

네트워크 기반 유도전동기 제어를 위한 고성능 LonWorks 제어모듈 개발

(Development of High Performance LonWorks Fieldbus Control Modules for Network-based Induction Motor Control)

김중곤 · 홍원표
(Jung-Gon Kim · Won-Pyo Hong)

Abstract

The interface between host processor and the ShortStack Micro Server may be a Serial Communication Interface(SCI). The LonWorks control module with a high performance is developed, which is composed of the 8 bit PIC Microprocessor for host processor and the smart neuron chip for the ShortStack Micro Server. This intelligent control board is verified as proceeding the various function tests from experimental system with an boost pump and inverter driving systems. It is also confirmed that the developed control module provides stably 0-10VDC linear signal to the input signal of inverter driving system for varying the induction motor speed. Thus, the experimental results show that the fabricating intelligent board carried out very well the various functions in the wide operating ranges of boost pump system. This developed control module expect to apply to industrial fields to require the comparatively exact control and monitoring such as multi-motor driving system with inverter, variable air volume system and the boost pump water supply systems.

1. 서 론

분산제어시스템이 도입되기 시작한 1980년대 초반부터 각종 제어 및 자동화 설비에 네트워크시스템이 도입되기 시작하였다. 그러나 초기의 네트워크시스템은 장비생산업체(vendor)들이 그들의 시장을 확보하기 위하여 개방화되지 않는 독자 모델의 vendor 중심의 프로토콜 버스를 사용하여 사용자들은 장비 생산업체에 기술적으로 종속당할 수밖에 없었다. 결국 커스텀 프로그램이나 게이트웨이를 사용하여 지능형빌딩이나 산업 애플리케이션을 구현하는 것은 가능하나 시스템 유지를 위한 비용과 신뢰도에서 효율적이지 못했다. 또한 다른 생산업체로부터 공급받은 장비들을 네트워크를 통하여 접속하기 위해서는 매우 고가의 Protocol 변환장치를 사용하지 않을 수밖에 없었다. 이러한 최소한의 공통 커뮤니케이션을 찾기 위한 노력이 시스템 통합자들에 의해 이루어 졌고 이에 따라 게이트웨이(gateway)가 사용되기 시작하였다. 게이트웨이는 제한적인 상태값과 제어정보를 전달할 뿐이어서 오류상태(fault status)에 대한 정보가 공유될 수 없고 게이트웨이의 조합형 프로그램은 서로 다른 센서로부터의 정보에 접근할 수 없으며 따라서 시스템 전반적인 상태 값에 대해서 리얼타

임으로 대처할 수 없다. 더욱이 게이트웨이는 서버 시스템 중 한 개만 바뀌어도 수정되어야 했고 이는 시스템 통합자 및 최종 사용자에게 까다로운 유지보수의 문제점을 야기시켰다. 이러한 시스템을 사용하는 동안에는 커스텀 프로그램이나 게이트웨이를 제공하는 제조업체에 종속될 수밖에 없었다. 최근 네트워크의 표준화와 개방화된 Protocol을 사용한 상호 운용성을 구현하기 위해 LonWorks과 BACnet 등의 Protocol을 사용하고 있다. ASHRAE의 BACnet과 Echelon사가 개발한 LonWorks등의 국제 표준에 따른 개방 시스템 아키텍처를 지향하고 있다. 특히 LonWorks시스템은 모든 Device에 의해 공유되는 Protocol을 통하여 통신할 수 있는 기술이다. 통신용 트랜시버와 매카니즘이 표준화되어 있고 오브젝트(object)모델, 프로그래밍/문제해결도구가 LonWorks Device간의 보다 빠르고 상호운용 가능한 설계 및 구현을 가능케 한다. 또한 LonWorks의 지능분산, 상호운용 가능성 및 수평적 구조(flat architecture)는 멀티 벤더의 환경을 구축해주고 시스템 제공자 및 사용자 모두에게 융통성과 유지보수의 용이성을 제공해 준다. 따라서 폐쇄적이고 하나의 벤더에 의존적이던 종래의 시스템이 보다 저렴한 설치비용, 보다 저렴한 유지보수비용 및 강화된 기능과 확장성이 보장되는 새로운 제어용 개방 네트워크 시스템이

