

게이트웨이 설계 및 네트워크 기반 복수전동기 제어

김 관 수, 정 의 현, 이 흥 희
울산대학교 전기전자정보시스템공학부

Gateway Design and Network Based Multi-Motor Control

Kim Gwan-Su, Eui-Heon Jung, Lee Hong-Hee,
University of Ulsan, School of Electrical Engineering

ABSTRACT

오늘날 산업현장에서는 다양한 필드버스 제품들이 사용되고 있으며 여러 필드버스들 사이의 통신에 관한 문제가 이슈화 되고 있다. 본 논문에서는 현재 사용되고 있는 필드버스들 가운데 가장 많이 사용되고 있는 Profibus와 CAN 네트워크를 대상으로 이기종 필드버스 시스템 사이의 실시간 데이터 전송을 위한 PC 기반의 게이트웨이 장치를 구현하고 그 타당성을 복수전동기 구동실험을 통하여 검증해 보았다.

1. Introduction

최근 제조업을 둘러싸고 있는 환경이 크게 바뀌고 생산시스템의 그로벌화가 진행되면서 생산라인의 감시/제어에서 정보전송에 이르기까지 모든 정보를 통합시키기 위한 네트워크 시스템의 중요성이 날로 증가하고 있다.^[1] 그러나 제조시스템의 종류가 다양하고 이러한 제품들이 제작사들이 이미 오래전부터 독자적인 디지털 통신기술을 개발하여 사용해왔기 때문에 제품들간의 통신 호환성이 없어 통합 환경의 네트워크 시스템을 구축하는데 어려움이 많다. 그래서 많은 필드버스 제조업체들이 표준화의 필요성을 느끼고 1990년대 초반부터 표준화 작업을 진행해 왔으나 여러 가지 기술적인 문제와 이해 관계로 명쾌한 결론을 내리지 못하고 있다. 오늘날 디지털 통신기술은 날로 발전하고 있고 미래형 공장은 네트워크 구축이 필수적이다. 국내에서는 한국프로피버스 협회를 만들어 특정 필드버스 보급에 노력하고 있지만 전문인력의 부족으로 서로 다른 프로토콜에 대한 통합 환경 구축은 아직 엄두를 못내고 있는 실정이다.

본 연구는 국내에서 많이 사용되고, 세계적으로 그 시장성이 큰 Profibus, CAN에 대한 프로토콜의

분석을 통해 상호 호환성이 없는 프로토콜 사이의 인터페이스를 가능하게 하여 제조시스템에서의 네트워크 환경을 구축하고자 하는 것이다.^{[2],[3]}

2. Profibus Communication

산업용 네트워크 시스템에서 적용되고 있는 Profibus 프로토콜은 OSI참조 모델을 기반으로 하고 있으며 물리계층, 데이터 링크 계층, 그리고 응용계층으로만 구성되어 있다. Profibus는 독일의 Bosch, Siemens, Klockener-Moller에서 개발하여 자국의 표준안인 DIN 19245로 지정된 프로토콜로 최근에 WorldFIP, P-NET과 함께 유럽의 표준안 EN50170으로 지정되어 생산자동화, 공정제어, 빌딩 자동 분야에서 필드 장비들 간의 실시간 통신을 위하여 사용된다. Profibus 제품군은 그림 1과 같이 Profibus-FMS, Profibus-DP, Profibus-PA으로 나뉘어진다.

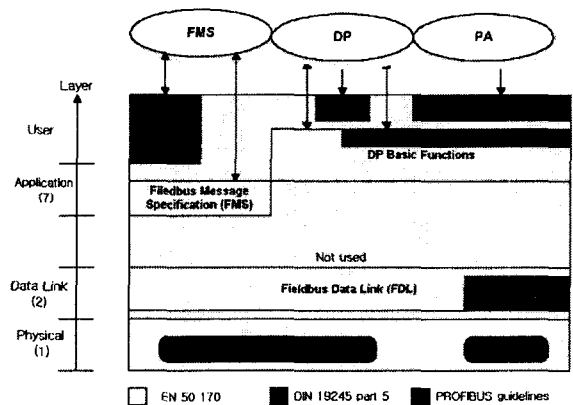


그림 1 Profibus 프로토콜의 구조
Fig 1 Architecture of Profibus protocol

본 논문에서 적용된 Profibus 프로토콜은 마스터 슬레이브 방식으로 동작하는 필드버스이다. 따라서

이러한 필드버스를 이용하여 통신을 수행하기 위해서는 먼저 두 노드간의 통신인증 과정을 거쳐야 한다. Profibus는 진단(diagnosis) 텔레그램을 사용하여 장비의 인증과 에러 검출을 수행하여 마스터 노드로 그 정보를 보낸다. 진단 텔레그램은 표 1과 같이 구성된다.

