

유도무기와 점검장비간 MIL-STD-1553B 통신의 실시간성 지원을 위한 미들웨어 설계 및 구현

김종진*, 조한무*,이철훈**

*LIG넥스원 유도항공연구소, **충남대학교

jjkim1086@lignex1.com, chlee@cnu.ac.kr

Design and Implementation of middleware for Real-Time support of MIL-STD-1553B communication between Missile System and Test Equipment

Jong-Jin Kim*, Han-Moo Jo*, Cheol-Hoon Lee**

*PGM2 R&D, LIG Nex1, **Chung-Nam Uvi.

요 약

유도무기는 점검장비와 MIL-STD-1553B 통신 시 많은 양의 데이터를 처리하고, 전달 받은 데이터에 대해 정확성과 빠른 응답성을 요구하기 때문에 점검장비에서 데이터를 실시간으로 처리하고 판단하는 것이 매우 중요하다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 기존에 개발된 실시간 이식 커널을 이용하여 미들웨어에서 MIL-STD-1553B 통신에 요구되는 주기적인 통신의 마감시간을 보장하도록 설계 및 구현하였다.

1. 서론

최근 전장 환경은 네트워크 중심으로 이루어지며 복합 유도무기체계 형태로 진보하고 있다. 유도무기체계는 실시간으로 데이터의 통신이 이루어져야하기 때문에 실시간성 제공을 요구하고 있다. 또한 무기체계 개발 시 무기의 성능검증을 위해 사용되는 점검장비의 경우 데이터의 정확성과 응답성을 검증하기 위해 실시간성을 요구한다. 이러한 점검장비의 경우 개발의 편의성을 제공하기 위해 범용운영체제인 윈도우즈를 사용하는데, 범용운영체제인 윈도우즈의 경우 실시간성을 제공하지 못하는 문제점이 있다. 따라서 점검장비에 실시간성을 제공하기 위해서는 RTX와 INtime과 같은 고가의 상용솔루션을 통해 실시간성을 제공해야한다.

본 논문에서는 윈도우즈를 사용하는 점검장비의 1553B 통신에 실시간성을 제공하기 위해 기존에 개발된 실시간 이식 커널인 RTiK(Real-Time implant Kernel)을 이용하여 실시간성을 제공받아 미들웨어에서 처리하도록 설계 및 구현하였고, 범용 계측기를 이용한 성능을 측정 통해 실시간성 제공여부를 검증하였다.

본 논문은 2장에서는 관련연구로 상용솔루션과 RTiK에 대해 설명하고, 3장에서 미들웨어에서 MIL-STD-1553B 통신의 주기적인 실시간성 보장을 위한 설계 및 구현에 대해 설명한다. 4장에서는 구현한 미들웨어의 검증 및 결과를 보이고, 마지막으로 5장에서는 결론에 대해 기술한다.

2. 관련연구

2.1 서드파티

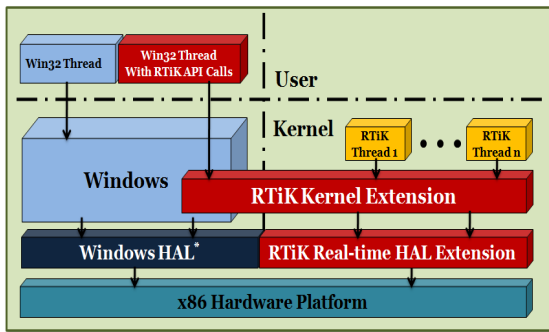
IntervalZero의 RTX(Real-Time Extension)는 윈도우즈 XP 및 윈도우즈 7에 실시간 제어가 가능한 실시간 운영체제의 기능을 부가해주는 확장 소프트웨어로서, 대부분의 개발자들이 익숙하고 편리한 윈도우즈 환경에서 RTX를 이용하여 기존의 실시간 운영체제의 장점을 최대한 사용할 수 있도록 제공해 준다. 즉, RTX는 순수 실시간 운영체제가 아니라 윈도우즈의 대중성 및 풍부한 GUI Library의 장점을 최대한 이용하여 실시간성 등의 제약사항들을 보완해 주는 소프트웨어이다[1].

