

항공기 무장/장착물 연동성 검증을 위한 실시간 통합점검시스템 설계 및 구현

Design and Implementation of a Real-time Integrated Inspection System to Verify Aircraft Armament/Equipment Interoperability

김효중* · 김지민 · 박찬영 · 권병기

LIGNex1Co 연구개발본부

Hyo-Joung Kim* · Ji-Min Kim · Chan-Young Park · Byung-Gi Kwon

R&D Center of LIGNex1Co, Ltd, Gyeonggi-do, 13488, Korea

[요 약]

국내에서 한국형 전투기 개발이 진행되고 있다. 이로 인해서 다양한 임무를 수행할 수 있는 무장/장착물이 장착되어 운용될 예정이다. 항공기와 무장/장착물이 성공적으로 장착되고, 운용되기 위해서는 연동 표준을 기반으로 설계 및 개발되어야 하며, 절차에 따라 동작해야 한다. 그리고 많은 반복 시험을 통해서 안전성과 신뢰성을 검증해야 한다. 본 논문에서는 반복 시험을 수행함에 따라 소모되는 자원(시간, 비용, 인력)을 최적화하기 위해서 항공기 표준 인터페이스(MIL-STD-1760)에 따라 점검시스템을 제작하고, 항공기 연동 절차를 실시간으로 수행할 수 있는 소프트웨어를 개발해서 무장/장착물의 연동성을 사전에 검증할 수 있는 통합점검시스템을 확보하였다.

[Abstract]

Recently, the development of a Korean fighter jet is underway in Korea. As a result, various armament/equipment that can perform various missions will be installed and linked. In order for the aircraft and armament/equipment to be successfully installed and linked, they must be designed and developed based on linkage standards and must operate according to the operating procedures. In addition, safety and reliability must be verified through many repeated tests. In this paper, in order to optimize the resources (time, cost, manpower) consumed by performing repeated tests, an inspection system was developed according to the MIL-STD-1760, an airborne standard interface, and software that can perform aircraft linkage procedures in real time was installed to secure an integrated inspection system that can verify the linkage of armament/equipment in advance.

Key word : Aviation armament/equipment, MIL-STD-1760, MIL-STD-1553B, Real-time inspection system.

<http://dx.doi.org/10.12673/jant.2024.28.5.664>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 27 September 2024; Revised 26 October 2024

Accepted (Publication) 29 October 2024 (31 October 2024)

