

## 실시간 운영체제를 이용한 Embedded Network Adapter Card 개발

김희승, 신두진, 박익동, 허욱열  
인하대학교 전기공학과

### Development of Embedded Network Adapter Card with Real Time Operating System

Kim, Hi Seung Shin, Doo Jin Park, Eik Dong and Huh, Uk Youl

Dept. of Electrical Engineering, Inha University, Incheon, 402-751, Korea  
Tel: +82-32-860-7394 Fax: +82-32-863-5822 E-mail: himi@netsgo.com

**Abstract** - 제어시스템이 복잡해지고 다변화 되어짐에 따라 네트워크의 중요성이 부각되어지고 있다. 특히, 원격지 제어시스템의 정보를 실시간으로 수집, 분석함은 물론 제어할 필요가 있다. ENA(Embedded Network Adapter) 카드는 원격지에 있는 대상 제어시스템의 정보를 수집 및 명령을 전달하기 위한 장치이다. 이 카드의 개발은 네트워크의 발달로 인한 다양한 인터페이스 요구를 만족시키기 위하여 개발되었으며, 기존의 장치에는 없는 기능인 실시간 운영체제를 탑재하였으며, 우선순위에 의한 방법과 Round-Robin에 의한 방법으로 각각의 TASK들을 관리하도록 하였다. TCmini라는 PLC 상용제품에 적용실험을 통하여 그 성능을 입증하였다.

Memory는 Block을 분할하여 사용하므로 보드 동작 중에도 프로그램을 Update 할 수 있다. PLC와 통신은 Dual Port RAM(DPRAM)을 이용하여 명령 및 데이터를 송수신 한다. 모든 프로그램은 XINU 운영체제를 사용하여 개발하였다. [5][6]

### 1. 서 론

최근 급속한 산업발전으로 복잡해진 제어 시스템은 높은 안정성, 신뢰성 및 동시 다발적으로 발생하는 사건들을 제한된 시간 내에 처리가 요구되어짐과 동시에 제어 시스템에서의 네트워크 또한 매우 중요한 부분으로 부각되어지고 있다. [4]

특히, 원격지의 정보를 실시간으로 수집하여, 정보를 분석함은 물론 원격지 대상시스템을 실시간으로 제어하기 위하여 Embedded 네트워크의 필요하다. 이러한 Embedded 제어 시스템에서 시리얼 통신을 사용함에 있어 몇 가지의 문제점 있다. [1][2]

거리의 제한이 있다는 것과 다중 접속을 위해서는 별도의 통신용 제어기가 필요하며, 통신용 제어기들은 Daisy-Chain 방식의 구성이 이루어 있어야 한다. 이 중에서 원격지에서 대상 시스템을 제어 및 모니터링을 위해서는 거리의 제한성이 가장 큰 문제점이다.

시리얼 통신의 문제점을 보완하는 네트워크 방식은 이더넷을 이용하는 방법입니다. 이더넷을 이용하면 기존의 통신망인 인터넷을 활용 할 수 있습니다.

인터넷과 같이 많은 사람이 사용하는 공용망에서는 원격 제어가 실행되는 순간마다의 망 지연 시간(Network time delay)을 정확하게 계산할 수 없기 때문에 제어 대상 시스템의 오동작을 유발시킬 수 있다. 이러한 오동작을 극복하고자 실시간 운영체제를 갖는 ENA (Embedded Network Adapter) Card를 개발하게 되었다.

### 2. 본 론

#### 2.1 ENA Card 개발

ENA는 Programmable Logic Controller (PLC) 와 Server간에 Ethernet을 통한 통신기능을 제공한다. ENA와 Server간에는 Ethernet 통신이 연결되면 PLC와 ENA간에는 Dual Port SRAM (이하 DPRAM) 또는 RS232C Serial 통신을 통하여 연결된다.

MPC860 CPU를 사용하여 1개의 Ethernet Port와 2개의 Serial Port 제공하며, 4~16M Bytes의 Flash Memory를 Booting Device로 사용한다. 또한 Flash

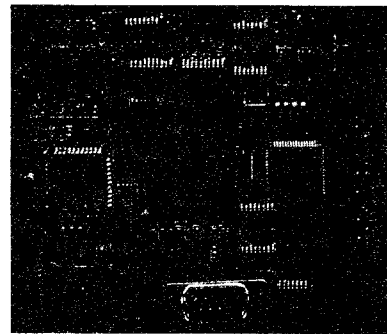


그림 1. ENA Card

#### 2.2 ENA Card의 하드웨어(Hardware)

ENA Card의 하드웨어의 구성은 그림 2.과 같으며, 구성품은 MPC860 CPU와 Flash 메모리, SDRAM, DPRAM의 3종류의 메모리를 사용하였으며, 통신 포트는 1개의 이더넷과 3개의 시리얼 포트로 구성되어 있다.

