

지능적 필드버스 Master/Slave 네트워크 모니터링 시스템

Intelligent Fieldbus Master/Slave Network Monitoring System

조영임* · 심재홍**

평택대학교 컴퓨터과학과 · 한국산업기술대학교 제어계측공학과

Young Im Cho · Jae Hong Shim

*Dept. of Computer Science, Pyongtaek University

**Dept. of Control and Measurement Engineering, Korea Polytechnic University

E-mail : yicho@ptuniv.ac.kr, jhshim@kpu.ac.kr

요 약

최근 반도체 물류자동화 시스템 또는 발전소, 제철소와 같은 대단위 프로세스 자동화 시스템 등에서 센서 및 계측 단말기로부터의 계측 데이터를 산업용 Ethernet/IP 또는 블루투스와 같은 상위네트워크를 통해 원격지에서 제어계측 할 수 있도록 하기 위한 필드버스 기반의 산업용 유무선 계측 시스템 기술의 연구가 활발히 진행되고 있다. 그러나 Ethernet/IP라는 산업용 프로토콜 기반이 상용화될 경우 웹 환경에서 시스템에 독립적인 Master/Slave간 지능적 모니터링 시스템의 개발은 매우 필요하다.

따라서 본 논문에서는 시스템 독립적으로 Master/Slave 네트워크 모니터링 소프트웨어 및 통합 운영프로그램 개발하여 실시간으로 모니터링 할 수 있는 시스템을 개발하여 사용자들이 쉽게 사용할 수 있도록 하였다. 본 논문에서 구현한 환경은 윈도우 기반의 Java를 이용하였다.

1. 서론

전세계적으로 공장자동화, 공정자동화, 빌딩자동화를 위한 DCS(Distributed Control System) 시스템과 전력, 가스, 환경설비를 위한 SCADA(Supervisory Control And Data Acquisition) 시스템 및 의료, 교통 등의 다양한 분야의 자동화 설비에서 개방형 통신망 기술의 사용이 크게 확산되고 있다. 이러한 추세는 자동화시스템이 대형화하고 기능이 복잡화함에 따라 멀티제조사들의 자동화요소들 간에 상호 동작성을 보장하여 시스템 통합이 용이하도록 하며, 시스템의 유연성, 확장성, 유지보수성을 제고시키는 표준화 및 개방형 통신망의 요구가 증대되고 있기 때문이다[1].

또한 인터넷이 급속도로 보급됨에 따라 일반 사무환경에서는 Ethernet이 여러 컴퓨터를 연결하는 가장 대표적인 매체로 자리 매김하고 있다 [2]. 국내의 경우 필드버스를 기본으로 하는 산업용 네트워크에 있어서 실시간성을 요하는 제어기

나 드라이버를 제외하고는 간혹 Ethernet이 사용되고 있다. 가장 대표적인 것으로 필드버스와 Ethernet신호를 상호 변환하는 프로토콜 변환기나, PLC[3]등의 전용장비에 내장하여 사용되는 전용 Ethernet 모듈 등이 출시되고 있다. 그러나 Ethernet은 사무실 환경에서 시작되었기 때문에 많은 부품들이 산업현장에 적합하지 않게 성능이 떨어진다. 이는 Ethernet이 단순 변환의 과정을 거치거나 표준화 등이 되어 있지 않아 성능에 제한적일 수밖에 없고, 시간에 제한을 받지 않는 제어 명령이나 계측 등의 모니터링에 한정적으로 사용할 수밖에 없기 때문이다.

따라서 이러한 문제를 해결하기 위해 산업 현장인 공장에서도 Ethernet을 기반으로 하여 산업용 개방형 통신 네트워크를 구축하기 위한 연구가 진행 중에 있으며 그 결과 제안된 프로토콜이 그림 1의 Ethernet/IP(Industrial Protocol)이다[4]. Ethernet/IP는 TCP/IP와 UDP/IP상에서 구동되는 산업용 자동화 프로토콜로 정의된다.

