

인터넷과 CAN을 이용한 원격 분산 Embedded System 설계

Design of a Remote Distributed Embedded System Using the Internet and CAN

이 현 석, 임 재 남, 박 진 우, 이 장 명
(Hyun-suk Lee, Jae-nam Lim, Jin-woo Park, Jang-Myung Lee)

Abstract : A small size and light-weight DSP board is newly designed for a real time multi-distributed control system that overcomes constraints on time and space. There are a variety of protocols for a real-time distributed control system. In this research, we selected CAN for the multi distributed control, which was developed by the BOSCH in the early 1980's. If CAN and Internet are connected together, the system attains the characteristics of a distributed control system and a remote control system simultaneously. To build such a system, The TCP/IP-CAN Gateway which converts a CAN protocol to TCP/IP protocol and vice versa, was designed. Moreover, the system is required to be small and light-weighted for the high mobility and cost effectiveness. The equipment in remote place has a TCP/IP-CAN Gateway on itself to be able to communicate with another systems. The received commands in the remote site are converted from TCP/IP protocol to CAN protocol by the TCP/IP-CAN Gateway in real time. A simulation system consists of a TCP/IP-CAN Gateway in remote place and a command PC to be connected to Ethernet.

Keywords : internet, TCP/IP, CAN, gateway, network

I. 서론

산업이 고도로 발전하면서 원격제어에 관한 연구는 많은 성과를 거두고 있다. 빠르게 발전하고 있는 인터넷 기술과 인터넷의 급속한 보급이 지리적, 공간적 제약을 극복하게 만들었다[1][2]. 하지만 인터넷만을 이용한 원격제어 시스템에서는 다중의 장비를 효율적으로 제어하기에 어려움이 있으며 다수의 장비를 제어하기 위해서 분산제어 네트워크만을 사용한다면 장거리 원격제어에 어려움이 있다.

이에 본 연구에서는 인터넷의 원격제어 장점과 분산 네트워크의 다중 실시간 제어 장점을 함께 구동할 수 있을 뿐만 아니라, 소형·경량으로 제작하여 컴퓨터가 아닌 프로세서 레벨에서 인터넷과 분산 네트워크를 변환하는 시스템을 개발하였다[3][4][5].

본 연구에서 사용한 분산 네트워크는 1980년대 초반에 BOSCH사에서 개발한 CAN을 이용하였으며[6], CAN 내장형 칩인 TI사의 TMS320F243을 이용하여 TCP/IP, ICMP, ARP 및 UDP 프로토콜을 구현하여 인터넷 컨트롤러(RTL8019)와 인터페이스 할 수 있도록 하였다. 또한 RTOS로서 소용량이며 실시간 처리가 가능한 $\mu C/OS$ 를 채택하여 구현하였다[7].

구현된 TCP/IP-CAN Gateway의 성능을 확인하기 위해 본 연구에서는 2개의 BLDC모터의 구동을 이용하여 실험하였다. 로컬 컴퓨터에서 속도 지령치를 주었을 때 원격지에서 각 CAN ID를 할당 받은 2개의 BLDC 서보모터가 서로 다른 속도 지령치에 도달하는 것을 보이고, TCP/IP-CAN Gateway

의 송신 여부를 파악하기 위해 모터에서의 속도 데이터를 TCP/IP-CAN Gateway가 로컬 컴퓨터로 데이터를 전송하여 확인하였다.

본 연구에서 서로 다른 프로토콜간의 변환을 보이는 TCP/IP-CAN Gateway의 설계 과정을 보이기 위해 2장에서는 $\mu C/OS$ 와 CAN 프로토콜 및 TCP/IP 프로토콜의 전반적인 사항을 다루고 3장에서 시스템의 구성을 다루며, 4장에서 실험결과에 관해 논의하고 5장에서 결론을 맺도록 한다.

