

## 필드버스를 이용한 자동화 네트워크의 구현

### Implementation of Network for Automation Using Fieldbus

° 이석원, 장석호, 최형섭, 남부희

강원대학교 제어계측공학과

(Tel: 0361-50-6341; Fax:0361-242-2059; E-mail: boonam@cc.kangwon.ac.kr)

**Abstracts** This paper presents the implementation of network for automaton using fieldbus. The RS-485 and twisted pair are used in layer 1 and medium, respectively. We implement FDL(Fieldbus Datalink Layer) using C-language and use token-bus network. The network consists of polling system and token passing system. The amount of transmitted data frames is evaluated with various transmission speeds that affect the system stability.

**Keywords :** Fieldbus, Tokenbus, FDL, master/slave station

#### 1. 서 론

1980년대 초반부터 생산자동화 환경에서 이기종의 자동화 장비들 간의 통신을 위한 표준화된 네트워크시스템으로 MAP (Manufacturing Automation Protocol)이 개발되었다. 그러나 MAP은 OSI(Open System Interconnection)표준모델에서 제시하는 7계층을 모두 가지고 있어 생산현장의 필드에 설치된 각종 장비들간의 실시간 통신에는 적합하지 않아 OSI의 7계층중 물리층, 데이터링크층, 응용층의 3개의 계층으로만 이루어진 Mini-MAP이 개발되었으나 가격 면에서도 저렴하지 않다. 생산현장의 필드에 설치된 각종 제어 및 자동화 관련장비들에서 생성되는 데이터들의 실시간 통신을 가능하게 하고 가격도 저렴한 네트워크시스템이 요구되게 되었으며, 이러한 요구를 만족시켜주기 위해 개발된 네트워크시스템이 필드버스(Fieldbus) 네트워크이다. 필드버스네트워크는 CIM의 계층구조상 가장 하위 계층인 생산 field의 분산 공정 제어(distributed process control)에 있어서 센서, 공정제어기, 액츄에이터, 계측기기, 스위치, 분석기 등과 같은 제어기기들 간의 통신을 위하여 비교적 저가이고 통신에 필요한 최소한의 기능을 수행하는 단순한 구조를 가지면서도 빠른 응답시간으로 실시간 처리에 적용될 수 있는 네트워크 시스템이다. 종래의 생산현장에서 많이 사용되던 RS-232, RS422와 같은 점대점 (point-to-point)의 1:1방식의 통신은 센서, 액츄에이터 등의 장치들과 통신을 위해서 많은 배선이 필요하게 되고 또한 유지보수측면에서 보았을 때

어려움이 있고, 새로운 장비를 추가 할 때마다 새로운 배선이 필요하다. 그러나 필드버스시스템에서는 하나의 버스(bus)를 사용하는 네트워크시스템이므로 배선량을 상당히 줄일 수 있고 새로운 시스템의 확장이나 장비 등을 추가할 때 용이하고 유지 보수면에서도 용이하다. 본 논문에서는 필드버스의 독일표준인 Profibus를 기준으로하여 필드버스 시스템을 구현한다. 필드버스 시스템의 기존의 연구들을 보면 버스의 효율을 높이기 위하여 대역폭(bandwidth)할당에 관한 연구[2], FDL에 우선순위(Priority) 개념[3]의 도입 등 많은 연구들이 이루어져 왔다. 본 논문에서는 실험시스템을 구성하여 여러 가지의 전송속도에서의 데이터 처리량을 구한다. 구성된 실험시스템은 기존의 연구와 같이 우선순위개념은 도입하지 않았고 추후 우선순위 개념을 넣어 구성하고자한다. 현재의 실험시스템에서의 전송속도는 9.6kb/s에서 56k b/s까지 변화시켰을 때의 데이터 처리량을 구한다. 본 논문의 구성을 보면 2장에서는 필드버스시스템의 일반적인 구성요건, 프로토콜의 기능등에 대하여 설명하고, 3장에서는 실험내용과 결과에 대해서 설명하고, 마지막으로 4장에서 결론을 맺는다.