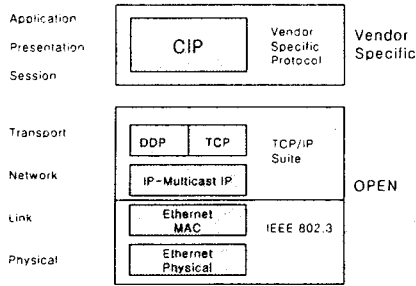


그림 1. Ethernet/IP

EtherNet/IP는 TCP/IP로 캡슐화된 Ethernet에 공통된 응용계층을 적용한 Ethernet TCP/IP의 산업화된 확장이다. TCP/IP 캡슐화란 Ethernet 메시지상에서 DeviceNet 노드가 이 메시지를 데이터로 캡슐화하게 한다. 그리고 노드가 그 메시지(메시지를 내장한 TCP/IP 프로토콜)를 Ethernet 통신 칩인 데이터링크층으로 보낸다. 자동화를 구현하기 위해 표준응용계층이 Ethernet상에 있는 자동화기기 및 제어 장치들간의 상호 동작성과 상호 호환성을 구현한다.

최근에는 계측 데이터를 산업용 Ethernet/IP 또는 블루투스와 같은 상위네트워크를 통해 원격지에서 제어계측 할 수 있도록 하기 위한 필드버스 기반의 산업용 유무선 계측 시스템 기술의 연구가 활발히 진행되고 있다. 그러나 Ethernet/IP라는 산업용 프로토콜 기반이 상용화될 경우 웹 환경에서 시스템에 독립적인 Master/Slave간 지능적 모니터링 시스템의 개발은 매우 필요하다.

따라서 본 논문에서는 EtherNet/IP기반의 개방형 네트워크에서 사용자의 편의성에 중점을 둔 GUI를 통해서 실시간으로 센서들의 데이터를 계측하고 모니터링하며, 사용자로부터 구성정보를 받아 각 서버들의 네트워크 파라미터를 구성할 수 있는 시스템을 개발하고자 한다.

본 논문의 구성은 다음과 같다. 2장은 Master/Slave 모니터링 시스템에 대해서 설명하고, 3장은 실제 사용방법을 설명하며, 4장은 성능평가를 하고 5장에서 결론을 맺고자 한다.

2. Master/Slave모니터링 시스템개요

본 논문에서 구현하고자 하는 EtherNet/IP Master/Slave 네트워크 모니터링 시스템은, EtherNet/IP 통신 프로토콜을 이용하여, EtherNet/IP 디바이스와 Master와의 통신으로 각 Slave에서 보내어지는 전류, 전압, 온도의 데이터를 Master가 메모리 맵에 저장하고, 이 맵으로부터 EtherNet/IP 구성자(Configurator)가 그 맵을 통하여 각각의 데이터를 추출하여 전압, 전류, 온도의 I/O 모니터링하고 Slave를 구성하는

시스템이다.

본 논문에서 구현한 시스템의 특징은 그림 2와 같다. 각 제조사로부터 EDS(Electric Data Sheet) 파일을 받아서 자동으로 디바이스 구성을 설정하고 디바이스 연결이 해제되었거나 문제 발생시 실시간으로 모니터링하며, I/O 모니터링 정보 요청시 사용자에게 보여줌으로써 GUI방식으로 설계하였다. 본 논문에서는 지능적 에이전트 시스템에 의한 네트워크 모니터링 및 트래픽 모니터링 시스템을 구현한 점이 특징이다.

