

○ \* \* \*  
 김 동길                  장은영                  김정선  
 \* 한국항공대학교 항공전자공학과

A study on the Design and Implementation of Local Area Network  
 for Factory Automation

○ \* \* \* \*  
 Dong Kil KIM, Eun Young CHANG, Jung Sun KIM

\* Dept. of Avionics Eng., HanKuk Aviation Univ.

**ABSTRACT :** It has discussed the hardware and software structure of LAN controller aquired in tranfer robots applied to factory automation. And through the design and implementation, the sharing of communications and resources among robots and CNC machines etc has been investigated.

1. 서론

사회의 다변화에 따라 대량의 정보들이 산출되었고 정보처리의 중심이 되는 컴퓨터와 통신기술의 발전과 함께 정보처리의 다양한 요구가 생겨나게 되었다.

이러한 현상들을 배경으로 하여 정보화 사회의 실현을 위한 필수적 단계중의 하나로 근거리 통신망(LAN: Local Area Network)<sup>(1)</sup>이 출현하게 되었으며 통신제어 프로그램의 정비, LAN 응용프로그램의 개발 및 이 기종간의 호환성문제로 LAN의 보급이 지연되어 오다가 최근 여러 표준화 기구들로부터 LAN의 표준안들이 제안되면서 그동안 널리 보급되어온 개인용 컴퓨터의 접속과 함께 LAN이 확산되기 시작하였다 (1)(2).

본 연구는 최근 무인화 공장을 목표로 원자재의 운반, 공급 및 가공품의 토출등에 많이 적용되고 있는 트랜스퍼 로봇에 LAN 컨트롤러를 설계, 시험제작하여 로봇 및 공장기계간의 통신과 자원의 공유 및 앞으로의 FMS(Flexible Manufacturing System)에의 적용을 검토하기 위한 것이다.

본 논문에서는 공장내 각각의 장치들을 호스트로 보고 이들을 실제의 네트워크에 접속할 수 있는 LAN Controller를 설계하고 시험제작하였다.

II. 시스템의 구성

1. 개요

전자산업의 발전으로 오늘날의 공장에서는 컴퓨터를 비롯하여 CNC(Computer Numerical Control), 산업용 로봇, PC(Programmable Controller), 인텔리전트한 계측기, 영상처리 시스템, CAD/CAM 장비, 통신기기등 수많은 첨단 장비들이 사용되고 있다. 이들 장비들은 그 자체가 하나의 컴퓨터이거나 대부분 마이크로 프로세서를 내장하고 있어 생산현장에서 필요한 정보를 수집하고 처리할 수 있다. 그러나 기계를 제어하고 공장을 개선하는데에 무한한 잠재력을 갖고있는 이들 장치들은 대부분 독립적으로 동작하고 있어 자동화의 고밀을 형성하고 있다.

이러한 환경으로부터 효율을 개선하고 다양한 생산요구

에 부응하려면 각각의 공정간에 일련의 연속성이 부여 되어야 하며 이를 위해서는 장비들 사이에 데이터통신을 가능하게 하는 네트워크가 구축되어야 한다. 그림 1은 이러한 네트워크의 예를 보여준다.

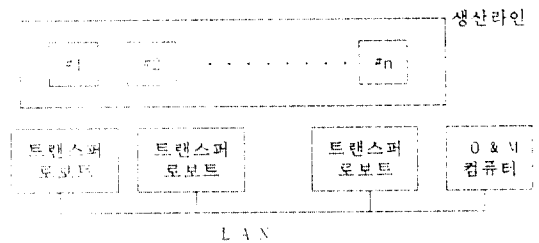


그림 1. 네트워크의 예

2. 설계 개요

LAN Controller는 각 호스트가 위와 같은 기능을 수행할 수 있도록 네트워크와 호스트의 중간에서 하드웨어 인터페이스를 제공하며 데이터 송수신을 수행한다. 본 연구에서의 LAN Controller 설계 개요는 다음과 같다.

