

공장자동화용 데이터 버스(EFbus)의 설계에 관한 연구

이전우 황선호 김현기 이혁희 채영도

한국전자통신연구소 자동화시스템연구실

A Study on the Design of Data Bus(EFbus) for Factory Automation

○J.W.Lee S.H.Hwang H.K.Kim H.H.Lee Y.D.Chac

Electronics & Telecommunications Research Institute

Automation Systems Section

ABSTRACT

This paper describes the design of EFbus(ETRI Fieldbus). EFbus was designed from the interim results of international Fieldbus standardization in IEC (International Electrotechnical Commission) and proposals of many companies. The design was performed and described on the basis of three layered architecture. Application layer S/W runs on IBM PC and provides services which is similar to MMS in MAP to user application. Data linklayer runs on Intel's iDCX96 real time executive and uses centralized media accesscontrol method. Physical layer uses manchester encoding & decoding, twisted pair lines and RS-485 electrical standard.

1. 서론

현재 공장용 네트워크는 통신특성과 주변 환경에 따라 세 레벨로 분류될 수 있다.[1][2][3] 가장 높은 레벨의 네트워크는 MAP(Manufacturing Automation Protocol)이고 중간레벨의 네트워크는 MiniMAP/Proway이며, 최 하위레벨 네트워크는 Fieldbus이다. 하위레벨 네트워크일수록 작은 size의 데이터를 교환하면서 보다 더 실시간 처리를 필요로 하는 응용에 적합하다.

Fieldbus는 저가이고 저성능이면서 빠른 응답시간을 요구하므로 MAP이나 MiniMAP/Proway를 사용하기에 부적당한 센서나 액추에이터(actuator), 소형 제어기기들을 연결하기 위하여 제안되었다. Fieldbus의 주된 개념은 기존의 analog signalling 기술을 사용하는 point to point 연결방식을 다중화된 digital signal 전송방식의 serial data bus로 대체한다는 것이다. 이렇게 함으로써 기존방식보다 케이블 설치비용을 줄일 수 있을 뿐 아니라 기기들 간에 보다 정확하고 많은양의 데이터를 교환할 수 있으며, 기기를 새로 접속하거나 분리시키는 것이 쉬워지는 등 여러가지 장점이 있다.[4]

Fieldbus는 최초 공정제어(process control) 분야에서 그 필요성이 대두되었으나 IEEE Project 1118(Microcontroller system serial control bus)와 협력하고, MAP의 MMS(Manufacturing Message Specification) 도입 및 MiniMAP과의 연결고려등을 통

하여 생산자동화(Manufacturing Automation) 분야로 영역이 확대되고 있다. 이에 대한 기술이 개발되면 생산현장에서 사용되고 있는 센서, 액추에이터, 제어기기 등에 많은 영향을 미칠 것으로 예상된다.

이 때문에 미국의 ISA(Instrument Society of America), 프랑스의 FIP(Factory Instrumentation Protocol) club, 유럽의 EUREKA 및 독일의 ZVEI(German Electrical Equipment manufacturer를 의미함) 등과 같은 단체 및 Rosemount, Simens, Foxboro 등의 기업체가 자국 및 자사의 Fieldbus를 개발하고 있으며, IEC(International Electrotechnical Commission)에서는 1985년부터 이의 국제표준화를 추진중에 있다.[5]

전자통신연구소에서는 공장자동화에서 Fieldbus가 차지하는 비중을 인식하여 이에 관한 자료수집 및 국제 표준화 동향 파악을 수행하고 있으며, Fieldbus 특성을 갖는 EFbus(ETRI Fieldbus)의 개발을 추진중에 있다. EFbus는 지금까지의 국제 표준화 추진결과를 근간으로 하면서, 아직 미정이거나 논란 중인 부분은 각 국가나 단체의 표준화 안을 참고로 하여 규격이 결정되었다.[6]

본 논문에서는 이 EFbus의 설계와 관련된 사항을 소개한다.

