https://doi.org/10.5392/JKCA.2017.17.10.001

윈도우 운영체제 기반의 실시간 점검장비 소프트웨어 설계 및 성능검증

The Design and Performance Verification of Real-Time Inspection Equipment Software based on Windows Operating System

김효중, 허용관, 권병기
LIG넥스원

Hyo-Joung Kim(hyooung.kim@lignex1.com), Yong-Kwan Heo(yongkwan.heo@lignex1.com),
Byung-Gi Kwon(byunggi.kwon@lignex1.com)

요약

최근 군용장비의 첨단화가 가속됨에 따라 장비의 성능을 실시간으로 검증하는 점검장비 역할이 중요해지고 있다. 대부분 점검장비가 개발 편의성, 개발기간 등을 고려해서 윈도우 기반으로 개발되었다. 하지만 윈도우 기반 점검장비가 실시간성을 제공하지 않아 이기종간의 데이터 통신에 대한 주기를 만족해주지 못하는 단점이 있으며, 이러한 문제를 해결하기 위해 고가의 상용 솔루션을 이용하여 윈도우 기반의 점검장비에 실시간성을 보장해주고 있다. 본 논문에서는 고가의 상용 솔루션을 대체하는 실시간 이식 커널인 RTiK-MP를 기반으로 하는 실시간 점검장비 소프트웨어를 설계하는 방법을 제시하며, 성능을 검증하기 위해 고속으로 통신하는 유도탄과 연동시험을 통해서 실시간성과 데이터 정확성을 측정하고 이를 검증하였다.

■ 중심어 : | 실시간 운영체제 | 점검장비 | 실시간 통신 |

Abstract

As the recent advancement of military equipment has been accelerated, it is becoming more important to act as an inspection device that verifies the performance of equipment in real time. Most of the inspection equipments were developed on the Windows OS based system. considering development convenience and development period. However, sice the data communication between these models occurs asynchronously, there is a problem that it is difficult to guarantee real-time performance of the window-based inspection equiment. To solve these problems, we use real-time commercial solutions to guarantee the real-time performance of Windows-based inspection equipment. In this paper, we propose a method of designing and implementing the inspection equipment software based on Real-Time implanted Kernel-Multi Processor (RTiK-MP) operating in Windows environment. In addition, real-time performance data accuracy was measured through a high-speed communication tool and interlocking test to verify the performance of the inspection device based on the real-time porting kernel.

■ keyword : | Real-Time Operating System | Device Driver | Network |

접수일자 : 2017년 07월 06일
수정일자 : 2017년 08월 10일
심사완료일 : 2017년 08월 13일
교신저자 : 김효중, e-mail : hyooung.kim@lignex1.com

I. 서 론

최근 군용장비의 첨단화가 가속됨에 따라 장비의 성능과 안정성을 검증하는 점검장비의 역할이 중요해지고 있다. 즉, 다양한 환경에서 운용되는 첨단 군용장비의 성능과 안정성을 검증하기 위해서는 점검장비의 실시간성이 보장되어야 한다. 실시간성(Real-Time)이란 정확한 주기에 정해진 일을 수행해야 하고, 그 결과가 정확하게 전달되어야 하는 것을 의미한다[1]. 이러한 실시간성의 지원여부는 운영체제에 종속적으로 결정된다. 현재 운용되고 있는 점검장비의 대부분이 개발자의 편의성과 다양한 프로그램을 제공하는 확장성때문에 범용 운영체제인 리눅스나 윈도우 기반으로 개발되었다[2][3]. 하지만 최근에는 리눅스 기반의 점검장비보다 개발 기간과 편의성 그리고 개발 환경 등에서 사용자에게 우수한 기능을 제공하는 윈도우 기반의 점검장비를 선호하고 있는 추세이다[2][4]. 윈도우는 편리한 개발환경 및 GUI 지원 등의 장점을 가지고 있지만 라운드로빈 스케줄링 정책을 사용하기 때문에 실시간성을 보장하지 못하는 단점이 있다. 이는 이 기종 컴퓨터 간의 데이터 통신을 수행할 때, 세밀한 주기를 지원하지 않아 주기 오차가 발생하고, 이에 따른 데이터 손실이 발생한다. 이와 같은 문제점을 해결하기 위해 윈도우에서는 RTX나 INtime과 같은 고가의 서드파티 상용 솔루션을 구입한 후 구현해야 하기 때문에 개발 복잡도 및 비용이 상승하게 된다[5][6].