II. 소개

1. $\mu C/OS$

$\mu C/OS$ 는 1992년에 Jean J. Labrosse의해 개발된 RTOS이다. $\mu C/OS$ 는 선점형 기반의 OS로서 총 64개의 priority가 있다. 하지만, 상위 4개와 하위 4개는 $\mu C/OS$ 가 Virtual Task를 사용하여 OS를 구동하고 있기 때문에 사용자가 사용가능한 priority는 56개이다. 그래서 사용자가 사용 가능한 56개의 priority를 이용하여 Task로 사용한다. 각 Task는 priority를 가지고 있어서 먼저 semaphore를 차지했는지라도 priority가 높은 Task가 priority가 낮은 Task 보다 먼저 처리된다. Task에 priority를 부여함과 동시에 스케줄링이 시작되어 Task를 처리한다.

2. CAN 프로토콜

CAN은 원래 차량 내에서 콘트롤러, 센서, 액츄에이터, ABS, 오디오 시스템, 엔진제어 시스템 등을 연결하기 위해 BOSCH사에서 개발한 시리얼 통신 프로토콜이다.

CAN은 거리에 따라 다르지만 최고 1Mbps의 높은 데이터 전송률과 충돌한 메시지의 자동 재전송과 15bit Cycle Redundancy Check(CRC)를 이용한 오류 검출등과 같은 안정

접수일자 : 2001. 10. 22., 수정완료 : 2002. 2. 16.

이현석, 임재남, 박진우, 이장명 : 부산대학교 전자공학과(eleclhs@koreaaero.com/ jaenam74@hanmail.net/ jwpark@iita.re.kr/jmlee@pusan.ac.kr)

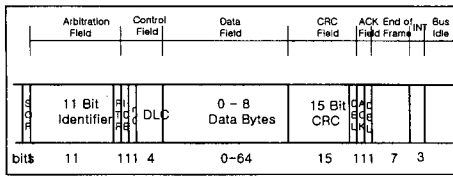


그림 1. CAN 표준 프레임 형식.
Fig. 1. CAN Standard Frame Format.

성을 제공하고 있어서 다수의 전자 제어장치를 상호 연결하여 실시간 제어를 효율적으로 지원할 수 있는 분산제어 네트워크이다.

CAN의 주요 특징은 모든 스테이션이 네트워크에 대한 동등한 제어권을 갖는 분산접속통제(distributed access control)와 Content-based addressing으로 인한 시스템 구성의 유연성, 메시지의 중요도에 따른 우선 순위 할당, 비과파적 버스 액세스 등이 있다.

3. TCP/IP

TCP(Transmission Control Protocol)과 IP(Internet Protocol)는 각각 OSI 모델의 계층 4과 계층 3에 대응되어진다. TCP/IP는 ARPA 프로젝트를 통하여 개발되어졌고 미국방성의 표준으로 정착되었다.

IP는 스테이션간에 데이터그램 서비스를 제공하기 위하여 설계된 네트워크 계층의 프로토콜이며 TCP는 사용자들의 사이의 Connection-oriented 서비스를 제공하며 Connection 설정, 유지와 해제를 전담하는 프로토콜이다.

III. 시스템 구성

1. 전체 시스템 구성

시스템의 구성은 TCP/IP-CAN Gateway을 기준으로 로컬 컴퓨터에서 인터넷을 이용하여 접속되어 있으며, 다수의 장비와 TCP/IP-CAN Gateway간에는 CAN네트워크로 연결되어 있다. 로컬 컴퓨터에는 사용자 위주의 응용프로그램을 이용하여 사용자가 TCP/IP-CAN Gateway에 접속하여 다수의 장비를 제어 가능하도록 하였다. 실제 CAN에서는 1000여개의 ID를 할당하여 분산네트워크를 구성할 수 있다. 그러나 본 연구에서는 TCP/IP-CAN Gateway에 2개의 BLDC모터를 장착하여 로컬 컴퓨터에서 제어가 가능하고, 데이터를 전송 받을 수 있도록 하고 있다.