라 할 수 있다. 시스템의 사용자는 단일 공급자가 제공하는 시스템을 사용하게 되면 그 공급자에 의존적이고 값비싼 업그레이드 계약이 수반되는 시스템에 묶여있게 되는 반면, LonWork Device 시스템을 사용하게 되면 신뢰성을 향상시키면서 유지보수비용도 절감할 수 있고, 시스템 공급자의 Device를 사용자가 입맛에 맞게 고를 수 있는 통합시스템 개발의 필요성이 절대적으로 필요하여 이 연구를 진행하게 되었다[1-3]. 최근 고성능과 다기능 범용프로세서의 산업계 적용이 광범위하게 이루어지고 있어 필드버스와의 네트워크를 구성하는 기술이 매우 주목을 받고 있다. 특히 LonTalk 통신 Protocol을 사용한 LonWorks Network는 Device의 네트워크 구성을 위한 강력한 소프트웨어[4]를 제공하고, 빌딩설비 등 다양한 산업계에 적용되고 있는 Protocol로 자리 잡아가고 있다. 따라서 Echelon사에서 제공하는 ShortStack소프트웨어[5]를 이용하여 범용프로세서의 하나인 PIC Microprocessor[6,7]에서 제어를 위한 응용프로그램을 수행하고 Neuron Chip에서 통신을 담당하게 하는 시스템을 개발함으로써 LonWorks 네트워크 및 노드의 산업계 적용범위를 확대할 수 있는 기반을 구축하였다[8-19].

2. 고성능 모듈 구성 프로세서

2.1 LonWorks 필드버스

LonWorks는 시스템의 설계, 구축, 운용, 유지를 위하여 4가지 구성요소가 있으며 통신프로토콜로서의 LonTalks, H/W로서 뉴런칩, 전송매체와의 연결을 위한 트랜시버(Transceiver), 통신망운용시스템으로서 LNS(LonWorks Network Service)가 있다.

이 제어용 네트워크 시스템은 Profibus, CAN 등 기존의 필드버스가 OSI 1,2, 7 층(layer)을 커버하는 데 비하여 LonWorks는 OSI 7개 층 모두를 커버하는 오픈 프로토콜로서 현장에서 제어네트워크를 구성하는 데 필요한 모든 요소는 물론이고 일관된 네트워크 관리 솔루션까지 갖춘 제어용 네트워크이다. 또한 통신 프로토콜 자체가 이미 뉴런칩에 H/W, F/W(firmware)형태로 포함되어 있기 때문에 디바이스 개발자는 LonTalk 프로토콜의 자세한 패킷 규격을 모르더라도 애플리케이션 프로그램을 쉽게 작성할 수 있다. i·LON 서버를 이용하여 인터넷에 쉽게 연결될 수 있어 빌딩의 원격 및 군 관리까지도 용이하게 구성할 수 있다. 론 포인트 시스템은 크게 연결되는 하위센서와 액추에이터의 종류에 따라 디지털 입력/출력, 아날로그입력/출력 모듈과 보일러 가동 등에 필요한 스케줄 작성을 위한 스케줄러 모

듈 또는 각 입/출력 점의 데이터 트랜드를 저장하는 데이터 로깅 모듈, 그리고 론 워크 네트워크의 트래픽 관리를 위한 라우터 모듈 등으로 나누어진다. 론 포인트 시스템은 8가지 모듈과 전원선/네트워크선 및 입출력선 연결을 위한 베이스플레이트, 네트워크 터미네이터로 구성되어 있다. 론 포인트는 지능화된 노드를 구성하기 위하여 각 디바이스마다 론 포인트를 연결시켜 완전 분산제어를 구성하는 것이 최종 목표라 할 수 있다.