2. EFbus의 구조

EFbus는 생산 현장의 제어기와 센서 및 액추에이터들 간의 통신을 위한 "low level data bus"로서 실시간 통신이 요구된다. 이러한 실시간 통신을 위해서 EFbus는 컴퓨터 통신 표준 모델인 ISO의 OSI 7 layer 구조에서 1,2,7 layer만을 사용한 통신 표준 모델을 정하였다. 공장자동화용 데이터 버스인 EFbus는 그림 2.1과 같이 물리 계층, 데이터링크 계층 및 응용 계층으로 구성된 3계층 구조를 가진다.

2.1 응용 계층

EFbus의 응용 계층은 FMS(Fieldbus Messaging Service) 부계층과 LLI(Lower Layer Interface) 부계층으로 나누어진다. FMS 부계층은 다른 응용 계층과 통신하기 위한 서비스를 제공하고, LLI 부계층은 응용 계층과 데이터링크 계층을 연결시켜

주는 것으로 acyclic, cyclic 및 broadcast 데이터 전송을 위한 functionality를 제공한다.

2.2 데이터링크 계층

데이터링크 계층에서 수행하는 기능은 다음 두가지 이다.

- Medium access 관리 기능
- 데이터 교환 기능

EFbus 에서의 medium access 관리는 centralized media access 방식으로 Bus Arbitrator가 system configuration 시 제공되는 scan table에 따라 access 관리를 수행한다. 데이터 교환 기능은 producer와 consumer의 개념으로 이루어진다.

2.3 물리 계층

물리 계층은 EFbus 구조상 Layer 1에 해당하는 최하위 레벨로서 MIS(Medium Independent Sublayer)와 MDS(Medium Dependent Sublayer)의 두 부계층으로 나누어진다. MIS는 medium에 전송할 신호를 Manchester coding 방식으로 인코딩하고 전송매체로부터 받은 신호를 디코딩하며 비트레벨에서 에러를 점검한다. MDS는 EFbus에 신호전송을 위한 전송매체로 shielded twisted pair 케이블을 사용한다.

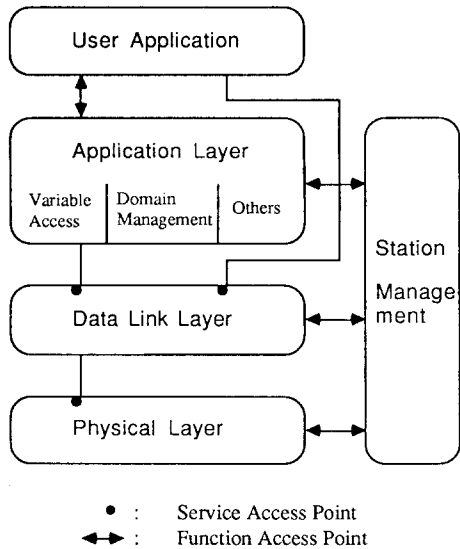


그림 2.1 EFbus의 구조

3. EFbus의 계층별 설계

3.1 응용 계층

응용 계층은 그림 3.1과 같이 FMS 부계층 및 LLI 부계층으로 구성된다. FMS 부계층은 사용자가 EFbus 네트워크를 access 하는데 필요한 user I/F 기능, FMS 서비스들을 사용하기 전에 FMS 환경을 구축하기 위한 FMS environment 기능, FMS 서비스들의 상태전이를 처리하는 FMS 서비스 state machine, lower layer에 전달되는 FMS PDU의 생성 및 해석을 처리하는 FMPM 및 error processing과 이에 대한 report 기능 등이 필

요하다. 그리고 LLI 부계층은 FMS 데이터를 데이터링크 계층에 전송하며 타 노드와의 통신을 위한 mapping을 하는 것이다. 이러한 기능을 수행하기 위해서 LLI 부계층은 초기화를 위한 Associate 서비스, 공정 데이터 전송을 위한 Data Transfer 서비스 및 association을 해제하기 위한 Abort 서비스를 제공한다. 또한 긴 message 전송을 위한 flow control 및 에러를 처리할 수 있는 error control 기능도 제공하여야 한다.

