

## 技術論文

DOI:10.5139/JKSAS.2010.38.6.620

## 모듈화 소형위성의 Data Bus 표준화 방안 연구

장연욱\*, 장영근\*\*

## A Study on Standardization of Data Bus for Modular Small Satellite

Yun-Uk Jang\* and Young-Keun Chang\*\*

## ABSTRACT

Small satellites can be used for various space research and scientific or educational purposes due to advantages in small size, low-cost, and rapid development. Small Satellites have many advantages of application to Responsive Space. Compared to traditional larger satellites, however, Small satellites have many constraints due to limitations in size. Therefore, it is difficult to expect high performance. To approach maximum capability with minimal size, weight, and cost, standard modular platform of Small satellites is necessary. Modularity supports plug-and-play architecture. The result is Small satellites that can be combined quickly and reliably using plug-and-play mechanisms. For communication between modules, standard bus interface is needed. Controller Area Network(CAN) protocol is considered optimum data bus for modular Small satellite. CAN can be applied to data communication with high reliability. Hence, design optimization and simplification can also be expected. For ease of assembly and integration, modular design can be considered. This paper proposes development method for standardized modular Small satellites, and describes design of data interface based on CAN and a method of testing for modularity.

## 초 록

소형위성은 저비용으로 단기간에 개발이 가능하기 때문에 이를 통해 폭 넓은 우주 연구를 수행할 수 있고 다양한 분야의 활용 가능하다. 소형위성은 빠른 개발이 가능한 이점이 있다. 따라서 신속 대응 우주 구현이 주목됨에 따라 소형위성의 역할이 급속히 증가하고 있다. 하지만 소형위성은 기존의 위성에 비해 크기나 전력의 제약이 많기 때문에 고성능의 안정적인 시스템을 기대하기는 어렵다. 이러한 문제점을 해결하고 빠른 개발을 위해 표준화와 모듈화 설계가 필요하다. 모듈화는 플러그 앤 플레이(Plug-and-Play)를 지원하여 수일 내에 인공위성 제작 및 시험을 가능하게 한다. 이러한 모듈 간의 원활한 데이터 통신을 위해서는 데이터 버스의 표준이 요구된다. CAN 통신 방식은 플러그 앤 플레이에 가장 효과적으로 대처할 수 있는 통신방식으로 꼽힌다. CAN은 높은 신뢰성을 가지며 분산 시스템을 지원하여 위성의 호환성을 높여준다. 따라서 시험이 용이해지고 짧은 기간에 고성능의 안정적인 위성 개발이 가능하게 된다. 본 논문에서는 모듈화 방식의 소형위성 개발 방안에 대해 제안하고, CAN을 데이터 버스로 적용하여 소형 위성 내부 데이터 인터페이스 버스를 설계하고 시험을 통해 적합성을 분석하여 기술하였다.

**Key Words** : Controller Area Network(CAN), Plug-and-Play(플러그 앤 플레이), Responsive Space(신속 대응 우주), Modularization(모듈화), Standardization(표준화), Small Satellite(소형위성)

† 2009년 12월 14일 접수 ~ 2010년 5월 26일 심사완료

\* 정회원, 한국항공대학교 대학원

교신저자, E-mail : yunuk1018@naver.com

경기도 고양시 덕양구 화전동 200-1

\*\* 정회원, 한국항공대학교 항공우주 및 기계공학부

## I. 서 론

90년대 이후 기존의 위성개발의 막대한 비용과 개발 기간을 최소화 하려는 노력이 나타나게 되었다. 이에 따라 소형위성 개발이 주목되고 있다. 소형 위성은 중·대형 위성에 비해 적은 예산과 짧은 개발 기간을 가지고 많은 기술적인 효과와 경험을 얻을 수 있기 때문에 미국, 유럽, 일본 등과 같은 우주 선진국에서 소형 위성 개발에 많은 관심을 기울이고 있다. 비용과 개발 기간에서의 효율성뿐만 아니라 중·대형 위성에 비해 보다 많은 발사 기회가 주어지므로 임무 수행에 있어서도 많은 이점을 가진다[1].

신속 대응 우주 구현이 주목 됨에 따라 위성의 빠른 개발이 요구된다. 소형위성은 빠른 응답성으로 그 역할이 급속히 증가하고 있지만 크기 및 전력의 제한을 가진다. 따라서 고성능을 구현하기 어려우며 낮은 신뢰성을 가지게 된다. 이러한 소형위성 개발의 문제를 해결하기 위한 방안으로 표준화와 모듈화 설계가 대두되고 있다[2].

표준화된 위성 플랫폼은 개발의 간소화와 빠른 응답성을 지원한다. 그리고 표준 플랫폼을 기반으로 새로운 기술의 적용이 용이해 진다. 또한 모듈화는 플러그 앤 플레이를 지향함으로써 개발뿐만 아니라 수정이나 보완 등이 용이하게 된다[3].