2.2 범용프로세서 PIC

PIC16F877 프로세서는 8 Bit MOS One Chip Micro Controller로 RISC와 하바드 아키텍처 구조를 가지고 있으며, 8K Word의 Flash Memory와 368 Byte의 데이터 메모리, 256 Byte의 EEPROM을 가지고 있다. 프로그램 메모리가 플래쉬로 되어 있어서 여러 번 라이팅 하는 것이 가능하며 프로그램 메모리는 14비트, 데이터 메모리는 8비트버스, 40핀의 패키지로 되어있고, 입출력포트는 양방향지정이 자유로운 33개의 I/O 포트에 되어 있다. 8개의 스택을 가지고 있고, 14개의 인터럽트 소스 그리고 8비트와 16비트의 타이머 내장과 10비트 AD컨버터가 8채널이 내장되어있고, SPI, I2C, SSP통신 모듈 및 8 Bit 병렬 슬레이브 포트가 내장되어 있다.

3. ShortStack을 이용한 지능분산제어 Device 개발

3.1 ShortStack Device 설계

값이 저렴하고 기능이 많아 사업계에 널리 사용되고 있는 8Bit Microprocessor인 PIC16F877과 FT3150 Smart Transceiver를 사용하여 다기능을 갖춘 지능형 디바이스를 개발하였다. Host Microprocessor인 PIC16F877내에 응용 프로그램을 정리하고 ShortStack Program내에 Host와 Micro Server가 SCI 나 SPI로 Interface 를 할 수 있는 Serial Driver가 내장되어 있어 구현이 용이하다. 그림1은 Neuron Chip과 PIC16F877로 구성된 고성능 모듈의 블록다이어그램이다.

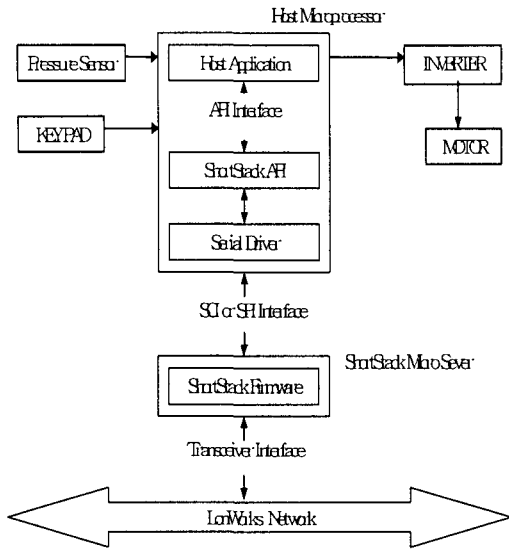


그림 1 ShortStack Server과 Host Microprocessor의 block diagram

통합을 위한 회로도로서 SPI와 SCI Interface 방법이 있는데, 본 논문에서는 SCI를 이용한 Interface를 사용하였고, SCI Interface에 관한 내용을 정리한다. SPI에는 더 많은 IO 연결이 필요하기 때문에 SCI가 Host Processor의 입장에서 구현이 용이하다.

3.2 AO 출력 디바이스 회로도

그림2는 AO 출력 Device 회로도로 PIC16F877의 DE0~2번 포트를 사용하여 시리얼로 데이터를 보내 데이터 값을 아날로그 신호 값 4~20mA를 출력하는 회로를 구성하였다.

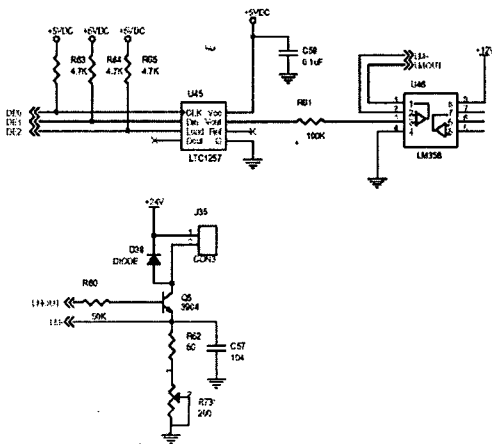


그림 2. AO 출력 디바이스 회로도

4. 고성능시스템 구축 및 고찰

4.1 ShortStack 시스템 구축 방법

(1) ShortStack Micro Server에 ShortStack3150.net을 다운로드한다. 본 논문에서는 ShortStack Micro Server 펌웨어인 ShortStack3150.net을 AT29C512에 다운로드하였고, 그림 3은 ShortStack 통합보드의 그림이다.