이와 같은 문제점을 극복하기 위해서 x86 기반의 윈도우에 실시간성을 제공하는 RTiK-MP(Real-Time implanted Kernel for Windows Multi-Processor)가 연구되었다. RTiK-MP는 윈도우와 독립적이며 주기적인 타이머를 갖는 인터럽트를 통해 실시간성을 제공하고 있으며, 이를 통한 태스크 및 이벤트 관리를 통해 실시간성을 제공할 수 있다[7]. 하지만 실시간 이식 커널인 RTiK-MP의 기본적인 태스크 수행 성능은 연구실 환경에서 검증되었으나 실제 산업환경에서 RTiK-MP의 태스크 수행 성능은 검증되지 않았다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로 RTiK-MP와 RTX, INtime 및 점검장비를 설명한다. 3장에서는 실시간 점검장비 소프트웨어에 대한 설계 및 성능검증 방안을 기술한다. 4장에서는 설계한 소프트웨어에 대한 성능 검증 실험을 위한 방법을 설명하고 이를 통한 실험 결과를 기술하며, 5장에서 결론 및 향후 연구를 통해 결론을 맺는다.

II. 관련연구

1. RTiK-MP

x86기반의 멀티코어 환경에서 윈도우에서 실시간성을 제공하는 RTiK-MP는 [그림 1]과 같이 디바이스 드라이버 형태로 이식되어 윈도우 커널 자원 및 하드웨어 자원을 이용할 수 있다.

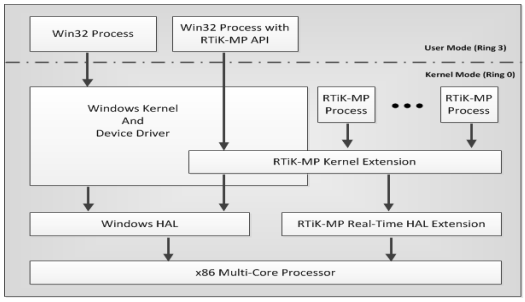

그림 1. RTiK-MP가 설치된 윈도우 운영체제 구성도

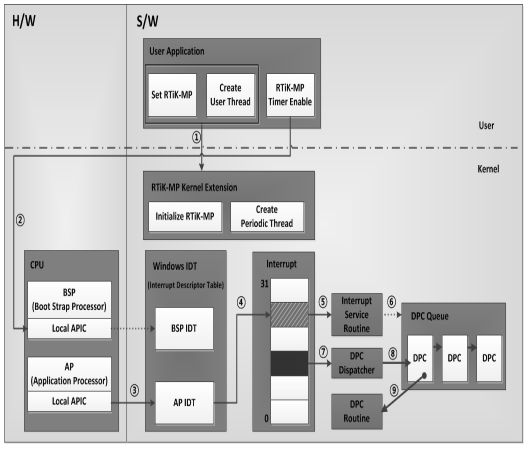

그림 2. RTiK-MP의 인터럽트 처리 과정도

RTiK-MP Kernel Extension은 윈도우의 커널과 디바이스 영역에 드라이버 형태로 확장 및 이식된다. 확장된 RTiK-MP Kernel Extension은 RTiK-MP Real-Time HAL(Hardware Abstraction Layer)를 통해서 x86 멀티코어 프로세서와 실시간 쓰레드(RTiK-MP Process)를 접근하고 관리한다. 또한 RTiK-MP API를 통해서 실시간 쓰레드와 RTiK-MP의 커널을 제어할 수 있다[6][7].