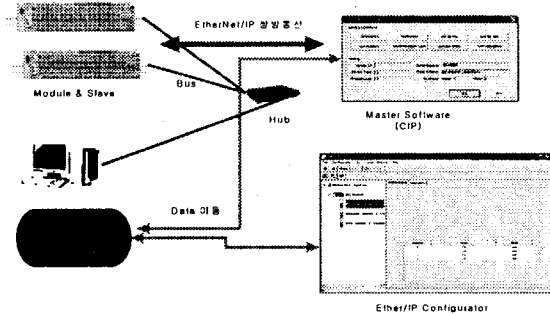


그림 2. Master/Slave 네트워크모니터링 시스템

그림 2의 시스템 각 구성 모듈을 설명하면 다음과 같다. ① Master소프트웨어는 Ethernet/IP와 DeviceNet이 공유하는 CIP(Control and Information Protocol)[5]를 소프트웨어로 개발하여 EtherNet/IP와 TCP, UDP를 통하여 서로 통신을 할 수 있게 해주는 시스템이다. Master는 공유메모리의 데이터를 가지고 TCP/IP통신을 통하여 디바이스와 연결하고 UDP통신을 통해서는 디바이스에 명령(Set, Reset, End)을 한다. Master는 Ethernet에 연결된 디바이스 각각의 센서에 대한 데이터를 공유메모리에 저장한다. ② 디바이스 구성자(Device Configurator 또는 EtherNet/IP Configurator)는 CIP에 의해 획득된 데이터를 통하여 모듈의 구성을 설정한다. 또한 모듈과의 연결이 되었으면 CIP를 통하여 획득된 데이터를 바탕으로 Slave와의 연결여부를 결정하고 어떠한 Slave와 연결되어 있는지 통신이 가능한지를 획득된 데이터를 통하여 표현한다. 이때 네트워크에 접속하는 방법은 IP를 통하여 디바이스에 연결된 각 모듈과의 연결을 설정한다. 각 디바이스는 각각 고유의 IP를 가지고 있거나 사용자가 임의로 결정하거나 고정된 방식을 선택하여 EtherNet/IP 통신프로토콜을 구성하게 된다. 네트워크의 접속은 BooTP[6]를 사용하여 접속하게 한다. BooTP란 디스크가 없는 시스템이 부트스트랩시 자신의 IP주소 등의 정보를 알기위해 사용하는 프로토콜로서 주로 UDP를 사용한다. 디바이스와 Slave의 접속은 필드버스를 통하여

연결되어 각 모듈은 각각의 데이터를 통합하여 EtherNet/IP를 통하여 데이터 사용자에게 보내어진다. ③ 공유메모리는 하나의 메모리 파일을 만든 다음, 이 메모리 파일을 두개 이상의 모듈에서 공유하여 데이터를 서로 전달하고 받을 때 사용한다. 메모리를 파일처럼 열고 한쪽에서는 데이터를 기록하고 한쪽에서는 데이터를 읽는 방식으로 데이터를 공유한다. 서로 다른 모듈이 메모리를 공유하는 서로 다른 프로그램에서 데이터를 전송하고 전송받기 위한 방법이다(그림 3).

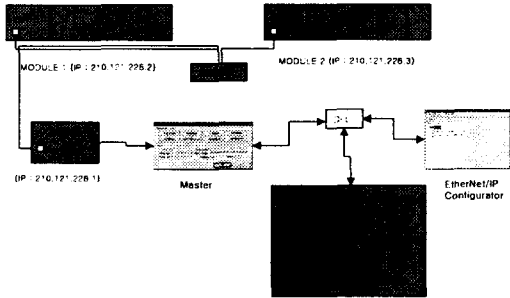


그림 3. 메모리 맵 연결구조

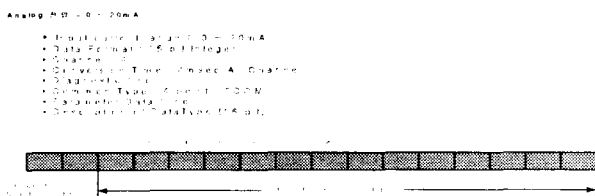
<표 1>에서와 같이 CIP Master 슈퍼바이저와 EtherNet/IP 구성자는 공유메모리를 통하여 서로 데이터를 주고받는다. CIP Master 슈퍼바이저는 C++로 작성되어 있고, EtherNet/IP 구성자는 운영체제와 독립적으로 수행위해 자바로 작성되어 있으므로 서로 다른 언어와의 통신을 위해서 이 공유메모리는 Visual C++을 통하여 DLL로 구성하였다.

표 1. 공유메모리 주소구조

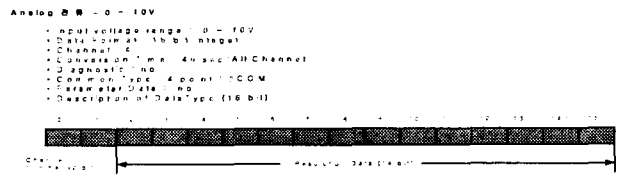
0000h 0FFh	4096 bytes (4 Kbytes)	Output Process Image	Host Master Bus
1000h 1FFh	4096 bytes (4 Kbytes)	Input Process Image	Bus Master Host
2000h 23Fh	1024 bytes (1 Kbytes)	Send Mail Box	Host Master
2400h 27Fh	1024 bytes (1 Kbytes)	Receive Mail Box	Master Host
2800h 29Fh	512 bytes 4byte(32bit)*128	Slave IP Addresses	Max slave:128
2A00h 2BFh	512 bytes	Slave Output Configuration Table	
2C00h 2DFh	512 bytes	Slave Input Configuration Table	
2F00h 2F0Fh	16 bytes	Parameterized Status	
2F10h 2F1Fh	16 bytes	Connection Status	
2F20h 2F2Fh	16 bytes	Alarm Status	

본 논문에서 사용한 I/O 디바이스에서의 Slave가 표현하는 전압, 전류, 온도의 표현범위는 다음 그림 4와 같다.