INtime은 인텔사가 개발한 iRMX커널을 사용하고 있는 인텔의 x86/CPU에 특화된 리얼타임 시스템 소프트웨어이다. INtime은 윈도우 XP/2000을 플랫폼으로 한 제어·계측 시스템에 INtime을 부가해 시스템 상에서 실시간성이 필요한 경우를 INtime을 사용하여 처리하게 되고, 윈도우즈의 유연성은 그대로 유지하면서 보다 높은 신뢰성과 정밀한 리얼타임 퍼포먼스를 보충할 수 있다. INtime은 산업용 컴퓨터나, x86 보드 컴퓨터, 범용 PC로 윈도우즈와 현보 동작하는 실시간 운영체제이며, 동일한 하드웨어 상에서 윈도우즈와 동시에 동작하는 실시간 운영체제이다[2].

2.2 RTiK

RTiK은 LIG 넥스원과 충남대학교에서 공동 개발한 실시간 이식 커널로써, x86 기반의 윈도우즈와는 별도의 우선순위 기반 라운드로빈(Round-Robin) 스케줄링에 독립

적으로 수행되고, 윈도우즈 프로세스보다 먼저 실행되는 실시간 스레드를 제공한다. RTiK은 실시간성을 제공하기 위해 윈도우즈의 Local APIC Timer를 이용하여 윈도우와는 별도의 타이머 인터럽트를 발생시켰다. (그림 1)는 윈도우즈 디바이스 드라이버 형태로 이식된 RTiK을 나타낸 그림이다. RTiK은 커널 레벨에서 동작함으로써 하드웨어 접근이나 윈도우즈 커널 자원의 접근을 가능하게 한다 [3-8].

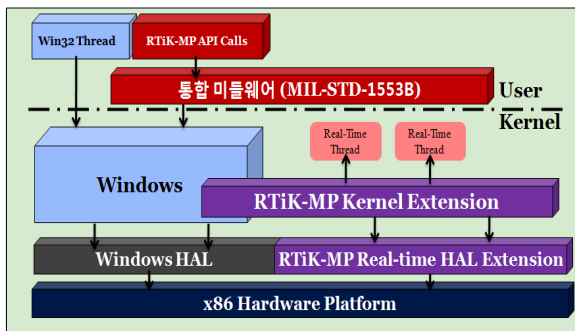


(그림 1) RTiK 구조

3. 설계 및 구현

3.1 1553B 실시간 통신을 위한 미들웨어의 설계

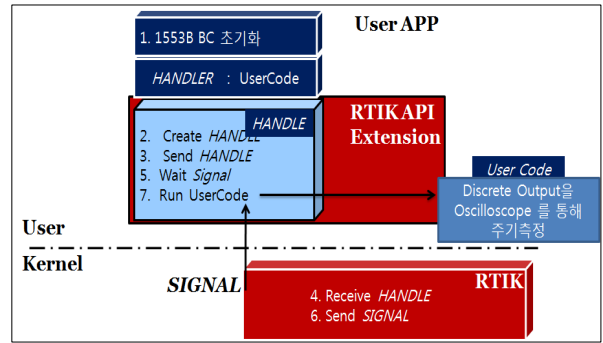
디바이스 드라이버 형태로 구현된 RTiK은 윈도우와는 독립적인 커널을 가진다. 이를 통해 윈도우즈의 커널영역에서 하드웨어 자원인 Local APIC의 제어를 통해 실시간성을 제공한다.