첫째, LAN Controller와 접속하는 호스트의 부하를 줄이기 위해 Controller에는 CPU를 내장하며 호스트와 송수신을 담당하는 공유 RAM을 둔다.

둘째, LAN Controller에는 데이터 송수신 속도를 제한받지 않는 범위내에서 소프트웨어로 처리하며 모니터 ROM을 내장한다.

셋째, LAN Controller의 프로세서 절차 및 데이터 포맷은 Ethernet 사양을 참조한다.

3. 동작

LAN Controller의 동작은 네트워크 계층구조의 기능으

로서 설명할 수 있다.

본 연구에서 설계한 네트워크의 계층구조는 그림 2와 같다(7).

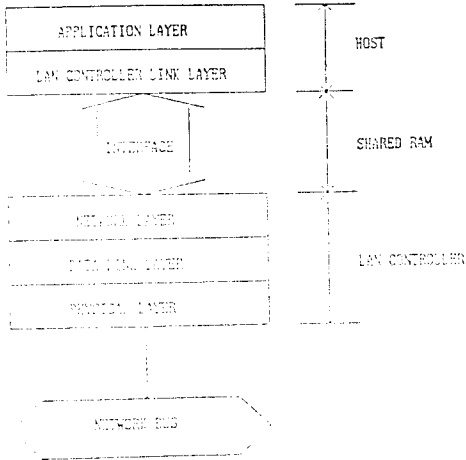


그림 2. 네트워크 계층구조

### 3.1. 물리 계층

네트워크를 지탱하는 최하위 계층으로 실제의 2진 데이터 0,1이 전송되며 모든 기능이 하드웨어로 실현된다.

논리적인 바이트 단위의 데이터와 네트워크상의 데이터 직렬, 병렬 변환을 행하며 네트워크 액세스를 한다. 그리고 현재 네트워크가 사용되고 있는 여부를 조사하여 네트워크를 다른 노드에서 점유하고 있으면 이를 상위계층에 알린다.

### 3.2. 데이터 링크 계층

LAN Controller의 송수신 버퍼에 있는 데이터 송수신을 행하며 에러의 검출 및 정정, 데이터의 논리적 단위인 프레임의 구분한다. 이 계층에서는 ISO 모델 제 2 계층의 표준안으로 채택된 HDLC(High Level Data Link Control) 프로토콜을 사용한다(6).

이 프로토콜의 프리미티브는 그림 3과 같으며 ADLC(Advanced Data Link Controller)가 수행한다. 주요 프로토콜 특성은 다음과 같다.

- Flag의 생성, 검출 및 삭제
- Zero "0"의 삽입 및 삭제
- FCS(Frame Check Sequence)의 생성 및 검출
- Abort 검출 및 전송
- Idle 검출 및 전송

#### (가) Flag의 생성, 검출 및 삭제

프레임의 경계를 구분하는 일정한 2진수의 패턴(01111110)으로 데이터가 ADLC의 Tx Stack에 기록되면 자동적으로 Opening flag가 생성되고 마지막 데이터가 기록된 후에 Closing flag가 생성되어 전송된다. Flag의 검출은 Flag 검출출력 또는 상태 레지스터의 상태비트에 의해 표시되며 프레임의 경계 및 각 Field를 구분할 뿐 Rx FIFO에 전달되지 않는다.

#### (나) Zero "0"의 삽입 및 삭제

Flag를 제외한 Frame 내부에서 연속적으로 1이 5개 나타나면 송신단에 의해 0이 삽입되며, 수신단에서는 연속적으로 1이 5개 수신된 다음에 나타나는 0을 제거한다.

#### (다) FCS의 생성 및 검사

HDLC 프레임에서는 16비트의 FCS를 가지며 순환잉여도 검사 CRC(Cyclic Redundancy Check)를 사용한다. 이 때의 송수신을 위한 생성 다항식은  $X^{16} + X^{12} + X^5 + 1$  이 된다. FCS의 생성, 전송, 체크는 ADLC 송수신단에서 자동적으로 수행되지만 FCS Field는 Rx FIFO에 전송되지 않는다.