#### 2. 필드버스의 구성

일반적으로 OSI의 7계층중 물리계층(physical layer), 데이터링크계층(data link layer), 응용계층(application layer)의 3계층과

그 위에 추가된 사용자계층(user layer)으로 구성되어있다. 본 논문의 물리층에는 RS-485가 사용되고 스테이션들간의거리는 리피터 없이 1200m까지 접속 가능하다. 전송속도는 네트워크구조와 선로에 따라 다르지만 필드버스시스템의 독일표준인

사용자계층(User Layer)

7계층 : 응용층(Application Layer)

2계층 : 데이터링크층(Data Link Layer)

1계층 : 물리층(Physical Layer)

물리매체(Physical Wire Medium)

그림 1. 필드버스 프로토콜 스택  
Fig. 1. Fieldbus protocol stack

Profibus는 9.6k에서 500 kbit/s까지 가능하다. 토큰을 얻은 스테이션만이 버스를 사용할 수 있는 권한을 갖는 토큰버스 방식을 사용하고, 스테이션의 종류에는 토큰을 가질 수 있는 마스터

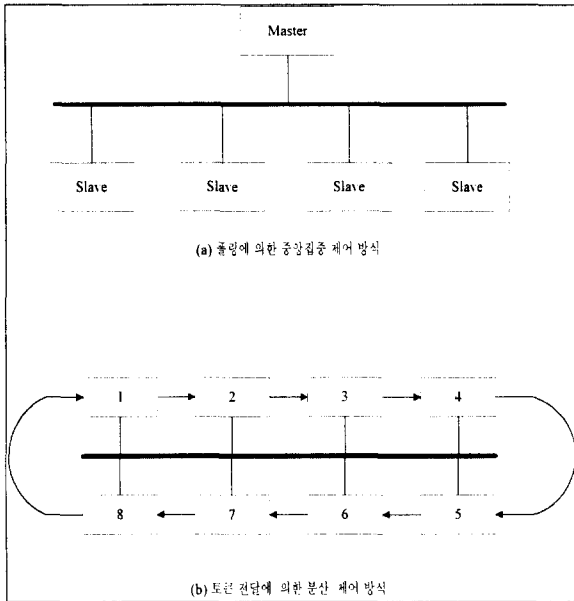


그림 2. 폴링시스템 / 토큰전달시스템  
Fig. 2. Polling system / Token passing system

스테이션과 가질수 없는 슬레이브 스테이션의 2종류 스테이션이 있다. 스테이션들이 갖는 주소범위는 0 - 127까지이며, 127번은 전역주소로 사용되고 버스관리는 마스터스테이션과 슬레이브스테이션에의한 중앙집중방식과 마스터스테이션과 마스터스테이션간의 토큰전달에 의한 분산제어방식이 사용된다. 그림2(a)에서 보듯이 폴링방식은 하나의 마스터스테이션이 여러개

의 슬레이브스테이션들과의 데이터전송을 수행한다. 즉 하나의

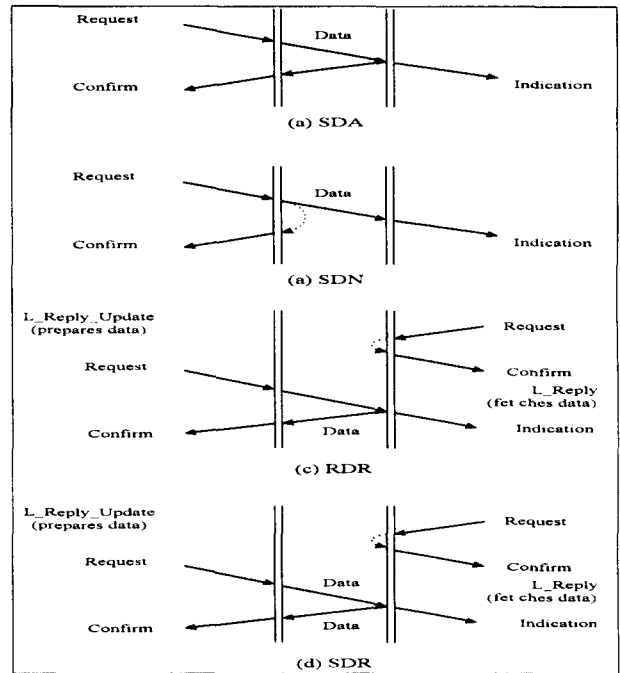


그림 3. 필드버스의 논리링크  
Fig. 3. Logical link of fieldbus

마스터가 버스운용을 전담하므로 폴링순서, 버스사용시간 등을 쉽게 결정할 수 있으므로 네트워크 시스템운용이 용이하다. 그러나 마스터 스테이션이 고장나면 네트워크시스템 전체를 사용할수 없는 단점이 있다. 반면에 그림2(b)와 같이 토큰전달에 의한 분산 제어방식은 마스터스테이션간에 토큰을 전달하므로서 버스를 사용할 수 있는 권리를 나누어 갖는다. 그러나 이러한 토큰전달절차는 그림2(a)의 폴링에 의한 중앙집중제어방식에 비해서 복잡하지만 마스터 스테이션 중의 하나가 고장이 나더라도 나머지 마스터 스테이션들은 계속동작을 할 수 있으므로 장애에 강하다. 필드버스의 데이터 전송서비스에는 주기적인 방식과 비주기적인 방식이 있다[1].