(그림 2) 미들웨어의 구조도

(그림 2)에서 볼 수 있듯이 MIL-STD-1553B 디바이스 드라이버에 실시간성 제공과 개발자에게 접근의 용이성을 제공하기 위해 미들웨어로 설계하였다. 즉, 1553B API를 사용하여 실시간 통신 제어가 가능하도록 설계함으로써, MIL-STD-1553B 디바이스 드라이버에 대한 구조적인 이해가 불필요하며, 개발 비용을 감소시키는 효과를 얻을 수 있다. (그림 3)는 RTiK의 사용자 영역에서의 동작과정을 나타낸 그림이다. 그림에서 볼 수 있듯이 커널영역에 RTiK이 이식되어있고, 이를 사용자 영역의 프로세스에서 신호를 받은 뒤 사용자 코드가 동작하도록 설계 및 구현하였다. RTiK의 성능 검증을 위해 사용자 코드의 내부(Handle)에서 범용계측기에 신호를 출력하도록 코드를 작

성하였다.



(그림 3) RTiK의 동작 과정

3.2 1553B 실시간 통신을 위한 미들웨어의 구현

(그림 4)는 미들웨어에서 1553B 디바이스 드라이버에 실시간성을 제공하기 위한 소스코드의 모습이다. 그림에서 볼 수 있듯이 RTiK_CreateThread API를 통해 1553B 디바이스 드라이버에 실시간성을 제공하도록 구현하였다. 이를 통해 사용자는 RTiK과 1553B의 내부구조에 대한 이해가 없어도 실시간성을 제공받을 수 있다.

```
#include "RTiK-MP_starter.h"
int main(int argc, char* argv[])
{
    RTiK_Initialization();
    while(TRUE) {
        printf( "%d : RTiK-MP User_Thread\n",
               getpid() );
        printf( "%d : RTiK-MP Enable\n",
               getpid() );
        printf( "%d : RTiK-MP Disable\n",
               getpid() );
        printf( "%d : Quit\n",
               getpid() );
        //1553B Setting Code
        //...
        ch = getch();
        switch(ch)
        {
            case '1':
                RTiK_CreateThread("RTiK_1553B_BC", 1, 1000);
                break;
            case '2':
                RTiK_Enable(1);
                break;
            case '3':
                RTiK_Disable(1);
                break;
            case '4':
                RTiK_Event_Clear(1);
                break;
            case '5':
                RTiK_End();
                break;
            default:
                printf( "!!!! Invalid Command !!!\n\n");
                break;
        }
        if( (exitFlag == TRUE) )
            break;
    }
    return 0;
}
```

Argument	Meaning	Value
void (*Function)(void)	실시간성 제공이 필요한 함수	RTiK_1553B_BC RTiK_1553B_RT RTiK_RS232_SEND RTiK_RS232_RECEIVE RTiK_LAN_SEND RTiK_LAN_RECEIVE
int time	실행 주기	1~1000 (ms)
int count	실행 횟수	0~65535 0 : 실행 횟수 무한대

(그림 4) 미들웨어를 통한 1553B에 실시간성 제공

실시간 이식 커널인 RTiK을 통해 미들웨어가 MIL-STD-1553B 통신 보드를 제어하기 위한 API는 (그림 5)과 같다.

RTiK_Initialization(DWORD RTiK_Process)			
Argument	Meaning	Value	Meaning
DWORD RTiK_Process	인스턴스RTiK-MP CPU 번호	RTiK-MP CPU 번호	
RTiK_CreateThread(void (*Function)(void), int times, int count)			
- 실시간 스레드 생성			
Argument	Meaning	Value	Value Meaning
void (*Function)(void)	동작 시키고자 하는 실시간 스레드	RTiK_1553B_BC	1553B BC 통신 스레드
		RTiK_1553B_RT	1553B RT 통신 스레드
		RTiK_RS232_SEND	RS232 Send 스레드
		RTiK_RS232_RECEIVE	RS232 Receive 스레드
		RTiK_LAN_SEND	LAN Send 스레드
RTiK_LAN_RECEIVE	LAN Receive 스레드		
int times	스레드 동작 주기	1 ~ 1000	ms 단위 주기
int count	스레드 동작 횟수	0 ~ 65535	0 : 무한대
RTiK_Enable(int nRequiredResolution)			
- RTiK_MP 활성화			
Argument	Meaning	Value	Meaning
int nRequiredResolution	RTiK-MP 활성화 주기	1 ~ 1000	ms 단위 주기
RTiK_Disable()			
- RTiK_MP 비활성화			
RTiK_Event_clear()			
- 동작된 RTiK-MP Event 제거 (Kernel 영역)			
RTiK_End()			
- 초기화 시에 설정된 EventHandle 제거 (User 영역)			