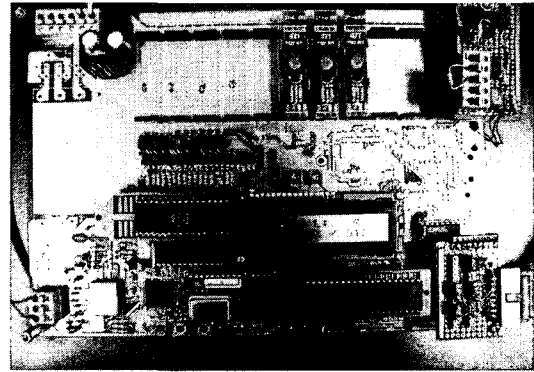


그림 3. ShortStack 통합보드

(2) Neuron Device에 PIC16F877을 통합하여 하드웨어 구성을 하였고, SPI에는 더 많은 IO 연결이 필요하기 때문에 SCI가 Host 프로세서의 입장에서 구현이 용이하다. 본 논문에서는 SCI를 이용한 Interface를 점퍼 핀을 이용하여 구현하였다.

(3) NB3 Resource Editor를 사용하여 리소스를 생성하였다. NB3 Resource Editor는 ShortStack Wizard에 포함이 되어있으며, Node Builder를 사용해도 된다. Network Variable, Configuration Property Neuron C model file을 생성하여 파일로 저장을 하였다.

(4) ShortStack Wizard를 구동하여 interface data와 device interface file을 생성하였다. ShortStack Wizard에서 Micro Server에 대한 몇 가지 질문에 대답하면 Wizard가 전체 NV(Network Variable)에 대한 Table을 작성해 주며 Micro Server를 초기화하고 설정하는 코드를 자동 생성해준다. CP(Configuration Property)의 경우, Direct memory R/W 방식은 지원하지 않는다.

(5) ShortStack Wizard를 구동하게 되면 생성되는 5 가지 파일이 있다.

“platform.h” : Platform dependent flags, basic data type

“NvTypes.h”: NV type 에 대한 구조와 enumerations

“LonDevice.h” 와 LonDevice.c : NV definition 과 utility functions

“filedir.h” : FTP 트랜잭션을 위한 구조로 된 파일이 그것이다.

(6) ShortStack에서 사용될 LonWorks의 Event Application을 수정 및 보완하여 Hi-Teck Comfile을 사용하여 Comfile을 한 후 Host Processor에 버닝 작업을 완료한다. 대부분 4KB의 코드공간과 200bytes이내의 RAM 사용을 한다.

(7) LonMaker for Windows를 사용하여 Device를 테스트한다.

(8) Device Plug-In을 개발한다. PIC 계열의 Micom은 8개 이상의 Nested Function을 지원하지 않음에 주의하도록 하고, PIC 계열의 Micom 사용시에는 Interrupt 루틴을 주의하여야 한다.

4.2 ShortStack을 이용한 장치 구성

(1) 그림 4는 ShortStack 통합보드를 이용한 고성능 LonWorks 모듈의 성능을 실험하기 위한 시스템 구성도이다. ShortStack 통합보드와 외부에 H사 인버터(N200 벡터 인버터) 그리고 Pump(Induction Motor - 2.2KW)를 설치 구성하여 외부의 아날로그 값으로 펌프속도를 가변 기동하는 장치를 구성하였고, 그 구성요소에 대해 LonMaker for Windows를 이용하여 모니터링을 하였다. 그림 5는 H사의 인버터(N200 벡터 인버터)이다.