- 비주기적 방식 :  
Send Data with/without Acknowledge  
Send and Request Data with Reply
- 주기적 방식(Polling)  
Send and Request Data with Reply

데이터전달방식에는 그림3과 같이 상대방에게 데이터를 전송한 후 확인을 요구하지 않는 SDN (Send Data with No Acknowledge), 상대방에게 데이터를 전송한 후 확인을 요구하는 SDN(Send Data with Acknowledge), 상대방에게 데이터 전송을 요구하는RDR(Request Data with Response), 상대방에 데이

터 전송하는 동시에 상대방 노드로부터의 데이터 전송을 요구하는 SDR(Request Data with Response)이 있다.

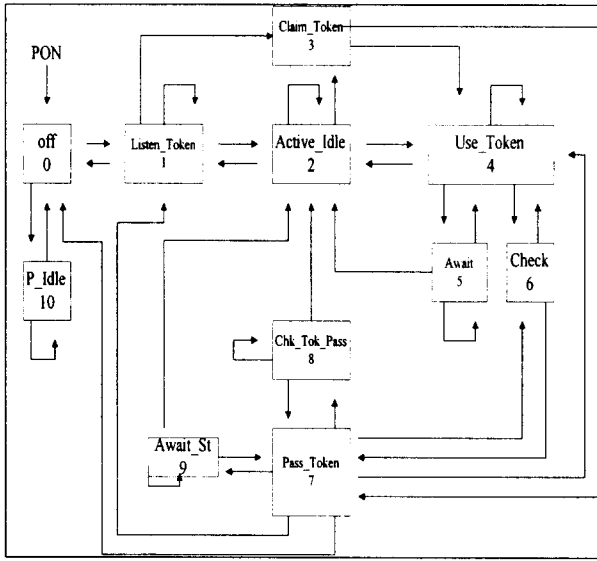


그림 4. FDL상태도  
Fig. 4. FDL status

그림4는 필드버스 데이터링크 ( FDL : Fieldbus Datalink Layer)층의 상태를 나타낸다[1]. 슬레이브스테이션은 0과 10의 상태를 갖고 마스터 스테이션은 10을 제외한 나머지 상태중의 하나를 갖는다. FDL의 상태1은 스테이션에 전원 인가시 자체 기능을 테스트한 후 논리적 링에 가입할 수 있도록 버스를 감시하다가 논리적 링에 새롭게 가입하고자 하는 스테이션이 있는지 검사하는 신호가 왔을 때 응답을 하므로써 논리적 링에 가입하고 상태2로 간다. 상태2는 버스를 감시하고 있다가 자신에게 전달되는 토큰이 있는지 혹은 자신에게 데이터를 요구하는 신호가 있는지 검사하고 있다가 응답하고, 토큰을 받으면 상태4로 변환된다. 상태4는 토큰을 획득한후 버스를 이용할 권리를 받은 상태이며 이 상태에서 일을 수행한다. 즉 슬레이브스테이션이나 마스터스테이션들과 데이터를 주고 받을 수 있다. 허용된 토큰사용시간이 초과되면 토큰을 다음 스테이션으로 넘기게 되는데 이때 상태4에서 이일을 수행하게 된다. 상태8을 토큰을 전달하는 과정을 검사하는 상태로서 정상적으로 토큰이 전달되면 다시 상태2로 가게 된다. 그렇지 못하면 상태7로 다시 가게 된다. 상태3은 상태1에 있던 스테이션이 논리적링에 가입하려고 버스를 감시하고 있다가 일정시간 동안 버스상에 신호가 감지 되지 않으면 현재 동작중인 스테이션이 하나도 없는 것으로 인식하고 스스로 토큰을 사용할 수 있는 상태인 상태6으로 변환된다. 상태6은 토큰을 사용할 수 있는 시간이 얼마나 남아있는지를 검사하는 상태이다. 이렇게 스테이션은 토큰을 받아서 일을 수행하고 토큰을 다음 스테이션에 전달하고 다시 토큰이 돌아 올때까지 버스를 감시하는 기능을 수행한다. 토큰의 전달순서는 스테이션 주소의 오름차순으로 전달된다. 최상위 번지에서는 다시 최하위 번지의 스테이션으로 토큰을 되돌려준다.

