

FMS를 위한 필드버스 시스템 구현

Implementation of Fieldbus System for FMS

°최형섭, 장석호, 이석원, 남부희

강원대학교 제어계측공학과(Tel : 0361-250-6341 ; Fax : 0361-242-2059 ; E-mail : boonam@cc.kangwon.ac.kr)
(Tel : 0361-243-4347 ; Fax : 0361-242-2059 ; E-mail : hyung@pine.kangwon.ac.kr)

Abstract The fieldbus is one of the communication networks in control and automation systems and allows the exchange of information between sensors/actuators and controlling elements. By means of the fieldbus, the various field devices can exchange their information among the nodes in the system. In this paper, we implemented the protocol of profibus, a kind of fieldbus for communication in FMS plant model. We applied this fieldbus to the FMS plant model which consists of three master stations and eight slave stations.

Keywords Fieldbus, Profibus, FMS

1. 서론

현대사회는 소비자들의 기호와 욕구가 다양하고, 기업간 국가 간의 경쟁이 치열해 짐에 따라 소비자의 다양한 욕구를 충족시키고, 수시로 변화하는 주위환경에 능동적으로 대처하기 위해 이에 맞는 생산시스템을 필요로 하고 있다. 유연생산시스템(FMS)과 통합생산시스템(CIM)이 바로 그것이다. 유연생산시스템은 많은 수의 공정들로 이루어져 있다. 따라서 이러한 많은 수의 공정 사이에는 원활하게 데이터를 전송해 줄 수 있는 네트워크가 필요하다. 이러한 네트워크 시스템에는 MAP, mini-MAP, TOP, 필드버스등이 있으며 이 중에서 필드버스는 OSI 7계층중 물리층, 데이터링크층, 응용층의 3개의 계층으로 이루어져 있어 생산현장의 필드에 설치된 각종 제어 및 자동화 기기에서 생성되는 데이터들의 실시간 통신이 가능하고, 가격도 저렴하여 각광받고 있는 네트워크 시스템이다.

본 연구에서는 FMS공장의 조립셀 라인의 네트워크 시스템용 필드버스 시스템중 profibus로서 구현하였다. 구현한 필드버스 시스템은 종래의 생산현장에서 많이 사용되던 RS-232와 같은 점대점(point to point)의 통신과는 달리 공통버스 방식(RS485)을 사용함으로써 많은 배선 절감효과를 가져왔고, 재래의 시스템보다 훨씬 덜 복잡하게 되어 유지 및 보수비용을 절감하게 되었으며, 온라인으로 각 필드기기의 상태를 점검할 수 있게 되었고, 새로운 시스템의 확장이 간단해 졌다. 특히, 소품종 대량생산의 과거와는 달리 다품종 소량생산이 요구되므로 제조 공정의 빈번한 변화를 가져올 수 밖에 없다. 이러한 변화에 대처하기 위해서는 생산 자동화용 네트워크가 필수적이라 할 수 있다. 필드버스는 적은 비용으로 구현가능할 뿐만 아니라, 네트워크를 통하여 공정변화를 쉽게 할 수 있고, 필드 기기의 지능화를 통하여 분산 제어를 할 수 있으므로 주 제어기의 부담을 줄일 수 있다.

2. 필드버스(PROFIBUS)

2.1 Profibus의 물리적 구성 및 특징

Profibus는 낮거나 중간정도 성능을 가지는 digital 필드 기기 또는 시스템들의 상호연결을 목적으로 하며, OSI 모델의 1,2,7층(물리층, 데이터링크층, 응용층)으로 구성되어 있다. 그렇게 때문에 상위 계층의 컴퓨터 network간에 전송을 자유롭게 할 수 있다. 물리층에서 profibus는 RS-485를 사용하며 station들 간의 거리는 리피터 없이 1200m 까지 가능하고, network 구조는 terminator가 있거나 없는 bus형을 사용한다. 전송속도는 network 구조와 선로길이에 의존하며 대략 9.6K ~ 500Kbit/s 까지 가능하다.

