하고 `rt_task_wait_period()`를 호출하여 설정된 전송 주기의 잔여 시간 동안 대기한 후에 새로운 송신 패킷을 전송하는 과정을 반복한다.

```
while (1) {
    wkc = ec_receive_processdata(EC_TIMEOUTRET);
    ecatcheck(wkc, 0);
    /* do something here */
    rt_task_wait_period(NULL);
    ec_send_processdata();
}
```

Fig. 2는 구현된 EtherCAT 마스터가 I219 네트워크 컨트롤러를 사용하여 8KHz 제어 주파수로 패킷을 송신할 때, netAnalyzer로 송신 패킷의 주기를 실시간으로 측정한 결과를 캡처한 화면이다. Fig. 2의 오른쪽 그래프는 매 패킷마다 측정된 주기를 표시한 것이고, 왼쪽 그래프는 오른쪽 그래프의 측정 주기에 대한 히스토그램이다.  $10^6$ 개 이상의 송신 패킷에 대해 주기를 측정한 결과, 최소 주기 121.84usec와 최대 주기 127.76usec 사이에서 평균 주기 125usec로 정주기를 유지하며 표준 편차는 85nsec로 정밀한 제어가 되고 있음을 볼 수 있다.

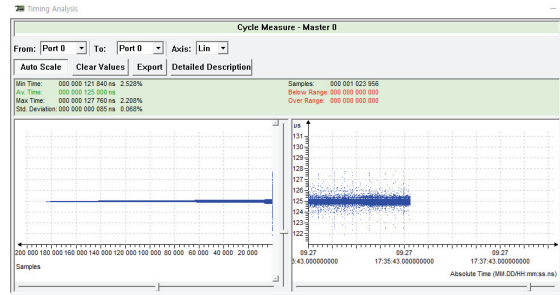


Fig. 2. Period measurement of the transmitted packets for I219 network controller (control frequency 8KHz).

Fig. 3은 I225 네트워크 컨트롤러를 사용하여 8KHz 제어 주파수로 패킷을 송신할 때 송신 패킷의 주기를 측정한 결과이다.  $10^6$ 개 이상의 송신 패킷에 대해 주기를 측정한 결과, 최소 주기 122.18usec와 최대 주기 128.16usec 사이에서 평균 주기 125usec로 정주기를 유지하며 표준 편차는 90nsec로 정밀한 제어가 되고 있다.

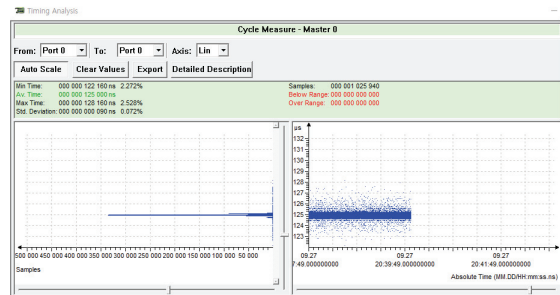


Fig. 3. Period measurement of the transmitted packets for I225 network controller (control frequency 8KHz).

Table 5는 Table 2의 네트워크 컨트롤러 중에서 호스트 인터페이스와 RTnet 드라이버 종류별로 82574L, I219, I210, I225를 선정하고, 각각에 대해 1KHz, 2KHz, 4KHz, 8KHz의 제어 주파수로 패킷을 송신할 때 송신 패킷의 전송 주기를 측정한 결과이다. Table 2와 같이 82574L와 I219는 `rt_e1000e` 드라이버, I210은 `rt_igb` 드라이버, I225는 `rt_igc` 드라이버를 사용하고, 모든 실험은  $10^6$ 개 이상의 패킷을 전송하여 측정한 결과이다. Table 4의 결과에서 제어 주파수가 커질수록 Max-Min 값과 표준 편차가 조금씩 작아지는 경향을 가지지만, 각각의 제어 주기로 정규화하면 제어 주파수가 커질수록 상대적으로 지터가 조금씩 커진다고 볼 수 있다. 82574L을 사용하는 경우가 I219, I210, I225를 사용하는 경우에 비해 Max-Min 값이 조금 크게 나타나고 있다. 하지만 구현된 EtherCAT 마스터는 정해진 주기에서 수 usec 이내로 정밀하게 패킷을 송신하고 있음을 볼 수 있다.