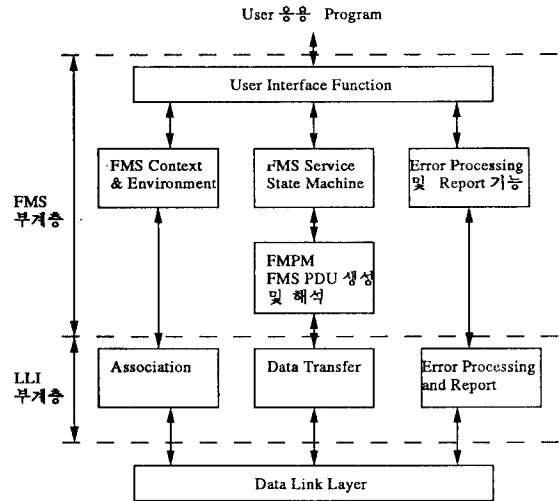


그림 3.1 응용 계층 구조

1) FMS 부계층

(1) Data Structures

이 데이터 구조는 사용자의 응용 코드와 FMS 부계층 사이에 전송되는 데이터를 처리하는 데 사용하며, requests, responses, indications 및 confirmations과 같은 서비스 primitive들에 대한 정보를 처리하는 데 사용하는 queue data structures와 FMS서비스에 대한 정보를 처리하는 데 사용하는 operation-specific data structures로 구성된다.

a. Queue data structures

사용자에 의해서 서비스 request가 발생되면 FMS 부계층은 그 request에 대한 정보를 추적하기 위한 queue data structure를 제공하며, 이 때 사용된 pending queue data structure는 그 request에 대한 confirmation을 track하는 데 사용된다. 또한 하나의 indication이 수신될때 FMS는 PDU를 parse하고 그것에 대하여 indication queue data structure를 하나 지정하며, 그 구조의 요소들은 indication이 수신될 때 queue에 data를 저장하기 위하여 assign된다.

b. Operation-specific data structures

응용계층 내에 있는 모든 FMS서비스 하나당 최소한 하나의 operation-specific data structure가 할당되며, 이러한 구조의 각각의 format은 사용된 서비스의 형태에 의존한다. 사용자가 primitive request 기능을 call하기전에 그 request에 대한 데이터를 포함하는 operation-specific data structure를 생성하여야 한다.

그리고 indication이 수신되면 그 request 에 대한 데이터를 포함하는 operation-specific data structure에 대한 pointer가 제공되어야 한다. 그 후 데이터를 access, process하며, primitive response function을 call 하기전에 response에 대한 operation-specific data structure를 생성한다.

(2) Function module description

a. Primitive request function module

이 기능은 사용자가 네트워크에 연결되어 있는 다른 노드들에 FMS request primitives 을 전달하는 데 사용되며, 이 module 의 동작은 다음과 같다. FMS 서비스 request가 수신되면 FMS 부계층은 서비스 invoke_id결정 및 FMSREQ_PEND의 구조가 선택되고, Info_ptr에 의해서 지시된 정보를 사용하여 FMS request PDU를 생성한후 LLI 부계층에 FMS request PDU를 전달하기 위하여 LLI기능을 call 한다.

b. Primitive response function module

이 기능은 네트워크에 있는 다른 노드로 부터 수신된 indications에 대하여 response 를 전달하는 데 사용되며, module 의 동작은 다음과 같다. FMS 서비스 indication 이 수신되면, info_ptr와 invoke_id에 의해서 지시된 정보를 사용하여 response PDU를 생성하고 response PDU 가 LLI부계층에 전달 되도록 LLI부계층의 기능을 call 한다.

c. Primitive user confirmation function module

이 기능은 primitive request기능에 대한 confirmation 이 responding 노드로 부터 수신될 때 FMS부계층이 Confirmation에 대한 결과를 CRT나 printer로 display하는 데 사용된다.

d. Primitive user indication function module

이 기능은 remote 노드에서 indication이 수신될 때 FMS부계층이 수신된 노드의 사용자에게 indication에 대한 결과를 알리는 기능이다.

e. FMS support function

이 기능은 specific FMS 서비스를 처리하는 데 사용하며, EFbus 시스템을 reset 및 setup하는 데 사용한다.

f. Error response function

이 기능은 네트워크에 있는 다른 노드로 부터 수신된 indications에 대한 error response 를 전달하는 데 사용된다.