표 1 Profibus 진단 텔레그램
Table 1 Profibus diagnosis telegram

Byte	Meaning
1	Station_Status 1
2	Station_Status 2
3	Station_Status 3
4	Diag.Master_Add
5,6	Ident_Number
7	1 Ex_Diag Header
8	2 CAN_Status
9	3 CAN_Address
10	4 CAN_Data rate

Profibus의 노드 주소는 0번지부터 125번지까지 가능하며 마스터 노드의 주소를 1번으로 하고 슬레이브 노드의 주소는 2번으로 하였다. 데이터 전송 속도는 9.6KBaud에서 최대 12MBaud까지 이다. Profibus 슬레이브 노드의 식별자(Identifier)는 표2와 같다. 표2의 식별자에 대한 포맷은 그림2와 같다.

표 2 Profibus 슬레이브 노드의 구성
Table 2 Profibus Slave node Configuration

Byte	Identifier
1	I/O Data
2	I/O Data
B7	For 8 Bytes I/O Data, consistent
37	For 8 Bytes I/O Data, nonconsistent

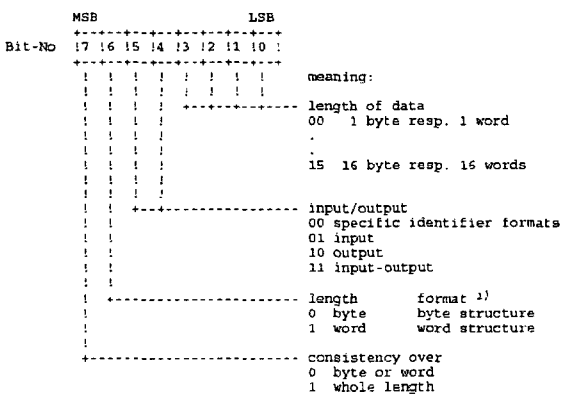


그림 2 식별자 포맷
Fig 2 Identifier Byte Format

CAN I/O 데이터와 Profibus-DP 텔레그램의 데이터 맵핑은 다음과 같다. 이 과정은 Profibus의 관

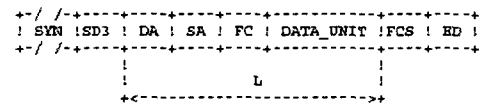
점에서 CAN I/O 데이터를 처리하는 것이다. 테이블 3에 CAN I/O 데이터와 Profibus데이터와의 맵핑 관계를 나타내었다.

표 3 CAN데이터와 Profibus-DP 텔레그램의 맵핑
Table 3 CAN data to Profibus-DP telegram mapping

Profibus input data	CAN output data
Byte 0	Byte 0
...	...
Byte 7	Byte 7
Profibus output data	CAN input data
Byte 0	Byte 0
...	...
Byte 7	Byte 7

표 3에서는 CAN 프로토콜의 최대 I/O Byte수는 8Byte이기 때문에 Profibus의 Byte수를 CAN과 같게 맞추었다. CAN데이터 프레임 주에서 I/O 부분만 가져와서 Profibus의 프레임중 DATA_UNIT 부분에 넣어서 Profibus의 프레임을 구성하게 된다. Profibus의 프레임 구조는 그림 3과 같다.^[4]

A) Format of the Send/Request Frame:



B) Format of the Response Frame:

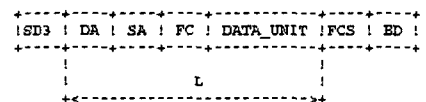


그림 3 Profibus 프레임 포맷

Fig 3 Profibus Frame Formats

3. CAN Communication

차량내의 배선상의 문제등으로 인해서 개발된 프로토콜인 CAN은 현재 여러 차량용 프로토콜중 가장 널리 사용되고 있다. CAN은 BOSCH 사에서 제안한 차량용 네트워크로서 ISO 11898 국제 표준 규격으로 제정되었으며 CSMA/CD+AMP(Carrier Sense Multiple Access/Collision Detection + Arbitration on Message Priority)프로토콜을 가진다.^{[5][6]} 이는 IEEE 802.3 CSMA/CD 프로토콜과 유사하며 ISO/OSI 7 계층중 물리계층과 데이터 링크 계층만으로 제안되었다. 이러한 CAN 프로토콜을 간단히 설명하면, 연결된 모든 노드는 전송해야 할 메시지가 있는 경우 버스의 상태를 감지해 버스 비활성화 상태에서 이를 전송한다. 만약 하나 이상의