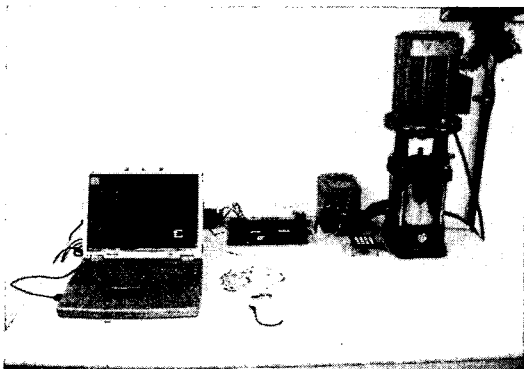


그림 4. ShortStack 통합보드를 이용한 시스템 구성

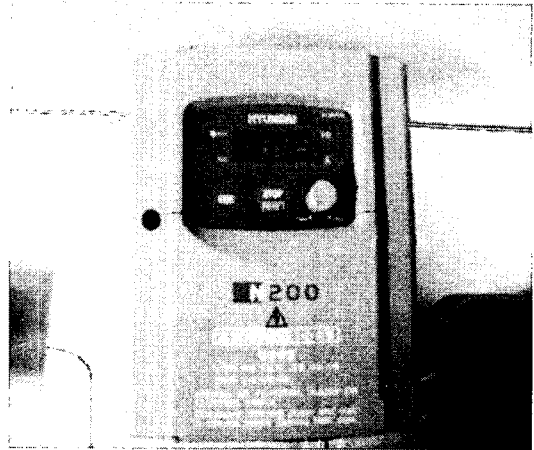


그림 5. H사 인버터(N200 벡터 인버터)

그림 6은 LonMaker for Windows에서 Device와 Function Block내의 NV를 바인딩한 후 모니터링 하기 위한 S/W 시스템 구성도를 나타낸 것이다.

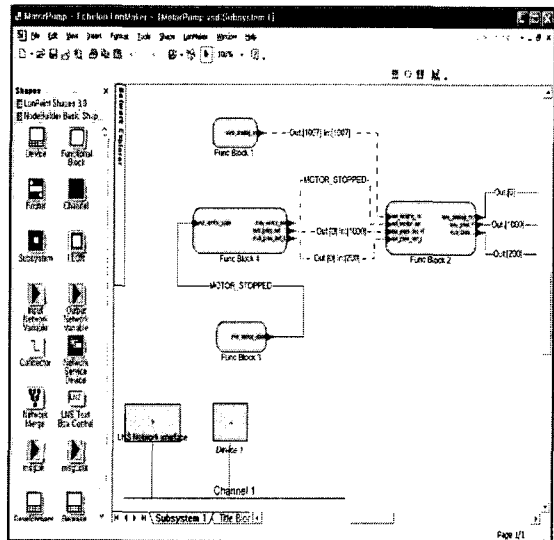


그림 6 LonMaker for Windows에 의한 바인딩

그림7은 LonMaker Browser에서 디바이스내에 있는 평선블록의 입출력 네트워크 변수의 값을 모니터링 한 그림이다.

Subsystem	Device	Functional Block	Network Variable	Config Prop	Min	Value
Subsystem 1	Device 1	Func Block 1	inv_analog_in	#	1007	
Subsystem 1	Device 1	Func Block 2	inv_analog_in	#	1007	
Subsystem 1	Device 1	Func Block 2	inv_motor_set	#	MOTOR_STOPPED	
Subsystem 1	Device 1	Func Block 2	inv_pres_set_H	#	1000	
Subsystem 1	Device 1	Func Block 2	inv_pres_set_L	#	200	
Subsystem 1	Device 1	Func Block 2	inv_analog_out	#	#	
Subsystem 1	Device 1	Func Block 2	inv_pres_H	#	1000	
Subsystem 1	Device 1	Func Block 2	inv_pres_L	#	200	
Subsystem 1	Device 1	Func Block 3	inv_motor_state	#	MOTOR_STOPPED	
Subsystem 1	Device 1	Func Block 4	inv_motor_state	#	MOTOR_STOPPED	
Subsystem 1	Device 1	Func Block 4	inv_motor_set	#	MOTOR_STOPPED	
Subsystem 1	Device 1	Func Block 4	inv_pres_set_H	#	#	
Subsystem 1	Device 1	Func Block 4	inv_pres_set_L	#	#	

그림 7. LonMaker Browser에 의한 네트워크 변수 모니터링

4.3 실험 시스템 구성 및 고찰

(1) Host 마이크로프로세서의 기능 중 10Bit의 아날로그 입력을 받을 수 있는 포트가 내장이 되어 있어 이를 이용하여 아날로그 입력을 압력센서로부터 출력되는 아날로그 값 (0~5V)을 직접 받아서 1024로 분할, 디지털신호로 변환했다.

