Foundation Fieldbus에서 주기적 데이터 스케듈링 기법 및 실험적 평가

A Scheduling Method of Periodic Data in the Foundation Filedbus and Experimental Evaluation

송 승 민, 홍 승 호*
(Sung Min Song and Seung Ho Hong)

Abstract : This paper experimentally validates the efficacy of scheduling method that satisfies the performance requirement of periodic data in the Foundation Fieldbus. The scheduling method allocates periodic data traffic to a bandwidth-limited fieldbus medium. The scheduling method generates schedule list that records schedule starting time and schedule period of each periodic data. This study developed an experimental model of a Foundation Fieldbus network system. Using the experimental model, this study showed that the scheduling method can be easily implemented in a practical Foundation Fieldbus network system. The results obtained from the experimental evaluation showed that the scheduling method restricts the delay of periodic data to a pre-specified bound.

Keywords : Foundation Fieldbus, scheduling method, periodic data, experimental model, performance evaluation

I. 서론

필드버스는 자동화 및 공정제어 시스템의 필드에 설치된 각종 센서, 제어기, PLC, 모터, 밸브 등의 단위 필드 장비와 이러한 장비들을 제어 및 관리하는 컴퓨터 기반의 자동화 기기에서 생성되는 데이터를 실시간으로 전송하는 통신망이다. 필드버스의 응용 프로세스에서 생성되는 데이터는 크게 비주기적 데이터와 주기적 데이터로 구분된다. 비주기적 데이터에는 각종 사건이나 경고를 전달하기 위한 메시지와, 프로그램 파일 및 데이터를 전달하기 메시지 등이 포함된다. 주기적 데이터는 주로 피드백 제어시스템에서 센서 데이터를 주기적으로 샘플링하거나 주기적인 명령을 액튜에이터에 전달하기 위해서 사용되며, 반드시 데이터 생성 주기의 제한 시간 이내에 데이터 전송이 완료 되어야하는 특징을 가지고 있다. FF(Foundation Fieldbus)[1-7]는 사용자계층에서 요구하는 주기적 및 비주기적 데이터의 전송 서비스를 모두 지원하도록 고안된 프로토콜이다. FF의 데이터링크 계층은 이러한 사용자계층에서 요구하는 통신서비스를 수행하기 위하여 스케듈링과 토큰-패싱에 의한 데이터 전송 방식을 지원한다. 선행 연구[8]에서는 FF의 데이터링크계층 프로토콜에서 산발적 및 주기적으로 발생하는 실시간 데이터의 통신 요구사항을 만족하는 동시에 통신망의 대역폭을 충분히 활용하는 대역폭 할당 기법을 제시하였다. 선행 연구에서는 그러나 데이터링크계층까지 만으로 구성되는 FF 통신망 시스템을 대상으로 하였으며, 대역폭 할당 기법을 실제 FF 통신망의 사용자 환경에서 어떻게 구현할 수 있는가에 대한 문제는 추후 연구과제로 남아있었다. 실제 산업 현장에서 대역폭 할당 기법이 활용될 수 있도록 하기 위하여서는 사용자 환경에서 이러한 기법을 구현할 수 있는 방안이 제시되어야 한다. 본 연구에서는 실제 산업 현장에서 사용되는 FF 통신망 장비를 이용하여 실험모델을 구성하고, 이를 통하여 대역폭 할당 기법 가운데 주기적 데이터의 스케듈링 기법을 실제 시스템에서 구현하는 방법을 제시하였다. 본 연구의 실험모델을 통하여 얻어진 결과에 의하면 스케듈링 기법은 FF 통신망 시스템에서 주기적 데이터 지연시간 요구사항을 만족시키는 것으로 나타났다.

본 논문의 전체 구성은 6장으로 되어있다. 2장에서는 FF 프로토콜[1-7]의 구조에 대하여 간략히 기술하였으며, 3장에서는 대역폭 할당 기법 가운데 스케듈링 기법을 요약하여 기술하였다. 4장에서는 사용자계층에서 스케듈링 기법을 구현하기 위한 실험모델의 구성과 이를 이용한 스케듈링 기법 구현 방법을 제시하였다. 5장에서는 실험 결과를 통하여 스케듈링 기법의 타당성을 검증하였으며, 마지막 6장에서는 본 논문의 결론이 기술되었다.