그림 2는 전체 시스템의 구성을 간략하게 보여 주고 있다. 앞에서 설명한대로 TCP/IP-CAN Gateway을 중심으로 다수의 장비와 로컬 컴퓨터를 연결하여 주고 있다. 본 시스템은 실험에서 실제의 시스템 특성을 분석하도록 한다. 한번에 실행되는 버퍼링의 양은 데이터 메모리의 크기가 64Kbyte로

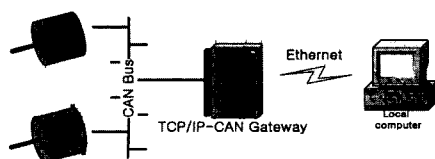


그림 2. 전체 시스템 구성도.
Fig. 2. The total system formation.

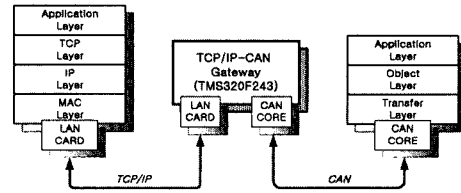


그림 3. TCP/IP와 CAN의 데이터 전송 경로.
Fig. 3. A data transfer path of TCP/IP and CAN.

제한되어 있고, 한번에 보내는 패킷은 이더넷을 사용하고 있으므로 1500 byte의 크기로 제한되어 있다. 또한 가변적인 각 계층의 헤더를 임시 저장해야 하고, CAN으로부터의 데이터 또한 저장해야 한다. 이런 것을 고려하였을 때, 버퍼링 사이즈를 32Kbyte로 제한하여 사용하면, 충분한 버퍼링 공간이 되어 패킷을 처리하는데 문제가 발생하지 않는다.

그림 3은 인터넷과 CAN의 프로토콜 계층구조의 변환을 나타낸 것이다. 서로 다른 네트워크 계층구조를 가지고 있기 때문에 TCP/IP-CAN Gateway는 TCP/IP와 CAN을 서로 다른 물리층으로 전송하기 위해 프로토콜을 변환한다[8].

2. 소프트웨어의 구성

발생하는 이벤트에 따라 데이터의 흐름에 맞게 구성하여, μ C/OS에서 할당된 우선 순위에 따라 발생된 이벤트들을 처리한다. μ C/OS의 최우선 순위는 이더넷 상에서 수신되는 데이터를 처리하는 것으로 하고 있다. 또한 CAN과의 신뢰성있는 데이터 전송을 위해서 TCP 프로토콜을 이용하였다 [9][10].

TCP/IP-CAN Gateway은 그림 4에서와 같이 CAN에서 수신한 데이터, 즉 CAN의 송신 요구를 받아서 이더넷으로 패킷을 전송하도록 구성되어 있다.

그림 5는 TCP/IP-CAN Gateway에서 접속 요청을 받아서 커넥션 되어지는 과정과 데이터를 수신하여 처리하는 과정을 나타내는 순서도이다.

그림 5에서 보는 것과 같이 패킷을 처리하는 도중에 다시 패킷이 수신되면, 패킷을 임시 저장하고 있다.

IV. 실험

TCP/IP에서 TCP/IP-CAN Gateway로 수신되는 데이터는 각 CAN에 대한 정보로서 2바이트의 헤더와 6바이트의 데이터로 구성되어 전송한다. 여기서 헤더는 CAN ID의 정보를

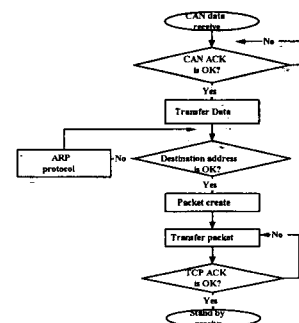


그림 4. CAN 에서 TCP/IP으로의 변환.
Fig. 4. The transformation from CAN to TCP/IP.

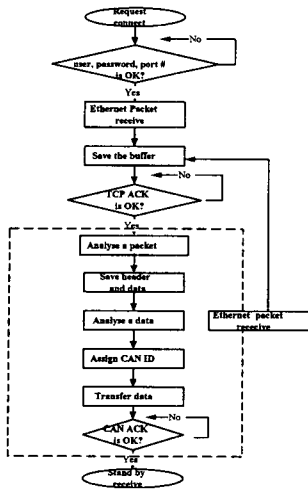


그림 5. TCP/IP에서 CAN으로의 변환.
Fig. 5. The transformation from TCP/IP to CAN.