RTiK-MP는 윈도우에 실시간 처리 기능을 제공하기 위하여 로컬 APIC를 이용해 윈도우와는 독립적인 타이머 인터럽트를 발생시킨다. [그림 2]에서 볼 수 있듯이 RTiK-MP는 BSP(Boot Strap Processor)가 아닌 AP(Application Processor)의 로컬 APIC를 이용하여 윈도우와는 독립적인 타이머 인터럽트를 발생시켜 실시간 쓰레드의 주기적인 동작을 보장한다. 로컬 APIC 타이머 인터럽트가 발생되면, AP의 IDT(Interrupt Descriptor Table)에 인덱스 역할을 하는 백터(Vector)를 통해 IDT에 등록되어 있는 인터럽트 핸들러로 분기하게 된다. 이때 수행되는 핸들러는 인터럽트의 지연시간을 최소화시키기 위해 지연처리함수(Deferred Procedure Call, DPC) Queue에 등록된다. 인터럽트 핸들러가 완료되고, IRQL(Interrupt Request Level)값이 Dispatch 레벨이 되면 인터럽트 핸들러에서 등록된 핸들러를 Dispatcher가 수행하게 된다. 이와 같은 과정을 통해 RTiK-MP는 윈도우에서 실시간 처리 기능을 제공한다.

III. 실시간 점검장비 소프트웨어 설계 및 성능 검증 방안

1. 실시간 점검장비 설계 및 검증방안

새롭게 개발되는 유도무기의 성능을 검증하기 위해서는 점검장비의 실시간성이 보장되어야 한다. 하지만 윈도우 기반의 점검장비는 별도의 솔루션 없이 실시간성이 보장되지 않는다. 이를 극복하기 위해 RTiK-MP를 이용해서 점검장비에 실시간성을 부여했다. 본 장에서는 TCP/IP 데이터 통신 기반의 점검장비 소프트웨어 설계 및 구현방법에 대해서 기술하겠다.

우선 TCP/IP란 네트워크 전송 프로토콜로 서로 다른 운영체제를 쓰는 이 기종 컴퓨터 간에도 데이터를 전송할 수 있어서 데이터 전송을 위한 표준 프로토콜로 쓰이고 있다. 이런 TCP/IP 통신 방법에는 TCP[8]와 UDP[9]로 구분할 수 있고, 이를 연결성 프로토콜과 비연결성 프로토콜로 구분할 수 있다. UDP는 비연결성 프로토콜로써 별도의 통신 세션에 대한 접속을 설정하지 않고, 고속으로 데이터 통신을 수행할 수 있다는 장점 때문에 최근 유도무기와 점검장비 간의 데이터 통신에 많이 이용되는 추세이다[10-12].

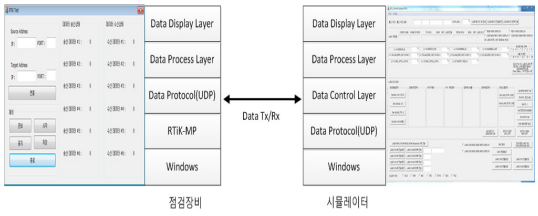

그림 3. 점검장비의 통신 구조

[그림 3]은 본 논문에서 구현한 점검장비와 시뮬레이터간의 소프트웨어 구조를 도식화한 것이다. 점검장비는 윈도우 기반에 RTiK-MP를 적용해서 구현하였다. 점검장비는 최종적으로 유도무기와 실시간 데이터 통신이 보장되어야 한다. 이를 위해 우선적으로 시뮬레이터와 점검장비를 연결한 후 점검장비의 성능을 1차적으로 검증하였다. 시뮬레이터를 통해서 1차적으로 성능을 검증한 이유는 잘못된 통신 프로토콜로 인해 유도무기에 영향을 주지 않고, 점검장비의 신뢰성과 안정성을 확인하기 위해서다.

본 장의 2절에서 점검장비를 위한 기능 요구사항을 밝히고, 3절에서는 점검장비에 대한 설계 및 구현 사항을 설명한다.

2. 실시간 점검장비의 기능 요구사항 분석

2.1 태스크 지원 요구사항

점검장비는 여러개의 송/수신 채널을 이용하여 데이터를 유도무기와 주고 받는다. 점검장비는 다중 통신 데이터 및 내부 명령 처리를 위해 다수의 태스크가 사

용되며, 각 태스크는 정해진 주기를 기반으로 동작하게 된다. 본 연구에서 설명하는 점검장비는 총 10개의 중요 통신 태스크가 동작하게 된다.