#### (라) Abort 검출 및 전송

데이터 전송 도중에 처리가 종료됨을 Abort라 하며 최소한 8개의 1이 전송된 후 Frame을 Abort 한다.

수신단에서는 7개 또는 그 이상의 1이 연속되었을 때 Abort가 인식된다.

#### (마) Idle 검출 및 전송

Tx FIFO가 Frame을 전송하고 있지 않은 상태를 Idle이라 하며 Flag가 연속되는 Time-fill 상태와 1만이 연속되는 Mark idle 상태가 Idle state에서 선택된다.

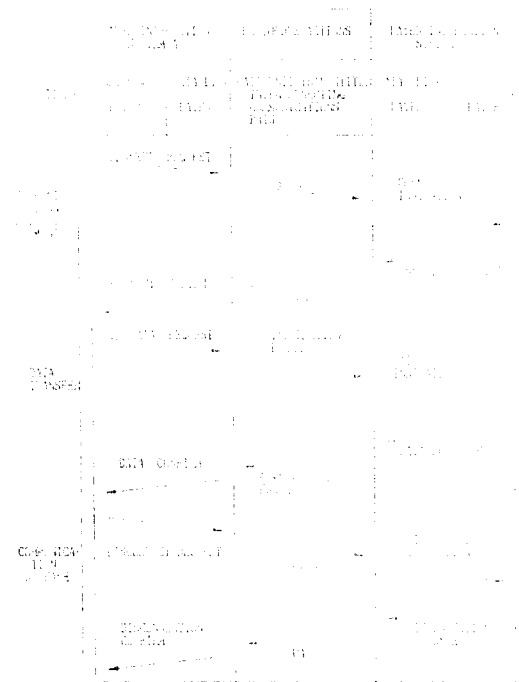


그림 3. HDLC 프로토콜의 프리미티브

### 3.3. 네트워크 계층

경로 탐색이 주 기능인 네트워크 계층은 노드 번호를 설정 하는 스위치에 하나의 어드레스를 할당해 놓고서 각 패킷마다 전송되어오는 목적지 주소를 비교하는 데이터그램 방식을 취한다.

이상 3개의 계층이 LAN Controller상에서 실현되며 상위의 계층은 호스트 컴퓨터가 Controller와의 공유 RAM을 이용하여 송수신 데이터를 처리한다.

3.4. LAN Controller 링크 계층

LAN Controller를 호스트에 인터페이스하는 계층이 된다. 호스트와의 데이터 송수신을 위하여 Controller상에 설계한 공유 RAM과 이를 사용하기 위해 각각 호스트와 Controller로부터 구동되는 인터럽트 발생회로가 하드웨어 구조를 이룬다. 그리고 이 공유 RAM에 데이터를 써넣으면서 상태 Flag를 발생시키고 이로부터 발생하는 인터럽트를 처리하는 루틴이 소프트웨어 구조를 이루며 호스트와 LAN Controller의 모니터 프로그램 내에 상주한다(8).

3.5. 응용 계층

호스트와 LAN Controller의 맨 상위 계층으로서 네트워크 서비스를 실행한다. 즉, 다른 노드와의 데이터 송수신을 수행함에 있어 최초 혹은 최종의 데이터가 상주하는 계층이 되며 모든 기능 및 제어가 호스트 컴퓨터의 ROM에 내장되어 있는 모니터 프로그램에 의해 수행된다.

11. 하드웨어 및 소프트웨어의 설계

1. 하드웨어

전술한 바와 같이 네트워크 계층을 갖고서 각각의 기능을 수행하는 LAN Controller의 하드웨어 블록 다이어그램은 그림 4와 같다. 이 하드웨어는 호스트 컴퓨터와의 공유 RAM과 이를 위한 제어부, 인터럽트 발생부, 데이터 링크 기능을 수행하는 ADLC와 송수신라인의 드라이버단, DMA부 및 CPU로 이루어진다. 물리계층의 기능을 수행하는 라인 드라이버단은 정류과 부극을 동시에 보내는 EIA RS-422 차동형 평형송수신 회로를 선택하였으며 트위스티드 페어 케이블을 사용하였다(3).