노드에서 버스 사용권을 요구했을 경우 메시지나 노드에 부여된 식별자를 이용해 우선순위가 높은 노드에서 버스 사용권을 얻는다.(이러한 버스사용권 획득방법은 NBA(Non-destructive Bitwise Arbitration)라 불린다.) 우선순위가 낮아 버스 사용권을 획득하지 못한 노드는 버스의 상태를 감시하다 비활성화 되면 메시지를 재전송한다.

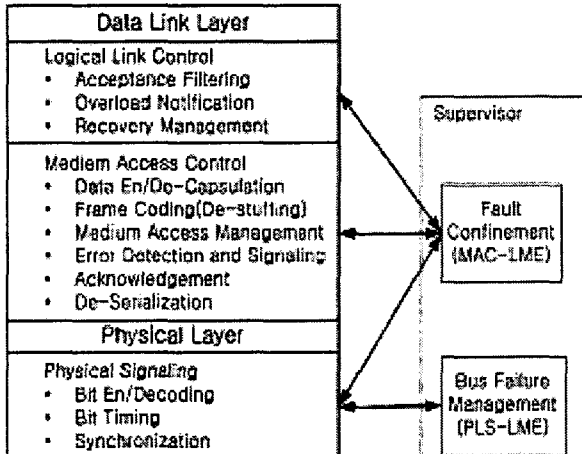


그림 4 CAN의 계층적 구조
Fig 4 Layered architecture of CAN

Profibus I/O 데이터와 CAN 텔레그램의 데이터 매핑은 다음과 같다. 이 과정은 CAN의 관점에서 Profibus I/O 데이터를 처리하는 것이다. 표 4에 Profibus I/O 데이터와 CAN 데이터와의 매핑 관계를 나타내었다.

CAN-Header										
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	
Byte 0	0	0	1	1	0	0	0	0	30	
Byte 1	0	1	0	0	1	0	0	0	48	

CAN-Data	
Byte 2	Profibus output data byte 0
Byte 3	Profibus output data byte 1
Byte 4	Profibus output data byte 2
Byte 5	Profibus output data byte 3
Byte 6	Profibus output data byte 4
Byte 7	Profibus output data byte 5
Byte 8	Profibus output data byte 6
Byte 9	Profibus output data byte 7

CAN-Header										
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	
Byte 0	0	1	0	0	0	0	0	0	40	
Byte 1	0	1	0	0	1	0	0	0	48	

CAN-Data	
Byte 2	Profibus input data byte 0
Byte 3	Profibus input data byte 1
Byte 4	Profibus input data byte 2
Byte 5	Profibus input data byte 3
Byte 6	Profibus input data byte 4
Byte 7	Profibus input data byte 5
Byte 8	Profibus input data byte 6
Byte 9	Profibus input data byte 7

표 4 Profibus 데이터와 CAN 텔레그램의 매핑
Fig 4 Profibus data to CAN telegram mapping

전송버퍼의 글로벌 Layout는 표 5와 같다. 이 버퍼는 SJA1000에서 데이터를 전송하고자 할 때 마이크로컨트롤러로부터 받은 메시지를 저장하는 기

능을 한다. 이 버퍼는 Descriptor(Head)와 Data Field로 나누어 진다. 이 전송버퍼는 동작모드에서 마이크로컨트롤러에 의해서만 읽고 쓰기가 가능하다.

CAN ADDRESS	FIELD	NAME	BITS							
			7	6	5	4	3	2	1	0
10	descriptor	identifier byte 1	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
11		identifier byte 2	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
12	data	TX data 1	transmit data byte 1							
13		TX data 2	transmit data byte 2							
14		TX data 3	transmit data byte 3							
15		TX data 4	transmit data byte 4							
16		TX data 5	transmit data byte 5							
17		TX data 6	transmit data byte 6							
18		TX data 7	transmit data byte 7							
19		TX data 8	transmit data byte 8							

ID 10 ... ID 0: CAN identifier
The CAN identifier consists of the object code (ID 10 ... ID 7) and of the node address (ID 6 ... ID 0).