[표 1]은 점검장비에서 사용하는 주요 태스크를 나타낸 것이다. 5개의 송/수신 채널을 이용하기 위한 태스크를 나열하고 있으며, 5ms의 주기마다 송신 태스크가 대상 기기에 명령을 내리도록 하고 있으며, 1ms의 주기마다 대상 기기에서 발생하는 정보를 수신할 수 있도록 하는 기능을 수행한다. 각 태스크의 정확한 주기를 지키기 위해 실시간성이 요구된다.

2.2 통신 요구사항

기존 개발된 RTiK-MP는 RS-232 통신 및 LAN 통신을 이용하여 윈도우에 실시간성을 제공하였다. 하지만 LAN 통신에서 사용된 TCP 통신은 연결지향 방식으로 통신을 위한 "연결-송신-송신확인-재전송" 단계를 거치기 때문에 작은 단위의 주기를 가지는 반복되는 통신에서는 태스크가 호출되어도 통신을 위한 주기를 만족하지 못하는 단점이 있다. 이를 해결하기 위해 본 논문에서는 RTiK-MP의 LAN 통신에 UDP 통신 기능을 추가 구현하였다. 이를 기반으로 점검장비는 비연결성 프로토콜인 UDP를 통해서 통신 주기를 만족하는고속의 데이터 통신을 수행할 수 있어야한다.

3. 실시간 점검장비 소프트웨어 설계 및 구현

새롭게 개발되는 유도무기의 성능을 검증하기 위해 점검장비의 실시간성이 보장되어야 한다. 본 논문에서 제안하는 점검장비는 윈도우 운영체제에 RTiK-MP를 이용하여 실시간성을 부여했다. 본 장에서는 TCP/IP 데이터 통신 기반의 점검장비 소프트웨어 설계 및 구현 방법에 대해서 기술한다. 점검장비는 [그림 4]와 같은 구조로 설계 및 구현되었다.

점검장비는 RTiK-MP를 사용하기 초기화, 프로세스 생성, 프로세스 제어함수를 순차적으로 호출한다. [그림 5]는 RTiK - MP가 프로세스를 다중으로 생성한 후 각 프로세스를 이벤트 핸들러에 등록시키는 과정을 확인할 수 있다. 생성된 프로세스는 유도무기와 다중채널로 데이터 통신을 수행하게 된다.

