

CAN 기반 제어 시스템 분석을 위한 인터페이스 유닛 설계 및 구현

Design and Implementation of an Interface Unit for Analysis of a CAN-Based Control System

박 병 료*, 정 구 민**, 안 현 식***, 김 도 현***

Byung-Ryuel Park, Gu-Min Jeong, Hyun-Sik Ahn, and Do-Hyun Kim

Abstract - In this paper, an interface unit is designed to efficiently monitor transmission data in Controller Area Network(CAN)-based control systems. The CAN uses a serial multi-master communication protocol that efficiently supports distributed real-time control with a very high level of data integrity, and communication speeds of up to 1Mbps. The interface unit is composed of a DSP controller which collects data on the CAN bus and transfers data to a personal computer via serial communication to save and display of interesting signals. The experimental system consists of three DSP controllers which represent electronic control units of a vehicle, an interface unit for analysing the data on the bus, and a graphic monitoring program coded on the Windows platform. The validity and the effectiveness of the proposed simple type of CAN interface unit are shown through the experimental results.

Key Words : Controller Area Network(CAN), monitoring system, distributed network, serial communication

1. 서론

최근 네트워크 기술의 발달 및 보급에 따라 자연스럽게 네트워크를 이용한 시스템 제어에 대한 관심이 증가하게 되었다. 네트워크를 이용한 제어 시스템에서는 효율적인 자원의 공유 및 정보교류가 가능해짐에 따라 설비 및 점유 공간 감소에 따른 경제성, 설치 및 운용의 편의성 등 여러 가지 장점을 확보할 수 있다. 따라서 공정 시스템, 발전 시스템, 자동차 및 항공기 산업분야 뿐만 아니라 군사용 장비분야에서도 네트워크를 활용하여 경제성 및 제어성능을 제고하고 있다 [1].

해외 고급 자동차부터 적용되기 시작하여 국산 자동차에도 적용되고 있는 대표적인 자동차 내 통신 프로토콜인 CAN(Controller Area Network)은 적용 범위를 빠르게 넓혀가고 있다. '보쉬'사에 의해 규격이 정의된 CAN은 최근 차량내 전장 시스템의 복잡도와 활용도가 증가하고 정보 공유의 필요성이 증가하면서 제안된 시리얼 통신 프로토콜로서 이제 자동차 업계 표준으로 자리 잡고 있다 [2][3]. CAN 프로토콜은 차량에서의 실시간 요구를 만족시키며, 트리 구조와는 다르게 전자기적 간섭에 의해 발생하는 전송에러를 발견하여 수정할 수 있다. 또한, 전체 시스템 구성의 용이성과 중앙 집단이 가능하다는 장점이 있다. 그러나 CAN 기반 제어 시스템은 기존의 point-to-point 방식과 다른 네트워킹 방식이기 때문에 전압/전류계, 오실로스코프등의 기존의 테스트 장비로는 시스템의 동작 및 성능 분석이 어려운 문제점도 갖고 있다 [4].

본 논문에서는 CAN 시스템을 도입하거나 유지, 보수하기 위하여 필요한 간이 인터페이스 장치를 제안한다. 우선 CAN 프로토콜에 대하여 개략적으로 알아본 후 인터페이스 유닛의 하드웨어 및 관련 소프트웨어를 설계하여, 자동차를 중심으로 한 다양한 분야에 적용할 수 있는 CAN 시스템용 인터페이스 유닛을 구현한다. 또한, 연료전지 차량의 파워트레인부에 대한 간이 시뮬레이터에 직접 적용하여 설계된 인터페이스 유닛의 동작 성능을 검토한다.

2. Controller Area Network(CAN)

CAN은 실시간 네트워크 기반 제어에 효과적으로 적용될 수 있는 직렬 네트워크 통신 프로토콜이다. 1986년 차량 내에 전자제어장치(Electronic Control Unit, ECU)간의 통신을 위해 차량 부품 업체인 보쉬에 의해 개발되었다. 최근에는 차량뿐만 아니라 산업 전 분야에 폭넓게 적용되고 있다 [2]. CAN 프로토콜은 다미스간의 데이터 교환 방식을 ISO의 OSI 7계층 참조 모델(7 Layer Reference Model)에 의거하여 7개의 계층 중에는 하위 두 계층인 물리적 계층과 데이터 링크 계층에 걸쳐 정의하였다.

CAN 프로토콜은 데이터 프레임(Data Frame), 리모트 프레임(Remote Frame), 에러 프레임(Error Frame), 오버로드 프레임(Overload Frame)이라는 4개의 프로토콜 프레임 형태를 제공한다.

