

# PREEMPT\_RT Linux에서 SOEM을 이용하는 임베디드 EtherCAT 마스터 성능 평가

강성진<sup>\*†</sup> · 김외철<sup>\*\*</sup>

<sup>\*†</sup>한국기술교육대학교 전기전자통신공학부, <sup>\*\*</sup>(주)네오헬스테크놀로지

## Performance Evaluation of an Embedded EtherCAT Master with SOEM on PREEMPT\_RT Linux

Sung Jin Kang <sup>\*†</sup> and Oe Cheol Kim <sup>\*\*</sup>

<sup>\*†</sup>School of Electrical, Electronics & Communication Engineering,  
Korea University of Technology and Education,

<sup>\*\*</sup> Neo Health Technology, LTD.

### 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, an EtherCAT master is implemented on an embedded board with Arm based 64-bit quad-core processor and its jitter performance is evaluated at the output of the network interface to include all the effects of the entire system in the results. For the EtherCAT master system, an open source EtherCAT master stack, Simple Open EtherCAT Master (SOEM), is installed on PREEMPT\_RT patched Linux operating system for real-time operation. The results show that the jitter performance is comparable to that of Xenomai-based master and the EtherCAT master with two master instances has similar jitter performance to the EtherCAT master with one master instance.

**Key Words** : EtherCAT, Linux, PREEMPT\_RT, Real-time, SOEM

## 1. 서 론

EtherCAT은 Ethernet기반의 필드버스 시스템으로 2007년에 국제 표준인 IEC 61158과 국제 반도체 장비 재료 협회인 SEMI의 표준으로 채택되었고, 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 분야에서도 활발하게 적용되고 있다[1,2].

일반적인 Linux 커널은 시스템의 전체 처리 능력을 최

대화하도록 설계되어 있기 때문에, 실시간 처리가 요구되는 환경에 적합하지 않다. 따라서, Linux용 실시간 프레임 워크인 Xenomai를 설치하여 사용하거나, PREEMPT\_RT 패치를 적용하여 사용해야 한다[3]. Xenomai를 사용하는 경우가 실시간 반응성과 정주기 제어 성능이 더 우수하지만, 네트워크 디바이스 드라이버를 RTnet 드라이버를 사용하지 않고 리눅스 커널의 generic 디바이스 드라이버를 사용하는 경우에는 실시간 반응성이 떨어져서 제어 주기에 큰 지터가 발생한다[2]. 따라서, Xenomai의 RTnet이 지원되지 않는 네트워크 컨트롤러를 사용하는 임베디드 보드에서는 Xenomai의 장점을 유지하기가 어려워진다.

본 논문에서는 ARM 기반 64-bit Quad-core 프로세서를 사용하는 임베디드 보드에 PREEMPT\_RT 패치된 Linux 운영

<sup>†</sup>E-mail: sjkang@koreatech.ac.kr

체제와 공개 소스 EtherCAT 마스터 스택인 Simple Open EtherCAT Master (SOEM)을 설치하여 EtherCAT 마스터 시스템을 구현하고 지터 성능을 평가한다. 또한, 한 프로세서에 2개의 마스터 인스턴스(instance)를 가지는 EtherCAT 마스터를 구현하고 지터 성능을 평가한다. 지터 성능 평가는 임베디드 보드의 네트워크 포트에서 출력되는 packet의 주기를 측정하여 EtherCAT 마스터 시스템에서 발생하는 모든 영향이 반영되도록 한다.

## 2. 임베디드 EtherCAT 마스터

본 논문에서 EtherCAT 마스터로 사용한 임베디드 보드는 NXP LS1043A 프로세서의 reference design board인 LS1043ARDB이며 주요 사양은 Table 1과 같다[4].