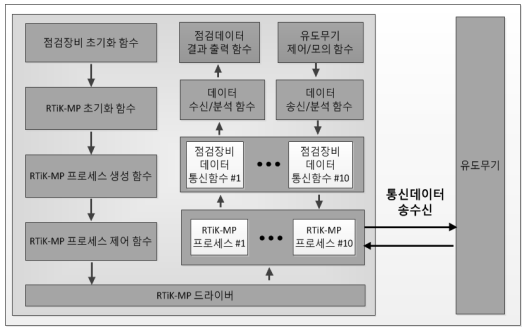

그림 4. 점검장비 소프트웨어 구조

그림 5. 다중채널 데이터 통신을 지원하는 쓰레드 생성

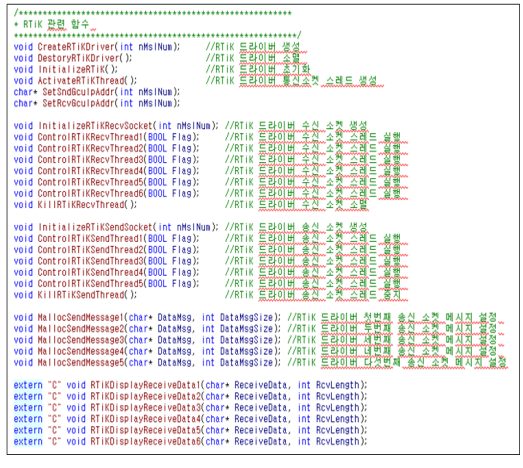

그림 6. 데이터 처리 및 통신 계층 관련 함수

표 1. 점검장비에서 사용하는 주요 태스크

이름	주기	개수	비고
송신 태스크	5ms	5개	5채널 통신
수신 태스크	1ms	5개	

점검장비는 [그림 6]의 함수를 통해서 데이터를 송수신, 처리, 출력 기능을 수행한다. 즉, 유도무기로부터 데이터를 수신한 점검장비의 RTiK-MP 프로세스는 점검장비 데이터 통신함수에 전달한다. 점검장비 데이터 통신함수는 수신한 데이터를 구조체에 복사하는 기능을 수행한다. 데이터를 구조체에 복사한 후 데이터 수신/분석함수에 전달한다. 데이터 수신/분석함수는 데이터의 무결성을 검증하고, 데이터 헤더를 분석해서점검장비와 유도무기 간의 통신 프로토콜 기반으로 데이터를 분석 및 분류한 후 점검데이터 출력함수를 통해서 사용자에게 출력된다. 또한 점검장비는 유도무기를 제어 및 모의하기 위해서 유모무기 제어/모의함수를 통해서 송신데이터를 생성한다. 생성된 데이터는 데이터 송신/분석함수를 통해서 무결성을 검증하고, 통신 프로토콜 기반으로 데이터 헤더를 작성하게 된다. 작성된 데이터는 점검장비 데이터 통신함수를 통해서 구조체에 복사한다. 복사된 데이터는 RTiK-MP 프로세스를 통해서 유도무기에 전달된다.

IV. 성능검증 실험

1. 성능검증 방법 및 성능검증 과정

본 논문에서 사용하고 있는 RTiK-MP는 서론에서 기술한 것과 같이 연구실 환경에서 통신의 주기를 CPU 카운터 값을 측정해서 통신수행 시간을 측정하였다. 즉, VxWorks, RTX, INTime 이 적용된 다른 실시간 장비와 통신을 수행하면서 성능을 측정하지 못했다. 따라서 본 장에서는 실시간 통신이 가능한 유도무기와 RTiK-MP 가 적용된 점검장비를 연동해서 실시간 성능을 측정하고 분석하였다.

우선적으로 데이터 수신 성능을 확인하기 위해 두 가지 단계로 실험을 수행하였다. 첫 번째 단계는 유도무기 시뮬레이터와 연동해서 점검장비의 성능을 1차적으로 검증했다. 이를 통해 안정적으로 동작하는 것을 확인한 다음 두 번째 단계로 유도무기와의 연동을 통한 성능 검증을 수행하였다.

2. 시뮬레이터 기반 성능측정

본 장에서는 유도무기 시뮬레이터 소프트웨어와 연동해서 성능을 검증했다. 실험환경 및 구성은 다음과 같다.

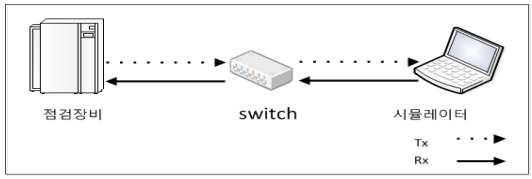

그림 7. 시뮬레이터 기반 성능측정 구성도

표 2. 시뮬레이터 기반 성능측정 환경

설정장비	설정항목	설정값
점검장비	데이터 송신주기	5ms
	데이터 수신주기	1ms
	송신 패킷 크기	1500 byte
	쓰레드 갯수	Tx : 5 개 Rx : 5 개
	실시간성 지원	지원
시뮬레이터	데이터 송신주기	5ms
	송신 패킷 크기	1500 byte
	실시간성 지원	지원 불가

[그림 7]과 같이 실험환경을 구성하였다. 점검장비와 시뮬레이터의 데이터 통신환경은 [표 2]와 같이 설정하였다. 점검장비는 5ms마다 데이터를 송신하는 5개의 채널과 1ms마다 데이터를 수신하는 5개의 채널을 동작시킨 후 시뮬레이터와 데이터 통신을 수행한다. 단, 시뮬레이터는 일정한 주기마다 인덱스 값이 입력된 패킷 5만개를 순차적으로 송신하게 된다. 이를 총 4차례 반복한 후 점검장비에 기록된 로그데이터를 분석한 결과는 [표 3]과 [표 4]와 같다.