데이터 프레임의 경우 송신단에서 수신단으로 데이터를 보내는데 사용되고 SOF(Start of Frame) Field, Arbitration Field, Control Field, Data Field, CRC(Cycle Redundancy

저자 소개

- * 國民大學 電子工學科 碩士課程
- ** 國民大學 電子工學科 助敎授・交通著者
- *** 國民大學 電子工學科 敎授・工博

Code) Field, Acknowledge Field, EOF(End of Frame) Field로 구성된다. 리모트 프레임은 같은 식별자를 갖는 데이터 프레임의 송신을 요청하는데 사용되며 데이터 프레임에서 Data Field가 제외된 것이다. 에러 프레임은 버스상의 에러와 관련된 기능을 처리하기 위해 사용되며 에러 플래그와 에러 구분자로 구성된다. 오버로드 프레임은 이전 프레임과 다음 프레임 사이의 추가지연을 제공하기 위해 사용되며 오버로드 플래그와 오버로드 구분자로 구성된다.

CAN 프로토콜은 표준 CAN(버전 2.0A)과 확장 CAN(버전 2.0B)의 두 가지 형태의 메시지 프레임들을 지원하며 그림 1은 확장 CAN 메시지 프레임들을 나타낸다.

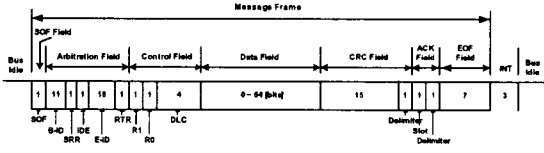


그림 1. 확장 CAN 메시지 프레임.

3. CAN 인터페이스 유닛의 설계 및 구현

3.1 설계 요구사항

그림 2에서와 같이 3개 또는 그 이상의 ECU들은 각각의 노드를 형성하여 하나의 CAN 버스를 공유하면서 서로 통신하고 있다. 이렇게 CAN 기반 네트워킹을 도입하여 시스템을 설계할 경우 시스템의 네트워킹에서 오는 여러 가지 장점(배선 절감, 멀티마스터 방식의 분산제어 시스템 구현 등)을 기대할 수 있다. 그러나 반대로 복잡한 CAN 기반 네트워킹의 프로토콜을 다루기 위해서는 설계과정에서 네트워킹의 동작을 확실히하고 성능을 측정할 수 있는 시스템이 필요하다.

CAN 프로토콜이 자동차의 각종 전기장치 및 제어장비의 통신을 위하여 개발되었기 때문에 본 논문에서는 연료 전자 자동차에 대한 시뮬레이터 시스템을 대상으로 인터페이스 유닛을 설계하고, 이를 CAN 기반 네트워킹에 적용하여 시스템의 성능 분석 및 개발후의 유지보수에 활용한다.

CAN 인터페이스 유닛의 주요 기능은 그림 2에서와 같이 각각의 노드들이 풀러있는 CAN 버스를 공유해서 제어 시스템에서 이루어지는 데이터의 흐름 및 통신 상태를 모니터링 하는 것이다. 이러한 통신과정에서 송·수신되는 데이터들과 각 노드들의 식별자 정보, 통신이 완료되는 시간에 관한 정보들이 주요 모니터링 대상이 되며 이를 CAN 인터페이스 유닛에 적용한다.

3.2 CAN 인터페이스 유닛의 구성

본 논문에서 제안하는 시스템은 CAN을 지원하는 하나의 컨트롤러를 이용하여 CAN 버스를 모니터링 하는 형태로 구성된다. CAN 인터페이스 유닛을 담당하는 컨트롤러는 이미 많은 사이트에서 신뢰성이 검증되고 CAN 기반의 리얼 타임 시스템의 구현과 다양한 주변장치들이 지원되는 Texas Instrument사의 제어용 칩인 TMS320F2812를 사용하였다.

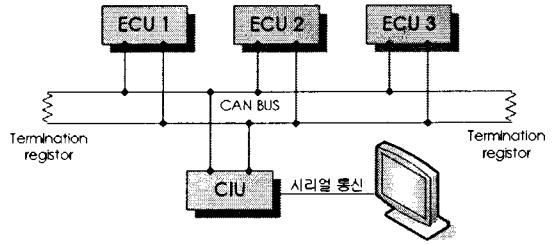


그림 2. CIU를 포함하는 CAN 기반 네트워크 구성도.