① 전압: 0 ~ 20 mA - 4 ~ 20 mA



② 전류: 0 ~ 10V - -10 ~ 10V



③ 온도: RTD - Thermocouple

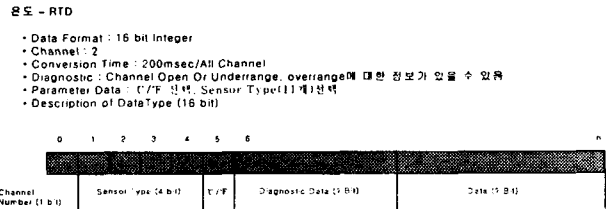


그림 4. 전압, 전류, 온도의 표현범위

CIP Master 슈퍼바이저는 각 모듈과 CIP프로토콜을 통하여 접속 및 통신을 총괄한다. 컴퓨터와 각 모듈이 Ethernet 허브(TCP/IP 통신)와 연결되어 있고 EtherNet/IP 구성자를 실행하기 전에 이 통신이 이루어져 있게 먼저 실행되어져 EtherNet/IP 구성자에서 설정 즉, 공유메모리에 기록된 IP를 통하여 EtherNet/IP(CIP) 연결한다.

통신이 이루어지면 이 Master는 공유메모리에 통신이 이루어지는 각 데이터를 기록하고 EtherNet/IP 구성자에 의해 명령되어지는 Connection, Reset 등의 데이터를 공유메모리에서 읽어 모듈에 CIP통신을 통하여 전달한다.

3. 모니터링 시스템의 실제사용방법

본 절에서는 필드버스 기반의 시스템에서 실제 사용방법을 단계별로 설명하고자 한다.

- ① 프로그램 시작: 네트워크 모니터링 시스템을 동작시키기 전에 Mater를 먼저 실행해야 한다.
- ② Slave 설정: 통신 어댑터를 이용하여 Slave에 대한 설정을 한다.
- ③ 슬롯 설정: 해당모듈에 연결되어진 Slave의 제품을 선택한다. 제품 선택이 정확치 않으면 I/O 모니터링에서 올바른 값을 보기 어려워진다(그림 5).

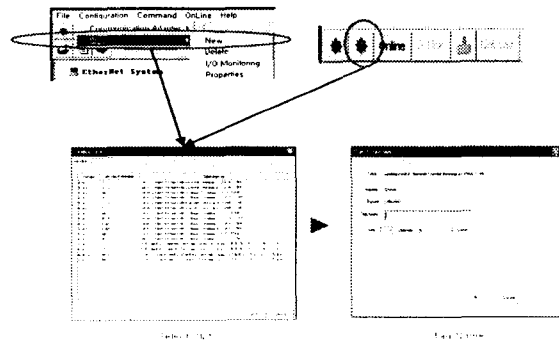


그림 5. Slave 및 슬롯 설정

④ 공유메모리 저장: 구성이 완료되면 정보를 공유메모리에 저장한다. 공유메모리를 할당하고 저장하는 기능이 온라인과 다운로드 기능이다. 온라인은 공유메모리를 생성하는 기능이며, 온라인 후 다운로드 버튼을 눌러 구성정보를 공유메모리에 쓰거나 공유메모리내 데이터 위치를 결정하게 된다. 다운로드 기능이 실행되고 나면 자동적으로 모듈과의 통신연결 여부를 체크하게 된다. 통신이 이루어져 있으면 I/O 모니터링기능과 각 모듈의 상태 및 Slave의 정보를 사용자에게 나타낸다. ⑤ I/O 모니터링: Menu - Configuration - Slot - I/O Monitoring을 선택하거나 모듈트리에서 직접 더블클릭 하면 그림 6과 같이 I/O 모니터링이 실행된다. 이때 데이터를 다양하게 그래픽하게 보여주는 기능을 갖는다.