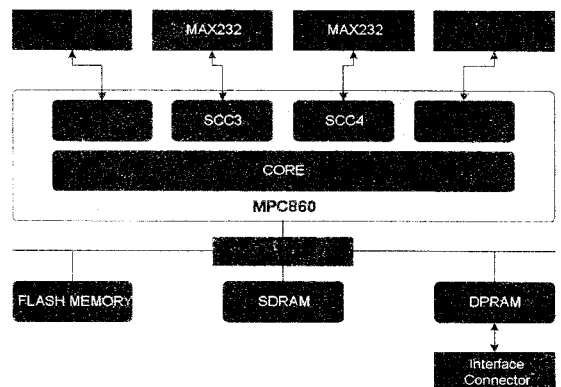


그림 2. ENA Card의 Hardware 구성도

#### 2.3 ENA Card의 소프트웨어(Software)

실시간 운영체제를 사용하여 다양한 통신방식을 지원 가능하도록 프로그램 하였으며, 프로그램의 저장 위치와 실행위치를 이중화하여 ENA Card 동작 중에도 실시간으로 프로그램 Update가 가능한 기능을 구현하였다.

또한 기존에 시리얼 통신만 지원하는 제어 대상 시스템을 수정 없이 사용하기 위하여, 시리얼 통신으로 입력된

데이터를 변환하여 이더넷으로 송신하고 이더넷으로 입력된 데이터를 시리얼 형태의 데이터로 변환하여 제어 대상 시스템에 송신하는 프로그램을 개발하였다.

FLASH 메모리에 저장되어 있는 설정 파라메타에 따라 각각의 포트 속성이 결정되며, 이 설정 파라메타는 사용자가 용도에 따라 변경이 가능하다. ENA Card의 보안성을 위하여 Time-Out에 의한 auto logout 기능과 패스워드 기능이 내장 되어있다. 또한 TCP, UDP, IP, FTP 프로토콜과 rlogin, telnet 터미널 프로토콜을 지원함은 물론 간단한 ping, arp 등 이더넷의 상태 확인을 위한 기능 제공한다. ENA Card와 LAN 상의 호스트 컴퓨터간에 remote login 통신 기능과 LAN 상의 호스트에서 ENA Card 자체를 원격 관리 및 제어 기능을 구현하였다.

위에서 언급한 설정 파라메타와 프로그램의 저장은 FLASH 메모리에 저장되며, 프로그램의 동작은 SDRAM에서 동작된다. 이와 같이 프로그램의 저장과 동작 매체의 이중화로 ENA Card가 동작 중에도 대상 제어 시스템의 동작을 중단함 없이 ENA Card의 프로그램을 Update를 할 수 있다. 소프트웨어의 구성은 TCP, UDP/IP 테스크를 비롯한 6개의 테스크로 구성되어 있다.

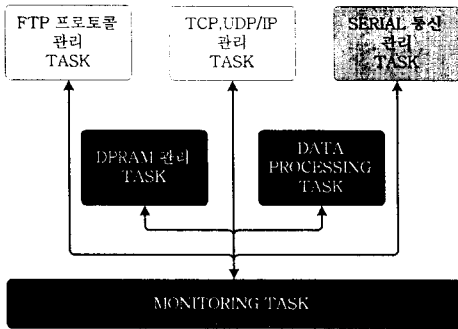


그림 3. 소프트웨어 구성도

### 2.4 실시간 운영체제에서의 사건처리 방법

동시 다발적인 사건에 대해 빠른 응답 시간과 제한된 시간 내에 사건 처리를 요구하는 시스템을 충족하는 운영체제를 실시간 운영체제라 한다.

범용의 운용체제에서는 순차적으로 프로그래밍 하는 것이 대부분인데 반해 실시간 운영체제는 각 사건의 데드라인(Deadline)과 우선 순위를 가지고 프로그램을 한다.

실시간 운영체제의 사건처리는 우선 순위에 의한 방법, 데드라인에 의한 방법과 두 방법을 상호 보완한 복합적인 방법으로 프로그램 한다.

우선 순위에 의한 방법에서는 각 사건에 우선 순위를 주고 높은 우선 순위 사건을 먼저 실행하며, 동일한 우선 순위에서는 Round-Robin 방식으로 사건을 처리한다. 그림 4.와 그림 5.에서는 이 방식들을 도식화 하였다.