시뮬레이터 기반의 점검장비에 대한 실시간 성능을 분석한 결과 최대 송신 주기와 최소 송신주기의 오차가 0.1ms 차이가 나며, 5만개의 패킷을 송신하는 동안 표

준편차가 0.01ms인 것으로 확인할 수 있다. 이는 기준 주기에 영향을 주지 않는 범위인 것으로 확인할 수 있었고, 통신에 대한 실시간 주기 성능이 보장된다는 것을 알 수 있었다. 또한 점검장비는 총 4차례 실험을 반복하면서 시뮬레이터로부터 수신한 데이터를 저장기능과 출력기능을 선택적으로 수행한 결과 5만개의 패킷을 100% 수신한 것을 확인할 수 있었으며, 이는 패킷 전송에 대한 누락이 없는 실시간 통신을 제공하는 것을 확인할 수 있었다.

표 3. 시뮬레이터 기반의 점검장비 송신 성능 검증

검증항목	측정값
데이터 송신 평균 주기	5.00 ms
데이터 송신 최대 주기	5.07 ms
데이터 송신 최소 주기	4.91 ms
데이터 송신 표준 편차	0.01 ms

표 4. 시뮬레이터 기반의 점검장비 수신 성능 검증

구분	점검장비 설정	수신률
1차 실험	데이터 저장	100 %
2차 실험	데이터 저장	100 %
3차 실험	데이터 저장/출력	100 %
4차 실험	데이터 저장/출력	100 %

3. 실제 유도무기 기반 성능측정

앞서 실험한 시뮬레이터 기반으로 점검장비의 성능을 검증하였으나 송수신 주기가 일반적인 윈도우 타이머에서도 충분히 처리할 수 있는 주기라고 판단되어 실제 유도무기와 연동해서 실시간 데이터 처리에 대한 성능을 측정하였다.

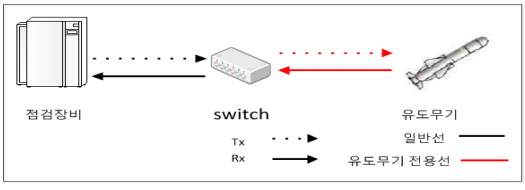

그림 8. 유도무기 기반 성능측정 구성도

[그림 8]은 유도무기 기반 성능측정에 대한 구성도를 나타낸 그림이다. 실제 유도 무기를 기반으로 하는 성능 측정을 위해서 시뮬레이터 장비와는 다르게 유도무기와의 통신을 위한 전용선을 사용하였다. 이를 switch를 통해서 점검장비로 데이터를 전송할 수 있도록 구성하였다. [표 5]는 유도무기 기반의 점검장비 성능을 측정하기 위한 환경에 대한 구성을 나타낸 표이다. 기존 시뮬레이터와의 차이점은 시뮬레이터를 활용한 통신 주기에서 5ms의 주기로 통신을 실시하였으나, 본 실험에서는 1ms의 통신 주기를 사용하였다. 짧아진 전송주기로 인해 데이터량이 약 5배 증가하게 되고 이로 인해 점검장비의 데이터 수신 및 처리 영역의 부하가 증가하게 된다. 이는 실제 유도무기 점검환경과 가장 유사한 환경이며 점검장비는 이와 같은 환경에서 실시간성을 보장해야한다. 또한 RTiK-MP를 적용한 점검장비와 윈도우 타이머를 적용한 점검장비 간의 실시간 성능을 비교하였다.

표 5. 유도무기 기반 성능측정 환경

설정장비	설정항목	설정값
점검장비	데이터 송신주기	비주기
	데이터 수신주기	1ms
	송신 패킷 크기	1500 byte
	쓰레드 갯수	Tx : 5 개 Rx : 5 개
	실시간성 지원	지원
유도무기	데이터 송신주기	1ms
	송신 패킷 크기	1500 byte
	실시간성 지원	지원

표 6. 윈도우 타이머 기반의 점검장비 실시간 수신 성능

구분	점검장비 설정	수신률
1차 실험	데이터 저장	99.635 %
2차 실험	데이터 저장	98.892 %
3차 실험	데이터 저장	99.532 %
4차 실험	데이터 저장	99.357 %
5차 실험	데이터 저장/동영상 1개 재생	89.885 %
6차 실험	데이터 저장/동영상 1개 재생	92.988 %
7차 실험	데이터 저장/동영상 1개 재생	92.953 %