**Table 1.** LS1043ARDB specifications

Processor	LS1043A (64-bit quad-core processor based on Arm® Cortex®-A53 @1.60GHz)
Memory	DDR4 2GB (32-bit DDR4 bus)
SD Card	16GB
Network controller	RGMI: RTL8211FS QSGMI: VSC8514XMK-11 XFI: AQR105-B1
Ethernet port	RGMI: 1 Gigabit Ethernet x2 QSGMI: 1 Gigabit Ethernet x4 XFI: 10 Gigabit Ethernet x1
Serial port	RS232 x2 with RJ45 jack
Size	170mm x 170mm

프로세서LS1043A는 Arm Cortex-A53 기반의 64-bit quad-core이며 최대 동작 주파수는 1.6GHz이다. LS1043ARDB에는 512MB NAND 플래시 메모리와 128MB NOR 플래시 메모리가 있지만 개발의 편의를 위해 16GB SD card 플래시 메모리를 사용한다. 이더넷 포트는 EtherCAT버스용 2개, 호스트 연결용 1개로 총 3개를 사용하며, 시리얼 포트1개는 호스트와 연결하여 시리얼 콘솔로 사용한다.

### 2.1 Target 이미지 생성

크로스 컴파일을 위한 호스트 PC는 Ubuntu 20.04 LTS에 gcc-aarch64-linux-gnu패키지를 설치하여 사용하였다.

Layerscape Software Development Kit (LSDK)는 NXP사의 Layerscape와 QorIQ 계열 프로세서에서 리눅스 사용이 가능하도록 드라이버, 툴, 라이브러리를 제공하는 SDK로서, LSDK 21.08이 최신 버전이며 flexbuild\_ldsk2108.tgz를 [5]에서 다운로드할 수 있다. 이 파일을 압축 해제한 후 flexbuild 툴

을 사용하여 소스를 빌드하면 Target 이미지를 생성할 수 있다. PREEMPT\_RT Linux Kernel을 사용하기 위해 configs/sdkyml에서 커널의 git repository tag를 LSDK-21.08-RT로 수정하였다[6]. 본 논문에서는 Target의 루트 파일 시스템을 Ubuntu:main으로 선택하고, 커널 옵션을 Table 2와 같이 추가로 설정하여 Linux 커널을 빌드하였다.

**Table 2.** Linux kernel configurations

Kernel Features --->
Timer frequency --->
(x) 1000Hz
[ ] Xen guest support on ARM64 (Disable)
ACPI Support --->
<> Processor (Disable)
CPU Power Management --->
CPU Idle --->
[ ] CPU idle PM support (Disable)
CPU Frequency scaling --->
[ ] CPU Frequency scaling (Disable)
[ ] Virtualization (Disable)
Memory Management options --->
[ ] Allow for memory compaction (Disable)

커널을 빌드한 후에 부트 이미지, 루트 파일 시스템, 부트로더(U-boot) 이미지를 생성하고, flex-installer를 이용하면 생성된 이미지를 SD card 메모리에 퓨징할 수 있다[6]. 이 SD card 메모리를 Target 보드에 삽입 후 [7]에서와 같이SD card 부트 모드로 설정하여 부팅하면 Linux 커널 버전이 5.10.35-rt39임을 확인할 수 있다.

### 2.2 EtherCAT 마스터 스택

공개 소스 EtherCAT 마스터 스택인SOEM은 사용자 영역에서 동작하는 응용 프로그램이며, 소스 코드는 [8]에서 다운로드하고 Target 보드로 옮겨 native환경에서 빌드하였다. PREEMPT\_RT linux 커널에서는 소스 코드의 변경없이 다음과 같이 빌드할 수 있다[9].

```
# tar -zxf SOEM-1.4.0.tar.gz && cd SOEM-1.4.0
# mkdir build && cd build
# cmake ..
# make && make install
```

여기에서, '#'는 Target 보드의 리눅스에서 수행되는 명령어임을 의미한다. 위와 같이 명령어를 수행하면 install 디

렉토리에 라이브러리, 헤더, 실행 파일이 생기며, 적절한 디렉토리로 옮기고 환경 설정을 마치면 EtherCAT 마스터 스택을 사용할 수 있다. 본 논문에서는 편의상 다음과 같이 /root 디렉토리 아래에 설치하여 사용하였다.

```
# mkdir /root/soem-1.4
# cp -rf ../install/* /root/soem-1.4
# echo 'export PATH="/root/soem-1.4/bin:$PATH"'
>> ~/.bashrc
```