모듈 간의 데이터 통신을 위해서는 내부 데이터 버스의 표준이 요구된다. 동일한 내부 데이터 버스를 구축함으로써 서로간의 원활한 데이터 통신이 가능해진다. 이러한 예로 미 공군 연구소(Air Force Research Lab.)에서 개발한 SPA가 있다. SPA는 플러그 앤 플레이를 지원하여 각 서브시스템의 모듈화를 가능하게 한다. SPA는 USB와 Spacewire 기반의 표준 데이터 인터페이스로 중앙 집중형 구조를 갖으며, 모듈 간 통신 제어를 위한 버스 제어기가 필요하다.

중앙 집중형 구조는 안정적이지만 중앙 시스템에 많은 부하(Load)가 걸리게 된다. 이에 따라 시스템의 신뢰성 향상을 위해 잉여 시스템(Redundancy)이 필요하게 된다. 이로 인해 크기, 무게 그리고 비용이 모두 증가하게 되어 위성의 소형화에 어려움이 있다. 그 외에도, 시스템을 수정하거나 변경해야 하는 경우 해당하는 시스템 이외에 모든 시스템의 변경이 필요하게 된다.

분산 시스템 구조의 모듈화 위성 개발을 통해 이러한 문제를 해결할 수 있다. 중앙 시스템에 집중된 부하를 분산시키고 데이터 관리와 제어가 용이해져서 효율적으로 내부 시스템을 운용할 수

있게 된다. 분산 위성시스템의 모듈 간 데이터통신을 위해서는 동일한 규격의 데이터 버스 표준이 요구된다. 플러그 앤 플레이에 효과적으로 대처 가능한 내부 데이터 통신 방식으로 CAN(Controller Area Network)을 꼽을 수 있다.

CAN 통신 방식은 신뢰성이 검증된 통신 방식으로 분산화 시스템에서 사용이 용이하다. 그리고 높은 데이터 전송률과 높은 확장성, 뛰어난 데이터 전송 오류 검출 기능 등은 위성 시스템에서의 데이터 통신 신뢰성을 더욱 향상 시켜준다.

CAN을 데이터 버스로 적용한 예로 SSTL(Surrey Satellite Technology Limited)의 SNAP-1이 있다. SNAP-1은 6.5kg의 나노위성으로 350×288의 저해상도의 CMOS 카메라를 탑재체로 가진다. 영상 데이터는 CAN을 통해 전송되어 지상국으로 보내진다. CAN을 기반으로 위성 설계의 간소화를 제공하여, 개발 및 시험이 용이하게 된다. 이를 통해 알 수 있듯이, 소형위성 개발에서 CAN은 빠른 응답성과 고신뢰성 개발을 제공하고, 분산형 구조의 시스템 설계를 지원하여 유기적인 시스템 구축이 가능하게 된다.

## II. Data Bus 표준화 방안 분석

### 2.1 분산형 구조

기존의 대부분의 위성은 중앙 집중형의 시스템 구조로 구성되어있다. 중앙 집중형 구조는 제한된 기능을 수행하는 위성에는 적합하지만 고성능의 복잡한 기능을 요구하는 현대의 위성에는 적합하지 않다. 정지궤도의 통신위성이나 기상위성은 기능이 복잡하지 않고 지상국과 항상 통신이 가능하므로 지상국에 의해 관리, 제어될 수 있다. 따라서 중앙 집중형 시스템 구조도 적합하다.

하지만 저궤도의 위성들은 탐사, 첩보, 관측 등이 목적으로 사용 된다. 따라서 그 기능이 복잡하고 고성능을 요구하게 된다. 이러한 위성의 경우 중앙에 집중된 기능을 여러 서브시스템으로 분산시켜 보다 효율적인 제어 및 관리가 필요할 것으로 판단된다.

분산 시스템은 여러 개의 데이터 프로세싱 모듈들 사이에 공통의 통신 버스를 연결한 구조로, 중앙시스템에 집중된 기능들을 여러 모듈에 분산시켜 효율적인 데이터 처리와 관리를 가능하게 한다. 그리고 중앙 시스템에 집중된 부하를 균등하게 각 서브시스템으로 분배할 수 있게 한다. 이러한 부하 분산(Load Balancing)을 통해 보다 효율적인 내부 시스템 운용이 가능해 진다.

Table 2-1. Comparison of Data Buses

	USB	Firewire	Spacewire	CAN
전송속도	12Mbps	100/200/400Mbps	~400Mbps	1Mbps
연결 가능 장치 수	최대 127개	최대 63개	-	110개
Plug-and-Play 기능	-	지원	-	지원
네트워크 구조	중앙 집중형	분산형	중앙 집중형	분산형
통신 방식	마스터-슬레이브	멀티 마스터	마스터-슬레이브	멀티 마스터
분산화 구현	어려움	쉬움	어려움	쉬움

## 2.2 통신 방식

분산화를 통해 위성 시스템을 독립적인 모듈 단위로 나누어 개발할 수 있다. 모듈 간의 데이터 통신을 위해 다중 방식의 내부 인터페이스 구조가 필요하다. 다중 방식에는 마스터-슬레이브 방식과 멀티 마스터 방식이 있는데 분산화 시스템 구현을 위해서는 멀티 마스터의 방식이 적합하다. 마스터-슬레이브 방식은 데이터 제어를 위한 중앙 시스템이 필요하고 이는 중앙 집중형 시스템 구조를 가지게 된다. 현대의 고성능의 소형 위성 개발을 위해서는 분산 시스템 구조가 적합하고, 이를 위해 멀티 마스터 방식의 통신 방식이 요구된다.