ID 10	ID 9	ID 8	ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID 0
objectcode						node address				

RTR: Remote Transmission Request bit
DLC 3: DLC0: Data Length Code, length of the user data

표 5 전송버퍼 Layout
Table 5 Transmit buffer layout

4. Gateway System

Profibus와 CAN 노드의 통신을 수행하기 위한 게이트웨이 시스템의 기본 블록도는 그림 5와 같다. 여기서, CAN에서 Profibus로의 데이터 통신은 CAN의 수신버퍼에 저장된 데이터는 Profibus의 송신버퍼로 이동한다. 그리고 게이트웨이 장치는 표3에서 알 수 있는 바와 같이 Profibus의 관점에서 CAN 입력 데이터를 Profibus-DP 출력 텔레그램으로 매핑한 후 Profibus 목적지 노드로 데이터를 전송한다. 역으로, Profibus에서 CAN으로의 데이터 전송은 반대 과정으로 수행되는데, 표4와 같이 CAN의 관점에서 Profibus-DP입력 데이터를 CAN 출력 텔레그램에 매핑한 후 CAN 노드로 전송한다.

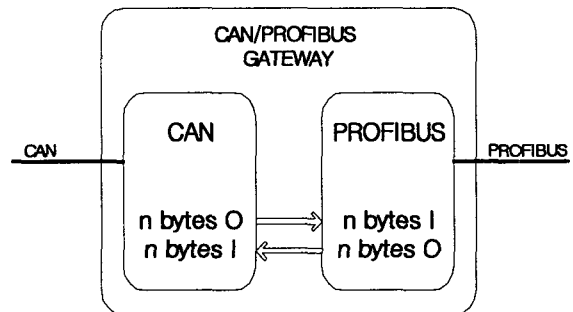


그림 5 게이트웨이 시스템 블록도
Fig 5 Block diagram of gateway system

그림 6은 Profibus와 CAN의 게이트웨이 기능을 가지는 이기종 네트워크 시스템을 나타낸 것이다.

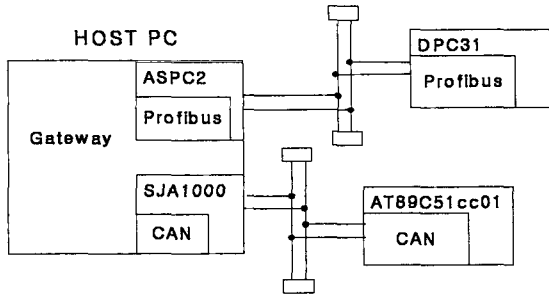


그림 6 게이트웨이 기능을 가진 이기종 네트워크 시스템
Fig 6 Heterogeneous network system with gateway function

여기서 HOST PC 기반의 필드버스 게이트웨이 장치를 구현하기 위해 Profibus와 CAN PCI 카드를 각각 장착하였다. Profibus PCI 카드는 Siemens의 CP5614 Profibus-DP 마스터 카드를 사용하여 설치하였다. 그리고 Profibus 슬레이브 노드는 Siemens의 Profibus-DP 슬레이브 개발 키트를 사용하여 구현하였다. CAN의 독립노드는 ATMEL사의 CAN 컨트롤러가 내장된 AT89C51cc01 칩을 사용하여 구현하였다. 이러한 게이트웨이 장치를 구현하기 위한 소프트웨어로는 HOST PC상에서는 Visual Studio를 사용하여 VC++로서 구현하였다. 그리고 Stand alone 노드중 Profibus 슬레이브 노드는 V1SL DPC31 Profibus-DP 펌웨어를 기반으로 하여 응용 프로그램을 구현하였고, CAN노드는 Keil C 컴파일러를 사용하여 펌웨어 및 응용 프로그램을 구현하였다.

5. Experimental Results

구현한 필드버스 게이트웨이 시스템을 이용하여 네트워크 기반 복수전동기를 제어하기 위한 실험 구성도는 그림 7과 같다. 이 실험에서는 Mitsubishi사의 0.4kW의 정격출력과 3000rpm의 정격속도를 가진 서보 앰프와 전동기를 사용하였다.