[표 6]는 윈도우 타이머 기반의 점검장비에서 수신 성능을 측정한 것이다. 실험은 총 7차 실험까지 진행하였으며 수신된 패킷을 저장하는 기본적인 측정과 동영상 재생을 통해 시스템에 부하를 주면서 패킷을 저장하는 측정하는 방식으로 진행하였다. 측정결과를 보면 윈도우 타이머 기반의 점검장비는 데이터 저장만 했을 경

우 데이터 평균 저장률이 99.353% 인 것을 확인할 수 있다. 하지만 유도무기를 점검하는데 소요되는 평균 시간이 약 30분인 것을 감안했을 때 유도무기에서 송신하는 데이터의 갯수는 약 180만개이다. 이 중 0.647% 데이터가 손실되었을 경우 누락되는 데이터의 개수는 약 12000개정도이다. 대부분의 유도무기의 중요 신호들의 동작시간이 짧게는 100ms, 길게는 500ms 동작하기 때문에 12000개의 데이터가 누락되었다는 것은 유도무기의 성능을 정확하게 분석할 수 없다는 의미로 볼 수 있다. 이와 동일한 환경에서 RTiK-MP 기반의 점검장비의 수신 성능을 측정하였다. 실험은 총 15차까지 진행하였다. [표 7]를 보면 데이터 저장만 하는 환경에서는 모든 패킷을 놓치지 않고 100% 수신하는 것을 확인할 수 있다. 다음으로 시스템에 부하를 줄 수 있는 동영상 재생을 진행할 때, 임의의 상황에서 0.01% 패킷이 손실되었다. 이와 같은 비교실험을 통해서 윈도우 기반의 점검장비보다 RTiK-MP 기반의 점검장비의 실시간 성능이 뛰어나다는 것을 확인할 수 있다. 하지만 수신한 데이터를 즉시 화면에 출력할 경우에는 평균 수신률이 30%인 것으로 보아 RTiK-MP는 시스템 부하가 커질수록 성능이 나빠지는 것을 확인할 수 있었다.

표 7. RTiK-MP 기반의 점검장비 실시간 수신 성능

구분	점검장비 설정	수신률
1차 실험	데이터 저장	100 %
2차 실험	데이터 저장	100 %
3차 실험	데이터 저장	100 %
4차 실험	데이터 저장	100 %
5차 실험	데이터 저장/동영상 1개 재생	100 %
6차 실험	데이터 저장/동영상 1개 재생	99.994 %
7차 실험	데이터 저장/동영상 1개 재생	100 %
8차 실험	데이터 저장/동영상 2개 재생	99.994 %
9차 실험	데이터 저장/동영상 2개 재생	100 %
10차 실험	데이터 저장/동영상 2개 재생	100 %
11차 실험	데이터 저장/출력	30.481 %
12차 실험	데이터 저장/출력	30.477 %
13차 실험	데이터 저장/출력	24.878 %
14차 실험	데이터 저장/출력	27.529 %
15차 실험	데이터 저장/출력	29.523 %

V. 결론 및 향후연구과제

본 논문에서는 윈도우 운영체제 기반의 점검장비에 실시간성을 부여하고, 이를 기반으로 하는 실시간 점검장비 소프트웨어에 대한 설계 및 성능 검증에 대한 연구 및 실험을 진행하였다. 기존 윈도우 기반의 점검장비에서는 윈도우 운영체제의 특성상 빠른 주기의 태스크 호출이 보장되지 않는다. 이를 개선하기 위해 윈도우 운영체제에 실시간 확장 커널인 RTiK-MP를 이식하여 실시간성을 제공하였다.

구현된 실시간 점검장비 소프트웨어의 실시간 성능을 검증하기 위해 5개의 통신채널에 데이터 송수신하였다. 유도무기에 직접 실험하기에 앞서 성능 검증을 위해 시뮬레이터를 통한 실험을 진행하였다. 시뮬레이터 환경에서는 5ms 주기의 통신을 설정하여 주기대로 인덱스 값이 입력된 패킷 5만개를 점검장비에 순서대로 송신한다. 점검장비에서는 각 5개의 송수신 쓰레드를 사용하여 측정하였으며 송수신 오차가 0.1ms, 5만개의 패킷을 송신하는 동안 표준편차가 0.01ms인 것으로 주기성능이 보장되었으며 5만개의 패킷을 100% 수신한 것으로 주기 성능을 확인할 수 있었다.