Table 2-1은 플러그 앤 플레이가 가능한 다중 방식을 지원하는 통신방식들을 비교하여 정리한 것이다. USB와 SpaceWire 데이터 버스는 중앙 집중형 네트워크 형태를 가진다. 이 두 통신 방식은 플러그 앤 플레이를 위한 모듈 간 다중통신을 위해서 라우터와 같은 버스 제어가 필요하다. 이는 취약점인데, 버스 제어기에 고장이 발생하였을 경우 전체 데이터 버스가 고장 상태가 되어버리는 문제점이 있다. FireWire와 CAN의 멀티 마스터 방식은 이러한 취약점을 보강하고 버스 제어기 없이 개별적인 모듈들이 데이터 버스에 연결되어 통신하는 구조의 형태를 갖는다. 분산 시스템을 지원하며 플러그 앤 플레이 모듈화 시스템 구현을 보다 용이하게 한다.

FireWire는 애플에서 만든 데이터 인터페이스 규격으로 요즘은 많이 사용하지 않는 추세로 범용성이 낮은 단점이 있다. FireWire에 비해 CAN은 범용성이 높고 다음과 같은 장점들이 있다

### ● CAN의 특징

- ① 표준 통신 프로토콜(Protocol)로 뛰어난 시장성을 가지며 범용성이 좋다.
- ② 플러그 앤 플레이를 지원한다.
- ③ 노이즈(Noise)에 강하다.

- ④ 하드웨어적인 오류 보정이 있다.

오류 발견과 보정은 CAN 컨트롤러 자체에서 처리되므로 그에 따른 처리가 현저히 감소하게 된다.

- ⑤ 우선순위가 있다.

컨트롤러의 ID에 따라 버스를 사용할 수 있는 우선순위가 결정된다. 이에 따라 하드웨어적으로 데이터 전송이 중재되기 때문에 데이터의 손실이나 지연에 대해 고려할 필요가 없다.

- ⑥ 사용되는 전선의 양을 단순화 할 수 있다.

CAN은 자동차의 분산 제어 시스템을 위해 처음 개발되었으며 열악한 환경에서의 낮은 에러율로 신뢰성이 높은 통신 방식이다. 자동차를 비롯한 항공, 철도, 우주항공 등 다양한 분야에서 적용되어 사용되고 있다. 이처럼 CAN을 기반으로 한 데이터 버스 표준화가 분산 시스템 구조의 모듈화 위성 개발에 더 적합할 것으로 생각된다. 각 서브시스템의 전장보드 크기, 전력, 무게 등의 설계 요소의 규격화를 통해 CAN 데이터 버스 시간의 모듈화 된 위성을 개발하여, 위성 개발의 확장성을 보다 높일 수 있다.

## III. CAN 기반의 Data Bus 설계

### 3.1 데이터 프레임(Data Frame)

본 연구에서는 플러그 앤 플레이 모듈화를 위해 데이터 인터페이스로 CAN을 사용하였다. CAN은 분산 시스템 구현이 용이한 멀티 마스터 구조를 가진다. 각각의 서브시스템은 데이터 통신을 위해 자신의 고유한 ID를 갖는다. 메시지 충돌의 중재는 ID를 통하여 우선순위 결정으로 이루어진다. 수신 노드는 ID를 확인한 후, 전송된 메시지를 무시하거나 저장하게 된다. 메시지를 수신하면 ACK 신호를 발생시켜 송신측이 데이터

수신을 확인 할 수 있도록 한다. 한 번에 전송 가능한 데이터 수는 최대 8 바이트(Bytes)이고 최대 1Mbps의 전송속도를 제공한다. CAN 컨트롤러는 CAN 버스 상에서 프레임(Frame)이라고 불리는 패킷(Packet)으로 데이터를 전송하게 된다. CAN 프레임은 중재 ID, 데이터 바이트, ACK 비트 등으로 구성된다.

CAN은 하드웨어적으로 데이터 프레임을 제공하여 사용자는 데이터 버퍼에 데이터를 쓰고 전송만 하면 그 외 모든 처리는 CAN 컨트롤러가 알아서 수행하게 된다. 따라서 분산 제어 분야에 매우 용이하다. Fig. 3-1은 CAN의 데이터 프레임을 나타낸다.

CAN의 메시지 프레임은 데이터 프레임, 리모트(Remote) 프레임, 에러(Error) 프레임, 오버로드(Overload) 프레임이 있다. 데이터 프레임은 일반적으로 전송되는 데이터 메시지의 프레임으로 CAN 버스 상의 모든 노드에 의해서 생성 가능하다. 0~8byte의 데이터를 CAN 버스로 전송하게 된다.