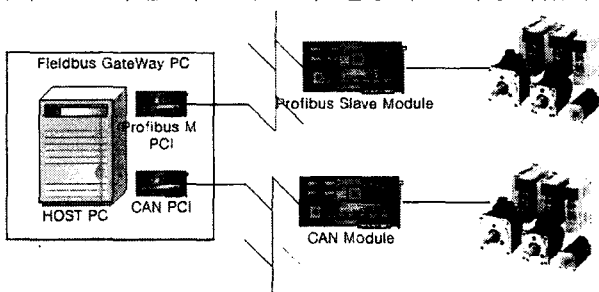


그림 7 게이트웨이 시스템의 실험 구성도
Fig 7 Experimental configuration for network system

그림 7과 같은 실험장치에서 CAN과 Profibus 네트워크의 통신속도는 각각 1Mbps와 12Mbps로 설정하였다. 본 논문에서는 2개의 전동기를 구동하

는 시스템에서 하나의 속도는 다른 것의 속도에 종속된다는 가정을 두고 복수 전동기의 속도 동기 제어 실험을 수행하였다. 전동기 하나는 마스터로 동작하고 나머지 하나는 마스터의 속도를 추종하는 슬레이브로 동작한다. 속도 지령에 의해 마스터 전동기가 동작할 때 슬레이브 전동기는 마스터의 실제 속도를 추종한다. 즉, 슬레이브 전동기의 속도 지령은 마스터의 속도 출력이다. HOST PC에서 마스터 전동기의 속도정보를 받아서 제어 알고리즘에 의해 슬레이브 전동기로 속도지령을 보낸다.

그림 8은 게이트웨이 시스템의 동작 상태를 모니터링하는 윈도우 환경을 나타낸 것이다. 이 프로그램은 CAN과 Profibus의 이기종 프로토콜 사이의 데이터 통신을 수행한다.

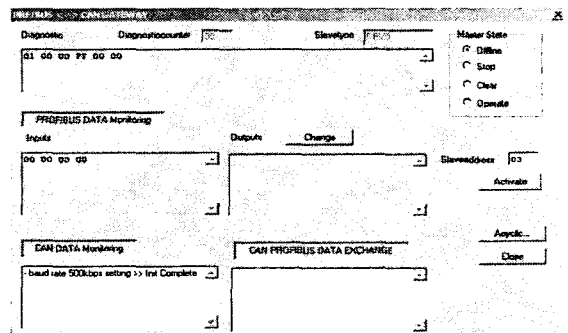


그림 8 게이트웨이 동작 프로그램 환경
Fig 8 Window of Gateway Operating Program

그림 9는 Profibus 노드에서의 데이터 전송 파형이다. 그림 9-(a)는 Profibus 노드에 데이터를 가지고 있을때의 파형이고, (b)는 확대한 파형을 나타낸다. (c), (d)파형은 데이터를 가지고 있지 않을때의 파형과 그것을 확대한 파형을 나타낸다.

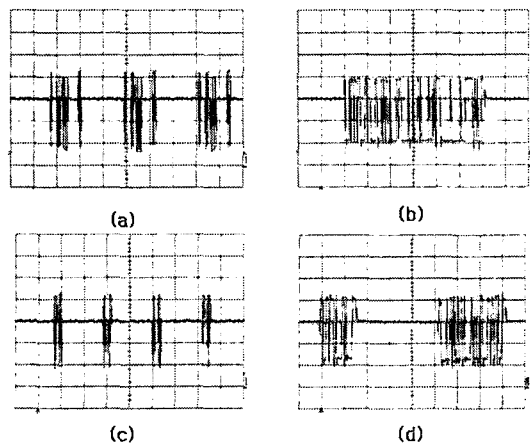


그림 9 프로피버스 데이터 파형
Fig 9 Profibus data waveforms

그림 10은 그림 9와 같은 파형으로 CAN 노드에서의 데이터 전송 파형이다. 그림 10-(a), (c)는 데이터가 있을때와 없을때의 파형을 나타내고 (b), (d)는 이것들을 각각 확대한 파형이다.

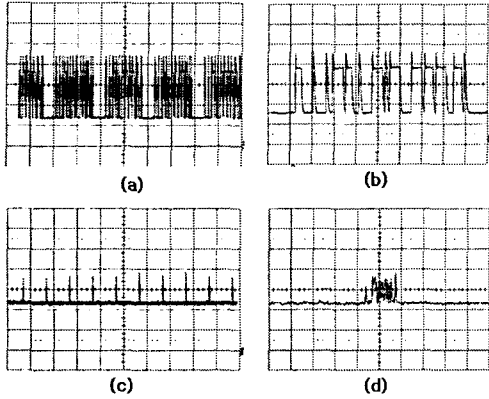


그림 10 CAN 데이터 파형
Fig 10 CAN data waveforms

그림 11은 PC 기반의 게이트웨이 시스템의 성능을 확인하기 위해 나타낸 파형이다. (a)파형은 CAN 노드에서 Profibus 노드로 전송 될 때를 나타낸 것이고, (b)는 Profibus 노드에서 CAN 노드로 전송될때를 나타낸 것이다.