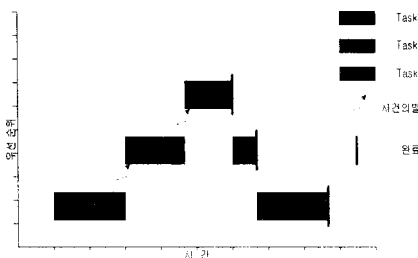


그림 4. 우선 순위 의한 사건 실행

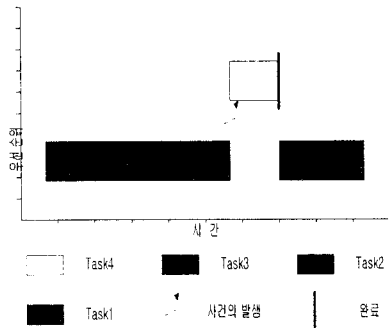


그림 5. Round-Robin 방식에 의한 사건 실행

그러나 우선 순위에 의한 프로그램은 우선 순위를 잘못 지정하면 시스템이 Down되는 현상이나 시스템에 오류가 발생할 수 있다. 이러한 문제점을 해결하기 위하여 그림 6.의 우선 순위 반전 방식과 그림 7.의 우선 순위 계승 방식을 이용하면 우선 순위에 의한 프로그램 시 발생할 수 있는 오류를 보완 할 수 있다.

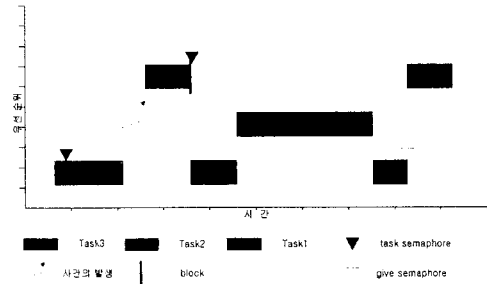


그림 6. 우선 순위 반전 방식

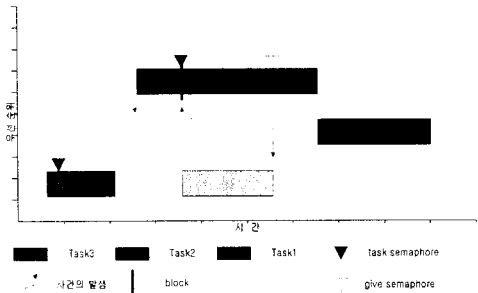


그림 7. 우선 순위 계승 방식

데드라인에 의한 방법은 각 프로세서에 데드라인 시간을 주어 각 시점에서 가장 가까운 데드라인 Time을 확인하여 그 프로세서를 먼저 처리하는 방법이다.

특정 기능 수행을 위해 생성되는 소프트웨어로서 스케줄링의 대상이 되는 실시간 테스크의 기본적인 시간제한은 그림 8.과 같다.

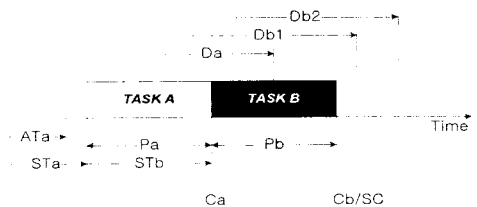


그림 8. Time constraint of real time task

$P_i$  : 테스크  $i$ 의 수행시간       $ST_i$  : 테스크  $i$ 의 스케줄링 시작시간  
 $Di$  : 테스크  $i$ 의 데드라인       $C_i$  : 테스크  $i$ 의 수행 종료시간  
 $SC$  : 테스크 집합의 종료시간     $Di/2$  : 테스크  $i$ 의 데드라인 1, 2  
 $ATI$  : 테스크  $i$ 의 도착시간

시간 제한 중  $Di/2$ 는 동일 테스크의 데드라인 긴급도를 나타낸다.

하드 실시간 시스템상의 시간 제한은 시스템상의 시간 제한은 최악의 경우를 고려하여야 하므로 수행시간은 예상되는 최대의 요구시간, 데드라인은 가장 긴급한 경우를 고려하여야 한다.

$$0 \leq AT_i \leq ST_i \leq Di - C_i$$

테스크의 데드라인은 그 발생성격에 따른 구분이 가능하다.

그러나, 발생가능 테스크의 주기 혹은 비주기 구분은 개념적이며 테스크를 모두 주기적인 테스크로 보는 것보다 비주기적인 테스크로 보는 것이 동적인 환경에 보다 적합하며 전자의 경우 비주기적인 테스크의 처리가 어렵다.