### 2.3 EtherCAT 마스터 응용 프로그램

EtherCAT 마스터 응용 프로그램은 SOEM 라이브러리 libsoem.a와 헤더 파일의 위치를 포함하도록 다음과 같이 Makefile을 작성하여 빌드하였다.

```
TARGET = test
CFLAGS = -I/root/soem-1.4/include/soem
LDFLAGS = -L/root/soem-1.4/lib -lsoem -lpthread -lrt
SRC_TARGET = $(TARGET:.c)
OBJ_TARGET = $(SRC_TARGET:.c=.o)
OBJS = $(OBJ_TARGET)
all : $(TARGET)
$(TARGET) : $(OBJS)
    $(CC) -o $$@ $(CFLAGS) $(LDFLAGS)
    rm $(OBJ_TARGET)
```

다중 마스터 인스턴스 (multiple master instances)를 갖는 응용 프로그램을 구현하기 위해서는 아래와 같이 EC\_VER2를 정의한 후에 SOEM 헤더 파일을 포함시키며, 'ec\_'로 시작하는 API 대신 'ecx\_'로 시작하는 API를 사용해야 한다.

```
#define EC_VER2
#include "ethercat.h"
```

본 논문에서는 다음과 같은 마스터 구조체를 선언하여 마스터 인스턴스를 구현하였다.

```
typedef struct _ec_master
{
    ec_slavet      ec_slave[EC_MAXSLAVE];
    int           ec_slavecount;
    ec_group      ec_group[EC_MAXGROUP];
    uint8         ec_esibuf[EC_MAXEEPBUF];
    uint32        ec_esimap[EC_MAXEEPBITMAP];
    ec_eringt     ec_elist;
```

```
ec_idxstackT    ec_idxstack;
ec_SMcommtypet ec_SMcommtypet[EC_MAX_MAPT];
ec_PDOassign    ec_PDOassign[EC_MAX_MAPT];
ec_PDOdesct     ec_PDOdesct[EC_MAX_MAPT];
ec_eepromSMt   ec_SM;
ec_eepromFMMUt ec_FMMU;
boolean         EcatError;
int64           ec_DCtime;
ecx_portt      ecx_port;
ecx_redportt   ecx_redport;
ecx_contextt   m_context;
char            IOMap[4096];
int             expectedWKC;
volatile int    wkc;
boolean         inOP;
} ec_mastert;
```

2개의 마스터 인스턴스를 갖는 경우는 다음과 같이 마스터 인스턴스를 선언하여 사용하였다.

```
#define MAX_EC_MASTER 2
ec_mastert masters[MAX_EC_MASTER];
```

### 2.4 실시간 동작을 위한 설정

PREEMPT\_RT Linux에서는 실시간 태스크가 동작할 CPU 코어를 스케줄러에서 분리(isolation)시켜 latency 성능을 향상시킬 수 있으며, 커널 부팅시에 isolcpus에 코어를 지정하여 CPU 분리가 가능하다[6]. 본 논문에서는 U-boot에서 환경 변수를 다음과 같이 설정하여 core 2, core 3을 분리하였다.

```
=> setenv othbootargs isolcpus=2,3
=> saveenv
```

분리된 CPU core를 제외한 core 0와 core 1에서 인터럽트가 수행되도록 아래 스크립트를 실행한다.

```
for I in $(ls /proc/irq)
do
    if [[ -d "/proc/irq/$I" ]]
    then
        echo 0-1 > /proc/irq/$I/smp_affinity_list
    fi
done
```

아래 명령은 실시간 태스크의 동작을 조절하는 *real-time throttling* (RTT)을 중지시킨다.

```
# echo -1 > /proc/sys/kernel/sched_rt_runtime_us
```