리모트 프레임은 수신을 원하는 데이터를 특정 노드에 요청하는 경우에 전송하는 프레임으로, 원하는 데이터를 가지는 해당 노드의 ID 정보를 실어 전송하게 된다. 데이터 프레임에서 데이터 필드가 없는 프레임 구조를 가지며 DLC 비트를 설정할 때 수신을 원하는 데이터의 프레임과 같은 값으로 설정해야 한다.

에러 프레임은 CAN 버스를 통해 데이터 통신 중 또는 종료 후 에러가 감지되었을 때 발생하게 된다. 에러가 발생한 노드는 에러 플래그(Error Flag)를 발생 시키게 된다. 다른 노드들의 정상적인 데이터 송·수신을 방해 하지 않도록 하며 에러 플래그가 발생하면 송신 노드는 데이터를 재전송 하고 수신 노드는 에러가 발생한 데이터는 삭제하고 새로 데이터를 수신하게 된다. 오버로드 프레임은 에러 프레임과 동일한 구조를 가지며 CAN 버스로부터 데이터를 수신할 때 다음 데이터 프레임을 받기 위한 준비가 되지 않았을 경우 발생하게 된다.

소프트웨어 개발자는 소프트웨어 설계 시 에러 프레임과 오버로드 프레임은 고려할 필요가 없으며 데이터 프레임과 리모트 프레임만을 사용하여 설계하게 된다.



Fig. 3-1. CAN Data Frame

### 3.2 CAN 소프트웨어

데이터 인터페이스를 위한 CAN 통신 소프트웨어는 기존의 CAN 통신 예제 코드를 기반으로 재작성 하였다. 기존의 CAN 통신 코드는 자동차 및 다른 산업 환경에서 사용되어 불필요한 코드가 많으며 복잡하고 적용이 쉽지 않다. 따라서 보다 간편하고 쉽게 사용할 수 있도록 편집하였다. C언어는 이식성과 호환성이 뛰어나고 테스트를 위해 컨트롤러에 적용이 용이하여 C언어로 제작하였다.

CAN을 적용하여 각 서브시스템의 소프트웨어를 작성 할 때 Fig. 3-2와 같은 기본 구조로 작성된다. CAN 통신과 관련된 헤더 파일을 링크시커 main함수에서 CAN 함수를 사용 할 수 있도록 한다. main함수 작성 전에 CAN 버스 상의 서브시스템들의 ID를 정의하도록 한다. main함수에서 해당 서브시스템의 동작에 대한 코드를 작성하기 전에 CAN 데이터 송·수신을 위한 데이터 저장 버퍼의 변수를 선언하고 CAN을 초기화 한다. CAN 수신 설정 코드를 통해 CAN 통신을 활성화 시킨 후 해당 서브시스템의 코드를 작성한다.

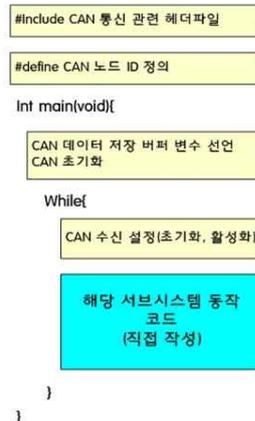


Fig. 3-2. Software Structure

## V. 시험 및 결과

### 4.1 시험 목적 및 내용

소형위성에서 플러그 앤 플레이의 모듈화를 위한 CAN 버스 적용이 설계 의도에 맞게 구현되었는지 평가하기 위해 테스트 베드(Test Bed)를 구축하여, 각 서브시스템 간의 데이터 통신을 시험하고자 한다.

모듈화를 위해 각 서브시스템은 각각의 CAN 컨트롤러가 필요하다. 인터페이스에 대한 성능

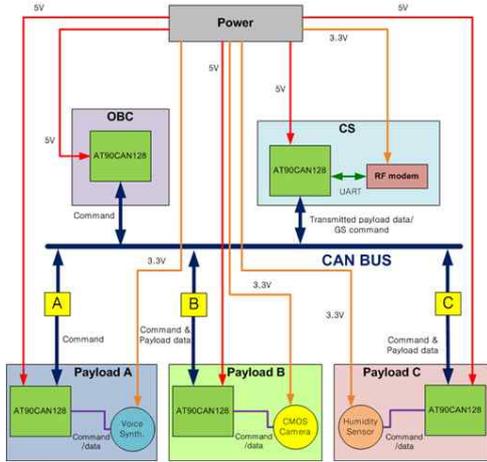


Fig. 4-1. Test Bed System Block Diagram

시험을 수행하기 위하여 사용이 용이한 컨트롤러로 트레이드-오프(Trade-off)를 진행하였다. 시험 환경의 안정성과 실제 소형위성에 장착 시 전장 보드의 소형화와 경량화를 위해 상용 부품을 사용하도록 하여, 최종적으로 ATMEL사의 CAN을 지원하는 AT90CAN128 칩을 선정하였다.

Fig. 4-1은 테스트 베드의 시스템 구성을 보여준다. 탑재 컴퓨터(OBC), 탑재체(Payload), 그리고 통신계(CS)를 구축하였고, 지상국과의 통신을 위해 송·수신기 모듈을 사용하여 무선 통신이 가능하도록 하였다. 장착되는 대부분의 칩 소자는 상용부품을 선정하였고 CAN 컨트롤러는 확장 보드를 선정하여 시험 환경 구축이 편리하도록 하였다.