가진 부분이고, 6바이트의 데이터는 각 명령 지령치를 가지고 있다. 반대로 TCP/IP-CAN Gateway에서 TCP/IP로 전송시에도 동일한 형태로 변환하여 전송한다.

그림 6은 BLDC 모터와 모터 드라이브, 컨트롤러로 구성되어지는 BLDC 서보 모터 모듈로서 CAN을 이용한 제어가 가능하다. 본 실험에서는 그림 10과 같은 BLDC 서보모터 모듈 2개와 본 논문에서 제안한

TCP/IP-CAN Gateway 모듈 1개를 이용하여 시스템을 구성하였다.

그림 7과 그림 8은 로컬 컴퓨터에서 리모트측의 모터 1과 모터 2로 TCP/IP Gateway를 통해 각각 500[rpm]과 1000[rpm]의 지령치를 전달하고 각 모터의 속도를 되돌려 받은 것과 TCP/IP Gateway를 통하지 않고 바로 CAN을 통해 모터를 제어한 것을 비교하고 있는 그림이다. 그림에서 알 수 있듯 시간 지연이 약 25ms가 일어나고 있다. 동일 이더넷 상에서 25ms의 시간 지연은 많은 것이다. 실제 실험에서 ICMP를 이용하여 이더넷 지연을 측정한 결과 약 5ms의 지연이 있었다. 이것과 비교하면 5배가 넘는 결과이다. CAN으로부터 받은 데이터를 TCP/IP-CAN Gateway에서 처리하여 다시 이더넷을 통하여 전송하고, 로컬 컴퓨터에서 데이터를 받아서 처리하는 시간이 모두 합산되었기 때문에 일반 컴퓨터와 컴퓨터

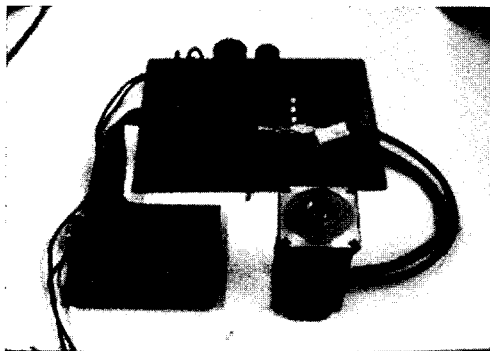


그림 6. 개별 BLDC 서보모터 모듈.
Fig. 6. A particular BLDC servo motor module.

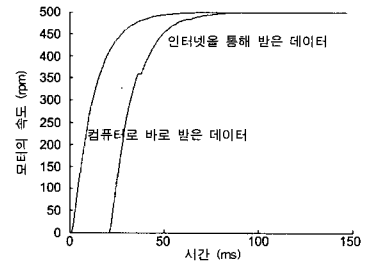


그림 7. 모터 1의 상태 곡선.
Fig. 7. A Status curve of motor 1.

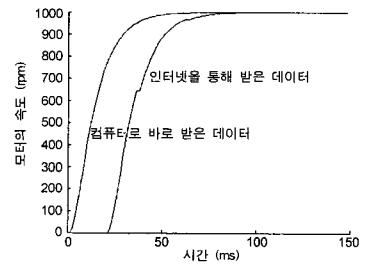


그림 8. 모터 2의 상태 곡선.
Fig. 8. A Status curve of motor 2.