*Corresponding Author; Hyo-Joung Kim

Tel: +82-42-718-3624

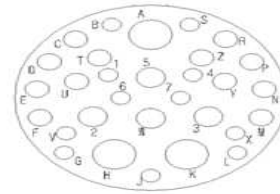
E-mail: hyojoung.kim@lignex1.com

1. 서론

한국형 전투기 개발 진행으로, 전투기의 다양한 임무 수행을 위한 항공 무장/장착물 역시 개발이 계획되거나 진행되고 있다. 항공기에 탑재되어 운용되는 무장/장착물은 각종 전자 계측장치부터 항공 무장에 이르기까지 그 종류가 다양하기 때문에 이를 제어하거나 검사하기 위해서 표준연동방식(MIL-STD-1760)을 적용하고 있다 [1],[2]. 표준연동방식이 제정되기 전에는 항공기와의 연동 인터페이스에 대한 기준이 없었기 때문에 제조사마다 독자적인 인터페이스를 개발해서 무장/장착물에 적용하였다. 그 결과, 항공기와 연동하는 과정에서 수정 및 개선 사항이 많이 발생하게 되었고, 시간과 비용이 증가하는 문제점도 발생하였다. 이러한 문제점을 극복하기 위해서 논리적, 물리적, 전기적 인터페이스와 커넥터 연결에 대해서 정의된 표준연동방식 기준을 마련하였고, 무장/장착물에 적용하였다. 그 결과, 하드웨어 및 소프트웨어 기능 추가 및 수정에 따른 문제 발생 확률 및 통합 비용도 낮추고, 상호운용성도 확보할 수 있었다[3]-[6]. 국내에서도 이와 같은 표준연동방식을 적용해서 무장/장착물 개발을 진행하고 있다. 하지만, 표준연동방식을 바탕으로 무장/장착물 개발을 수행해도 실제 항공기와 통합하고, 상호운용성을 검증하기 위해서는 많은 시간과 비용 그리고 인력들이 요구된다. 따라서 실제 항공기와 통합하기 전에 지상에서 각종 점검 및 시뮬레이션을 통해 무장/장착물에 장착되는 탑재 장비들의 성능을 실시간으로 점검하고, 상호운용성을 확인할 수 있는 통합점검 시스템이 필요하다. 이를 위해서 표준 연동방식에서 요구하는 전기적, 물리적, 논리적 인터페이스대로 통합점검 시스템의 하드웨어 및 소프트웨어를 설계하고, 구현해야 한다. 본 논문에서는 표준연동방식을 기반으로 무장/장착물의 상태를 점검하고, 연동 절차에 따라 항공기와 무장/장착물 간 상호운용성을 검증하고, 실시간으로 성능을 확인할 수 있는 통합점검 시스템을 설계하고 구현하였다.

II. 통합점검 시스템 하드웨어 설계 방안

본 장에서는 항공기와 무장/장착물 간의 상호운용성을 검증하고, 성능을 점검할 수 있는 통합점검 시스템의 주요 설계 방안을 기술하였다. 통합점검시스템은 서론에서 기술했듯이 전기적, 물리적, 논리적 인터페이스대로 설계해야 한다. 즉, 그림 1처럼 기능이 정의된 물리적 인터페이스 통해서 무장/장착물을 투하하기 전 상태 감시 및 제어를 위한 전원 공급과 통신주소 설정, 상태 확인, 연동 절차 수행 및 제어할 수 있는 전기적 인터페이스를 만족하는 배꼽 케이블(umbilical cable)을 설계 및 제작해야 한다.



K, H	Mux A & B (twinaxial) (MIL-STD-1553)
5	High Bandwidth 1 (20 Hz - 1.6 GHz)
W, 3, 2	HB 2*, 3 & 4* (20 Hz - 20 MHz)
(HB 1, 2 = 50	Ohm co-ax., HB 3, 4 = 75 Ohm co-ax)
A	Low Bandwidth (twinaxial) (Audio)
Y, U	Fiber Optic 1* & 2* (reserved)
1	Release Consent
B, S	Interlock & interlock return
L, X, 7, 4, V	Address bits 0 thru 4
G, 6	Address parity & return
T	Structure ground (safety, 10 Amp)
C, D	28 VDC power 1 & return (10 Amp)
F, E	28 VDC power 2 & return (10 Amp)
P, M, J, Z	115 VAC, 3 phase & return (10 Amp)
R, N	270 VDC power* & return* (10 Amp)

* = Not required for Class II interface

그림 1. 전기적/물리적 인터페이스
Fig. 1. Electrical/Physical Interface.

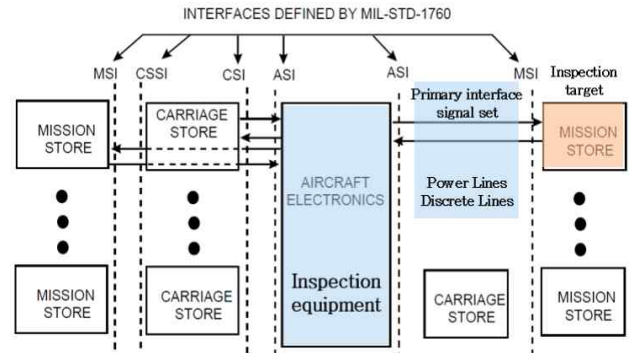


그림 2. 점검 시스템 설계 영역
Fig. 2. Inspection system design area.