2.2 Profibus의 station 구성

Profibus에서 스테이션은 토큰을 가질 수 있는 마스터스테이션과 응답만 가능한 슬레이브 스테이션등 2가지 종류가 있으며, 최대 127개의 스테이션까지 구성이 가능하다. 사용하는 주소범위는 0-127까지 가능하며, 127은 전역 주소로 사용한다. 마스터 스테이션은 32개 까지가 적당하나, time critical 하지 않은 경우 127개 까지 가능하다. 버스관리는 마스터-슬레이브에 의한 중앙집중방식과 토큰전달에 의한 분산제어 방식을 사용하고, 데이터 전송 서비스에는 비주기적인 방법과 주기적 방법이 있다.

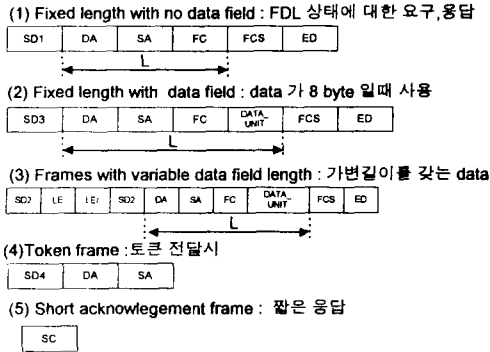
2.3 장·단점

Profibus는 특별한 부품을 요하지 않고, OSI에서의 기술을 수용하고 있으며, 각종 멀티플렉서나 드라이버와 같은 특정 용도를 위한 응용에 개방되어있다.

그러나 오직 마스터/슬레이브의 폴링동작을 통하여만 기능을 충분히 발휘할 수 있고, 시간(주기)의 엄격한 준수가 요구되는 시스템에 사용하기 곤란하다는 단점을 가지고 있다.

2.4 프레임

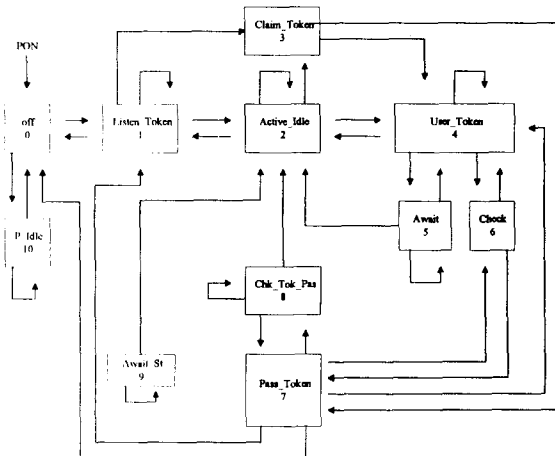
토큰은 마스터 스테이션들로 이루어진 논리적 링에서의 순환을 하며 토큰을 수신한 스테이션은 버스를 이용할 수 있다. profibus 에서 스테이션들간의 데이터 교환을 위해 전달하는 데이터들을 프레임이라고 하며, 프레임의 길이는 1 또는 3에서 최대 255 바이트이다. 프레임의 종류는 다음과 같은 것들이 있다.



(1)번 프레임은 데이터가 없을때, 즉 FDL상태에 대한 요구/응답 시 사용하며 (2)번 프레임은 데이터가 8byte로 고정되어 있을때, (3)번 프레임은 데이터의 길이가 가변일때 사용하며 (4)번 프레임은 토큰을 의미하는 프레임이며, (5)번 프레임은 짧은 응답이나 확인시 사용된다. 각 프레임에서 SD(start delimiter)는 프레임의 시작을 알리고, ED(end delimiter)는 프레임의 끝을 알린다. SA(source address)는 프레임 발신주소를, DA(destination address)는 목적지 주소를 나타낸다. 프레임은 FC(frame control)에 자신의 상태를 나타내는 중요한 정보를 가지고 있다. 예를들어 FC의 값이 0x49이면 "request FDL status with reply"를 나타내고, 0x00이면 "slave station"을, 0x10이면 "master station not ready"를, 0x20이면 "master station ready"를 0x4c이면 "send and request data"를 나타내는데 FC의 값은 그 프레임의 상태와 같은 control정보를 포함하고 있다.