II. FF 프로토콜 구조

FF는 물리계층, 통신 스택, 사용자 계층으로 구성된다. 통신 스택은 다시 데이터 링크 계층과 응용 계층으로 나뉘어진다. 그림 1에는 FF 통신망의 계층 구조가 나타나 있다.

FF의 물리(PHY) 계층은 IEC의 국제 표준 규격[9]을 따른다. 코딩 방법은 Manchester Biphase-L 기술을 사용하고, 31.25 Kbps, 1.0 Mbps, 2.5 Mbps의 세 가지 전송속도를 지원한다. 31.25 Kbps는 레벨이나 온도제어 같은 비교적 정적인 시스템에 사용되며 필드버스에게 내부 보호 기능을 지원한다. 1.0 Mbps, 와 2.5 Mbps는 고성능 프로세서 제어, 원격 입·출력 그리고 빠른 속도를 필요로 하는 자동화 시스템에 사용된다.

* 책임저자(Corresponding Author)
논문접수 : 2004. 1. 7., 채택확정 : 2004. 10. 5.
송승민 : LG전자 UMTS 시스템 연구소(smsong@lge.com)
홍승호 : 한양대학교 전자컴퓨터공학부(shhong@hanyang.ac.kr)
※ 본 논문은 과학재단 목적기초연구(R01-2002-000-00046-0) 지원으로 수행되었습니다.

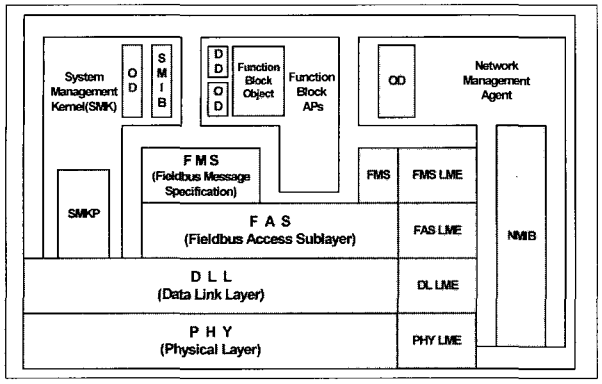


그림 1. FF 프로토콜의 계층 구조.
Fig. 1. Layered structure of FF protocol.

데이터링크(DLL) 계층[6,10]에서 프레임의 미디엄 접속 제어는 LAS(Link Active Scheduler)라고 불리는 버스 스케듈러에 의해서 관리된다. 데이터 링크 계층의 모든 메시지는 DLPDU(Data Link Protocol Data Unit)의 형태로 전송된다. 메시지 전송 방법은 PT(Pass Token) DLPDU를 사용하는 토큰 패싱과 CD(Compel Data) DLPDU를 사용하는 스케듈링의 두 가지 방식을 제공한다. 토큰 패싱 방식에서는 LAS가 미리 TCL(Token Circulation List)에 정해진 순서에 따라 차례로 PT(Pass Token) DLPDU를 일반 노드에 전송하고, 일반노드는 토큰을 소유하는 동안 데이터를 전송한 후 RT(Return Token) DLPDU를 이용해서 LAS로 토큰을 반환한다. 스케듈링 방식에서는 시스템 관리자에 의해서 미리 입력된 시간이 되면 LAS는 해당 노드에 CD DLPDU를 전송해서 미리 스케듈된 데이터가 전송될 수 있도록 하는 권한을 부여한다. CD DLPDU를 수신한 노드는 하나의 스케듈 데이터를 전송하고 RT DLPDU를 이용해서 LAS에 토큰을 반납하게 된다.

FF 프로토콜의 규격서에는 데이터링크 계층에서 처리되는 데이터의 우선순위를 긴급(urgent), 보통(normal), 시간허용(time-available)의 세 가지로 구분하며, 우선순위에 따라 데이터의 길이가 제한된다.

FAS(Fieldbus Access Sublayer) 계층은 데이터링크 계층의 토큰-패싱과 스케듈링의 두 가지 데이터 전송 방식을 상위 계층에서 요구하는 통신서비스에 매핑한다. FAS의 서비스 형태는 VCR(Virtual Communication Relationship)이라는 단축키 형식으로 제공한다. VCR의 종류에는 PT DLPDU를 사용하는 Client/Server 및 Report Distribution 서비스와 CD DLPDU를 사용하는 Publisher/Subscriber 서비스가 있다.