(2) 압력의 상한값과 하한값을 외부의 Key Pad로 설정하거나, LonMaker for Windows의 툴에서 설정할 수 있다. 상한값과 하한값 내에서 디지털신호를 아날로그 신호(0~10V)로 변환하여 인버터의 입력포트로 입력하여 Motor 속도를 지령하게 된다.

(3) 압력이 하한값에 가까워지면 모터펌프의 회전수는 높게 되고, 압력이 상한값에 가까워질수록 모터펌프의 회전수는 낮아지게 된다.

(4) 그림 33은 압력값에 따른 출력값을 비교한 그림이다. 아날로그 입력값에 따른 아날로그 출력 값을 전압(0~10V)으로 변환하여 측정하였다.

그림 8은 상한값과 하한값을 Key Pad나 LonMaker for Windows에서 설정하고 압력센서에서 들어오는 아날로그값을 Microprocessor에서 입력을 받아 보정하여 이에 상응하는 아날로그 출력값을 내보내는 순서도이다.

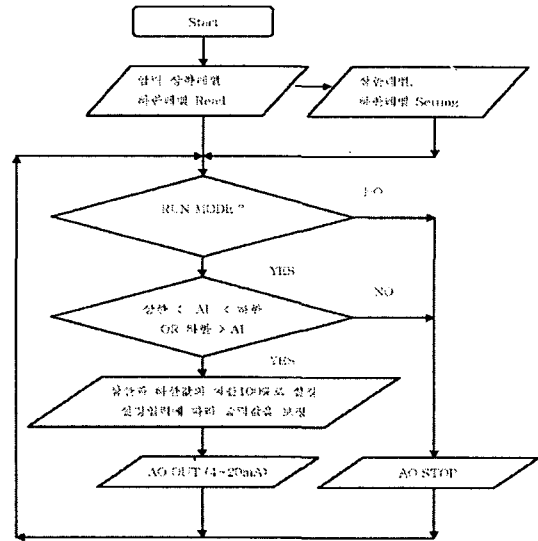
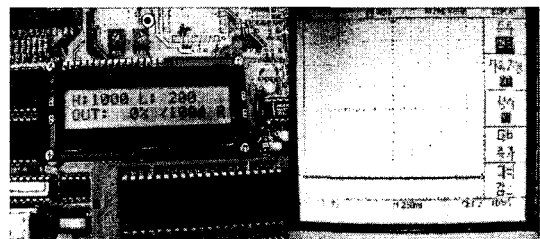
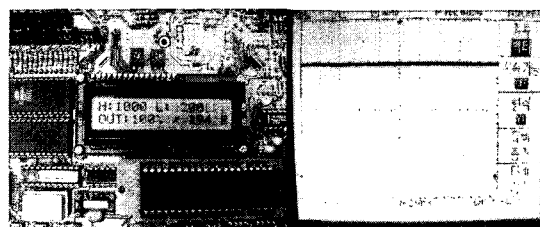


그림 8 아날로그 출력값을 내보내는 순서도

그림 9는 압력센서를 통한 브스터펌프의 회전속도의 변화를 주기위한 인버터 입력변화를 나타낸 것이다. 입력값이 상한값인 경우 인버터입력은 0[V]로 속도변화가 일어나지 않고, 그림 (b)는 압력센서의 값이 하한값으로 인버터의 입력에 10[V]를 주고 있어 속도가 가장 빠르게 회전하도록 안정적으로 운전하는 것을 보여주고 있다. 표 1은 실험 결과를 나타낸 것으로 전 영역에서 선형적으로 출력을 내고 있는 것을 알 수 있다.