2.5 FDL (Fieldbus Datalink Layer)

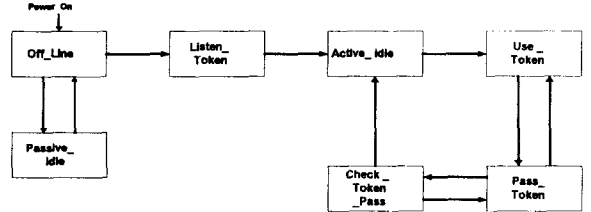
Profibus의 FDL은 <그림1>에서와 같이 마스터 스테이션의 10개의 FDL 상태와 슬레이브 스테이션의 2개의 FDL 상태를 가지고 있다.



< 그림 1 > profibus의 FDL 상태도

3. Reduced FDL

본 논문에서는 마스터 스테이션에서 6개의 FDL 상태를 사용하였고, 슬레이브 스테이션에서 2개의 FDL 상태를 사용하여 구현하였다.



< 그림 2 > 구현한 FDL 상태도

· Off_line state는 power on 후의 상태를 나타내며 이 상태에서 FDL은 모든 operating parameter를 초기화한다. station 주소(TS)를 지정하고, baud-rate을 설정한다. baud-rate은 모든 station을 동일하게 설정해 주어야 한다. 그리고 media를 single 이나 redundant로 설정해 주어야 하며, hardware와 software의 version을 지정해 준다. master station의 logical ring에 참여 여부를 결정해 준다. 또한 최상위 주소 HSA를 설정하고, 최대 재시도 횟수를 설정해 준다.

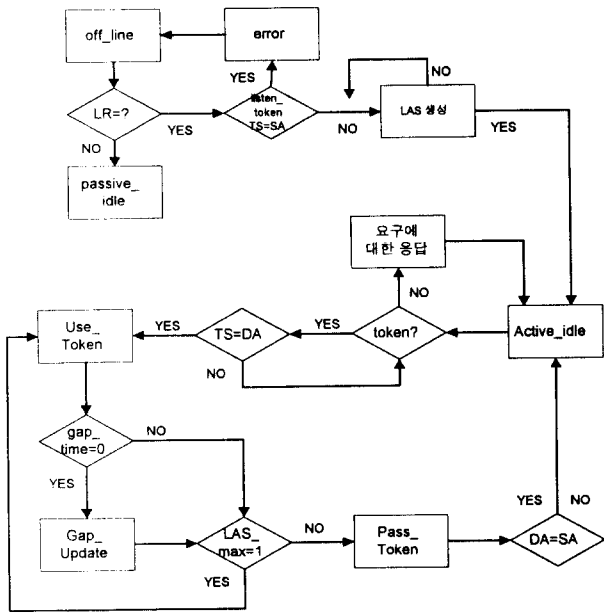
· Listen-Token state는 선로를 모니터링하며, LAS (List of Active Station)를 생성하여 logical ring에 들어갈 준비를 하는 단계이다. 선로를 모니터링하여 토큰 프레임을 분석하고, 토큰 프레임에 포함된 스테이션 주소로 LAS를 생성한다. 토큰 프레임의 SA(Source Address)로 LAS를 생성하다가 LAS에 기록된 주소를 갖는 토큰을 listen한 경우 LAS생성을 끝낸다. LAS 생성동안에는 어떤 "Request FDL status"에도 "not ready"라고 응답하고, 다른 프레임은 처리되지 않는다. LAS를 생성한 후 "Request FDL status"를 받으면 "ready to enter logical token ring"을 응답한후 "Active_Idle"상태가 된다. 그리고 마스터 스테이션으로 등록된후 SA로 자신의 주소와 같은 주소를 가진 토큰을 감지하면 심각한 에러로 인식하고 "Offline"상태로 되돌아간다.

· Active_Idle state는 선로를 listen 하며, active (Use token 상태) 되기 위한 대기 상태로서, 선로를 listen 하며 LAS를 갱신한다. 그리고 자신에게 보내진 적당한 액션프레임이 수신되면 확인/응답한다. 자신에게 토큰 프레임이 전달되면 토큰을 받아 "Use-Token"상태가 된다.

· Use-Token state는 실제 작업을 수행하는 state로서 마스터 스테이션이나 슬레이브 스테이션에 데이터를 요구하거나, 데이터를 보내 실제 메시지의 전송이 이루어지는 부분이다.

