

CAN을 이용한 Smart Sensor/Actuator Node 개발

이원무*, 정준홍*, 윤상진*, 조용석**, 박기현*
*성균관대학교, **건양대학교

Development of Smart Sensor/Actuator Node using CAN

Wonmoo Lee*, Joonhong Jung*, Sangjin Yoon*, Yongsuk Cho**, Kiheon Park*
*Sung-Kyun-Kwan Univ, **Kon-Yang Univ.

Abstract - 최근 분산제어시스템 설계시 CAN(Controller Area Network)을 적용하는 연구가 활발히 진행되고 있다. 본 논문에서는 분산제어에 필요한 각종 입출력 데이터를 CAN 버스 상에서 송수신할 수 있는 Smart Sensor/Actuator Node를 직접 개발하고, 이를 이용한 실험용 Network를 구축하였다. Sensor Node에서 획득된 외부의 Analog신호를 CAN Bus를 통해 전송한 후, 원격지의 Actuator Node에서 출력하는 Data 송수신 실험을 통해서 설계된 Network와 각 Node의 효율성을 입증하였다.

Intel, Philips, National, Motorola 등 유수의 반도체 회사에서 제어용 컨트롤러가 대량 생산되고 있어서, 그 시장성이 뛰어나다는 장점도 가진다[1],[2],[4].

2.2 CAN의 계층별 구조

CAN의 계층구조는 <그림 1>과 같이 OSI 7계층 모델 중 최하위 두 계층인 물리계층과 데이터링크 계층만으로 구성 되어 있으며, 상위계층에 적절한 응용프로그램을 작성하여 전체 네트워크를 구축한다.[1]

Application Layer	ISO15765
Data Link Layer	PCAN20200
Object Layer -Message Filtering -Message and status Handling Transfer Layer -Fault Confinement -Error Detection and Signaling -Message Validation -Acknowledgement -Arbitration -Message Framing -Transfer Rate and Timing	
Physical Layer	FCAN20200
	-Signal Level Bit Representation -Transmission Medium

그림 1. CAN의 계층구조

링크계층은 객체계층(Object Layer)과 전달계층(Transfer Layer)으로 구분되어 있다. 전달 계층은 CAN 버스상의 메시지를 받아 객체계층에 전달하는 역할을 하며 전송률, 비트타이밍, 메시지 캡슐화 및 중재, 에러 감지 등을 수행한다. 객체계층은 전달계층으로부터 받은 메시지를 확인자(Identifier)를 통해 필터링하고 핸들링 하는 역할을 한다.

2.3 CAN 프로토콜

2.3.1 메시지 종류

CAN 프로토콜은 데이터 프레임, 리모트 프레임, 오버로드 프레임, 그리고 에러 프레임으로 구성된다[1],[3],[5].

· **데이터 프레임(Data Frame)** : 각 노드(node)간의 데이터를 전송하는 역할을 하며 <그림 2>와 같이 7개의 세부 필드로 구성된다.

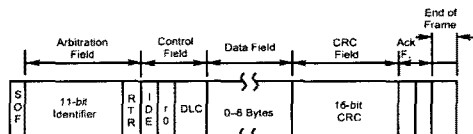


그림 2. 데이터 프레임 형식

1. 서 론

최근 활용범위가 증가하고 있는 CAN(Controller Area Network)은 초기 개발 당시 자동차 내부의 ECU(Electronic Control Unit)들을 서로 통합, 연결하여 제어할 수 있도록 하는 차량용 통신 프로토콜로 개발되었다. 이후, CAN은 데이터 전송상의 높은 안정성과 신뢰성을 바탕으로 차량용 네트워크뿐만 아니라, 로봇, 공정제어, 자동제어시스템 등 다수의 전자장치들을 상호 연결하여 제어하는 분산제어시스템(Distributed Control System) 설계시 널리 적용되고 있다. CAN을 이용한 분산제어방식은 저가의 하드웨어 장치로 효율성이 높은 네트워크 구성이 가능하며, 기존의 분산제어 방식에서 사용하던 일대일 직렬통신(RS-232) 방식에 비해 시스템 관리 및 유지보수를 용이하게 만드는 등의 장점을 가진다[1],[2].

본 연구에서는 CAN 프로토콜에 대한 전반적인 내용을 분석하고, 이를 일반적인 제어대상에 쉽게 적용할 수 있도록 설계 Smart Sensor/Actuator Node를 제작하였다. 또한, 실험용 CAN 네트워크를 구성한 후, 실시간으로 데이터를 송수신하는 실험을 통해서 제작된 Smart Sensor/Actuator Node가 실시간 제어가 요구되는 자동제어 시스템에 적용 가능함을 보였다.