(a) 압력값이 상한값 이상일 때 출력값 0V



(b) 압력값이 하한값 이하일 때 출력값 10V

그림 9 압력값에 따른 출력값의 모니터링 비교

표 6. 압력값에 따른 아날로그 출력값과 모터 속도 비교

상한압력 설정값	하한압력 설정값	압력 입력값	출력값 (V)	모터속도 (rpm)
1000	200	1004	0	0
1000	200	807	2.5	425
1000	200	602	5	850
1000	200	405	7.5	1275
1000	200	194	10	1700

5. 결론

본 연구에서는 Echelon사의 ShortStack을 이용하여 Microchip사의 PIC16F877 범용프로세서를 LonWorks 네트워크에 적용할 수 있도록 고성능 제어모듈을 개발하고 이 성능을 평가하기 위하여 부스터 펌프에 적용되는 유도전동기의 속도제어 시스템을 구축하여 실험한 결과 다음과 같은 결론을 얻었다.

(1) LonWorks Network기술을 산업계에 광범위하게 활용하기 위하여, LonWorks 네트워크에 LonTalk 통신을 하는 Neuron Chip과 Microchip사의 PIC16F877 범용프로세서인 Micro Processor를 소프트웨어적으로 연결하는 ShortStack을 이용하여 LonTalk 통신이 가능한 범용 고성능 제어모듈을 개발하였다.

(2) 개발된 고성능 LonWorks제어 모듈의 성능을 평가하기 위하여 부스터 펌프 시스템에 적용될 수 있는 Pump Control 알고리즘을 이용하여 시스템을 구축하였고 압력센서에서 입력되는 아날로그 출력값에 따라 인버터 구동시스템을 통하여 모터의 속도가 선형적으로 변화됨을 확인하였다.

앞으로 다양한 범용프로세서에 적용할 수 있는 알고리즘을 개발하고 모터의 속도를 피드백하여 이를 통하여 보다 정교한 시스템 제어에 사용할 수 있도록 연구를 진행할 예정이다.

Acknowledgment

The authors would like to gratefully acknowledge the financial support of KESRI (Korea Electrical Engineering & Science Research Institute) and ETEP of KEPRI under the project number A3050.

참고 문헌

- [1] 임구현, "Web Browser를 활용한 LonWorks제어·감시 시스템 구축에 관한 연구", 한밭대학교 석사학위 논문, 2002.
- [2] 백우진, "지능 분산제어를 위한 LON시스템 구축", 울산대학교 석사학위 논문, 1998. 12.
- [3] "론웍스의 이해와 응용(최종보고서)" 고려대학교 정보통신 기술 공동 연구소, 2001. 9.
- [4] Echelon Co, "LonMaker for Windows User's Guide", 2003.
- [5] Echelon, "ShortStack User's Guide", 2002.
- [6] Microchip, "16F877 Data Sheet", 2003.
- [7] Microchip, "H_TECH_PICC piclite User's Guide"
- [8] D. Loy, "Open Control Networks", Kluwer Academic Publishers, pp.217-229, 2001.
- [9] VBoard, "Networking and Integration of Facilities Automation Systems", CRC Press, pp.131-138, 1999.
- [10] M.J. Coffin, "Direct Control for Building HVAC System", John Wiley & Sons, Inc. pp.161-209, 1991.
- [11] Echelon, "LonBuilder User's Guide", 1999.
- [12] Echelon, "Neuron C Programmer's Guide", 2001.
- [13] Echelon, "Neuron C Reference Guide", 2001.
- [14] Echelon, "NodeBuilder User's Guide", 2001.
- [15] Echelon, "LonMaker for Windows User's Guide", 2002.
- [16] Motorola, "Neuron Chip Data Book", 1999.
- [17] Echelon, "Resource Editor Users Guide", 2001.
- [18] Echelon, "LonWorks Training", 1999.
- [19] Echelon, "LonWorks Network Design", 1999.