그림5는 스테이션과 스테이션 사이에서 일어나는 메시지 전달 과정을 나타낸 것이다. 메시지를 요구하고 상대 스테이션으로부터 응답을 받는 메시지 사이클은 아래 식(1)과 같이 나타낼 수 있다.

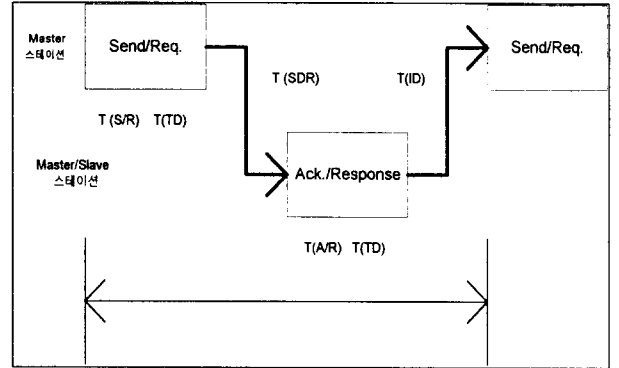


그림 5. 메시지사이클  
Fig. 5. Message Cycle

$$T_{MC} = T_{S/R} + T_{SDR} + T_{A/R} + T_{ID} + 2 \cdot T_{TD} \quad (1)$$

$T_{S/R}$  = request를 decoding 하고,

ack./response프레임조합하는데 필요한시간

$T_{SDR}$  = 스테이션 delay time

$T_{A/R}$  = acknowledge or response 프레임 전송시간

$T_{ID}$  = ack./response와 새로운 request가 올때까지의 시간

$T_{TD}$  = request/response 프레임 전송시간

### 3. 실험 및 결과

그림 6은 본 연구에서 실험한 시스템을 나타낸다. 이 시스템에서 물리층은 RS-485를 사용하였고, PC슬롯에 꽂아 사용하였다. FDL은 C-언어로 작성하였다. 마스터스테이션I은 슬레이브스테이션1, 2, 3를 제어하고 마스터스테이션II는 슬레이브스테이션4, 5를, 마스터스테이션III은 슬레이브스테이션6을 제어한다. 각각의 슬레이브스테이션에 해당하는 제어루프를 각각  $x_1, x_2, x_3, x_4, x_5, x_6$ 로 나타낸다.  $x_1=600ms, x_2=400ms, x_3=200ms, x_4=80ms, x_5=60ms, x_6=40ms$ 의 샘플링주기를 갖는다. 폴링시스템에서 시스템 전체가 안정하기 위해서는 토큰회전시간이 최소 샘플링 주기를 갖는  $x_6$ 루프의 샘플링타입보다 작아야한다[2]. 네트워크 시스템이 안정할 조건은 식(2)와 같다[5].

$$\sum_{j=1}^N \lambda_j b_j < 1 \quad (2)$$

$\lambda_j$  : 데이터의 스테이션j에의 평균도착률  
 $b_j$  : 스테이션 j 에서의 평균 서비스 시간  
 $N$  : 스테이션의수

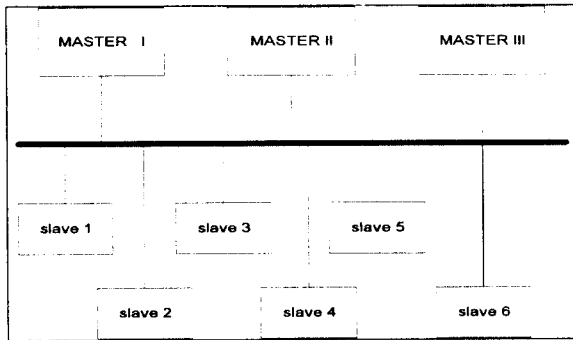


그림 6. 실험시스템  
 Fig. 6. Experiment system

모든 스테이션에서의 서비스시간은 전송속도에 따라 9.6kb/s 19.2kb/s 38.4kb/s 56kb/s 일 때 각각 22.9msec, 114.6msec, 5.7msec, 3.9msec이다. 데이터의 평균 도착률은 각 슬레이브스 스테이션의 샘플링시간과 같으므로 각각  $\lambda_1 = 1.66$ ,  $\lambda_2 = 2.5$ ,  $\lambda_3 = 5$ ,  $\lambda_4 = 12.5$ ,  $\lambda_5 = 16.7$ ,  $\lambda_6 = 25$  이다. 전송속도 9.6kb/s일 때 식(2)에 대입하면, 식(3)과 같이 나타낼 수 있다.