FMS(Fieldbus Message Specification)는 응용계층의 통신 서비스를 표준화된 방식으로 제공하기 위한 프로토콜이다. 필드버스를 통해서 교환되는 정보는 객체로 표현되며, 이 객체 표현을 구조화시킨 것이 객체 사전이다. 객체 사전은 가상 필드 장비에 등록된다. 가상 필드 장비는 실제 필드 장비의 자원과 기능을 추상화된 객체로 정의해 놓은 것으로 노드간의 데이터 교환은 이를 통하여 이루어진다. FMS가 제공하는 통신 서비스에는 가장 필드 장비의 관리, 연결 관리, 객체 사전 관리, 변수 접근, 사건 관리, 영역 관리, 프로그램 기동 관리 등이 있다. FF의 사용자 계층은 기능 블록(Function Block), 시스템 관리(System Management), 네트워크 관리(Network Management)의 세가지로 구성된다.

기능 블록은 다시 자원 블록(Resource Block), 기능 블록(Function Block), 트랜스듀서 블록(Transducer Block)의 세 가지로 나뉜다. 자원 블록은 필드버스에 접속되는 필드 장비의 하드웨어적인 특성을 나타내기 위해 각종 파라미터들을 정의해 놓은 부분이다. 기능 블록은 공정제어 시스템에서 자주 사용되는 기능들을 블록화 시켜 놓은 것으로써, 장비들 간의 기능블록 입출력 매개변수는 필드버스를 통하여 서로 연결될 수 있다. FF의 응용 프로세스에 연결된 각각의 자원들은 하나 이상의 기능블록을 가질 수 있으며, 기능블록은 다른 기능블록의 출력 또는 입력과 연결됨으로써 상호간에 정보를 교환하고, 이렇게 수신한 정보들을 처리하여 같은 자원내의 트랜스듀서 블록 채널로 출력한다. 기능블록의 실행은 시스템 관리(System Management)에 의해 스케듈링 되거나 혹은 다른 기능블록의 처리 종료에 연결되어 동작될 수 있으며 제조업자가 미리 지정할 수도 있다. 트랜스듀서 블록은 센서 값을 읽거나 출력하는 하드웨어로의 명령에 필요한 입·출력 함수로부터 기능블록을 보호하는 역할을 한다. FF 프로토콜은 각 디바이스에 대한 시스템 관리를 위해 SMK(System Management Kernel)를 정의하고 있다. SMK는 디바이스 어플리케이션의 상호동작 및 실행을 위한 분산 환경에 필요한 기본 정보들을 조정하고 관리하는 기능을 수행한다. SMK에 의해 유지되는 기본 정보로는 물리적인 태그, 디바이스 ID, 어플리케이션 클럭, 기능블록의 스케듈 등이 있으며, 이러한 정보들은 SMIB(System Management Information Base)로 정의되고 객체 사전을 이용하여 기술된다. 시스템 관리기능은 기능블록의 실행과 필드버스 상에서 일어나는 기능블록 파라미터들 간의 통신을 동기화 시키고, 날짜 정보 생성기능, 디바이스 주소의 자동할당, 필드버스상의 매개 변수나 태그 검색 등의 서비스를 수행한다.

FF는 별도의 네트워크 관리자를 이용하여 디바이스의 통신 시스템에 대한 관리 기능과 NMIB(Network Management Information Base)를 이용한 통신 스택에 대한 정보 제공 및 네트워크로의 접근 기능을 제공한다. NMIB는 관리대상 변수의 집합으로써 디바이스 내의 통신시스템에 대한 설정, 성능에 대한 정보 등과 전체 통신스택에 대한 정보들이 저장되어 있다. 네트워크 관리는 설정 관리, 성능 관리, 오류 관리의 기능을 제공한다.