(그림 5) 미들웨어에서 1553B의 제어를 위한 API

MIL-STD-1553B 통신 보드에서 BC와 RT를 초기화하기 위한 API는 (그림 6)과 같다.

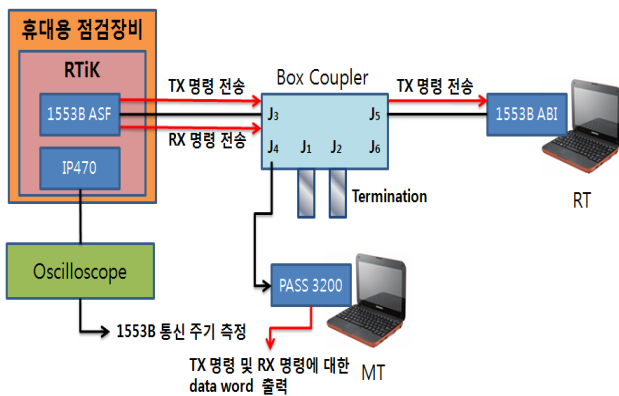
RTK_INL_1553B_BC(SBS16 DeviceNumber)				
- Simple BC test 초기화				
Argument	Meaning	Value	Meaning	
SBS16 DeviceNumber	디바이스번호	-	-	-
RTK_INL_1553B_RT(SBS16 DeviceNumber)				
- Multiple RT/SA data buffer를 사용하는 RT test 초기화				
Argument	Meaning	Value	Meaning	
SBS16 DeviceNumber	디바이스번호	-	-	-
RTK_INL_1553B_AddLink(SBS16 DeviceNumber, SBS_CHAIN_LINK_INFO *ctrlblk)				
- BC Chain Program에 Link 연결				
Argument	Meaning	Value	Meaning	
SBS16 DeviceNumber	디바이스번호	-	-	-
SBS_CHAIN_LINK_INFO ctrlblk	SBS_CHAIN_LINK_INFO 구조체에 대한 포인터	-	-	-
RTK_INL_1553B_DeleteLink(SBS16 DeviceNumber, SBS32 LinkID, SBS16 ChainID)				
- BC Chain Program에 Link 해제				
Argument	Meaning	Value	Meaning	
SBS16 DeviceNumber	디바이스번호	-	-	-
SBS32 LinkID	Link ID	-	-	-
SBS16 ChainID	Chain ID	-	-	-
RTK_END_1553B(SBS16 DeviceNumber)				
- 1553B 통신 close				
Argument	Meaning	Value	Meaning	
SBS16 DeviceNumber	디바이스번호	-	-	-