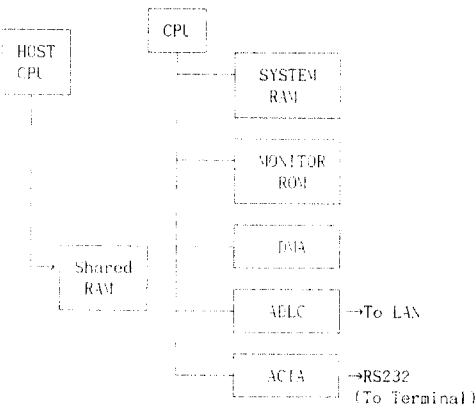


그림 1. LAN 컨트롤러 하드웨어 블록도

2. 소프트웨어

호스트 컴퓨터와 네트워크 사이에서 LAN Controller가 수행하는 소프트웨어는 주로 데이터링크 계층과 네트워크 계층이 실현하며 호스트와의 인터페이스를 위한 인터럽트 처리루틴과 ADLC의 초기화 프로그램이 포함된다.

위의 두 계층에는 각각 송신 프로세스와 수신 프로세스가 있으며 전체적으로는 GET-HOST, PUT-NETWORK, GET-NETWORK, PUT-HOST의 4개 프로세스로 구성된다. 그림 5는 LAN Controller 소프트웨어의 블록 다이어그램을 나타내고 있다(4).

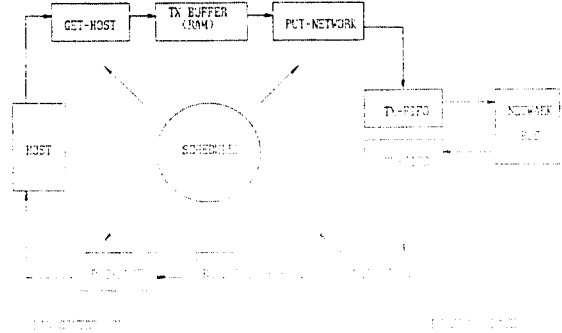


그림 5. LAN Controller 소프트웨어 블록 다이어그램

2.1. GET-HOST

응용계층의 서비스를 실행하는 호스트 컴퓨터의 응용 프로그램을 네트워크로의 전송에 앞서서 LAN Controller가 취하는 프로세스로서 호스트의 요구에 따라 이루어진다. 즉 호스트에 의하여 다른 노드로 전송될 데이터가 채워진 다음 인터럽트 발생회로로부터 요구되어 온 인터럽트를 처리한다. 호스트로부터 인터럽트를 수신하면 전송을 위한 데이터 임을 인식하고 공유 RAM의 데이터를 Controller의 시스템 RAM에 Move 한다. 이때 LAN Controller는 송수신 노드의 어드레스와 데이터의 길이를 기억하게 된다.

2.2. PUT-NETWORK

PUT-NETWORK는 호스트로부터 수신한 데이터 즉, 공유 RAM의 상위 2K 바이트에 있는 데이터를 LAN Controller의 시스템 RAM에 이동시킨 후 이 데이터를 목적지에 전송하기 위하여 네트워크를 점유하고 송출하는 프로세스로서 주 기능은 ADLC에서 이루어지며 DMA에 의해 데이터 전송이 수행된다.

먼저 ADLC를 초기화 한 후 Tx Reset이 해제된 후에 시스템 RAM의 데이터가 Tx FIFO에 로드될 때까지 Active 또는 Inactive Mark/idle 상태가 선택되어 계속된다.

Tx FIFO의 유용성은 TDRA(Transmit Data Register Available)상태 비트에 의하여 표시되며 이 상태는 Tx RS비트 또는 CTS입력에 의존한다. 그리고 TDRA가 High상태로 될 때 데이터의 전송이 가능해진다.