· Pass-Token state는 LAS를 참조하여 NS(Next Station)로 token frame을 전송하는 state로서 Gap_Update를 하고, NS로 token frame을 전송한다. 이때 LAS에 혼자만 있을때는 다시 Use-Token 상태로 돌아간다.

· Check-Token pass state는 토큰이 잘 도착했는지 확인하는 상태로서, NS이 token을 사용하고 있나를 관찰한다. 관찰하는 방법은 선로의 데이터 프레임이나, 토큰 프레임에서 SA를 감지하여 유효한 프레임 (SA = NS)이면 성공적으로 token이 전달된 것으로 판단하고, 유효치 않은 프레임이면 (SA ≠ NS) 다른 스테이션이 active 된 것으로 판단하고 Active_Idle 상태가 된다.



<그림 3 > FDL flowchart

4. 구현한 FDL

본 논문에서 구현한 FDL은 reduced FDL로서 데이터의 전송 부분을 인터럽트를 이용하여 처리하였다. 인터럽트는 UART(Universal Asynchronous Receiver Transmitter) 8250과 PIC(Programmable Interrupt Controller) 8259의 레지스터 값을 조정함으로써 가능하게 하였다. UART의 interrupt enable register의 receive data available을 1로 set 시키고, modem control register의 OUT2 를 1로 set 시켜 8250의 인터럽트를 가능하게 하였으며, PIC의 interrupt mask register의 값을 읽어 UART의 IRQ를 unmask 시키고, 그 값을 register OCW1(Operation Command Words)에 써 줌으로써 인터럽트를 가능하게 하였다.

그리고 off line state에서 여러가지 operating parameter를 설정해 준다. this station 주소 TS는 7,8,9, (MS)로 지정해 주었고, baud_rate은 모든 스테이션이 동일하게 9600 bps로 설정되었으며, 8250을 초기화 할때 설정해 주었다.

baud_rate설정은 8250의 line control register 의 8번째 비트인 DLAB를 set 시킨후 receiver buffer 와 interrupt enable register 에 분주할 숫자를 써넣으면 된다. 9600bps는 12 분주를 하면되므로 receiver buffer에 12를 써 넣고 interrupt enable register에 00을 써 넣으면 된다. 그 후에 DLAB를 reset 시키면 baud_rate 설정이 끝나게 된다. media선택은 single로 설정하고, hardware 와 software version은 상품화된 것이 아니므로 임의의 숫자로 동일하게 지정하여 주었으며, yes나 no로 logical ring에 참여 여부를 결정하였다. 최상위 주소 HSA는 0x20으로 설정하였으며, 최대 재시도 횟수는 2번으로 설정 하였다.

Use_Token state에서 마스터 스테이션 1 은 각 슬레이브 스테이션들에게 차례대로 센서의 값을 요구하고, 응답을 기다리다 응답(센서의 값)이 오면 모든 센서의 값을 한 바이트로 저장하고 token을 전달한다. 그후 마스터 스테이션 2, 3에서 센서값의 요구가 있을시 데이터를 전송한다. 마스터 스테이션 2는 마스터 스테이션 1에게 센서값을 요구하고, 받은 데이터를 분석하여 로보

트 R1, R2, R3에 작업을 지시하는 액션 프레임의 전송한다. 마스터 스테이션 3는 마스터 스테이션 1 에게 센서값을 요구하고 받은 데이터를 분석하여 AGV에게 작업을 지시하는 프레임의 전송하게 된다.

센서용 슬레이브는 데이터를 가지고 있다가 요구가 있을때 데이터를 전송하고 로봇과 AGV는 액션프레임이 전달되면 작업을 수행한다.

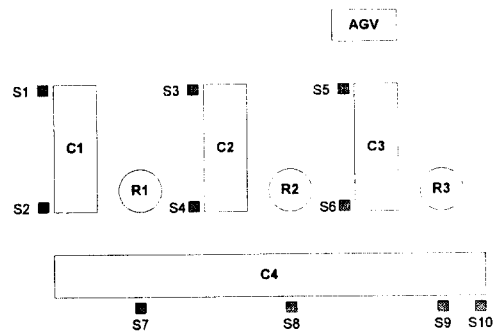
5. FMS공정 모델링

본 논문에서는 FMS 공장의 여러부분중 조립셀 라인을 대상으로 삼아 필드버스 네트워크로 모델링 하였다.