주기적인 테스크의 데드라인은 아래와 같이 정의될 수 있다.

$$\begin{aligned}
 A(i+1) &= D(i) \\
 D(i+1) &= A(i+1) + P \\
 A(i) &: \text{주기적 테스크의 } i\text{번째 도착시간} \\
 D(i) &: \text{주기적 테스크의 데드라인,} \\
 P &: \text{주기}
 \end{aligned}$$

비주기 테스크의 시간제한은 작업분석, 전용 소프트웨어 등으로 획득이 가능하다.

### 2.5 실험 및 고찰

ENA Card와 PLC의 실시간 인터페이스 실험을 통해 ENA Card의 하드웨어 및 소프트웨어의 적합성을 확인해 보았다.

PLC(Programmable Logic Controller)란, 종래에 사용하던 제어반 내의 릴레이 타이머, 카운터 등의 기능을 LSI, 트랜지스터 등의 반도체 소자로 대체시켜, 기본적인 시퀀스 제어 기능에 수치 연산 기능을 추가하여 프로그램 제어가 가능하도록 한 자율성이 높은 제어 장치로서 이 실험에서는 TCmini을 사용하였다.

TCmini는 일본의 도시바기계의 제품으로 PLC의 모듈화 되어있는 부분들을 하나의 보드에 집적하여 개발한 보드로 ladder 언어를 사용하여 프로그래밍이 가능한 미니 PLC보드이다. 이 보드는 간단한 제어 프로그램으로 제어가 가능한 다양한 산업현장에 활용되고 있다.

ENA Card와 TCmini는 다양한 인터페이스 중에서 DPRAM을 통한 인터페이스를 이용하였으며, 시스템의 인터페이스는 그림 9.에서 보여주고 있다.

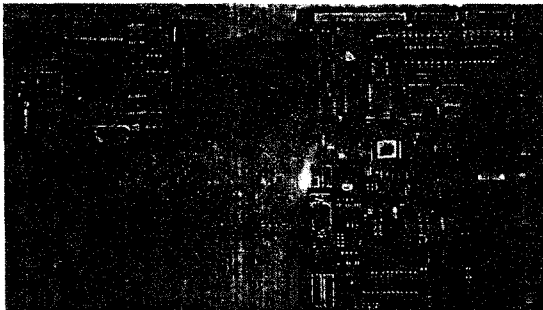


그림 9. ENA Card와 TCmini의 인터페이스

또한, TCmini와의 인터페이스 실험을 위하여 DPRAM을 System 관리영역, ENA / TCmini 인터페이스 관리 영역, ENA / TCmini 데이터 영역으로 구분하여 프로그램을 개발하였다.

이와 같이 개발한 프로그램의 실험은 그림 10.과 같은 구성을 갖는 시스템으로 구성하여 실험하였다.

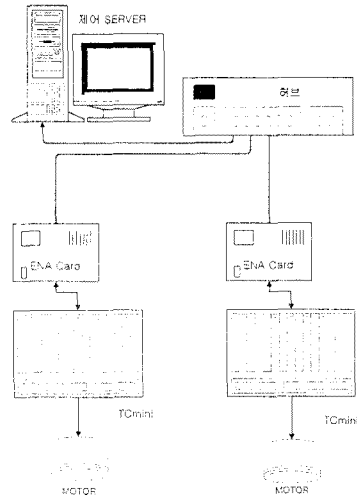


그림 10. 적용 실험 구성도

### 3. 결 론

본 논문에서는 복잡해지고 다변화된 제어시스템의 확장성과 빠른 응답성을 고려하여 ENA Card를 개발하였다.

ENA Card는 다양한 네트워크 지원으로 시리얼 통신만 지원하는 제어시스템은 물론 새롭게 제작되는 제어시스템까지도 네트워크 지원이 가능하다. 또한 실시간 운영체제를 적용함으로써 시간지연을 최소화하는데 중점을 두었다.

향후 연구는 실시간 운영체제를 적용하여 네트워크의 시간지연을 향상시킨 것에 대한 이론적인 연구가 필요하다.

#### [참 고 문 헌]

- [1] Phillip A. Lapante, "Real Time System Design And Analysis", IEEE Computer Society press, 1992
- [2] James F. Ready, Hunter & Ready Inc., "VRTX : A Real Time Operation System for Embeded Microprocessor Application", IEEE Micro, 1986
- [3] Jane W. S. Liu, "Real Time Systems", Prentice Hall, 2000
- [4] Stuart Bennett, M. J. Grimble "Real - Time Computer Control", Prentice Hall, 1994
- [5] Douglas Comer / Steven Munson, "Operating System Design Vol 1: The XINU Approach", Prentice Hall, 1989
- [6] Douglas Comer, "Operating System Design Vo 12: Internetworking with XINU", Prentice Hall, 1984
- [7] Wind River System, "VxWorks Programmer's Guide", 1993