III. FF에서 주기적 데이터의 스케듈링 기법

FF의 데이터링크 계층에서는 주기적 데이터의 전송을 지원하기 위하여 스케듈링에 의한 데이터 전송 방식을 제공한다. 그러나 프로토콜 규격서[10]에는 주기적 데이터의 전송을 위한 스케듈의 시작 시간과 주기가 정해진 경우에 LAS가 CD DLPDU를 통하여 이들을 전송하는 방식에 대하여서만 명시되어 있을 뿐이며, 주기적으로 생성되는 데이터의 성능 요구사항을 만족시키기 위하여 LAS에서 각 노드에 대한 스케듈의 시작시간과 주기를 어떻게 설정하여야 하는 가에 대하여서는 기술되어 있지 않다. 즉, 스케듈의 시작 시간과 주기의 결정은 필드버스 통신망의 설계자가 알아서 설정

하여야 할 통신망 파라미터로 주어져 있다. 대역폭 할당 기법[8] 가운데 스케듈링 기법은 주기적 데이터의 최대 허용 지연시간이 주어진 경우에 이에 대한 성능 요구사항을 만족시키도록 하는 기법이며, 본 장에서는 이를 간략히 소개한다. 대역 할당 기법에서는 FF의 대역폭을 주기적 데이터들의 생성 주기 가운데 최소 생성 주기인 T_1의 시분할 구간으로 나누고, 이를 다시 주기적 구간들과 비주기적 구간들로 나눈다. 주기적 구간에서는 미리 주기적으로 생성되도록 스케듈된 데이터만이 스케듈링 방식에 의하여 전송되며, 비주기적 구간에서는 산발적으로 발생되는 데이터가 토큰-패싱 방식에 의하여 전송된다. FF의 데이터 링크 계층에서는 토큰-패싱 방식으로 동작되는 도중에 스케듈 데이터를 전송해야 하는 시간이 되면 주기적 데이터가 우선적으로 전송되며, 따라서 비주기적 데이터의 전송은 일시 중지된다. 다음은 본 논문에서 사용되는 주요 기호들이다.

N_p : 주기적 데이터 생성 노드의 개수

L_p : 주기적 데이터의 전송시간 (*ms*)

N_a : 비주기적 데이터 생성 노드의 개수

L_a : 비주기적 데이터의 전송시간 (*ms*)

T_i : 노드 i에서 주기적 데이터의 발생 주기

ϕ_i : 노드 i의 주기적 데이터의 최대 허용 지연시간

k_i : 최소 생성 주기인 T_1에 대한 T_i의 비

γ : 최소 생성 주기인 T_1동안 생성되는 주기적 데이터 개수의 제한값

σ : LAS가 한 노드에서 토큰-패싱 또는 스케듈링 서비스를 처리하는데 소요되는 오버헤드 시간

R : LAS가 링크 내의 모든 노드를 한번 순환하는데 걸리는 오버헤드 시간

그림 1에는 FF에서 대역폭 할당기법을 통하여 필드버스의 대역폭을 할당하는 예가 나타나 있다. 스케듈링 기법의 기본개념은 N_p 개의 주기적 데이터를 생성하는 노드가 γ ($\gamma \leq N_P$)개의 윈도우를 서로 동적으로 공유하도록 하여 T_1 구간 동안 전송되는 주기적 데이터의 개수가 γ 개를 초과하지 않는 동시에 데이터의 전송지연시간이 최대 허용 지연 시간인 ϕ_i를 초과하지 않도록 각 노드에서 데이터의 생

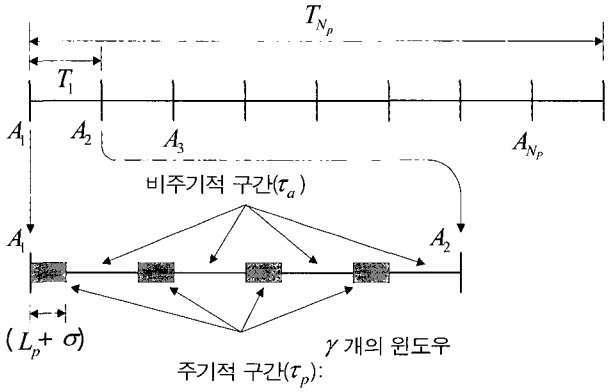


그림 2. FF에서 대역폭 할당기법의 예.

Fig. 2. Bandwidth allocation scheme in the FF.

성 주기인 $T_i(i=1$에서 $N_p)$와 각 노드에서 첫 번째 데이터의 생성시간 t_i ($i=1$에서 N_p)를 결정하는 것이다. LAS에서는 t_i와 T_i의 정보를 이용하여 CD DLPDU를 분배하는 스케듈 시작 시간 s_i와 스케듈 주기 S_i를 결정하며, 이는 LAS가 가지고 있는 스케듈 리스트에 기록된다.