4.2 시험 소프트웨어(Test Software)

앞서 설계한 내부 인터페이스 규격에 따른 서브시스템 간의 데이터 통신의 성능을 분석을 위해 시험 소프트웨어를 제작 하였다. 본 실험은 CAN 기반의 내부 통신의 효율적인 플러그 앤 플레이 검증을 위한 실험이기 때문에 CAN 버스를 통한 서브시스템 간의 명령 및 데이터 전송 과정의 모니터링과 분석에 초점을 두었다. 실제 위성 제작 시, 작성된 시험 소프트웨어를 확장하여 소형위성의 비행 소프트웨어로 활용하게 될 것이다.

또한 모듈간의 호환성을 알아보기 위해 CMOS 카메라, 음성합성기, 그리고 온습도 센서의 3가지 타입(Type)의 대체 가능한 탑재체 유닛(Unit)을 제작하였다. 세 가지 탑재체를 교체로 테스트 베드에 플러그-인(Plug-in)하여 시스템의 호환성과 확장성에 대한 시험을 수행하였다.

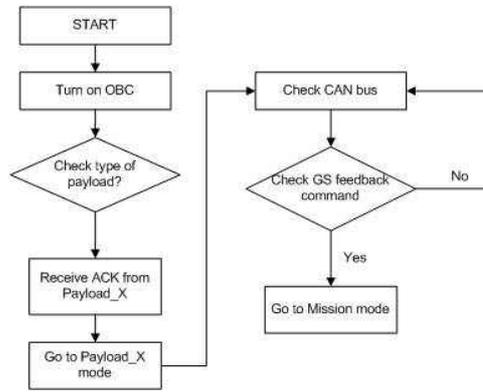


Fig. 4-2. OBC Normal mode Algorithm

시험 소프트웨어는 OBC와 CS의 운영 소프트웨어와 3가지 유닛 별 탑재체 소프트웨어로 구성된다. OBC와 CS의 시험 소프트웨어는 시험의 편리성을 위해 Payload Unit A, B, C 모든 경우에서 사용이 가능하도록 제작하였다.

OBC는 노멀 모드(Normal mode)와 미션 모드(Mission mode)로 구분된다. Fig. 4-2는 탑재체의 노멀 모드의 동작 알고리즘을 보여준다. 미션 모드는 탑재체 별로 구분하여 총 3가지 모드로 구성된다. 탑재체 타입 분석을 통해 연결된 탑재체를 확인하고 지상국의 명령에 따라 해당하는 미션 모드로 전환되어 동작을 수행하게 된다. CS는 지상국으로부터 명령을 수신 받아 CAN 버스를 통해 OBC로 전달한다. 탑재체는 OBC의 제어 명령에 따라 동작을 수행하게 된다. 카메라와 온습도 센서는 동작을 수행한 후 OBC로 결과 데이터를 OBC로 다시 전송하게 되고, OBC는 탑재체의 데이터를 수신하여 CS를 통해 지상국으로 전송하게 된다. Fig. 4-3은 탑재체 타입별 시험 소프트웨어 알고리즘을 보여준다.

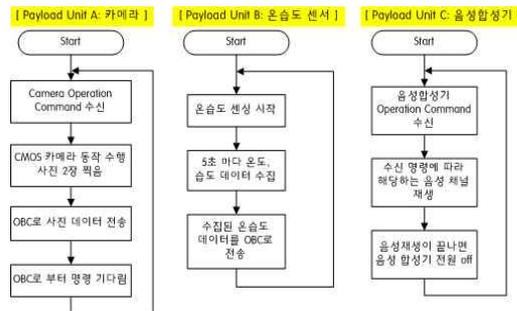


Fig. 4-3. Payload Test Software Algorithm

### 4.3 결과 분석

각 서브시스템은 CAN 버스를 통해 패킷 통신을 하게 된다. 시험은 각 서브시스템의 시험 알고리즘을 기반으로 테스트 소프트웨어를 제작하였고, 각 서브시스템의 CAN 버스를 통한 데이터 송·수신을 PC를 통해 확인해 보았다. Table 4-1은 CAN 버스에 연결되어 있는 서브시스템의 목록과 기능 및 성능시험 항목을 나타낸 것이다. CAN 버스 상의 데이터 송·수신은 랩뷰(Lab View)와 시리얼 통신 프로그램을 이용하여 확인하였다. 전원이 인가되면 CAN 버스를 통해 각 서브시스템 동작에 대한 명령이 전송된다. 탑재체는 OBC에서 동작 명령이 송신되면 작성된 알고리즘에 따라 임무를 수행하게 된다. OBC는 탑재체를 확인하기 위해 리모트 프레임을 탑재체로 전송한다. 탑재체는 OBC로부터 리모트 프레임을 수신하면 자신의 ID 정보를 실은 데이터를 OBC로 송신한다. OBC는 탑재체 확인한 후 해당하는 탑재체의 미션 모드로 전환하여 탑재체 명령을 전송하게 된다. 명령을 수신한 탑재체는 동작을 수행하여 Payload 데이터를 OBC로 송신 한다.