배꼽 케이블과 체결되는 통합점검 시스템은 그림 2의 구조와 같이 표준연동방식에서 정의된 항공기/장착물 간 전기적 상호연동체계 (AEIS; aircraft/store electrical interconnection system) 중 항공기 장착물 인터페이스 (ASI; aircraft store interface)를 기반으로 설계하였다.

2-1 전원 인터페이스 설계

본 절에서는 항공기와 유사하게 무장/장착물에게 다양한 전원을 공급할 수 있는 통합점검 시스템의 전원 인터페이스 설계 방안에 대해서 기술하였다. 전원 인터페이스는 전원 및 접지 인터페이스 기준(MIL-HDBK-1760)을 만족하기 위해서 1대의 3상 115VAC 전원과 2대의 28VDC 전원을 설계에 반영하였다. 또한, 표 1 공급 전원 및 정격 전류를 충분히 만족하는 검증된 상용 전원 공급기를 선정하였다. 또한, 접지는 단일 지점 접지(single point ground)를 적용하고, 개별적으로 전원을 제어하기 위해 릴레이(M83536 relay)를 그림 3처럼 설계에 반영하였다.

표 1. 전원 및 정격 전류

Table 1. Power and rated current.

Voltage	Current rating
28 VDC PWR	10 A
115 VAC 3phase	10 A/phase

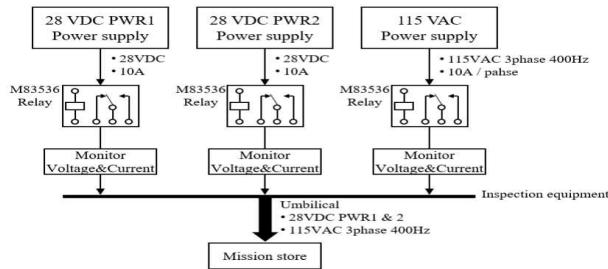


그림 3. 점검 시스템 전원 설계
Fig. 3. Inspection system power design.

2-2 이산신호 인터페이스 설계

본 절에서는 통합점검 시스템에 반영된 표2와 같은 4종의 이산신호에 대해서 설계 방안을 기술하였다. 주소 (address)는 무장/장착물의 1553 터미널 주소를 정의하기 위한 5개의 바이너리 비트(A0~A4), 1개의 패리티 비트 (parity) 그리고 1개의 공통 비트(return)로 구성되어 있다. 이때 신호는 그림 4처럼 ASI에서 구성을 제공하지만 귀환 경로(return path) 및 구조 접지(structure ground)와 완벽하게 절연이 되어야 한다. 본 논문의 통합점검 시스템은 연동 시스템과 절연된 상태의 주소 회로를 제공하기 위해 주소 비트, 패리티 비트 및 공통 비트까지 광 커플러(optical coupler)를 적용해서 무장/장착물에 다양한 주소를 모의할 수 있도록 그림 4처럼 설계하였다. 또한, 항공기와 무장/장착물 간의 물리적 장착 여부를 상시 확인하기 위해서 무장 분리 연동신호(interlock)와 28VDC를 그림 5처럼 직결하도록 MIL-HDBK-1760에서 권장하지만, 통합점검 시스템은 무장분리 연동신호 점검을 위해서 배꼽 케이블로 무장/장착물에 연결한 후 해당 신호를 최초 1회 확인하고, 제어를 통해서 물리적 접촉을 확인할 수 있도록 광 커플러를 Interlock과 Interlock return 신호에 각각 적용하여 그림 6처럼 설계하였다.

표 2. 이산신호 라인

Table 2. Discrete lines.