TMS320F2812에는 CAN 통신을 위한 트랜시버가 내장되어 있지 않으므로 CAN 프로토콜에 정의된 HI/LOW 신호의 생성을 위해서 CAN 트랜시버가 필요하다. 각각의 ECU들과 CAN 인터페이스 유닛에는 Texas Instrument사의 SN65HVD230 트랜시버를 사용하였으며 실제 통신 속도는 1Mbps이다. 또한 각각의 CAN 노드에 대해서 총괄적으로 데이터를 수집하고 분석해야 하기 때문에 빠른 연산처리 속도를 필요로 한다. 이를 위해 좀 더 고급 인터페이스가 요구되기에 모니터링에 대한 결과를 RS232 통신을 통하여 PC상의 윈도우 플랫폼에 디스플레이 하였다. 이러한 PC와의 통신을 위해 MAX3232가 필요하며 실제 통신 속도는 115200bps 이다.

통신을 통하여 전달된 데이터는 그 종류에 따라 구분되어져 각각의 형식에 맞추어 디스플레이 할 수 있는 소프트웨어를 필요로 한다. 모니터링 프로그램 소프트웨어는 Microsoft사의 Visual Studio .NET 2003 환경에서 Win32 API와 MFC를 이용하여 개발하였다. 그림 3은 이러한 CAN 인터페이스 유닛의 구성도를 보여주고 있다.

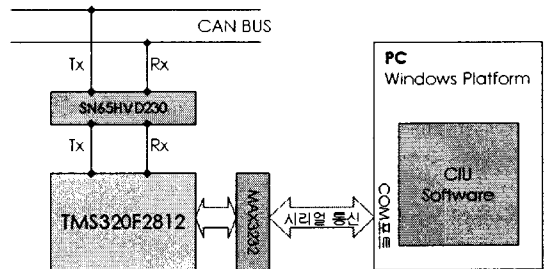


그림 3. CAN 인터페이스 유닛 구성도.

3.3 CAN 인터페이스 유닛의 설계 및 구현

CAN 인터페이스 유닛은 각각의 ECU들이 공유하고 있는 CAN 버스에 연결되어서 다른 ECU들이 송·수신하고 있는 미리 정의된 형태의 데이터들을 모니터링 한다. 이러한 기능을 수행하기 위해서는 먼저 모니터링 대상에 해당되는 CAN 노드들이 할당받은 ID를 PC상의 모니터링 프로그램에서 추가해 주어야 한다. 전달된 ID는 시리얼 통신을 통하여 CAN 인터페이스 유닛으로 전달된다. CAN 인터페이스 유닛은 전달 받은 ID를 포함하는 메시지 프레임 데이터들을 다시 PC상의 모니터링 프로그램으로 전달한다. 더불어 해당 노드의

송·수신 완료시점에 대한 시간 정보, 데이터를 수신하게 되는 주기 등의 부가적인 정보를 함께 전달하여 CAN 기반 제어 시스템의 분석을 용이하게 한다. 이러한 정보들은 시리얼 통신을 통하여 그림 4와 같이 미리 정의된 패킷 형태로 전송하게 된다.



그림 4. 시리얼 통신 패킷 구조.

패킷에는 시리얼 통신 도중에 발생하는 데이터들의 손실로 인해 잘못된 정보를 보여주는 가능성을 배제하기 위하여 프레임 체크 코드(Frame check code)를 포함한다. 식 (1)은 프레임 체크 코드에 대한 계산식을 나타낸다. 모든 데이터들을 1바이트 단위로 XOR(Exclusive OR, ⊕)하여 받은 데이터가 올바른지 판별하는 수단으로 사용될 수 있게 하였다.

PC상의 모니터링 프로그램은 COM 포트가 수신한 패킷을 저장하고 분석하여 데이터 종류에 따라 각각 디스플레이하여 CAN 네트워크 시스템에 대한 해석을 가능하게 해준다. 이러한 인터페이스 유닛은 네트워크에 의한 제어시스템에서 야기되기 쉬운 시스템의 오동작 및 설계 과정에서 발생하는 하드웨어 및 소프트웨어상의 오류를 감지하는데 좋은 수단이 될 수 있다.