송·수신 데이터는 CAN 버스를 통해 각 서브시스템 고유의 CAN ID에 따라 해당하는 서브시스템으로 전송되었고, 수신된 명령에 따른 각 서

브시스템의 동작을 통해 데이터 패킷이 정상적으로 송·수신 되는 것을 확인할 수 있었다.

CAN을 적용함으로써 각 서브시스템의 전장 보드는 단지 2개의 전선으로 연결되어 하니스(Harness)가 더욱 간결하게 되었고 신뢰성 또한 확보할 수 있었다. 그리고 탑재체의 호환성 시험을 통해 CAN을 기반으로 한 시스템에서의 확장성과 플러그 앤 플레이 구현의 용이성을 확인할 수 있었다. Fig. 4-4는 CAN 통신의 모니터링(Monitoring)의 결과를 보여준다. 랩뷰(Labview)와 시리얼 통신의 모니터링 결과를 비교 분석해 봄으로써 CAN 통신이 오류 없이 정상적으로 이루어졌음을 알 수 있었고, 명령에 따라 탑재체에 센서와 모듈들의 정상적인 동작을 확인할 수 있었다.

## V. CAN 적합성 분석

### 5.1 CAN 기반 CubeSat 설계

앞서 말한 바와 같이, CAN 버스를 적용한 모듈화 방식의 소형위성 개발은 위성개발의 신뢰성을 높여주고 빠른 개발을 가능하게 한다.

큐브셋(CubeSat)은 1999년 캘리포니아 폴리텍 주립대학(CalPoly State Univ.)과 스탠포드 대학교(Stanford Univ.)가 공동으로 개척한 큐브셋 프로그램을 시작으로 개발되었으며, 1kg의 한 변이 10cm인 정육면체로 위성 개발을 위한 기술 검증의 테스트 베드로 활용 되고 있다.

소형위성 개발에서의 CAN의 적합성과 이점을 알아보기 위해 큐브셋에 CAN을 적용하여 기존의 큐브셋과 비교하여 분석해 보았다.

#### Mechanical

모듈화 방식의 큐브셋 설계에서는 플러그 앤 플레이의 용이성을 위해 각 서브시스템 모듈을 슬롯(Slot)형태로 제작할 수 있다. 서브시스템 모듈 사이의 연결은 인터페이스 보드를 제작하는 방식과 전선 하니스로 연결 하는 방식이 있다. 인터페이스 보드를 사용하는 경우 복잡한 연결선을 제거 할 수 있지만 내부 공간의 손실이 발생한다. 하니스로 연결하는 경우 연결이 복잡할 뿐만 아니라 내구성이 약하기 때문에 끊어질 위험이 있다.

이 두 방식을 적절히 조합하여 Fig. 5-1과 같이 하니스를 제작할 수 있다. 위와 아래의 커넥터는 전선 리본(Wire-ribbon)으로 연결 되고 각 서브시스템의 전장보드는 가운데 위치한 커넥터에 꽂아서 연결 된다. 이를 통해 연결 보드나 선

Tab. 4-1. CAN Test Item

Item	Sub-Item	Description
CAN Interface Test	Payload A Packet Test	CAN interface, Packet 송수신
	Payload B Packet Test	CAN interface, Packet 송수신
	Payload C Packet Test	CAN interface, Packet 송수신
	CS Packet Test	CAN interface, Packet 송수신
	OBC Packet Test	CAN interface, Packet 송수신

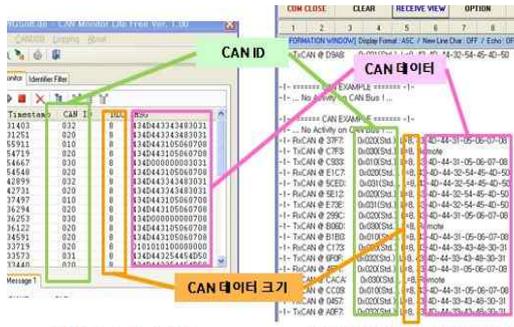


Fig. 4-4. CAN Test Monitoring

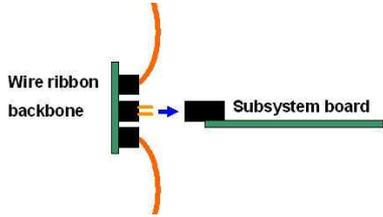


Fig. 5-1. Harness Design

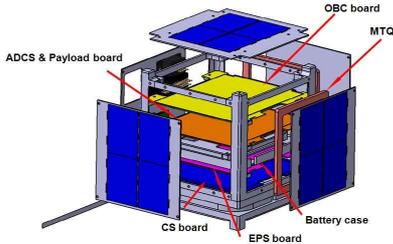


Fig. 5-2. CubeSat Conceptual Configuration

으로 인한 부피와 무게의 손실을 줄일 수 있고 시험이나 오류검출 뿐만 아니라 각 보드의 플러그 앤 플레이가 용이하게 된다. Fig. 5-2는 큐브셋의 구조물 설계와 각 서브시스템의 내부 배치를 나타낸다.