Signal	Description	
Address	A0~A4	Assign the store MIL-STD-1553 Address
	Parity	
	Return	
Interlock	Monitoring store mated status	
Interlock return		
Release consent	Store release consent (not a firing signal)	

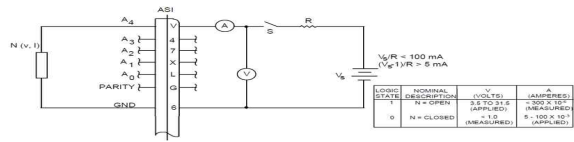


그림 4. ASI address 전기적 특성
Fig. 4. ASI address electrical characteristics.

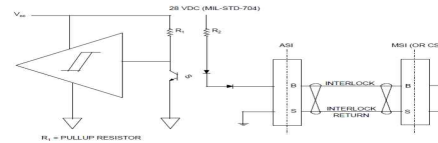


그림 5. Interlock 회로 (MIL-HDBK-1760)
Fig. 5. Circuits for the interlock function.

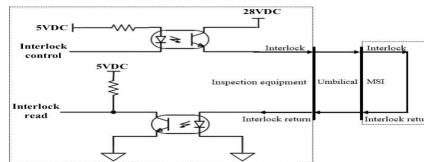


그림 6. Interlock 신호 설계
Fig. 6. Interlock signal design.

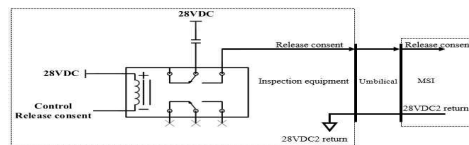


그림 7. Release consent 신호 설계
Fig. 7. Release consent signal design.

항공기가 무장/장착물에게 보내는 투하동의신호(release consent)는 단일 라인으로 28VDC/Open 규격을 갖는다. 이를 위해서 MIL-HDBK-1760에서 제안하는 전자식 릴레이 (electromechanical relay)를 적용하였으며, 공급 전원은 전원 라인의 28VDC Bus line이므로 해당 전원을 적용하고 과 전류 보호를 위한 5A 퓨즈를 그림 7처럼 반영하였다.

2-3 통신 인터페이스(1553B 버스) 설계

통합점검 시스템은 1553B 통신에서 버스 컨트롤러 기능을 수행하며, 이를 위해 MIL-STD-1760의 전송 전압 기준을 충족하는 1553B 통신 카드를 적용하였다. 통신 인터페이스는 통합점검 시스템의 특성상 무장/장착물 간 1:1 통신을 수행하기 때문에 다중 통신은 고려하지 않았다. 통합점검 시스템과 무장/장착물이 물리적으로 멀리 떨어진 점검 환경에서도 점검을 수행할 수 있도록 그림 8처럼 배꼽 케이블에 1553B 커플러를 적용해서 거리 이격에 따른 영향성을 최소화하였다.

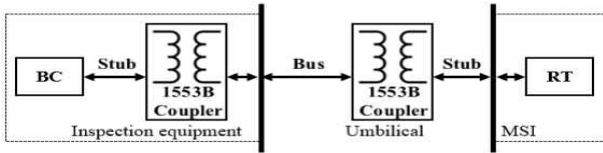


그림 8. MIL-STD-1553B 버스 설계
Fig. 8. MIL-STD-1553B bus design.

III. 통합점검 시스템 소프트웨어 구현 방안

서론에서 기술했듯이 항공기와 무장/장착물 간의 상호 운용성을 검증하기 위해서 연동 규격서에 정의된 절차와 타이밍에 따라 신호를 제어하고, 통신 명령을 인가해야 무장/장착물의 성능을 정밀하게 검증할 수 있다. 이를 위해서 통합점검 시스템은 실시간성을 보장해야 한다.