그림 2에서 주기적 구간 τ_p는 하나의 스케듈 데이터를 전송할 수 있는 윈도우로 구성되며, 윈도우의 크기는 주기적 데이터의 전송시간(L_p)과 주기적 데이터의 전송을 위한 LAS에서의 토큰 처리 시간(σ)의 합이다($\tau_p = L_p + \sigma$). 하나의 윈도우가 끝나게 되면 비주기적 구간 τ_a가 시작되며, 비주기적 구간에서는 새로운 윈도우가 시작될 때까지 토큰-패싱 방식에 의한 비주기적 데이터의 전송이 이루어진다. FF의 스케듈링 방식에서는 주기적 데이터의 전송 시간을 미리 적절히 스케듈링할 수 있는 기능이 지원되므로, 그림 2에서 보는 바와 같이 γ 개의 윈도우를 연속적으로 할당하는 것보다는 T_1의 주기 내에 골고루 분산시키는 것이 비주기적 데이터의 전송 지연시간에 대한 성능을 향상시킨다.

주기적 데이터를 생성하는 N_p개의 노드에서 최대 허용 지연 시간은 다음의 벡터 Φ로 주어진다.

$$\Phi = [\phi_1, \phi_2, \cdots, \phi_{N_p}], \quad \phi_i \leq \phi_{i+1} \tag{1}$$

또한 N_p개의 노드에서 생성되는 주기적 데이터의 생성 주기를 원소로 하는 벡터 T는 다음과 같이 정의된다.

$$\boldsymbol{T} = [T_1, T_2, \cdots, T_{N_p}], \quad T_i \leq T_{i+1} \tag{2}$$

여기서 T 와 Φ는 오름차순으로 정렬한다(즉, $T_i \leq T_{i+1}$, $\phi_i \leq \phi_{i+1}$). 그림 2에서 T_1은 N_p개의 노드 중에서 최소 생성 주기이고 T_{N_p}는 최대 생성 주기이다. 대역폭 할당기법으로부터 각 노드에서 스케듈 데이터의 생성 주기 T_i는 다음과 같이 결정된다[8].

$$T_1 = \phi_1, \quad T_i = k_i T_1, \quad k_i = 2^{\log_2(\phi_i / T_1)}, \quad \forall i = 2 \text{ 에서 } N_p \tag{3}$$

(3)으로부터 T_i는 최대 허용 지연 시간 ϕ_i을 초과하지 않는 범위 내에서 모든 스케듈 데이터의 생성 주기가 서로 배수의 관계를 갖는다. 즉,

$$\mathrm{Rem}\left[\frac{T_j}{T_i}\right] = 0, \ \forall i, j \quad (j \geq i) \tag{4}$$

따라서 네트워크 대역폭 할당은 T_{N_p} 주기로 반복된다. T_1의 주기 내에서 윈도우의 개수 γ은 다음과 같이 구할 수 있다.

$$\gamma = \lceil a_k \rceil, \quad a_k = \sum_{i=1}^{N_p} \frac{1}{k_i} \tag{5}$$

여기서 a_k는 T_1동안 생성되는 스케듈 데이터 개수의 평균값이다. T_1의 구간 동안 생성되는 주기적 데이터의 개수가 γ를 초과하지 않도록 하기 위하여 대역폭 할당기

법은 각 노드의 첫 번째 데이터 생성 시간 t_i를 적절히 선정한다. 그림 2에서 A_l (l=1 에서 N_p)을 T_{N_s} 구간 내에서 $l-$번째 T_1 슬롯이 시작되는 시점이라고 하자. 주기적 데이터를 생성하는 각 노드는 자신의 첫 번째 데이터가 생성되어야 할 시간인 t_i를 A_l에 대응시킨다. t_i는 대역폭 할당 기법으로부터 다음과 같이 결정된다.

$$t_i = \inf\,[A_l \geq A_{l-1}\,;u^i(A_l) \leq \gamma],$$

$$i=1 \text{ 에서 } N_p,\quad l=1 \text{ 에서 } T_{N_p}/T_1 \tag{6}$$

여기서 t_i ($t_i \leq t_{i+1}$)는 $i=1$번 노드부터 시작하며, $u^i(A_l)$은 i-번째 노드에서 생성된 주기적 데이터를 포함하여 A_l의 순간에 생성된 주기적 데이터의 개수이다. (6)의 조건은 T_1 구간 내에서 생성되는 주기적 데이터의 개수가 γ를 초과하지 않도록 제한한다. 대역폭 할당기법에서 T_1의 구간동안 γ개의 주기적 구간과 비주기적 구간이 존재하므로 주기적 데이터의 안정화 조건은 다음과 같이 주어진다.