모듈화 위성은 동일한 규격을 가지는 전장보드의 조합으로 이루어지게 된다. 큐브셋의 내부 공간을 효율적으로 사용할 수 있도록 하며 서브시스템 모듈의 플러그 앤 플레이가 용이하도록 고안하였다. 이에 따라 각 서브시스템의 보드의 크기를 규격화 할 수 있다. 전장 보드의 배치 위치에 따라 보드 외형은 차이가 있으나 실제 회로가 설계, 제작되는 영역은 90mm × 80mm 정도로 기존의 큐브셋의 전장보드에 비해 보다 넓은 영역을 제공할 수 있다.

**Data Interface**

CAN은 2개의 전선을 사용하여 데이터 전송이 이루어지기 때문에 하니스의 무게를 줄일 수 있고 연결선을 단순화 할 수 있다. 따라서 각 서브시스템의 연결이 쉬워지고 위성 내부의 공간 활용성을 높일 수 있다.

CAN을 통해 높은 호환성과 확장성을 가질 수 있다. 즉, 다양한 버전(Version)의 서브시스템을 제작하여 시스템 요구에 따라 적절한 모델을 선택하여 연결하여 하나의 위성 시스템을 구축, 제작 할 수 있다. Fig. 5-3은 CAN을 데이터 버스로 하는 위성의 서브시스템 모듈 간 내부 인터페이스와 호환성을 보여준다.

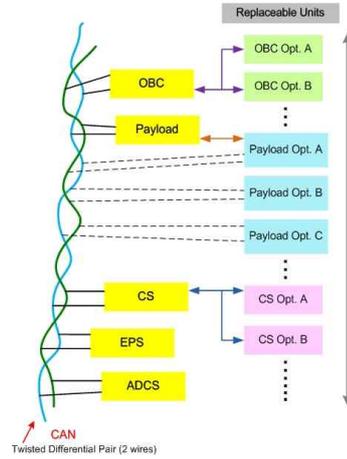


Fig. 5-3. CAN based Satellite Interface

**5.2 비교 분석**

대부분의 초소형위성은 센서와 서브시스템을 관리 및 제어하기 위해 탑재컴퓨터의 컨트롤러에서 여러 통신 방식을 사용하여 데이터 통신을 하게 된다. CAN을 적용하면 하나의 데이터 버스로 통합하여 간단한 시스템 구성을 가질 수 있다. 즉, 여러 종류의 데이터 버스를 하나의 CAN 표준으로 통합하게 되어 내부 인터페이스 설계가 간편해지고, 소프트웨어 개발이 편리해진다. 따라서 개발이 용이해지고, 각 서브시스템의 연결이 쉬워지고 단순해져서 내부 공간을 효율적으로 사용할 수 있다.

CAN은 분산제어 시스템을 제공하여 기능의 분산과 정보의 집중을 가능하게 한다. 탑재컴퓨터가 가지는 중앙 시스템에 집중된 부하를 각 서브시스템으로 분산하여 원활한 자료 처리 및 운영이 가능해지고, 탑재컴퓨터에 이상 발생 시 다른 서브시스템에 끼칠 과급효과를 최소화 하여 시스템의 신뢰성을 향상 시킬 수 있다.

Fault-tolerant 시스템은 일부 시스템에 고장이 발생하여도 다른 시스템에 영향을 주지 않도록 하기 위한 방안으로 시스템의 신뢰성을 향상시켜 준다. 대부분의 위성에서 오류 발생을 대비하여 잉여 시스템을 구축하게 된다. 하지만 잉여 시스템을 마련하게 되면 그에 따른 크기와 무게가 증가하게 되어 위성의 소형화에 어려움이 있다. 따라서 소형위성은 낮은 신뢰성을 갖게 된다.

CAN을 적용한 모듈화 구조는 Fault-tolerant에 보다 효과적인 구조를 가진다. 각 서브시스템은 독립적인 소단위의 시스템으로 분리되어 하나의 시스템에 오류가 발생하여도 다른 시스템에 영향을 미치지 않는다. 예를 들어, 탑재컴퓨터에 오류

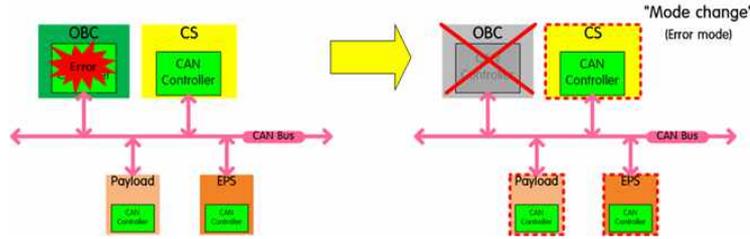


Fig. 5-4. CAN Fault-tolerant System

가 발생하는 경우 CS(통신계)에서 탑재컴퓨터를 대신하여 제어 통제권을 가지도록 할 수 있다. 다시 말해, 각 서브시스템의 소프트웨어를 제작할 때 에러 모드를 구분하여 설계하여, 탑재컴퓨터에 장애가 발생하면 각 서브시스템을 에러 모드로 전환하여 동작하도록 한다.