Frame check code =

$$\text{데이터 길이} \oplus \text{데이터 종류} \oplus \text{데이터 값} \quad (1)$$

3.4 CAN 인터페이스 유닛을 이용한 모니터링

앞서 기술한 바와 같이 CAN 인터페이스 유닛은 연료 전지 자동차에 대한 시뮬레이터 시스템을 이용하여 구현 하였다. 모니터링 프로그램은 CAN 버스를 통해 전달되는 중요 데이터들을 디스플레이하여 시뮬레이터 시스템의 성능 분석이 용이하게 한다. 또한 분산 시스템에서의 제어흐름에 대한 정보를 쉽게 파악할 수 있게 해준다. 그림 5는 CAN 인터페이스 유닛 모니터링 프로그램이다. 사용자는 화면에 나타나는 관심 노드들의 데이터들과 부가 정보들을 이용하여 CAN 기반 네트워크 시스템을 분석할 수 있다.

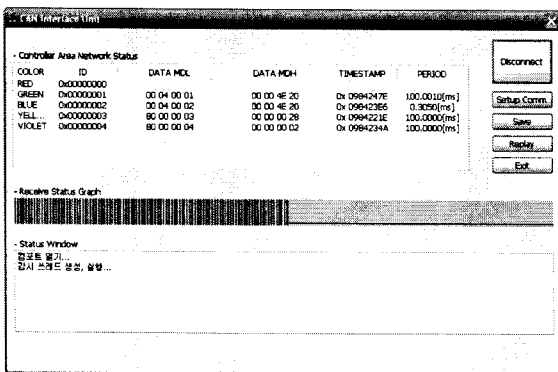


그림 5. CAN 인터페이스 유닛 모니터 프로그램.

각각의 ECU들은 CAN 버스를 이용하여 통신을 한다. 이에 반해 PC와의 통신은 시리얼을 이용하기 때문에 이들 사이에 통신 속도의 차이가 존재하게 된다. 만약 CAN 버스 상에서 데이터를 샘플링 하는 주기가 너무 짧아지면 PC의 COM 포트단에서 버퍼링이 심해져 화면에 데이터를 업데이트를 하지 못하는 현상이 발생한다. 이러한 현상은 CAN 통신 속도보다 낮은 속도의 시리얼 통신을 사용한 점, PC에서 COM 포트에 받은 데이터를 분석하고 화면에 뿌려주는 부하가 존재하기 때문에 발생하게 된다. 성능 개선을 위해서는 USB방식을 이용하는 등의 PC와의 통신 속도 향상, 그리고 패킷에 대한 정의와 데이터 과심에 대한 더 좋은 알고리즘 연구가 필요하다.

4. 결론

본 논문에서는 CAN 기반 제어시스템을 분석하기 위한 인터페이스 유닛을 설계하고 구현하였다. 우선, 연료전지 자동차 내 파워트레인부 서브시스템들의 역할을 정의하고 각 DSP 제어기에 분산 할당시킨 후 CAN으로 연결하여 간이 연료전지 자동차 시뮬레이터 시스템을 구현하였다. 또한, 제안된 인터페이스 유닛을 동일 버스에 연결하여 현재 전송중인 내용 즉, 목적지 ID, 데이터 내용, 수행주기 등을 확인할 수 있었고, 개인용 컴퓨터로 시리얼 전송된 관심 변수들을 원하는 형태로 그래픽 모니터링할 수 있었다.

세계적으로 자동차 및 제어 시스템에 네트워크 기술이 지속적으로 도입되고 있으며, 제어시스템을 통합하면서 분산처리를 실현하는 효과를 거두고 있다. 향후 제어용 네트워크 분야에서는 CAN의 이용 범위가 더욱 넓어질 것이고 관련 응용 시스템의 개발 및 유지 보수를 위한 측정 장비들의 수요도 많아질 것이다. 본 논문에서 제안한 시스템은 이러한 요구에 잘 부합할 수 있을 것으로 기대된다. 현재까지 전 세계적으로 형성된 CAN 응용제품 시장에 대응하기 위해서는 적합한 개발환경, 운영체제, 프로토콜, 전용 언어 등 고도의 기술력을 요구하는 기반환경 구축 분야에 더 많은 연구 활동이 있어야 할 것이다.

참 고 문 헌

- [1] R. S. Raji, "Smart network for control," *IEEE Spectrum*, vol. June, pp. 49-55, 1989.
- [2] *CAN Specification version 2.0*, Robert Bosch GmbH, 1991.
- [3] K. Etschberger, "Controller Area Network : An Introduction", TechOnline, http://www.techonline.com/community/related_content/22590, Apr. 7, 2003.
- [4] 권대현, 임준홍, "CAN 시스템을 위한 트래픽 분석/생성 시스템 설계 및 구현," *공학기술논문집 Vol.9 No.1* pp. 93-99, 2000. 8.
- [5] *TMS320C28x Serial Communication Interface(SCI) Reference Guide*, Texas Instrument, 2004.