2. 본 론

2.1 CAN(Controller Area Network)

CAN은 차량내부의 각종 전자장치간 통신에 사용하기 위해 1986년 독일의 Bosch GmbH사에서 개발하였으며, 이후 높은 데이터 신뢰성과 전기적 장애(noise 등)에 대해 쉽게 영향받지 않고, 통신상의 에러를 감지하고 교정하는 능력이 뛰어나며, 저가의 우수한 네트워크 구축이 가능하다는 장점 등으로 차량용 네트워크는 물론, 다양한 영역의 분산제어시스템 설계시 널리 적용되고 있다. 또한, 국제화 표준인 ISO 11898(high-speed application)과 ISO 11519(lower-speed application)로 공인 받은 표준 프로토콜로서

- **전송요청 프레임(Remote Frame)** : 한 노드에서 다른 노드에 데이터 전송을 요구하는 프레임으로서 그 형식은 <그림 3>과 같이 6개의 필드로 구성되어 있다. 프레임 형식은 데이터 프레임과 유사하지만 데이터필드가 없다는 것이 특징이다.

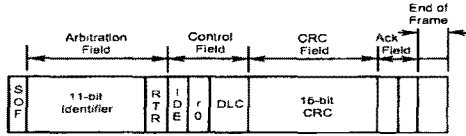


그림 3. 전송요청 프레임 형식

- **에러 프레임(Error Frame)** : CAN 버스 상에 각종 에러가 발생하였을 때 전체 노드에 오류가 발생했음을 알리는 역할을 한다. <그림 4>는 두 개의 필드로 구성되어 있는 에러 프레임의 형식을 보여준다.

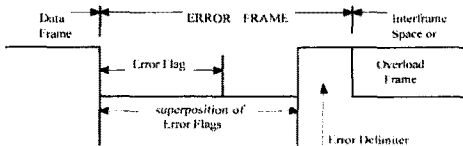


그림 4. 에러 프레임 형식

- **오버로드 프레임(Overload Frame)** : 과부하가 일어난 노드에서 다음에 오는 데이터 프레임이나 전송요청 프레임을 지연시키기 위해서 사용된다. <그림 5>는 오버로드 프레임의 형식을 보여주고 있다.

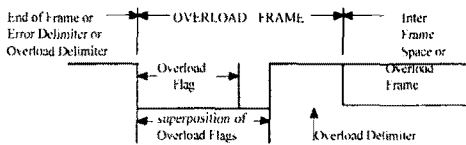


그림 5. 오버로드 프레임 형식

위와 같은 4가지 프레임 형식 이외에 각 프레임 형식을 구분하기 위한 인터프레임 스페이스(Interframe Space)가 있다. 즉, 데이터 프레임과 리모트 프레임의 사이에 3비트를 두어 각 프레임을 분리하는데 사용하며 단, 오버로드 프레임과 에러 프레임에는 인터프레임 스페이스가 존재하지 않는다.

2.3.2 주소지정 방식과 메시지 우선 순위

CAN은 메시지 필터링에 의한 주소 지정 방식을 사용한다. 데이터 프레임이나 전송요청 프레임의 중재필드에 11비트의 확인자(Identifier)를 갖고 있어 이 확인자 비트를 각 노드에서 필터링 하여 자신에게 온 메시지인지를 확인해서 받아들이고 그렇지 않으면 무시하게 된다. 또한, 확인자는 이러한 메시지 필터링 이외에 메시지 전송의 우선순위를 결정하는데도 사용되는데, 예를 들어 서로 다른 노드에서 동시에 메시지를 전송하려할 때 확인자의 비트에 따라 우선 순위가 높은 메시지가 CAN 버스를 우선적으로 점유하게 된다. CAN에서 메시지를 전송하는데 사용되는 이 같은 방식을 CSMA/AB (Carrier Sense Multiple Access with Arbitrary Bitwise)라 한다[1],[3].

2.3.3 에러 핸들링

CAN은 안정되고 신뢰성 있는 데이터를 전송하기 위해 전달계층에서 강력한 에러검출 기능을 제공한다. 이를 위하여 CAN 프로토콜에서는 아래와 같은 다섯 가지의 에러를 규정하고 이를 검출한다.