<그림 4> 에서 보는 바와 같이 조립셀은 로봇3대, 컨베이어 4대, 센서10개, 무인반송차 1대로 구성되어 있다. 부품을 실은 반송차는 각 부품 컨베이어에 부품을 내려놓고, 각 부품 컨베이어는 부품을 작업위치 까지 운반한다.

부품이 도착하면 조립셀에 설치되어 있는 3대의 로봇이 각기 맡은 작업을 수행하여 조립을 하게된다. 조립이 완료되면 무인 반송차는 완제품을 창고로 운반하게 된다.

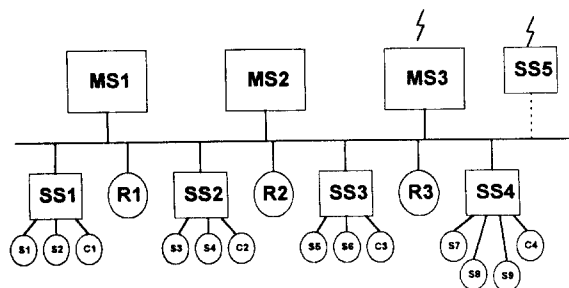
본 논문에서는 <그림 5>에서의와 같이 마스터 스테이션 3대와 슬레이브 스테이션 8대(로봇 3대)로 구성된 네트워크 시스템으



<그림4 > FMS 공장 조립셀

로 조립셀 라인을 모델링 하였다.

MS1 (마스터 스테이션 1)은 SS1, SS2, SS3, SS4 를 슬레이브 스테이션으로 하여 각 센서의 값을 관리, 감독하고 MS2, MS3 에서 필요한 센서의 값을 제공하여 준다. MS2 는 MS1 으로부터 센서의 값을 받아 로봇 R1, R2, R3를 제어하게 된다. MS3는 MS1으로부터 센서의 값을 받아 AGV(SS5)를 제어하게 된다. 슬레이브 스테이션 SS1, SS2, SS3, SS4 는 센서의 값을 가지고 자체적으로 컨베이어를 제어 하게 된다.



<그림5 > 필드버스 네트워크 시스템

6. 동작 알고리즘

1. Master station 1 이 토큰을 받는다.
2. Slave station 1, 2, 3, 4에게 센서값을 요구하고 받는다. 여기에 사용되는 프레임은 아래와 같다.

액션 프레임

68	04	04	68	0F	07	4C	DATA	FCS	16
----	----	----	----	----	----	----	------	-----	----

응답 프레임

68	05	05	68	07	0F	0C	DATA	FCS	16
----	----	----	----	----	----	----	------	-----	----

프레임에서 data부분에는 몇번째 슬레이브라는 값이 있으며 응답 프레임에는 센서값이 있다. 여기서 두,세번째 byte는 information field length로 DA (Destination Address), SA(Source Address), FC(Frame Control), Data field를 합한 길이를 나타낸다.

3. Master station 2로부터 센서값의 요구가 있을 때 응답을위해 7개의 센서값을 한byte로 압축한다.
4. Master station 2에게 토큰을 넘긴다.
5. Master station 2가 토큰을 받는다.
6. Master station 1에게 센서값을 요구하는 프레임을 보내고 응답을 받는다.

여기에 사용되는 프레임은 다음과 같다.

액션 프레임

68	04	04	68	07	09	4C	DATA	FCS	16
----	----	----	----	----	----	----	------	-----	----

응답 프레임

68	05	05	68	09	07	4C	DATA	FCS	16
----	----	----	----	----	----	----	------	-----	----

응답 프레임에서의 data는 마스터 스테이션1이 모이는 센서값이다.

7. 센서값에서 s2에 on, s7에 off 일 때 robot 1에게 액션 프레임을 보내고 응답을 받는다. 여기에 사용되는 프레임은 다음과 같다.