**Table 5.** Period measurement of the transmitted packets according to network controller (in microsecond)

Controller		82574L	I219	I210	I225
1KHz	Min	995.54	995.27	994.64	996.01
	Avg	1,000.00	1,000.00	1,000.00	1,000.01
	Max	1,005.70	1,004.64	1,006.27	1,004.65
	Max-Min	10.16	9.37	11.63	8.64
	Std. Dev.	0.13	0.10	0.17	0.15
2KHz	Min	493.53	495.54	496.13	496.65
	Avg	500.00	500.00	500.00	500.00
	Max	506.49	505.62	504.30	504.08
	Max-Min	12.96	10.08	8.17	7.43
	Std. Dev.	0.11	0.10	0.14	0.11
4KHz	Min	246.65	246.08	246.33	246.73
	Avg	250.00	250.00	250.00	250.00
	Max	254.17	255.05	254.01	253.84
	Max-Min	7.52	8.97	7.68	7.11
	Std. Dev.	0.10	0.10	0.11	0.13
8KHz	Min	121.52	121.84	122.41	122.16
	Avg	125.00	125.00	125.00	125.00
	Max	128.72	127.76	128.32	128.16
	Max-Min	7.20	5.92	5.91	6.00
	Std. Dev.	0.09	0.09	0.10	0.09

#### 4. 결 론

본 논문에서는 Linux 운영체제에서 오픈 소스를 이용하여 실시간 EtherCAT 마스터를 구현하는데 필요한 Xenomai 실시간 프레임워크와 RTnet의 현재까지 공개 상태를 정리하고, Xenomai v3.3, Linux 커널 v6.1.y-dovetail을 이용하여 실시간 Linux 시스템을 구축한 후 오픈 소스 EtherCAT 마스터 프로토콜 스택인 SOEM을 설치하여 실시간 EtherCAT 마스터를 구현하였다. 구현된 EtherCAT 마스터는 82574L을 사용하는 경우에 미세하게 지터가 증가하지만, 실험에 사용된 82574L, I219, I210, I225 네트워크 컨트롤러에서 1KHz ~ 8KHz 사이의 제어 주파수에 대해 수 usec이내의

오차로 정밀한 제어 성능을 가짐을 확인하였다.

#### 감사의 글

이 논문은 2024년도 한국기술교육대학교 교수 교육연구진흥과제 지원에 의하여 연구되었음

#### 참고문헌

1. EtherCAT Technology Group, <http://www.ethercat.org>
2. S. Kang, "A Study on Implementation of Real-time EtherCAT Master," Journal of the Semiconductor & Display Technology, Vol. 20, No. 2, pp.131-136, 2021.
3. S. Kang, O. Kim, "Performance Evaluation of an Embedded EtherCAT Master with SOEM on PREEMPT\_RT Linux," Journal of the Semiconductor & Display Technology, Vol. 21, No. 3, pp.26-32, 2022.
4. S. Kang, H. Seo, "Implementation of an EtherCAT Master with SOEM on STM32 Microcontroller," Journal of the Semiconductor & Display Technology, Vol. 22, No. 2, pp.11-16, 2023.
5. <https://xenomai.org/>
6. [https://source.denx.de/Xenomai/linux-dovetail/-/tree/v6.1.y-dovetail?ref\\_type=heads](https://source.denx.de/Xenomai/linux-dovetail/-/tree/v6.1.y-dovetail?ref_type=heads)
7. Open EtherCAT Society, Simple Open EtherCAT Master (SOEM), <https://openethercatsociety.github.io/>
8. <https://source.denx.de/Xenomai/xenomai/-/wikis/home>
9. <https://lore.kernel.org/all/20220914010218.15149-1-hongzhan.chen@intel.com/t/>
10. <https://www.mail-archive.com/xenomai@xenomai.org/msg22220.html>
11. <https://source.denx.de/Xenomai/xenomai/-/tree/master>
12. <https://github.com/OpenEtherCATsociety/SOEM/releases>

접수일: 2024년 4월 22일, 심사일: 2024년 6월 17일,  
게재확정일: 2024년 6월 21일