- **비트 에러(Bit Error)** : 메시지를 보내는 노드에서 하나의 비트를 보낼 때 CAN 버스 상에서 실제 나타나는 전압 레벨과 일치하지 않을 경우 발생하는 에러이다.
- **스터프 에러(Stuff Error)** : 전송되는 프레임에서 같은 레벨의 비트가 연속해서 6비트 이상 유지될 때 발생하는 에러이다.
- **CRC 에러(CRC Error)** : 프레임 내에 포함되어 있는 CRC 검사 값이 일치하지 않을 경우 발생하는 에러이다.
- **폼 에러(Form Error)** : CAN 프로토콜 상에 정의된 프레임 형식과 다른 메시지가 수신될 때 발생하는 에러이다.
- **ACK 에러(Acknowledgment Error)** : 메시지를 보낸 노드에서 ACK 신호(ACK Slot이 메시지를 받는 순간 trigger down)가 제대로 검출되지 않을 때 발생하는 에러이다.

모든 노드는 송신오류 카운터(Transmit Error Count)와 수신오류 카운터(Receive Error Count)를 갖고 있으며, 메시지 송수신시 오류가 발생할 경우 이 값을 증가시키고, 반대로 메시지 송수신이 정상적으로 종료되면 이 값을 감소시킨다. 그리고, 이 카운터 값에 따라 <그림 6>과 같이 각 노드에서의 에러상태(Error Active, Error Passive, Bus Off Station)가 정해진다[1],[3].

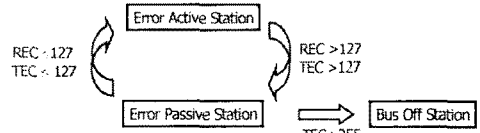


그림 6. 에러상태 변환규칙

2.4 Smart Node의 구성

본 연구에서는 CAN 버스에 직접 연결되어 실제 데이터를 송수신할 수 있는 말단시스템, 즉 제어응용에 적합한 Smart Sensor/Actuator Node를 설계하고 직접 제작하였다. 제작된 Smart Node의 블록선도는 <그림 7>과 같다.

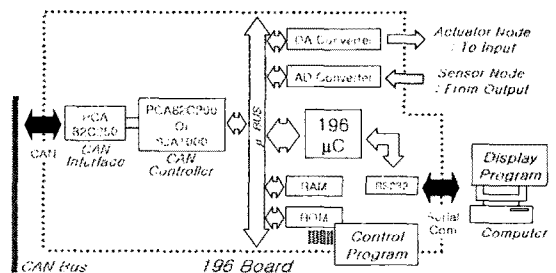


그림 7. Smart Node의 블록선도

제작된 Smart Sensor/Actuator Node의 특징은,

- Main Controller : Intel, 80C196KC20

- CAN Controller : Phillips, PCA82C200
- CAN Interface : Phillips, PCA82C250
- A/D Converter : Analog Device, AD1674(Sensor Node)
- D/A Converter : Analog Device, AD667(Actuator Node)
- Serial Comm. : RS-232(PC Serial Interface)

3. 실험

제작된 스마트 노드를 이용하여 CAN 네트워크를 구성하고, 임의의 아날로그 신호를 Smart Sensor Node에 입력한 후, CAN 버스를 통해 Smart Actuator Node로 출력되는 파형을 관찰하였다. 이를 위해, 응용계층에 해당하는 메인 컨트롤러의 롬(ROM)부분에 각각의 Node가 Sensor 및 Actuator로 동작하도록 프로그램 하였다. <그림 8>은 본 논문에서 실험에 사용한 전체 CAN 네트워크의 구성도이다.

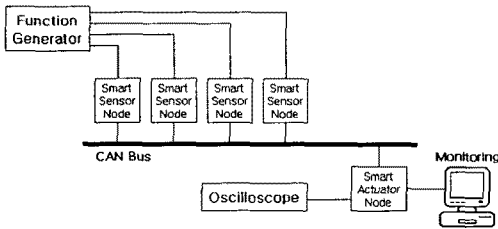


그림 8. 실험용 CAN 네트워크

먼저, 4개의 Sensor Node에 파형발생기를 통해 임의의 아날로그 신호를 입력하면, Sensor Node는 이 아날로그 신호를 A/D변환기를 통해 디지털 신호로 샘플링하고, 이 디지털 값을 CAN 데이터 프레임으로 캡슐화, 버스를 통해 전송하게 된다. 그리고, Actuator Node는 CAN 버스를 통해 수신한 데이터를 가지고 D/A 변환기를 이용하여 원래의 아날로그 신호를 복원, 출력한다.