(그림 6) 1553B 통신의 BC와 RT를 초기화를 위한 API

4. 실험 환경 및 결과

4.1 실험 환경

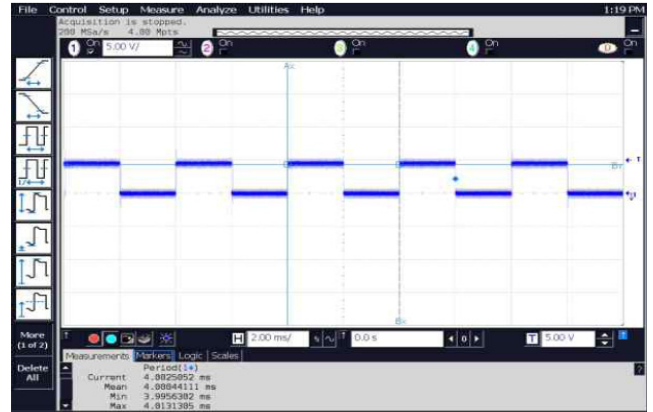
실험 방법은 커널 레벨과 유저 레벨에서 2ms의 주기를 가지는 실시간 쓰레드를 생성하여 실행 주기를 각각 측정하였다. 이때 윈도우즈에서 동작하고 있는 다른 쓰레드가 있는 경우, 쓰레드의 실행 주기에 영향을 주는지 측정하기 위해 무한으로 while문을 수행하는 프로세스를 생성하여 주기를 측정하였다. 성능검증은 매 주기마다 I/O포트를 사용하여 범용 계측기로 출력 신호를 확인하였다. 아래 (그림 7)는 MIL-STD-1553B의 실시간 통신 실험 환경을 나타낸 그림이다.



(그림 7) MIL-STD-1553B 통신 실험 환경

4.2 실험 결과

윈도우즈 기반의 점검장비에 RTiK을 이식하고 MIL-STD-1553B 통신의 주기적인 데이터를 범용 계측기로 측정된 출력 신호의 모습은 (그림 8)과 같다. 그림에서 볼 수 있듯이 주기를 벗어나지 않고 최소 주기인 2ms로 신호가 출력되는 것을 확인함에 따라 실시간 통신이 이루어짐을 검증하였다.



(그림 8) 1553 통신의 주기적 실시간성 점검결과

5. 결론

점검장비는 실시간으로 데이터를 획득하고 평가하기 위해 실시간성을 요구하고 있다. 그러나 점검장비에 많이 사용되는 범용 운영체제인 윈도우즈는 실시간성을 제공하지 못하는 문제점이 존재하기 때문에 실시간성을 제공하기 위해서는 RTX나 INTime과 같은 고가의 상용솔루션을 사용해야 한다. 본 논문에서는 개발 비용의 절감과 개발의 용이성을 제공할 목적으로 LIG 넥스원에서 산학으로 개발한 실시간 이식 커널(RTiK)을 사용하여 미들웨어를 구현하였고, 이를 통해 실시간성을 제공하는 1553B API를 제공하였다. 실시간성을 제공하는 1553B 디바이스 드라이버를 구현하기 위해 RTiK을 이용하여 생성된 실시간 1553B 통신 쓰레드에게 주기적인 신호를 보내도록 하여 실시간성을 보장하였다. 또한 성능검증을 위해 범용 계측기를 사용하였고, 주기적인 신호 출력하여 실시간성을 확인하였다.

향후 연구과제로는 점검장비에 사용되는 컴퓨터 발전 추세에 맞게 멀티 코어뿐만 아니라 Windows 8 환경을 지원하는 연구가 필요하다.

참고문헌

- [1] <http://www.intervalzero.com>
- [2] <http://www.tenasys.com>
- [3] 김중진, "윈도우 기반 점검장비의 실시간성 보장을 위한 확장 커널의 설계 및 구현" 충남대학교, 2011
- [4] Intel, "Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3 : System Programming Guide", September, 2009
- [5] 이봉식, "IT Expert 윈도우 디바이스 드라이버", 한빛미디어, 2009.
- [6] 장덕영, "Windows 구조와 원리", 한빛미디어, 2009
- [7] Johnson M.Hart, "Windows System Programming 3rd Edition", Addison Wesley, 2005.
- [8] 조문행, 김주만, 이철훈, "실시간 시스템을 위한 시간 결정적 메모리 할당 알고리즘", 한국차세대컴퓨팅학회논문지, 제6권, 제3호, pp41-49, 2010.