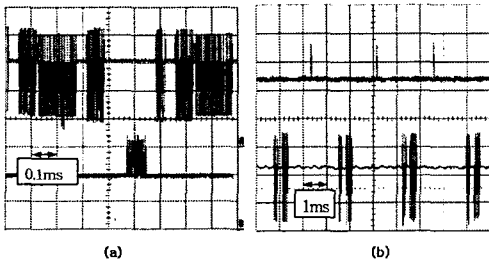


그림 11 Profibus와 CAN 노드의 전송파형
Fig 11 Profibus and CAN Data Transmission Waveforms

그림 11에 보여지는 것과 같이 지연 시간은 거의 0.1[ms] 이고, 복수전동기 구동이나 로봇 제어를 위한 네트워크 시스템에서 실시간으로 수행할 수 있는 충분한 시간이다.

그림 12는 게이트웨이 시스템을 사용한 마스터 전동기와 슬레이브 전동기의 속도 파형을 나타낸 것이다. 마스터 전동기의 실제 속도는 HOST PC(게이트웨이시스템)로 전송되고, HOST PC에서 슬레이브 전동기의 속도를 계산하여 슬레이브 전동기의 서보 앰프로 속도 지령을 전송한다. 마스터 전동기의 속도는 채널 1에 나타내었고, 슬레이브 전동기의 속도는 채널2에 나타내었다. 마스터 전동기의 속도를 잘 추종하는것을 볼 수 있다.

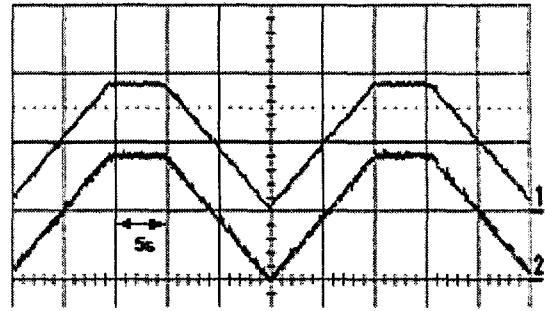


그림 12 마스터 전동기의 속도에 따른 슬레이브 속도지령
Fig 12 Slave speed command according to Master motor speed

6. Conclusion

본 논문에서는 CAN 노드와 Profibus 노드 사이의 실시간 통신을 위한 PC 기반의 게이트웨이 시스템을 개발하였다. 그리고, 개발한 게이트웨이 시스템의 성능을 실험을 통하여 확인하였다. PC 기반의 게이트웨이 시스템은 유연성이 좋고, 다른 독립형 게이트웨이 시스템에 비해 설계가 간편하다. 더구나 이러한 종류의 게이트웨이 시스템은 PC를 HOST 컴퓨터로 네트워크 시스템에 사용하기 때문에 추가적인 장비나 장치의 도움없이 구현할 수 있다. 또한, PCI 카드 형태로 구현할 수 있는 다른 필드버스 프로토콜에도 적용할 수 있다.

본 연구는 한국과학재단 지정 울산대학교 네트워크 기반 자동화연구센터의 지원에 의한 것입니다.

참고 문헌

- [1] Keines H., Zwoll K., Drochner J., Integration of industrial Automation Equipment in Experiment Control Systems via PROFIBUS-Developments and Experiences, *IEEE TRANS On NUCLEAR SCIENCE*, Vol. 47, 2000
- [2] Christian S., Thilo S., A Secure Architecture for Fieldbus/Internet Gateways, *IEEE*, pp. 279-285, 2001.
- [3] Philippe M., Alain C., A Wireless Gateway For Fieldbus, *IEEE*, pp. 105-109, 1995.
- [4] Profibus organization, PROFIBUS Specification Normative Parts of PROFIBUS-FMS,-DP,-PA, Profibus, 1998
- [5] Wlofhard Lawrenz, CAN System Engineering form Theory to Practical Application, Springer, 1997.
- [6] CiA, CAN Specification 2.0 PartA,B
- [7] Philips Semiconductor, SJA1000 Stand-alone CAN controller, Data Sheet, 2000.