(3) FMS service functions

a. Environment & General management services

이 서비스 그룹은 FMS 서비스를 사용하기 전에 FMS 환경 구축 및 FMS 서비스 종료를 담당하는 데 사용한다.

- Initiate service

- Abort service

b. Domain management services

이 서비스 그룹은 하나의 domain과 real 기기의 memory 영역을 mapping 하는 데 사용되며, Upload service와 Download service 가 있다.

a) Download domain service

- InitiateDownloadSequence service

- DownloadSegment service

- TerminateDownload Sequence service

b) Upload domain service

- InitiateUploadSequence service

- Uploadsegment service

- TerminateUploadSequence service

c. Variable access services

이 서비스들은 VFD내에 있는 variable 들을 access하는 데 사용된다.

- Read service

- Write service

- InformationReport service

d. VFD support services

이 서비스들은 VFD를 관리하는 데 사용된다.

- Status service

- Unsolicited status service

- Identify service

2) LLI(Lower Layer Interface) 부계층

LLI 부계층은 응용계층의 FMS 데이터를 데이터링크 계층에 전송하고 타 노드와의 통신을 위한 mapping 기능을 수행한다. 그림 3.2는 LLI 부계층의 개략도이다.

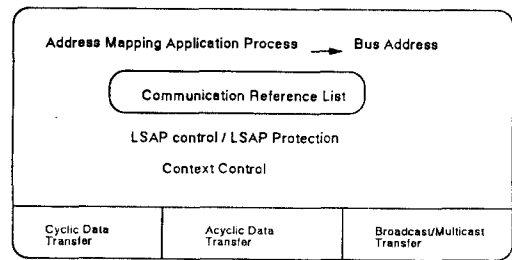


그림 3.2 LLI 부계층의 개략도

LLI 부계층은 컴퓨터 통신모델인 ISO 의 OSI 7 계층모델의 3-7 계층의 일부 기능을 수행하는데 주 기능은 AE(Application Entity)들 간의 통신을 위한 mapping을 하는 것이다. LLI 부계층의 구성은 그림 3.2와 같이 AE들에 대한 통신 정보와 실제 bus address 와 LSAP(Link Service Access Point) 를 기록한 CRL(Communication Reference List) 와 AE들 간의 통신을 위한 communication relationship을 유지하는 Associate 서비스 및 공정 특성에 따른 데이터 전송 functionality들을 제공하며 통신의 완료 및 에러가 발생 시 이를 처리하는 Abort 서비스들로 구성된다. 이러한 LLI 부계층의 동작은 AE들 간의 통신을 위한 communication relationship을 설정 후 communication relationship에 따라 FMS 데이터를 전송한다. 그리고 AE들 간의 통신이 완료되면 이러한 AE들 간의 communication relationship을 해제하여 통신을 완료한다. 이제 이러한 LLI 부계층을 구성하는 LLI 부계층 서비스들 즉 Associate서비스, Data Transfer 서비스 및 Abort 서비스에 대하여 살펴 보겠다.

(1) Associate 서비스

Station 들 간의 data transfer 를 하기전에 communication

initialization을 수행하여야 한다. Communication initialization은 communication reference 에 의해 지정된 AE에 대한 통신 관계를 install 하고 이에 대한 통신을 가능하게 하는 것이다. 이러한 통신 초기화 과정에는 LLI, FMS context를 검색하며 AE 들 간의 communication functionality 가 같을 경우 데이터 전송이 가능하다.