첫번째 바이트의 전송은 Frame Continue 어드레스에서 Tx FIFO로 로드되어야 하며 이에 따라 프레임 전송이 이루어진다. 이때 Tx Mark idle 상태이며 2-3개의 Tx 클럭에서 Opening flag를 발생한다.

Frame Continue 어드레스 데이터가 Tx FIFO에 계속 로드되는 동안에 하나의 프레임은 연속되고 Frame Terminate 어드레스에 마지막 데이터가 로드되면 이 데이터를 전송하고 FCS 필드와 Closing flag가 추가된다. 만일 Tx FIFO가 프레임용 전송하는 중에 Tx Shift 레지스터로 전송할 데이터가 없다면, Underrun 상태가 발생하고 Tx Abort에 의해 Tx가 종결된다. 그림 6은 PUT-NETWORK의 흐름을 보여준다.

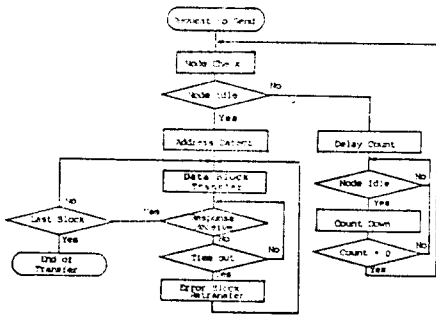


그림 6. PUT NETWORK 흐름도

2.3. GET-NETWORK

GET-NETWORK는 네트워크 상으로 데이터가 보내져 오면 프레임 동기를 취하고 \$8000번지로 설정되는 노드 확인 어드레스와 목적지 어드레스를 비교하여 어드레스가 일치하면 Rx FIFO의 상태에 따라 데이터를 수신하는 프로세서서 PUT-NETWORK와 같이 주 기능을 ADLC가 수행한다.

수신되는 데이터는 2진 비트의 연속되는 열로서 Abort flag 또는 Idle 조건없이 최대 1이 5개까지 발생한다. 한 Flag가 검출되면 Flag time에 프레임 동기를 설정하여 ADLC 내부 버퍼시간 만기이전에 프레임이 종결되면 프레임은 무시된다. 수신된 직렬 데이터는 Rx FIFO에 들어간다. CPU는 Rx 인터럽트 처리루틴으로부터 ADLC상태를 Read하고 RDA 또는 Address present로부터 수신데이터가 유효함을 확인한 후 수신 데이터를 읽어간다. 이후에 인터럽트와 상태비트는 자동적으로 Reset된다. GET-NETWORK의 흐름도는 그림 7과 같다.

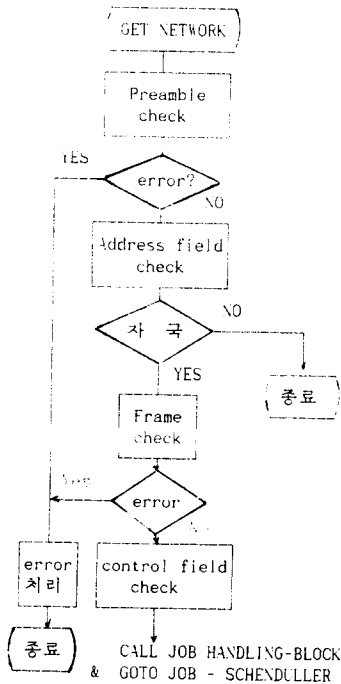


그림 7. GET NETWORK 흐름도

2.4. PUT-HOST

LAN Controller에 의해 네트워크로부터 수신된 데이터가 목적지 응용계층에서 수행될 수 있도록 호스트서 전송하는 PUT-HOST 프로세스는 GET-HOST 프로세스와 상대적인 절차를 갖는다. 호스트에 전송할 데이터가 LAN Controller의 RAM에 로드되면 공유 RAM의 하위 2K 바이트를 체크하고 \$00를 채운다음 이곳에 데이터를 전송한다. 그리고는 이를 호스트에 알리기 위해 인터페이스 레지스터에 인터럽트 벡터 \$81을 Write한다. 이에 따라 호스트는 공유 RAM의 하위 2K 바이트로 로드하여 실행하게 된다.