커널의 *write-back* 쓰레드를 core 0에서 수행하도록 다음과 같이 설정하였다.

```
# echo 1 > /sys/bus/workqueue/devices/writeback/cpumask
```

실시간 동작을 방해할 수 있는 불필요한 서비스는 다음과 같이 중지시킨다.

```
# systemctl disable docker.service
# systemctl disable containerd.service
# systemctl disable cron
# systemctl disable libvirtd.service
# systemctl disable libvirt-guests
# systemctl disable apt-daily.timer
# systemctl disable apt-daily.service
# systemctl disable apt-daily-upgrade.timer
# systemctl disable apt-daily-upgrade.service
```

*unattended-upgrade*가 수행되지 않게 다음과 같이 */etc/apt/*  
*unattended-upgrades*를 수정하였다.

```
APT::Periodic::Update-Package-Lists "0";
APT::Periodic::Unattended-Upgrade "0";
```

응용 프로그램에서는 *main()* 함수의 시작 부분에서 아래 함수를 호출하여 메모리 스와핑을 방지하게 하였다.

```
mlockall(MCL_CURRENT | MCL_FUTURE);
```

응용 프로그램에서 실시간 쓰레드를 생성할 때 스케줄링은 *SCHED\_FIFO*로 설정하고, 우선 순위는 다음과 같이 가장 높은 우선 순위보다 1이 작게 설정하였고 여기에서는 98이 된다.

```
priority = sched_get_priority_max(SCHED_FIFO) - 1;
```

응용 프로그램의 실시간 쓰레드가 CPU를 독점하도록 쓰레드 함수의 시작 부분에서 다음과 같이 *CPU affinity*를 설정하였다.

```
int core_id = 3;
cpu_set_t mask;
CPU_ZERO(&mask);
CPU_SET(core_id, &mask);
sched_setaffinity(0, sizeof(mask), &mask);
```

### 3. 실험 및 성능 평가

본 논문에서 구현한 EtherCAT 마스터의 정주기 제어 지터 성능을 평가하기 위해 임베디드 보드의 네트워크 포트 출력 신호를 *netAnalyzer*[10]에 연결하여 실시간으로 측정하였다. EtherCAT Slave는 Infineon XMC4800 EtherCAT Kit를 사용했으며, Output size는 8bits, Input size는 2bits, 4개의 Sync manager, 2개의 FMMU를 가지고 있다[11].

EtherCAT 마스터에서 송신되는 패킷이 일정한주기를 갖도록 하기 위해 [12]의 타이머 함수를 사용하였고, 응용 프로그램의 정주기 제어 루프는 다음과 같이 구현하였다.

```
init_period_info (&pinfo, PERIOD);
while (1) {
    ecx_receive_processdata_group (...);
    /* do something. */
    wait_rest_of_period (&pinfo);
    ecx_send_processdata_group (...);
}
```

임베디드 보드의 클럭 설정은 Fig. 1과 같이 부팅시 콘솔에 표시된다. EtherCAT 마스터 인스턴스가 1개인 경우와 2개인 경우에 대해 성능을 평가하였다.

```
U-Boot 2021.04 (Aug 01 2022 - 11:23:11 +0900)
SoC: LS1043AE Rev1.1 (0x87920011)
Clock Configuration:
  CPU0 (A53):1600 MHz  CPU1 (A53):1600 MHz  CPU2 (A53):1600 MHz
  CPU3 (A53):1600 MHz
  Bus: 400 MHz  DDR: 1600 MT/s  FMAN: 500 MHz
Reset Configuration Word (RCW):
00000000: 08100010 0a000000 00000000 00000000
00000010: 14550002 80040012 60040000 c1002000
00000020: 00000000 00000000 00000000 00038800
00000030: 00000000 00001100 00000096 00000001
Model: LS1043A RDB Board
Board: LS1043ARDB, boot from SD
CPID: V2.0
PCBA: V6.0
SERDES Reference Clocks:
SD1_CLK1 = 156.25MHz, SD1_CLK2 = 100.00MHz
DRAM: 1.9 GiB (DDR4, 32-Bit, CL=11, ECC off)
Using SERDES1 Protocol: 5205 (0x1455)
FS1_SDHC: 0
```

Fig. 1. Clock configuration of the embedded board.