(2) Data Transfer 서비스

FMS PDU(Protocol Data Unit) 를 전송하기 위한 data transfer 서비스는 크게 전송 여부를 확인하는 confirmed data transfer 서비스와 전송여부를 확인하지 않는 unconfirmed data transfer 서비스로 나뉜다. Confirmed data transfer 방법에는 데이터 전달 방법에 따라 acyclic data transfer 방식과 cyclic data transfer 방식이 있다. Acyclic data transfer 방식은 데이터 요청이 오면 데이터를 전송하는 비 주기적인 데이터를 전송하며 cyclic data transfer 방식은 EFbus 특유의 데이터 전송방식으로 buffer를 두어 주기적으로 공정 데이터를 update 시켜 데이터 요청시 buffer의 내용을 전송하여 주기적인 데이터의 전송을 가능하게 하는 방식으로 공정제어에 많이 사용된다. 이러한 data transfer 서비스는 acyclic data transfer, cyclic data transfer 및 broadcast data transfer 에 대한 data transfer functionality 들로 구성되며 Associate 서비스 수행에 의해 이들 data transfer 방식중의 하나가 선택되어 FMS PDU 를 전송한다.

(3) Abort 서비스

Station 간의 통신에 있어서 데이터 전송이 끝나거나 데이터 전송 시 에러가 발생하였을 경우 communication termination 을 수행하여야 한다. Communication termination 은 사용자 및 LLI 부계층에서 수행가능하다. LLI 부계층에서의 communication termination 은 protocol, context, life time 및 link layer에 에러가 발생하였을 경우에 수행한다.

3.2 데이터링크 계층

1) 설계시 제한사항

데이터링크 설계에 앞서 효율적인 개발을 위하여 몇가지 제한사항을 정하였다.

(1) 상위 계층에 제공하는 서비스는 SDN, SDA, RDR 서비스를 기본으로 한다.

지금까지 제안된 국제 표준안들은 각 계층간의 조정이 없이 별도로 작성되고 있어서, 응용계층이 기대하는 데이터링크 서비스와 데이터링크계층이 응용계층에 제공하고자 하는 서비스에 차이가 있다. EFbus spec. 에서도 이러한 문제점이 일부 존재하므로 우선 응용계층이 필요로 하는 서비스를 제공하는 것에 주안점을 둔다.

(2) 계층관리(Layer Management) 기능은 가능한 축소하여 설계한다.

표준안에서 계층관리 기능에 대해서는 거의 언급이 없는 상황이므로 EFbus spec 에서도 이를 명확히 정의하지 않았다. 따라서 이번 설계에서는 구현중에 시험을 위하여 꼭 필요한 정도의 기능만을 포함 시키도록 한다.

(3) 제어기기 노드는 BA기능이 포함된 형태로 설계한다.

BA를 포함하는 제어기기 노드는 모든 데이터링크 기능을 다 포함하므로 이 노드에 필요한 데이터링크가 개발되면 다른 노드들은 각 노드에서 제공하지 않는 서비스를 제외하면 된다. 또한, BA를 stand alone 형태로 개발할 경우 생기는 H/W 개발업무 추가 부담도 없다.

2) 데이터링크 계층 설계내용

(1) Protocol S/W 동작 환경

데이터링크 계층의 protocol S/W 는 iDCX96 real time multitasking executive상에서 동작한다. EFbus 규격에서 동시에(concurrently) 수행되는 것으로 기술된 entity들은 iDCX96 의 task 가 된다. iDCX96은 이들 task 를 우선순위(priority)와 관련 event에 따라 scheduling하고, task간의 통신수단을 제공한다.

(2) Protocol S/W

EFbus 에는 producer/consumer entity 와 BA 라는 두가지 종류의 entity 가 존재한다. Producer/consumer entity는 모든 데이터를 송 수신하며, frame type 분석기능, 송 수신 대상 데이터들의 identifier table, 데이터의 기록 보관기능들을 갖는다. BA는 다른 entity들의 medium access를 제어하며, link scheduling 을 위한 scan table 관리기능, medium access 권한을 부여할 다음 entity 결정 기능 및 frame type 분석기능을 갖는다. 이 두가지 entity 를 모두 포함하는 protocol S/W 는 다음과 같은 4 개의 task 모듈로 구성된다.