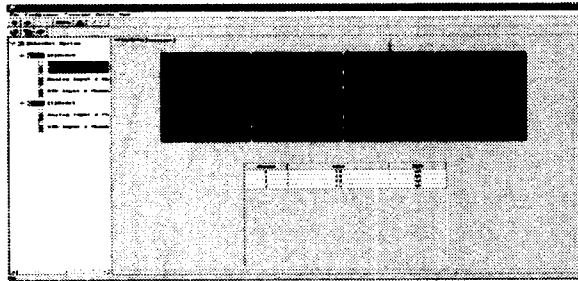


그림 6. I/O 모니터링 실행화면

본 논문에서 시스템 내부에 과부하가 발생하는 경우에 대한 처리를 위해 지능적 에이전트 개념을 도입하여 시스템을 구성하였다. 에이전트는 자동적으로 수행하는 소프트웨어인데[7], Master/Slave간 네트워크상에서 센싱된 데이터(전압, 전류, 온도)의 데이터를 실시간으로 표현하고, 데이터를 폴링하는 역할을 자동적으로 수행한다. 또한 과부하된 데이터는 그림 7과 같이 사용자의 우선순위에 따라 모니터링 한다. 사용자의 우선순위는 시스템 내부에서 사용자가 선택하도록 하였다.

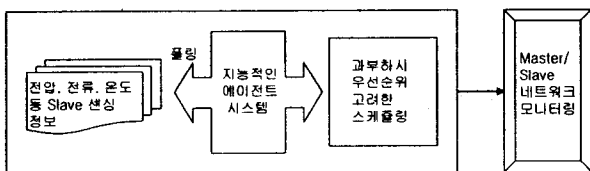


그림 7. 지능적 에이전트에 의한 모니터링 과정

4. 성능평가

본 논문에서 구현한 지능적 Master/Slave 네트워크 모니터링 시스템의 성능평가를 위해 다음 그림 8은 지능적 에이전트를 사용했을 경우와 에이전트가 없을 경우의 모니터링 정도(%)를 비교

한 것이다. 모니터링 정도(%)는 실시간적으로 모니터링되는 정도를 시간에 따른 수치로 나타낸 값이다. 시뮬레이션 결과, 시간에 따라 에이전트가 없을 경우는 Slave의 개수가 증가할수록 센싱할 데이터량이 많아서 네트워크 과부하가 발생하여 데이터 전송율이 불규칙하게 되므로 모니터링이 제대로 안될 경우가 많으나, 에이전트에 의한 방법은 사용자 우선순위를 고려하여 처리하므로 항상 일정한 모니터링 정도를 나타내고 있다.

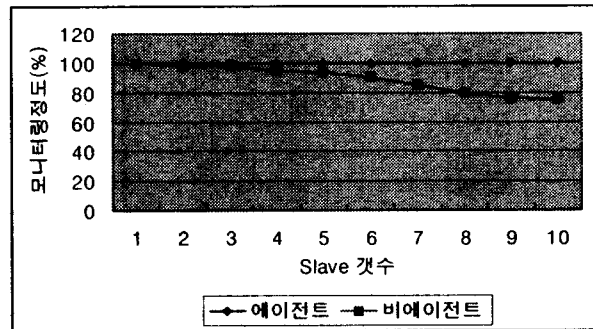


그림 8. 에이전트에 의한 모니터링정도 비교

5. 결론

본 논문에서는 최근의 산업용 Ethernet/IP 사용 경향에 따라 Master/Slave간 지능적 네트워크 모니터링 시스템을 개발하였다. 실험결과, 지능적인 에이전트에 의한 방법이 에이전트를 사용하지 않았을 때보다 모니터링정도가 매우 높은 것을 알 수 있었다.

앞으로 사용자 우선순위의 학습방법을 이용함으로써 좀더 지능적이며 자율적인 모니터링 시스템을 개발해야 한다. 또한 좀더 사용자 중심의 다양한 그래픽 기법을 도입해야 할 것이다.

6. 참고문헌

- [1] William A. Shay, Understanding Data Communications and Networks, PWS Publishing Company, 1995
- [2] W. Richard Stevens, TCP/IP Illustrated, Volume1 - The Protocols, Addison Wesley, 1997
- [3] 원태현외 6인, PLC 제어기술, 제2판, 복두출판사, 2001
- [4] Paul Izzo, Gigabit Networks, John Wiley & Sons, 2000
- [5] 조익영, 개방형 필드버스, 제어계측, 2003.5
- [6] <http://dwseo.x-v.net/bootp.htm>
- [7] Stuart Russel, Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall International Editions, 1995