#### 3.1 마스터 인스턴스가 1 개인 경우

EtherCAT 마스터 인스턴스가 1개인 경우는 Fig. 2와 같이 장치를 구성하고 주기를 측정하여 지터 성능을 평가하였다. Fig. 3은 제어 주파수를 4KHz(주기 250usec)로 설정하

고 타이밍 분석 결과를 캡처한 것이다. Table 3에서 PREEMPT\_RT는 제어 주파수가 각각 1KHz, 2KHz, 4KHz일 때, 10<sup>6</sup> 개 이상의 패킷에 대해 주기를 측정한 결과를 정리한 것이며, Max-Min은 주기 측정의 최대 값과 최소값의 차이를 의미한다.

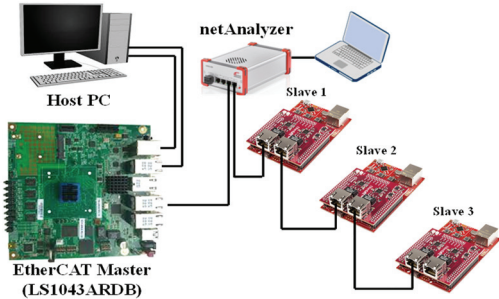


Fig. 2. Experimental setup with 1 master instance.

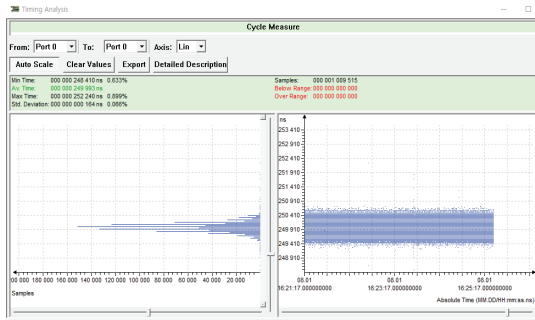


Fig. 3. Timing analysis @ 4KHz, 1 master instance.

Table 3. Period measurement results in micro-second, 1 master instance case

Control Frequency	1KHz	2KHz	4KHz
PREEMPT_RT			
Number of packets	1,001,425	1,004,980	1,009,515
Min	998.430	498.170	248.410
Average	999.975	499.987	249.993
Max	1,001.640	501.850	252.240
Std. Dev.	0.182	0.164	0.164
Max-Min	3.210	3.680	3.830
Xenomai [2]			
Min	994.70	496.61	246.25
Average	998.59	499.30	249.65
Max	1,003.14	503.61	253.37
Std. Dev.	0.23	0.22	0.19
Max-Min	8.44	7.00	7.12

Table 3의 결과로부터 구현된 EtherCAT 마스터는 제어 주파수와 무관하게 일정한 지터 성능을 가지며 정주기 제어가 잘 되고 있음을 알 수 있다. Table 3에서 Xenomai는 [2]의 결과이며, 마스터 사양이 달라서 직접적인 비교는 어렵지만 본 논문에서 구현한 EtherCAT 마스터의 지터 성능이 더 우수함을 볼 수 있다. 그러나, PREEMPT\_RT는 8KHz에서 정주기 제어가 잘 되지 않는 반면, [2]에서 Xenomai는 8KHz이상의 제어 주파수에서도 정주기 제어가 잘 되고 있다.

### 3.2 마스터 인스턴스가 2 개인 경우

마스터 인스턴스가 2개인 경우는 Fig. 4와 같이 master 0에는 1개의 slave를 연결하고, master 1에는 2개의 slave를 연결하고 주기를 측정하여 지터 성능을 평가하였다.