시뮬레이터 환경을 검증한 다음 실제 유도 무기를 통해 점검 장비의 성능을 검증하였다. 실제 유도 무기 환경에서는 1ms 주기의 통신 환경에서 성능을 측정 하였으며 데이터 저장 및 동영상 재생을 통한 부하 환경에서 실시간성이 보장되었으나 입출력 인터럽트 및 시스템 과부하가 발생할 경우 실시간성이 어긋나는 단점을 확인할 수 있었다. 향후 RTiK-MP를 이용하여 유도무기 점검장비 분야에 적용할 때, 시스템의 과부하에 영향을 최소화 시킬 수 있는 구조로 설계하며 다양한 장비에 실시간 처리 기능을 제공할 수 있는 확장성을 가질 수 있는 연구가 필요하다.

참 고 문 헌

[1] C. M. Krishna and Kang G. Shin, *Real-Time Systems,* McGraw-Hill, 1997.

[2] 김주만, 송창인, 이철훈, “RTiK-Linux: 리눅스용 실시간 이식 커널의 설계,” 한국콘텐츠학회논문지, 제11권, 제9호, 2011.

[3] 주민규, 이진욱, 김종진, 조한무, 박영수, 이철훈, “x86 기반의 윈도우 상에서 실시간성 지원 방법,” 한국차세대컴퓨팅학회논문지, 제11권, 제4호, 2011.
[4] 조아라, 송창인, 이철훈, “윈도우즈 상에서 실시간 디바이스 드라이버를 위한 통합 미들웨어,” 한국콘텐츠학회논문지, 제13권, 제3호, 2013.
[5] Real-Time Operating System(RTOS) Platform, http://www.intervalzero.com
[6] INTime® for Windows Platform, http://www.tenasys.com
[7] 송창인, 이승훈, 주민규, 이철훈, “멀티프로세서 윈도우즈 상에서 실시간성 지원,” 한국콘텐츠학회논문지, 제12권, 제6호, pp.68-77, 2012.
[8] IETF(Internet Engineering Task Force), RFC(Request for Comments) 793, “TCP(Transmission Control Protocol),” https://tools.ietf.org/html/rfc793
[9] IETF(Internet Engineering Task Force), RFC(Request for Comments) 769, “UDP(User Datagram Protocol),” https://tools.ietf.org/html/rfc768
[10] 장순건, 윤종호, "실시간 전송을 위한 동기식 이더넷 시스템의 구현," 대한전자공학회 학술대회, pp.56-58, 2011(6).
[11] 곽상은, 김형규, "유도무기를 위한 동기식 이더넷 기반의 실시간 제어 시스템의 설계," 한국통신학회 학술대회논문집, pp.1076-1077, 2016(6).
[12] 오진오, “비행시험 발사통제 시스템의 신호처리 알고리즘,” 한국정보통신학회논문지, Vol.19, No.8, pp.1965-1972, 2015(8).

저 자 소 개

김 효 중(Hyo-Joung Kim) 정회원

- 2007년 2월 : 충남대학교 정보통신공학부 컴퓨터전공(공학사)
- 2009년 2월 : 충남대학교 정보통신공학과(공학석사)
- 2009년 ~ 2011년 : SK텔레시스
- 2011년 ~ 현재 : LIG넥스원(주) 유도무기2연구소 선임연구원

<관심분야> : 유도탄 체계점검, 유도탄 시험평가

허 용 관(Yong-Kwan Heo) 정회원

- 1991년 2월 : 한양대학교 전자계산학과(공학사)
- 1997년 ~ 2005년 : 도화 엔지니어링
- 2006년 ~ 현재 : LIG넥스원(주) 유도무기2연구소 수석연구원

<관심분야> : 유도탄 체계점검, 유도탄 시험평가

권 병 기(Byung-Gi Kwon) 정회원

- 2007년 2월 : 경북대학교 전자전기컴퓨터공학부(공학사)
- 2006년 ~ 현재 : LIG넥스원(주) 유도무기2연구소 선임연구원
- 2016년 : 아주대학교 IT융합대학원(공학석사)

<관심분야> : 유도탄 체계점검, A/J GPS 수신기