IV. 구성 및 실험

본 연구에서는 호스트머신 1대와 LAN Controller를 접속하여 하나의 노드(node)로 보고 네트워크를 구성하였다. 이 네트워크의 파라미터는 버스형 토폴로지에 트위스트 페어 케이블을 버스로 사용하였고 변조방식은 ADLC 내부 NRZ 방식과 NRZI 방식 중 단류 NRZ 방식을 선택하였다.

첫번째 실험은 2개의 노드간 Point-to point 방식으로 데이터 및 파일의 송신과 수신을 실시하였다. 정보 전송량을 256 바이트로 설정하고 인터럽트 제어 방식에 의한 전송에서 전송속도는 31.6 Kbps를 얻었다.

두번째 실험은 CPU의 부하를 줄이기 위하여 DMA(Direct Memory Access) 방법을 도입하여 전송속도를 1 Mbps까지 얻었다.

5. 결론

FA 환경이 갖는 특성을 고찰하고 LAN을 구축하는데 수반되는 FA용 LAN Controller를 설계하고 시험 제작하였다.

LAN Controller상에는 호스트 컴퓨터와 공유하는 RAM을 두어 효율적인 인터페이스를 제공할 수 있게 하였다. ADLC를 사용하여 데이터의 송수신을 실행하므로써 네트워크의 신뢰성을 향상시킬 수 있었으며 이로써, 각 계층간의 상호작용을 보장하고 각 계층이 갖는 고유 기능들을 효율적으로 관리할 수 있었다.

실제 공장에서 데이터 통신을 위한 소규모의 LAN을 구성할 경우, 이를 제어하는 한 모델로서 이상의 LAN Controller를 사용할 수 있을 것으로 기대된다. 하지만 실험과정에서 발생한 다음 사항에 관하여 꾸준한 연구가 필요할 것으로 사료된다.

1) LAN 컨트롤러를 제어하는 슬레이브 CPU로 하여금 통신만을 전담토록 하였기 때문에 통신 자체는 실시간으로 처리 될 수 있지만 때때로 호스트측의 과부하로 호스트가 공유 RAM으로부터 확인이 읽어 오지 못하는 사례가 있었다.

2) LAN에 연결된 모든 프로세서들의 상태를 감시하고 망상태 정보를 관리할수 있는 체계 및 효율적인 방안이 연구되어야 한다.

3) 본 실험에서는 NODE를 3개로 제한되어 수행하였으나 NODE가 확장되는 경우의 트래픽 및 성능에 관하여 검토되어야 한다.

본 연구에서는 하드웨어 설계와 구현에 중점을 두고 LAN Controller의 운용과 Point-to-point 방식의 데이터 전송을 위한 소프트웨어를 설계하였으나, 더 나아가 용

용 프로그램의 개발과 각 계층의 프로토콜이 연구 보완  
되면 더욱 훌륭한 시스템으로 개선될 수 있을 것이다.

참 고 문 헌

- (1) Andrew S.Tanenbaum, " *Computer network* ", Prentice-Hall, Englewood Cliffs, 1981.
- (2) William Stallings, " *Local network* ", Macmillan publishing company, 1984.
- (3) Motorola, " *8bit microprocessor & peripheral data* ", Motorola inc., 1983.
- (4) Motorola, " *Microsystems products technical data* ", Motorola inc., DS4079, 1986.
- (5) Motorola, " *The complete MAP solution* ", Motorola inc., 1985.
- (6) H.Zimmermann, " *OSI reference model-the ISO model of architecture open systems interconnection.* " ,IEEE Trans. on Commun., com-28, No.4, pp.425-432., 1980.
- (7) William Stallings, " *The IEEE 802 local network standards* ", Tele- communications. pp.40-48, March, 1986.
- (8) 월간, Interface, No.55, pp.159-222, Dec., Japan, 1981.
- (9) 정진욱, 변옥환, " *근거리 통신망* ", ohm 사, pp.284- 294, 1986.