간의 지연시간과 비슷한 결과이다. 다시 말해서, CAN의 전송지연과 TCP/IP-CAN Gateway 에서 처리시간, 이더넷의 전송지연, 로컬 컴퓨터에서 처리시간이 모두 합쳐져서 생긴 지연시간이다. 또한 그림 7과 8에서 인터넷을 통해 받은 데이터에서 보면 찌그러짐 현상을 볼 수 있다. 찌그러짐 현상을 알기 위한 그래프를 분석하면, 동일 시간대에서 찌그러짐이 일어남을 알 수 있다. 이것은 패킷의 재전송에 의해 데이터가 수신되지 못하여 생기는 현상이다. 실험에서는 시간대에 따라 차이는 있지만, 약 17% 정도 재전송이 일어남을 알 수 있다.

V. 결론

본 논문에서는 다수의 장비를 원격지에서 분산제어가 가능하도록 하기 위해 CAN프로토콜과 TCP/IP프로토콜의 변환을 지원하는 소형·경량의 TCP/IP-CAN Gateway 을 설계하였다. 기존에 CAN네트워크가 구현되어 있는 시스템에 원격제어를 지원할 수 있으며, stand-alone방식을 취하여 소형·경량의 시스템을 요구하는 모바일 장비나 사람이 근접하기 어려운 장소에 장착된 장비와의 데이터 전송을 쉽게 할 수 있게 하였다. 또한 본 시스템은 원격제어에서 실시간 제어에 유리한 UDP를 구현하기 보다 신뢰성이 우수한 TCP를 구현하고 CAN에서 지원하는 신뢰성을 함께 이용함으로써 신뢰성에 우수한 시스템이 될 수 있게 설계하였다.

본 연구는 향후 UDP프로토콜과 TCP프로토콜을 동시에 구현하여 데이터를 전송시에는 TCP를 이용하고, 실시간을 구현할 때는 UDP를 구현할 수 있도록 하여, 신뢰성과 실시간성을 동시에 구현할 수 있는 알고리즘을 개발하여야 한다. 또한, TCP/IP-CAN Gateway 내부에서 보다 효율적인 메모리 관리 및 이벤트 관리를 위해 보다 개선된 스케줄링 알고리즘을 개발하여야 한다.

참고문헌

- [1] Robert E. Filman, RIACS, "Embedded internet systems come home," *IEEE Internet Computing*, vol. 5, pp. 52-53, Issue : 1, Jan.-Feb. 2001.
- [2] Burton H. Lee, "Embedded internet system : Poised for takeoff," *IEEE Internet Computing*, vol. 2, pp. 24 -29, Issue : 3, May-June 1998.
- [3] Gerhard Gruhler, Gerd Nusser, Dieter Buhlet, and Wolfgang Kuchlin, "Teleservice of CAN systems via internet," *Proc. of the International CAN Conference(ICC)*, Torino, Italy, Nov. 1999.
- [4] Gerhard Gruhler, Gerd Nusser, Dieter Buhlet, and Wolfgang Kuchlin, "A java client/server system for accessing arbitrary CAN open fieldbus devices via the internet," *South African Computer Journal*, no. 24, pp. 239-243, ISSN : 1015-7999., Nov. 1999.
- [5] Jack W, Szymanski, "Embedded internet technology in process control devices," *WFCS-2000*, Sep. 6-8, 2000.
- [6] M. Farsi, K. Ratcliff, and Manuel Barbosa, "An overview of controller area network," *Computing & Control Engineering Journal*, vol. 10, pp. 113 -120, Issue : 3, Jun. 1999.
- [7] 홍성수, "분산 실시간 제어 시스템의 개발," CASE 기술 특집 : 실시간 제어 시스템(3), ICASE, 1. 1998.
- [8] 차석근, "TCP/IP를 이용한 제어기기 인터페이스," 제어 · 자동화 · 시스템 공학회지 제2권 제4호, 7. 1996.
- [9] Jamahl W. Overstreet and Anthony Tzes, "An internet -based real-time control engineering laboratory," *IEEE Control Systems Magazine*, vol. 19, Issue : 5, pp. 19-34, Oct. 1999.
- [10] J.W. Park and J.M. Lee, "Transmission characteristic analysis and modeling for internet-based control," *IEEE International conference on Mechatronics and Machine Vision in Practice*, Hong-Kong, CD-ROM version, 2001.