3-1 리눅스 운영체제 기반의 소프트웨어 설계

최근 군용장비의 첨단화가 가속됨에 따라 장비의 성능과 안정성을 검증하는 통합점검 시스템의 역할이 중요해지고 있다. 즉, 다양한 환경에서 운용되는 첨단 군용장비의 성능과 안정성을 검증하기 위해서는 통합점검 시스템의 실시간성이 보장되어야 한다. 실시간성이란 단순히 정해진 일을 빠르게 처리하는 것이 아니라 정확한 주기에 정해진 일을 수행해야 하고, 그 결과가 정해진 시간에 전달되어야 하는 것을 의미한다. 이러한 실시간성 지원 여부는 통합점검 시스템의 운영체제 성능에 의해서 결정된다. 군에서 운용 중이거나 무기체계 업체에서 개발 중인 통합점검 시스템은 대부분 VxWorks 또는 윈도우 운영체제 기반으로 개발되었다. 윈도우는 편리한 개발 환경 및 GUI 지원 등의 장점을 가지고 있지만 라운드로빈(round-robin) 스케줄링 정책을 사용하기 때문에 실시간성을 보장하지 못하는 단점이 있다. 이는 이 기종 컴퓨터 간의 데이터 통신을 수행할 때, 세밀한 주기를 지원하지 않아서 주기 오차가 발생하고, 주기 오차가 누적됨에 따라 데이터 손실 확률도 높아져서 점검 신뢰성을 확보할 수 없게 된다. 반면에 VxWorks는 우선순위 선점형 스케줄러 방식으로 실시간성을 보장하지만, 윈도우 운영체제와 비교하면 GUI 기능이 부족하고, 구입 및 유지 비용이 비싸다는 단점이 있다. 또한, 운영체제 버전에 따라 하위 버전과 상위 버전 간 상호 호환이 안 되는 경우도 있다. 즉, 시스템 호환성과 유지성 측면에서 문제점이 발생하게 된다. 이와 같은 두 운영체제의 단점을 보완하고, 실시간 처리 성능을 확보하기 위해서 본 논문에서는 리눅스 운영체제 기반으로 소프트웨어를 설계하였다.

3-2 실시간 처리 성능 확보 방안

앞에서 기술했듯이 통합점검 시스템의 점검 소프트웨어는 실시간 성능을 보장하는 것이 중요하다. 실시간 성능은 운영체제의 인터럽트 지연시간(interrupt latency)을 측정해서 성능을 확인할 수 있다. 인터럽트 지연시간이란 하드웨어가 인터럽트를 발생시킨 시점부터 해당 인터럽트를 소프트웨어가 처리하는 시점까지의 측정 시간을 의미한다. 그 측정 시간이 짧을수록 운영체제의 실시간 처리 성능이 우수하다고 말할 수 있다. 실시간 처리 성능이란 정해진 작업 또는 이벤트에 대해서 운영체제가 신속하게 반응해서 정해진 시간에 마감하는 성능으로 정의할 수 있다. 이와 반대로 마감이 없는 처리를 범용 처리 성능으로 정의할 수 있다. 리눅스는 원래 범용 처리용 운영체제이기 때문에 실시간 운영체제에 비해서 실시간 처리 성능이 좋지 않다. 이를 해결하기 위해서 인터럽트 금지 상태를 짧게 하는 소프트웨어 패치를 통해서 실시간 처리 성능을 일부 확보할 수 있다. 즉, 리눅스 운영체제는 실시간 처리 동작을 우선순위로 처리하고, 범용 처리 동작은 뒤로 미루는 스케줄러를 생성해서 실시간성을 확보할 수 있었다. 그러나, 범용 처리 동작을 후순위로 처리해도 파일 처리나 네트워크 처리 같은 작업은 실행된다. 이러한 작업은 취급하는 데이터양에 따라서 인터럽트 금지 상태의 시간이 변하기 때문에 실시간 처리 성능을 악화시키는 경우도 발생할 수도 있다.

이를 해결하기 위해서 멀티코어 기반의 리눅스 운영체제로 소프트웨어를 구현하였다. 소프트웨어 패치를 통해서 실시간성을 일부 확보하고, 멀티코어를 통해서 범용 처리 작업과 실시간 처리 작업으로 구분해서 처리할 수 있도록 하였다. 또한, 인터럽트 금지 상태의 시간 길이가 실시간 처리 성능에 영향을 주지 않도록 CPU 보호(CPU shielding) 기법을 실시간 처리 코어에 적용하였다.