-응용계층 인터페이스 task

Producer/consumer entity에서 응용계층으로 부터 요구 primitive 들 받아서 그 내용을 분석 하고, 물리계층 인터페이스 task 가 필요한 처리를 수행할 수 있도록 데이터를 해당 버퍼에 보관한다. 또한, 이러한 요구의 처리결과나 새로운 데이터의 도착 등을 응용계층에 알리는 기능도 수행한다.

-물리계층 인터페이스 task

Producer/consumer entity와 BA 에서 공통적으로 수행하는 frame type 분석이 주된 기능이며, 분석 결과에 따라 응용계층 인터페이스 task 가 보관해 둔 데이터로 PDU를 만들어서 remote 로 전달하거나, remote로 부터 수신한 데이터를 버퍼에 보관한다.

-Macro cycle 관리 task

BA entity 의 scan table 관리기능중 주기적 데이터교환, 비주기적 데이터교환, 메시지교환 및 전체 cycle의 timing 조절을 위한 stuffing 으로 구성되는 macro cycle을 관리한다.

-Elementary cycle 관리 task

BA entity 에서 macro cycle관리를 제외한 모든기능을 수행하며, 이 기능들은 macro cycle 관리 task 가 정해주는 time slot에 맞게 수행된다.

3) Task 들 간의 통신

Task들 간의 통신은 공유메모리 사용과 메시지(message) 교환 방식을 이용한다. Task간에 사용된 통신방식은 그림 3.3과 같다. 그림에서 원은 task 를 나타내며, 네모는 공유메모리를 의미한다.

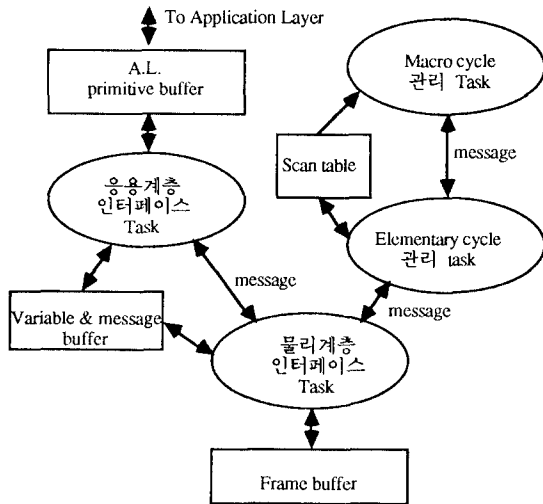


그림 3.3 Task간 통신구조

3.3 물리 계층

EFbus 의 물리 계층은 전송매체와 접속되는 부분으로 IBM-PC(또는 필드기기) 와 EFbus 사이의 하드웨어 접속을 제공한다. EFbus 인터페이스 보드는 Processor부(8096BH one-chip microcontroller) 및 메모리부, PC(또는 필드기기) 인터페이스부, medium 인터페이스부로 구성되어 진다. 그러나 본 논문에서는 EFbus 물리 계층과 직접적인 관련이 있는 medium 인터페이스부에 대해서만 언급하기로 한다. Medium 인터페이스는 EFbus의 물리 계층으로 IBM-PC(또는 필드기기) 와 medium 간에 데이터의 송수신을 담당한다. 이를 위한 기본회로의 구성은 그림 3.4 와 같다.

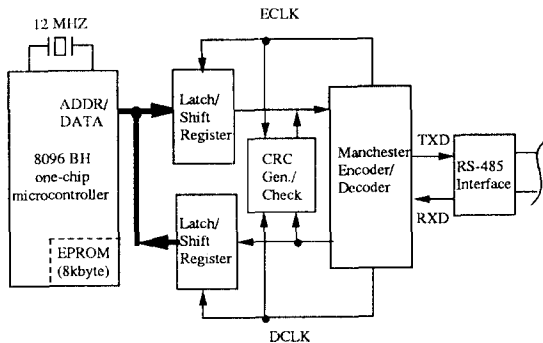


그림 3.4 EFbus의 물리 계층 구성도

위와 같이 medium 인터페이스 부분은 latch/shift register, CRC generator/ checker, Manchester Encoder/Decoder(MED), RS-485 인터페이스 및 이의 주변회로들로 구성되어 있다.

latch/shifter register 는 8 비트 병렬 입력 데이터를 직렬 출력 데이터로, 또는 직렬 입력 데이터를 병렬 출력 데이터로 바꾸어 주는 역할을 한다.