Fig. 5-4는 CAN을 적용한 위성에서의 Fault-tolerant 시스템 설계를 보여준다. CAN은 ID를 통한 데이터 통신이 이루어지기 때문에 이러한 Fault-tolerant 시스템 구현에 보다 용이하게 된다. 추가적인 장애 극복 소프트웨어가 필요 없기 때문에 탑재컴퓨터의 비행 소프트웨어의 간소화를 기대할 수 있고 효율적인 시스템 제어가 가능해진다. 따라서 소형위성의 신뢰성을 향상시킬 수 있게 된다.

## VI. 결 론

빠른 응답성을 가지는 안정적인 소형위성 개발을 위해 분산화 시스템 구조의 모듈화 개발이 필요하다. CAN을 기반으로 위성 데이터 버스 표준을 설계하여, 연결선을 간단하게 제작할 수 있었고 소프트웨어 개발도 쉽고 간단하게 할 수 있음을 알 수 있었다. 플러그 앤 플레이 기능시험과 모듈 간의 데이터 통신 검증시험을 통해 분산형 모듈화 소형위성에서의 CAN의 적용가능성을 확인 할 수 있었다.

CAN은 일반적인 위성 데이터와 저해상도의 위성 영상 전송에서 신뢰성 높은 통신을 보여주었다. 하지만 고해상도의 영상 데이터를 실시간으로 전송하기에는 어려움이 있다. 위성의 명령 및 데이터 통신 버스로써 CAN을 사용하고 고용량의 영상 데이터를 실시간으로 전송하기 위해 전송속도가 높은 영상 데이터 버스를 따로 구축하여, CAN 통신과 다른 통신방식을 혼용하는 방법을 사용하면 이러한 문제점을 개선할 수 있을 것이다.

하지만 소형위성은 빠른 응답성의 강점을 활

용하여 수개월 내에 제작하여 신속하게 우주 대응을 할 수 있도록 하는데 개발의 의의가 있다. 시스템의 변경과 수정의 용이성이 필요한 빠른 개발이 요구되는 임무에서 수개월 이내에 개발이 가능한 나노급(Nano-satellite; 1~10kg) 이하의 위성은 높은 활용성을 가진다. 따라서 CAN을 적용하면 개발의 간소화를 제공하여 응답성을 더욱 빠르게 하고, 신뢰성을 향상 시켜 주게 되므로 개발상 이점이 더 많음을 알 수 있다.

본 연구를 통해 구현된 CAN 데이터 인터페이스 소프트웨어는 차후 개발할 소형위성의 요구조건에 맞게 수정하여 적용할 수 있을 것이다. CAN의 데이터 전송속도를 향상시킬 수 있는 방안이 마련되면 고해상도의 영상 탑재체를 가지는 소형위성에서도 활용이 가능하여 더 넓은 범위의 소형위성 개발에서 CAN을 적용하여 신뢰성 있는 소형위성 개발이 가능해질 것으로 기대한다.

## 참고문헌

- 1) 이병훈, 장영근, “소형인공위성 시장의 현황과 전망”, *KSAS Spring Conference*, Apr. 2003.
- 2) Christopher J. McNutt, Robert Vick, Harry Whiting, and James Lyke, "Modular Nanosatellites - Plug-and-Play(PnP) CubeSat", *AIAA Journal*, Apr. 2009.
- 3) Jeffrey L. Janicik, "Implementing Standard Microsatellites for Responsive Space", *AIAA Journal*, Apr. 2003.
- 4) TM Davis, SD Straight, USV Directorate, "Development of the Tactical Satellite 3 for Responsive Space Missions", *AIAA 4th Responsive Space Conference*, April, 2006.
- 5) T Sorensen, G Prescott, M Villa, D Brown, J Hicks, "KUTESAT-2, A Student Nanosatellite Mission for Testing Rapod-Response Small

Satellite Technologies in Low Earth Orbit", *AIAA 3rd Responsive Space Conference*, April, 2005.

6) Jim Lyke, Don Fronterhouse, Scott Cannon, Denise Lanza and Wheaton Byers, "Space Plug-and-Play Avionics", *AIAA Journal*, Apr. 2005.

7) I.Galysh, K.Doherty, J.McGuire, H.Heidt, D.Niemi, G.Dutchover, "CubeSat: Developing a Standard Bus for Picosatellites", *Small Payloads in Space*, Vol. 4136, Nov. 2000, pp. 64~71.

8) Ctherine Erkorkmaz, Menachem Nimelman, Andrew Ogilvie, "Spacecraft Payload Modularization for Operationally Responsive Space", *AIAA 6th Responsive Space Conference*, April, 2008.

9) G.Maral, Michel Bousquet, "Satellite Communications Systems: Systems, Techniques and Technology, 3rd Edition", WILEY, 2000.

10) H.Helvajian, S.W.Janson, "Small Satellites: Past, Present, and Future", Aerospace Corporation, 2008.