3-3 CPU 보호 기법

리눅스 운영체제에서는 작업 처리를 특정 코어에 지정하는 친화성(affinity) 기능이 있다. 친화성 기능을 사용하여 실시간 처리를 전용 코어에 할당하고, 해당 코어가 다른 인터럽트에 의해서 영향을 받지 않게 하는 것이 그림 9와 같은 CPU 보호 기법이다.

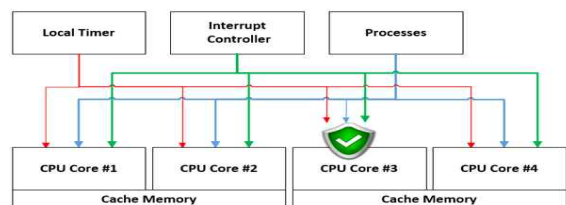


그림 9. CPU 보호 기법
Fig. 9. CPU shielding.

```
top - 05:44:07 up 12 min, 1 user, load average: 0.90, 0.32, 0.11
Tasks: 262 total, 2 running, 260 sleeping, 0 stopped, 0 zombie
%Cpu0 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5 :100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6 :  1.0 us,  1.0 sy,  0.0 ni, 98.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7 :  1.0 us,  0.0 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 128658.7 total, 125948.6 free,  1386.4 used,  1323.7 buff/cache
MiB Swap: 16384.0 total, 16384.0 free,  0.0 used, 125756.7 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	MEM	TIME+	COMMAND
7476	root	20	0	222520	3180	2892	R	100.0	0.0	1:21.37	bash
6496	root	20	0	4195996	200700	183676	S	1.0	0.2	0:07.37	gnome-shell

그림 10. CPU 비보호 상태 확인
Fig. 10. CPU non-shielding.

```
-----
CPUID  irqs  ltms  proc
-----
0      no    no    no
1      no    no    no
2      no    no    no
3      no    no    no
4      no    no    no
5      yes   no    yes
6      no    no    no
7      no    no    no