CRC generator/checker 는 직렬 디지털 데이터 처리 시스템

에서 데이터 송수신시 에러 점검을 위한 곳으로 데이터 전송 완료후 CRC checker frame 이 발생되고 데이터 수신 완료후 에러 여부를 check하게 된다.

MED는 EFbus에서 데이터링크 계층의 데이터를 물리 계층으로 전송하기 전에 디지털 신호를 coding 하는 부분이다. 이 coding 방식은 전송신호에 clock 신호를 함께 전송하는 방법으로 직렬 데이터 통신에서 MED는 nonreturn-to-zero(NRZ) 데이터를 manchester coder 로 변환하고 manchester coder 를 NRZ 로 decoder 한다.

EFbus 에서 데이터 송수신시 데이터의 전송속도는 31.25 kbps 에서 동작 하도록 한다. EFbus 에서 사용되는 전송매체는 네트워크에서 transmitter와 receiver간의 물리적 경로로서 shielded twisted pair 케이블을 사용한다. 이 케이블과의 인터페이스는 EIA의 직렬 통신에 사용되는 전기적 규격인 RS-485 방식을 사용한다.

- Bus 방식 : multi-drop
- 최대 접속 노드수 : 32 개
- 전송매체 : shielded twisted pair line
- Termination : 120 Ω
- 인터페이스 : SN75174/175 line driver/receiver

4. EFbus 네트워크시스템 구성

EFbus 네트워크시스템은 그림 4.1과 같이 감시/ 제어기능을 수행할 IBM PC, 이 PC를 EFbus 에 접속시키는 PC 인터페이스 보드, Field 기기를 EFbus 에 접속시키는 Field 기기 노드 및 이들을 연결하는 EFbus 네트워크로 구성된다.

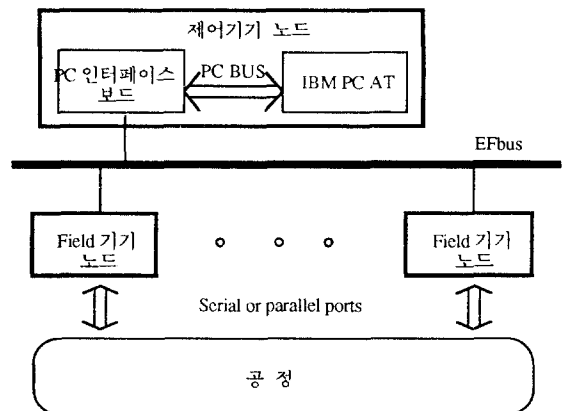


그림 4.1 EFbus 네트워크 시스템

4.1 제어기기 노드

제어기기노드의 하드웨어는 IBM PC 와 PC 에 내장되는 PC 인터페이스 보드로 구성된다. PC 인터페이스 보드는 자체의 local processor, local memory, PC와의 데이터 교환을 위한 공유 메모리 및 EFbus를 연결하는 medium 인터페이스 회로를 가지며, EFbus protocol S/W를 firmware 로 탑재하고 있다. Local

processor 로는 16 bit single chip 마이크로제어기인 8096BH를 사용하며 PC와의 통신용인 공유메모리는 dual port RAM을 사용하여 구성된다. Medium 인터페이스 부분은 physical medium으로 부터 데이터를 송수신하는 부분으로 Manchester encoder/decoder, CRC, generator/checker, line driver/receiver등을 포함하며, EFbus 에서의 physical layer이다.