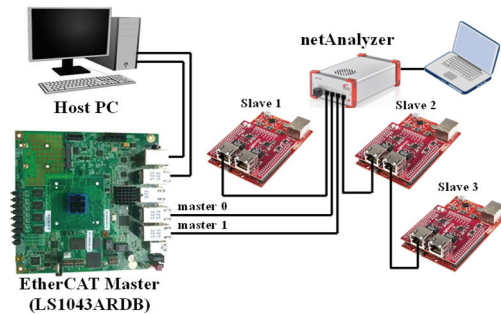


Fig. 4. Experimental setup with 2 master instances.

master 0와 master 1의 제어 루프는 각각 쓰레드를 생성하였고, master 0용 제어 루프 쓰레드는 core 2에, master 1용 제어 루프 쓰레드는 core 3에 각각 할당하였다. 이 경우에 쓰레드간 동기가 필요하기 때문에 아래와 같은 루틴을 각 쓰레드의 정주기 제어 루프 앞에 추가하여 시그널이 전달될 때까지 각 쓰레드가 대기상태에 있도록 하였다.

```
pthread_mutex_lock(&mutex);
pthread_cond_wait(&cond, &mutex);
pthread_mutex_unlock(&mutex);
```

main() 함수에서 각 쓰레드를 생성하고 일정 시간을 sleep한 후에, 아래 함수를 호출하여 대기상태인 쓰레드에 시그널을 브로드캐스팅해서 각 쓰레드가 동시에 정주기 제어 루프를 시작하게 하였다.

```
pthread_cond_broadcast(&cond);
```

제어 주파수가 4KHz (주기 250usec) 일 때 Fig. 5는 master 0의 타이밍 분석 결과이고 Fig. 6은 master 1의 타이밍 분석 결과이다. Fig. 7은 master 0에서 송신한 패킷과 master 1에서 송신한 패킷 사이의 시간 차이를 분석한 결과이다.

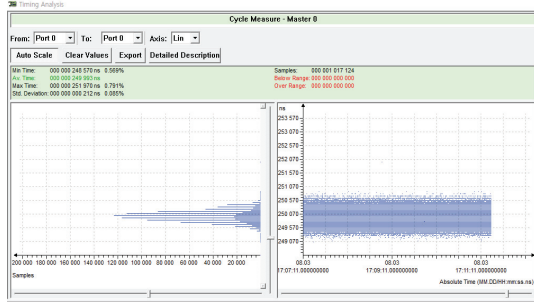


Fig. 5. Timing analysis @ 4KHz, master 0.

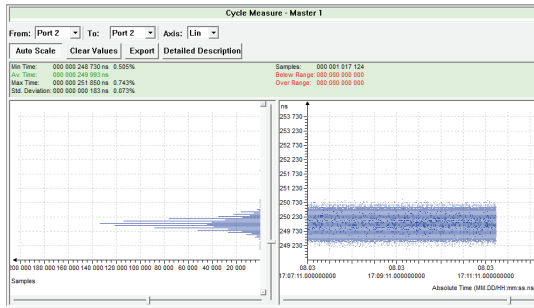


Fig. 6. Timing analysis @ 4KHz, master 1.

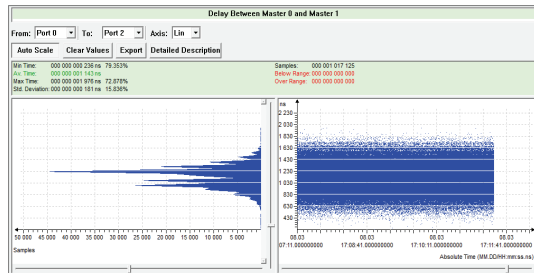


Fig. 7. Timing analysis of time difference between master 0 packet and master 1 packet @ 4KHz.