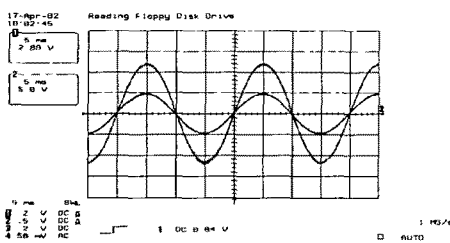


그림 9. 실험결과 (50Hz Sin파)

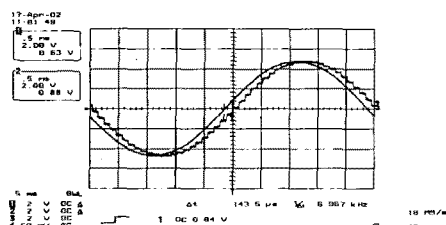


그림 10. 실험결과 (200Hz Sin파)

<그림 9>, <그림 10>은 하나의 Sensor Node에서 입력한 Sin파형을 Actuator Node에서 복원한 신호와 실시간 비교한

것이다. <그림 9>에서는 두 신호의 차이가 거의 없어 보이지만, 실제 <그림 10>과 같이 Sensor Node에서 입력되는 원 신호와 Actuator Node에서 출력되는 변환신호 사이에서는 지연시간이 발생하는데, 이는 A/D변환시간, D/A변환시간, 그리고 네트워크 상에서 프레임의 송수신 하는데 걸리는 시간 등이 원인이 되어 발생하는 것이다. 이러한 지연시간은 전송하는 프레임의 크기, Node 증가, 그리고 네트워크 버스의 경로 및 길이 등에 의해 달라질 수 있으며, 이는 네트워크 성능에 가장 많은 영향을 주는 요소중의 하나이므로, 실제 네트워크를 구성하는데 가장 크게 고려해야 한다.

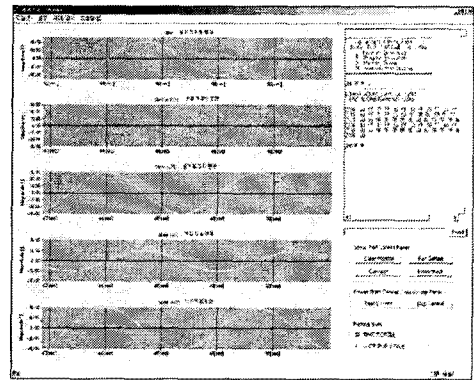


그림 11. PC를 통한 Smart Sensor/Actuator Node의 출력

<그림 11>은 4개의 Sensor Node에서 입력되는 파형과, Actuator Node에서 출력되는 파형을 직렬통신을 통해 PC로 전송, 그 값을 모니터링 한 것이다.

4. 결론

본 연구에서는 실제 제어시스템을 CAN 네트워크를 통해 구축하는데 필요한 Smart Sensor/Actuator Node를 직접 설계하고, 제작하였다. 이를 위해, CAN 프로토콜을 내장한 CAN 컨트롤러 칩과 마이크로 컨트롤러, 그리고 A/D, D/A변환기를 이용해 각 Node를 설계하였으며, 응용계층에 해당하는 프로그램을 통해 소프트웨어적으로 각 Node들의 데이터 전송속도, ID 등을 쉽게 설정할 수 있도록 설계하였다. 그리고, 실제 CAN 네트워크를 구성하여 실험을 한 결과, 제작된 Node들이 실시간 제어 응용에 적합한 성능을 가지고 있음을 확인하였다. 향후, 이러한 실험결과를 바탕으로 실제 네트워크를 통한 제어시스템 (Networked Control System) 설계시 개발한 Smart Sensor/Actuator Node를 적용하고자 한다.

[참 고 문 헌]

- [1] BOSCH. CAN Specification, Part A/B, 1991.
- [2] 이진영, "CAN 통신을 이용한 다중모터 위치제어기 구현", 전기학회논문지 D, 제 51권, 제 2호, pp55~60, 2. 2002.
- [3] SJA 1000 Stand-alone CAN Controller - Product Specification, Phillips Semiconductors, 4. 2000.
- [4] "CAN high-speed physical layer", CAN in Automation (CiA), <http://www.can-cia.de>, 2002.
- [5] 윤상진 외, "CAN(Controller Area Network)을 이용한 멀티플렉싱 기술개발", 대한 전기학회 하계학술대회 논문집 pp2206~2208, 7. 2001.