top - 05:43:10 up 12 min, 1 user, load average: 1.67, 0.34, 0.20
Tasks: 267 total, 2 running, 265 sleeping, 0 stopped, 0 zombie
%Cpu0 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2 :100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 128658.7 total, 125915.4 free,  1399.8 used,  1343.5 buff/cache
MiB Swap: 16384.0 total, 16384.0 free,  0.0 used, 125724.9 avail Mem
```

그림 11. CPU 보호 상태 확인
Fig. 11. CPU shielding.

CPU 보호 기법의 성능을 확인하기 위해서 무한 루프를 수행하는 프로그램을 백그라운드 형태로 실행시킨 후 CPU 자원(resource) 상태를 확인하였다. 그림 10을 통해서 무한 루프 프로그램이 5번 CPU의 자원을 100% 사용하는 것을 확인할 수 있다. 즉, 통합점검 시스템의 소프트웨어가 실행되는 5번 CPU에서 실행된다고 가정했을 때 다른 프로세스나 인터럽트가 해당 CPU에 영향을 주어서 실시간성이 보장되지 않을 수 있다는 것을 유추할 수 있다. 이와 같은 상태에서 CPU 보호 기법을 5번 CPU에 적용하였다. CPU 보호 기법을 적용한 결과 그림11처럼 5번 CPU는 로컬타이머, 인터럽트, 프로세스부터 영향을 받지 않는 상태로 변했고, 5번 CPU에서 동작하던 프로그램도 2번 CPU로 재할당된 것을 확인할 수 있다. 이를 통해서 통합점검 시스템의 소프트웨어가 실행되는 CPU를 지정하고, 해당 CPU에 보호 기법을 적용하면 실시간성을 보장할 수 있다.

3-4 통합점검 시스템 소프트웨어 구조 설계 및 구현

통합점검 시스템의 소프트웨어는 실시간성을 확보하기 위해서 CPU 보호 기법을 활용해서 통합점검 시스템의 소프트웨어 구조를 설계하였다. 그림 12에서 볼 수 있듯이 스케줄러(scheduler)를 통해서 응용 프로그램이 지정된 CPU에서 동작할 수 있도록 할당하였다. 스케줄러를 통해서 응용프로그램은 1ms 주기로 동작하면서 입력장치(input device)에 할당된 CPU로부터 데이터를 수집한다. 수집된 데이터는 응용프로그램의 입력 태스크(task)를 통해서 처리되고, 출력 태스크(output task)로 전달된다. 출력 장비(output device)에 할당된 CPU는 출력 태스크를 통해서 수신한 명령대로 장비를 제어하게 된다. 이와 같은 구조로 소프트웨어를 구현한 후 항공기와 무장/장착물 간의 소프트웨어에서 설정한 통신 메시지 주기에 따라 연동 절차를 수행하였고, 1553B 분석 프로그램을 통해서 항공기와 무장/장착물 간의 메시지 송수신 주기를 확인하였다. 그리고, 분석 프로그램을 통해서 통신 절차를 확인하였고, 통합점검 시스템과 무장/장착물 간의 통신 메시지의 주기를 측정하였다. 측정 결과는 표3과 같았고, 통합점검 시스템의 통신을 담당하는 소프트웨어가 실시간성을 유지하는 것을 확인할 수 있었다.

리되고, 출력 태스크(output task)로 전달된다. 출력 장비(output device)에 할당된 CPU는 출력 태스크를 통해서 수신한 명령대로 장비를 제어하게 된다. 이와 같은 구조로 소프트웨어를 구현한 후 항공기와 무장/장착물 간의 소프트웨어에서 설정한 통신 메시지 주기에 따라 연동 절차를 수행하였고, 1553B 분석 프로그램을 통해서 항공기와 무장/장착물 간의 메시지 송수신 주기를 확인하였다. 그리고, 분석 프로그램을 통해서 통신 절차를 확인하였고, 통합점검 시스템과 무장/장착물 간의 통신 메시지의 주기를 측정하였다. 측정 결과는 표3과 같았고, 통합점검 시스템의 통신을 담당하는 소프트웨어가 실시간성을 유지하는 것을 확인할 수 있었다.

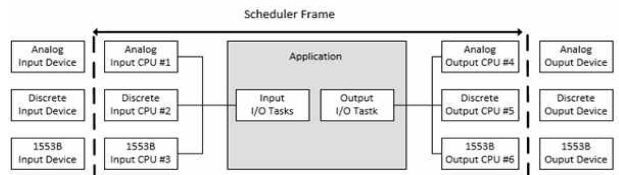
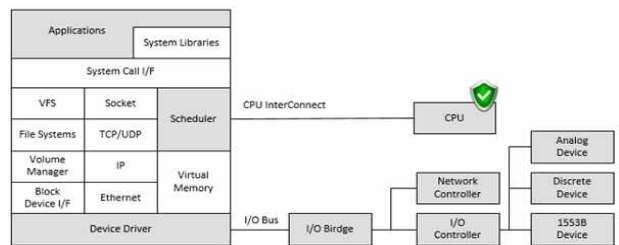


그림 12. 통합점검 시스템 소프트웨어 구조
Fig. 12. Test system software architecture.

표 3. 주기 측정
Table 3. Measurement cycle table.

Message name	Mode	Setup (Hz)	Measured (Hz)
2R	Tx	12.5 Hz	12.5 Hz
3R	Tx	1.5625 Hz	1.5625 Hz
4R	Tx	1 Hz	1 Hz
15R	Tx	1.5625 Hz	1.5625 Hz
22R	Tx	12.5 Hz	12.5 Hz
2T	Rx	1.5625 Hz	1.5625 Hz
11T	Rx	12.5 Hz	12.5 Hz
15T	Rx	12.5 Hz	12.5 Hz
22T	Rx	12.5 Hz	12.5 Hz

V. 결론