Table 4는 제어 주파수가 각각 1KHz, 2KHz, 4KHz일 때, 10<sup>6</sup>개 이상의 패킷에 대해 master 0, master 1의 주기를 측정 한 결과와 master 0에서 송신한 패킷과 master 1에서 송신한 패킷 사이의 시간 차이(Time Diff)를 측정 한 결과를 정리한 것이며, 구현된 EtherCAT 마스터는 정주기 제어가 잘 되고 있음을 알 수 있고, master 0와 master 1도 2~3usec내로 동기가 이루어지고 있음을 확인할 수 있다.

Table 4. Period measurement results in micro-second, 2 master instance case

Control Frequency		1KHz	2KHz	4KHz
Number of packets		1,006,774	1,005,661	1,017,124
Master 0 with 1 slave	Min	997.190	497.860	248.570
	Average	999.975	499.987	249.993
	Max	1,002.680	502.020	251.970
	Std. Dev.	0.169	0.379	0.212
	Max-Min	5.490	4.160	3.400
Master 1 with 2 slaves	Min	997.880	497.900	248.730
	Average	999.975	499.987	249.993
	Max	1,002.000	501.580	251.850
	Std. Dev.	0.164	0.220	0.183
	Max-Min	4.120	3.680	3.120
Time Diff.	Min	1.076	0.016	0.236
	Average	1.911	1.301	1.143
	Max	2.726	2.536	1.976
	Std. Dev.	0.167	0.278	0.181

### 4. 결론

본 논문에서는 ARM 기반 64-bit Quad-core 프로세서를 사용하는 임베디드 보드에서 PREEMPT\_RT Linux와 SOEM을 이용하여 EtherCAT 마스터를 구현하고 정주기 제어의 지터 성능을 평가하였다.

1개의 마스터 인스턴스를 가지는 경우와 2개의 마스터 인스턴스를 가지는 경우 모두 1KHz, 2KHz, 4KHz 제어 주파수에 대해 수usec 이내의 오차로 정주기 제어가 가능함을 확인하였다. 또한, 2개의 마스터 인스턴스를 가지는 경우에 수usec 이내로 마스터 간의 동기가 이루어짐을 확인하였다.

### 감사의 글

이 논문은 2022년도 한국기술교육대학교 교수 교육연구진흥과제 지원에 의하여 연구되었음.

### 참고문헌

1. EtherCAT Technology Group, <http://www.ethercat.org> [accessed August 8, 2022]
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. C. Huang, C. Lin, C. Wu, "Performance Evaluation of Xenomai 3," Available at [http://wiki.csie.ncku.edu.tw/embedded/xenomai/rtlws\\_paper.pdf](http://wiki.csie.ncku.edu.tw/embedded/xenomai/rtlws_paper.pdf) [accessed August 8,

- 
- 2022]
4. QorIQ LS1043A Reference Design Board Reference Manual, Rev. 4, Nov. 2017
  5. <https://www.nxp.com/design/software/embedded-software/linux-software-and-development-tools/layer-landscape-software-development-kit-v21-08:LAYERSCAPE-SDK> [accessed August 8, 2022]
  6. Layerscape Software Development Kit User Guide: LSDK 21.08, January 2022
  7. QorIQ LS1043A Reference Design Board Getting Started Guide, Rev. 8, May 2019
  8. <https://github.com/OpenEtherCATsociety/SOEM/releases> [accessed August 8, 2022]
  9. <https://github.com/OpenEtherCATsociety/SOEM> [accessed August 8, 2022]
  10. <https://www.hilscher.com/products/product-groups/analysis-and-data-acquisition/ethernet-analysis/nanl-b500g-re/> [accessed August 8, 2022]
  11. [https://www.infineon.com/cms/en/product/evaluation-boards/kit\\_xmc48\\_relax\\_ecat\\_v1/](https://www.infineon.com/cms/en/product/evaluation-boards/kit_xmc48_relax_ecat_v1/) [accessed August 8, 2022]
  12. <https://wiki.linuxfoundation.org/realtime/documentation/howto/applications/cyclic> [accessed August 8, 2022]
- 
- 접수일: 2022년 8월 12일, 심사일: 2022년 9월 6일,  
 게재확정일: 2022년 9월 6일