액션 프레임

10	0F	09	43	FCS	16
----	----	----	----	-----	----

응답 프레임

E5

E5 : short acknowledgement

8. 센서값에서 s4와 s8이 on일 때 robot 2에게 액션 프레임을 보내고 확인을 받는다. 여기서 사용되는 프레임은 step 7에서와 같다.
9. 센서값에서 s6 와 s9이 1일 때 robot 3에게 액션 프레임을 보내고 응답을 받는다. 여기에 사용되는 프레임은 step 7에서와 같다.
10. Master station 3에게 토큰을 전달한다.
11. Master station 3가 토큰을 받는다.
12. Master station 1에게 센서값을 요구하는 프레임을 보내고 응답을 받는다. 여기서 사용되는 프레임은 다음과 같다.

여기에 사용되는 프레임은 다음과 같다.

액션 프레임

68	04	04	68	07	08	4C	DATA	FCS	16
----	----	----	----	----	----	----	------	-----	----

응답 프레임

68	05	05	68	08	07	4C	DATA	FCS	16
----	----	----	----	----	----	----	------	-----	----

응답 프레임에서의 data는 마스터 스테이션1이 모이는 센서값이다.

13. 센서값에서 s10이 1 이면 AGV에게 c4로, s1이 0 이면 c1으로, s3 이 0이면 c2로, s5 가 0이면 c3로 가라는 명령을 준다. 마스터 스테이션 1에게 토큰을 전달한다.
14. 그외의 상황이면 바로 마스터스테이션 1에게 토큰을 전달한다.

7. 결과 및 고찰

< 표 1 > 네트워크 성능평가

	토큰 점유 시간	점유율 (%)	전송시간
MS1	21441/40620	52.8	9616초
MS2	9368/40620	23.1	3513초
MS3	9756/40620	24.0	2909초

토큰 점유 시간은 모니터 시간동안 마스터 스테이션의 use_token 시간을 말하며, 점유율은 총 모니터 시간과 각 스테이션의 토큰 점유시간의 비율, 순수 data 전송시간은 각 스테이션이 토큰을 가지고 있는 동안에 다른 스테이션에 액션프레임을 보내고 그 스테이션으로부터 응답을 받은 시간의 합을 말한다. 스테이션 1은 매번 각 센서에 값을 요구해야 하나 스테이션 2나 스테이션 3는 센서값에 따라 슬레이브 스테이션에 프레임을 발송하지 않을수도 있기 때문에 데이터 결과 값에서 나타나듯이 토큰 점유 시간과 순수 데이터 전송시간이 스테이션 1보다 적다.

본 논문에서는 OSI의 3계층(1, 2, 7계층)을 이용하지 않고 7계층없이 구현하였기 때문에 FDL에서는 상위 계층의 프리미티브 요구시 데이터 전송을 담당하는 것이 아니라 자체에서 데이터에 대한 연산을 수행하기 때문에 토큰 점유 시간에 비해 순수 데이터 전송시간이 작게 나타난다.

본 논문에서는 FMS plant model을 fieldbus network으로 modeling함으로써 실제 FMS공장에서 field device들을 fieldbus network으로 연결하여 단위공정을 자동화 시킬수 있을 것으로 판단된다. 그렇게 함으로써 전체공정을 일관되게 관리하여 생산성을 향상시키고, 비용을 절감시키는 동시에 자동화공정의 설계, 구축및 유지관리에 유연성과 신뢰도를 극대화 시킬 수 있을 것이다.

참고문헌

- [1] Profibus Trade Org., DIN 19245 Profibus Standard: Part1, PTO, 1993.
- [2] Profibus Trade Org., DIN 19245 Profibus Standard: Part2, PTO, 1993.
- [3] 박홍성, 권옥현, "산업용 네트워크 그 응용", 제어자동화 시스템 공학회지, 7월, 1996년.
- [4] 임용희, 김대원, "펄트리네트를 이용한 유연 로봇 조립셀의 시뮬레이션에 관한 연구"
- [5] 홍승호, 김기암, 김지용, 고성준, "분산제어 및 자동화 시스템과 펄트버스," 제어자동화 시스템 공학회지, 7월, 1996년.