본 논문에서는 무장/장착물의 상호운용성을 사전 검증하기 위해 실시간 통합점검 시스템을 제작하였다. 또한 검증된 무장/장착물과 연동을 통해서 통합점검 시스템의 절

차 수행 성능과 실시간성을 확인할 수 있었다. 추후에는 실제 항공기처럼 무장/장착물에 위성 신호, 공압을 모의할 수 있는 장비를 제작한 후 통합 점검시스템과 결합해서 더 정밀하게 무장/장착물을 점검하고, 검증할 수 있는 시스템으로 발전시킬 예정이다.

References

- [1] U.S. Department of Defense, *Aircraft/store electrical interconnection system*(MIL-STD-1760E), 24 Oct 2007. Retrieved from https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=37120.
- [2] U.S. Department of Defense, *Aircraft/store electrical interconnection system*(MIL-HDBK-1760A), 10 Mar 2004. Retrieved from https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=208391.
- [3] T. Kim, K. Park, J. Kim, J. Jung, and B. Kwon, "A case study on MIL-STD-1760E based test bench implementation for aircraft-weapon interface testing," *Journal of Advanced Navigation Technology*, Vol. 22, No. 2, pp. 57-63, Apr. 2018. DOI: <https://doi.org/10.12673/jant.2018.22.2.57>
- [4] H. Kim, Y. Kim, and B. Kwon, "Design and implementation of test system for safety separation and performance verification of aircraft weapons/equipment," *Journal of Advanced Navigation Technology*, Vol. 26, No. 2, pp. 47-53, Apr. 2022. DOI: <https://doi.org/10.12673/jant.2022.26.2.47>
- [5] S. Choi, J. Lee, J. Kim, S. Lee, and J. Jung, "Analysis of the correlation between armament/store integration criteria and aircraft launch missile development process," *Journal of Advanced Navigation Technology*, Vol. 22, No. 2, pp. 84-89, Apr. 2018. DOI: <https://doi.org/10.12673/jant.2018.22.2.57>.
- [6] S. Lee, S. Kim, H. Lee, and D. Cho, "A software update technique for aircraft missiles based on MIL-STD-1760," *Journal of the KIMST*, Vol. 21, No. 5, pp. 649-657, Aug. 2018. DOI: <https://doi.org/10.9766/KIMST.2018.21.5.649>
- [7] U.S. Department of Defense, *Digital time division command/response multiplex data bus*(MIL-STD-1553B), 5 Jan 1996. Retrieved from https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=36973



김 호 중 (Hyo-Joung Kim)

2009년 2월: 충남대학교 정보통신공학과 (공학석사)
2009년 2월 ~ 현재: LIG넥스원 수석연구원
※ 관심분야 : 유도탄 체계, 항공기 무장, 소프트웨어 설계



김 지 민 (Ji-Min Kim)

2014년 2월: 강원대학교 전기전자공학과(공학석사)
2014년 2월 ~ 현재: LIG넥스원 선임연구원
※ 관심분야 : 유도탄 체계, 항공기 무장, 신호처리



박 찬 영 (Chan-Young Park)

2019년 2월: 인하대학교 항공우주공학과(공학사)
2021년 7월 ~ 현재: LIG넥스원 선임연구원
※ 관심분야 : 유도탄 체계, 항공기 무장, 소프트웨어 설계



권 병 기 (Byung-Gi Kwon)

2016년 8월 : 아주대학교 IT융합대학원 (공학석사)
2006년 1월 ~ 현재 : LIG 넥스원 수석연구원
※ 관심분야 : 항공기 무장, 유도탄체계, 신호처리