제어기기 노드의 소프트웨어는 제어 알고리즘을 수행하는 응용 소프트웨어와 EFbus 를 통한 통신을 수행하는 protocol 소프트웨어로 구성된다. Protocol 소프트웨어는 노드 특성상 규모가 비교적 클 것이 예상되므로, PC 인터페이스 보드와 PC에 분산, 수행된다. PC 인터페이스 보드에서는 Intel 사의 8096 용 executive인 iDCX96 상에 EFbus protocol 소프트웨어 중 데이터링크계층 소프트웨어가 수행되고 PC에서는 MS-DOS상에서 응용계층 소프트웨어와 응용소프트웨어가 수행된다.

4.2 Field 기기 노드

하드웨어는 제어기기 노드에서의 PC 인터페이스 보드와 유사하다. 노드상에는 PC 인터페이스 보드에서와 같이 자체의 local processor, local memory를 가지며, EFbus 를 연결하는 medium 인터페이스 회로도 가지고, EFbus protocol S/W를 firmware 로 탑재하고 있다. 차이점은 IBM PC 에 연결되던 부분이 field 기기에 연결되므로, field 기기에 따라 인터페이스가 다르게 구성된다는 것이다. 소프트웨어는 DCX96, EFbus protocol 소프트웨어 및 응용이 모두 노드 상에 탑재될 수 있어야 하고, field 기기가 일부 intelligence 를 가질 경우 응용은 field 기기가 담당할 수도 있다. Protocol 소프트웨어는 각계층의 conformance level 에 맞게, 접속되는 field 기기가 필요로하는 기능만을 포함한다.

5. 결 론

본 논문에서는 현재 Fieldbus 국제 표준화 기관인 IEC SC65C 와 ISA SP50위원회 등에서 수행중인 Fieldbus 표준화 등향분석 및 이미 국제표준으로 결정된 Minimap 네트워크의 서비스와 프로토콜을 분석한 후 결정한 EFbus규격을 바탕으로 하여 EFbus 네트워크의 구현을 위한 계층별 설계내용을 기술하였다. EFbus는 process control 및 discrete manufacturing 분야에 응용되며, 이 응용 분야의 네트워크 적용면에서의 특징은 전달되는 데이터의 크기가 작고 속도가 빠르며, 또한 기기간에 전달되는 데이터는 일정한 주기를 갖고 cyclic 하게 전달되는 것이다. 따라서 응용계층에서 설계된 내용의 특징은 기존의 MMS 일부서비스에 cyclic 서비스와 broadcast 서비스의 추가이고 데이터 링크 계층의 특징은 centralized MAC 방식과 producer-consumer 개념의 데이터 교환방식의 사용이며, 물리계층의 특징은 Manchester encoding의 데이터전송방식과 RS-485 방식을 사용한 twisted pair cable 의 사용이다. 추후에 EFbus네트워크 시스템의 개발이 완료되면 공장자동화의 궁극적인 목표인 CIM 시스템 체계 구축 시 최하위 레벨의 센서 및 액츄에이터의 네트워크접속이 용이하게 되며, 공장용 네트워크 구성에 필요한 PD, 센서, 액츄에이터 등의 제작 표준화를 유도할 수 있

다. 그리고 향후 과제로 물리계층과 데이터 링크계층의 네트워크 S/W 및 H/W가 하나의 소형 chip으로 제작되어야 하며, 또한 network management 기능과 optical fiber medium사용에 대한 연구 및 Fieldbus 에 맞는 새로운 ASN.1 encoding rule에 대한 연구가 수행되어야 한다.

참고문헌

- [1] Leo Sintonen, "The Hierarchy of Communication Networks in the Programmable Assembly Cell : an Experimental Framework," IEEE Network, May 1988, pp 48-54
- [2] P. Pleinevaux 외, "Time Critical Communication Networks : Field Buses," IEEE Network, May 1988, pp 55-63
- [3] G. Wood, "Industrial LAN Architectures for Factory and Process Automation," European Conference on industrial LANs, November 1988, Brussels
- [4] 박기현 외, "공장자동화를 위한 인터페이스 기술 개발에 관한 연구," 최종연구보고서, 과학기술처, 10. 1989
- [5] G. Wood, "Progress in Fieldbus Standardization during 1989," Foxboro, UK
- [6] 황선호 외, "EFbus Specification," 한국전자통신연구소, TM, 4. 1990