$$\gamma(\tau_p + \tau_a) \leq T_1 \tag{7}$$

만일 이 조건이 만족되지 않는 경우에는 필드버스 통신망 설계자는 주기적 데이터를 생성하는 노드의 개수 N_p를 줄여야 한다. 네트워크 설계자는 주기적 데이터를 생성하는 노드 i의 첫번째 데이터 생성 시간인 t_i와 데이터 생성 주기인 T_i를 기반으로 하여 LAS의 스케듈 리스트를 작성한다. 스케듈 리스트에는 주기적 데이터를 생성하는 각 노드에 CD DLPDU를 전송하기 위한 스케듈 시작 시간과 스케듈 주기가 기록된다. 그림 2에서 보는 바와 같이 스케듈 데이터는 T_1의 구간에서 골고루 분산되어 전송되므로 LAS에서 CD DLPDU를 분배하는 스케듈 시작 시간과 스케듈링 주기는 다음과 같이 결정된다.

$$s_i = t_i + \frac{T_1}{\gamma}[u^i(A_l) - 1] \tag{8}$$

$$S_i = T_i \tag{9}$$

LAS는 이에 따라 해당 노드에 CD DLPDU를 전송한다. 대역폭 할당기법을 적용하는 경우 모든 노드에서 생성되는 주기적 데이터는 $T_1(=\phi_1)$ 시간 이내에 전송이 완료되며 따라서 모든 노드에서 최대 허용 지연시간 요구사항 $\phi_i(\geq \phi_1)$i을 만족한다.

IV. 실험 모델 구성

본 연구에서는 FF의 사용자계층에서 대역폭 할당기법적용을 하기 위하여 실제 산업 현장에서 사용되는 필드버스 장비의 Starter Kit을 가지고 실험 모델을 구성하였다[11-15]. 본 장에서는 먼저 III 장에서 제시한 대역폭 할당 기법을 실제 산업 현장의 필드 디바이스에 어떠한 방법으로 적용하였는지에 대하여 기술한다. 대역폭 할당기법을 적용하기 위해서는 디바이스에 대한 정보 및 기능블록이 디바이스에 탑재되어 있어야 하므로, 우선적으로 이를 탑재하는 방법에 대

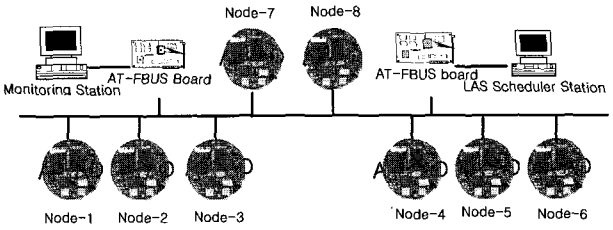


그림 3. 실험 모델의 구성도.

Fig. 3. Configuration of the experimental model.

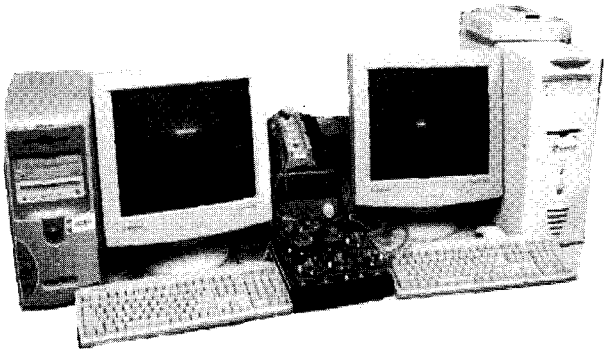


그림 4. 실험 모델의 사진.

Fig. 4. Picture of the experimental model.

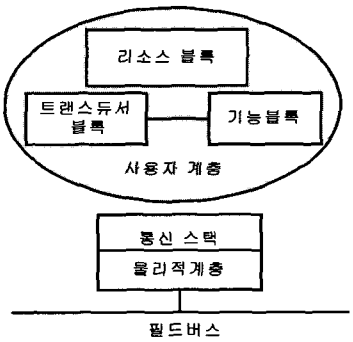


그림 5. FF 사용자 계층 구조도.