$$(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6) \cdot 22.9ms = 1.45 \quad (3)$$

식(3)의 결과는 식(2)을 만족하지 않으므로 전체 시스템이 안정하지 않음을 알 수 있다. 38.4kb/s일 때 식(2)에 대입하면, 식(4)과 같이 나타낼 수 있다. 식(4)는 식(2)를 만족시키므로 시스템

$$(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6) \cdot 5.7ms = 0.361 \quad (4)$$

지 안정함을 알 수 있다. 즉, 토큰이 전체 마스터 스테이션을 돌아서 오는 시간이 최소한 40ms 보다 작아야 모든 데이터를 처리할 수 있다. 실험시스템에서의 전송속도는 9.6kb/s, 19.2kbp/s, 38.4kbp/s, 56kb/s 의 네 가지의 전송속도를 갖는다. 실험 시스템에서는 샘플링타입이 큰 슬레이브1, 2, 3은 마스터 I에 의해 제어되고, 마스터II는 중간정도의 샘플링타임을 갖는 슬레이브4, 5를 제어하며, 마스터 III는 샘플링 타임이 아주 작은 슬레이브6만 제어한다. 토큰은 마스터들을 순환하며 토큰을 받은 마스터 스테이션은 처리할 데이터가 있으면 하나의 데이터만을 처리하고 처리할 데이터가 없으면 바로 다음 마스터스태이션으로 토큰을 전달한다. 그림7은 실험한 결과를 나타낸 것이다. 그림 7에서 보듯이 전송속도가 변화함에 따라 식(3), (4)와 마찬가지로 시스템에서 발생하는 데이터를 모두처리못하다가 전송속도가 올라감에따라 시스템에서 발생하는 모든 데이터를 처리하는 것을 볼수 있다.

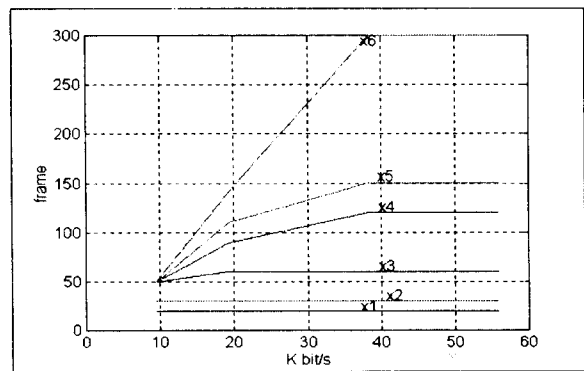


그림 7. 처리된 데이터의 양  
 Fig. 7. The amount of processed data

#### 4. 결론

본 연구에서는 물리층에는 RS-485를 사용하였고, FDL은 C-언어를 이용하여 필드버스 시스템을 구현하였다. 전송속도와 시스템안정성과의 관계를 보였다. 현재 많이 사용하고 있는 4-20mA 전류루프 시스템을 필드시스템으로 교체하기 위해서는 각종 센서 및 액츄에이터 등을 필드버스에 연결할 수 있는 제어장치들에 관한 연구가 계속 필요하다.

#### 참고문헌

- [1] PROFIBUS standard DIN 19 245 part I and II, 1992.
- [2] S. H. Hong, "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems," IEEE Tran. on Control System Technology, Vol. 3, No. 2, pp.225-230, June 1995.
- [3] S. Cavalieri, A. Di Sefano and O. Mirabella, "Optimization of Acyclic Bandwidth Allocation Exploiting the Priority Mechanism in the Fieldbus Data Link Layer," IEEE Trans. Industrial Electron., Vol. 40, No. 3, pp.297-306, June 1993.
- [4] Kang G. Shin, Chih-Che Chou, "Design and Evaluation of Real-Time Communication for Fieldbus-Based Manufacturing Systems", IEEE Trans. Robotics and Automation, Vol., 12, No. 3, June 1996.
- [5] O. C. Ibe and X. Cheng, "Stability Condition for Multiqueue Systems with Cycle Service," IEEE Tran. on Automatic Control, Vol. 33, No. 1, pp.102-103, January 1988.