Fig. 5. Structure of FF user layer.

표 1. FF에서 정의된 표준 기능 블록.

Table 1. FF standard function blocks.

기능블록 이름	심 볼
Analog Input	AI
Analog Output	AO
Bias/Gain	BG
Control Selector	CS
Discrete Input	DI
Discrete Output	DO
Manual Loadder	ML
Proportional/Derivative	PD
Proportional/Integral/Derivative	PID
Ratio	RA

하여 알아보고, 탑재된 기능블록사이에서 발생되는 주기적 및 비주기적 데이터의 발생 방법에 대하여 기술한다. 또한, 기능블록의 설정 방법과 주기적 데이터 생성 주기와 시작 시간을 설정하는 스케듈 리스트 작성 방법에 대해서도 기술한다. 실험모델은 그림 3에서와 같이 2대의 PC와 8개의 round card로 구성된다. Round card는 센서 또는 액튜에이터에 바로 접속되어 하나의 노드를 구성할 수 있는 FF 통신망 접속 장치이다. 2대의 PC에는 FF의 PC 인터페이스 카드가 탑재되어 있으며, 이들은 각각 LAS 스케듈러 기능을 수행하는 마스터 노드와 트래픽을 모니터링 및 분석하는 기능을 수행하는 모니터 노드의 역할을 수행한다. 실험 모델에서 모든 노드들은 주기적 데이터와 비주기적 데이터를 생성하며, 각 노드에는 기능블록에 의한 데이터 생성프로그램이 탑재되어 있다. 또한 마스터노드는 스케듈링 뿐만 아니라 데이터 링크에 포함된 모든 노드에 시간 동기화시키는 기능까지 수행한다. 모니터링 노드에서는 CD DLPDU에 의한 스케듈 데이터의 생성시간과 PT DLPDU에 의한 비주기적 데이터의 생성시간을 1 μsec 단위까지 측정할 수 있다. 그림 4는 실제 실험 모델을 구성한 사진이다.

1. 주기적 및 비주기적 데이터의 발생

본 절에서는 주기적 및 비주기적 데이터의 발생 방법에 대하여 알아보고자 한다. FF의 스케듈링 기법에 대한 타당성을 검증하기 위하여서는 사용자계층에서 임의로 주기적 데이터를 발생시킬 수 있어야 한다. FF의 사용자계층에서는 네트워크에 접속되는 디바이스에서 사용되어지는 데이터와 디바이스 자체의 기능을 정의하기 위해서 블록과 오브젝트를 사용한다. 즉, 그림 5와 같이 FF의 사용자계층에서는 일반적 통신 프로토콜처럼 일련의 명령을 통해서 각 디바이스와 인터페이스하는 것이 아니라, 미리 정의된 블록과 오브젝트의 정의를 통해서 사용자계층과 인터페이스 하게 된다. 사용자계층의 리소스 블록은 제조업자나 디바이스이름과 같은 다바이스의 일반적인 정보를 갖는다. 트랜스듀서 블록은 입력단과 출력단 사이의 물리적인 연결, 데이터의 전달 기능 및 다양한 데이터 전처리 과정을 수행한다. 기능블록은 제어 시스템의 기능을 블록화 시켜놓은 것으로써, 기능블록의 입출력 매개변수는 필드버스를 통하여 서로 연결될 수 있다. 하나의 사용자 어플리케이션 내에는 여러 개의 기능블록이 존재 할 수 있다. 이러한 기능블록은 FF 프로토콜에서 중요한 요소로써 표 1과 같은 10개의 표준 기능블록을 정의 해놓고 있다. 본 연구에서는 상위계층에서 데이터 발생을 시키기 위하여 AI블록과 AO블록 정보를 코딩하여 탑재하였다. 이러한 트랜스듀서 블록, 리소스블록, 기능블록을 상위계층의 설정 툴의 창에 올리기 위해서는 디바이스 템플릿과 사용자 프로그램을 작성하고, Starter Kit에서 지원하는 Configuration 코드 생성기를 사용하여 C 코드로 변환 시켜준 후 컴파일, 링크 과정을 거쳐 라운드 카드의 플래쉬 메모리에 다운로드하는 일련의 과정을 수행한다. 이러한 단계를 거치면 각각의 라운드카드의 플래쉬 메모리에 디바이스 정보와 기능블록이 탑재되게 된다. 아래의 그림 6은 마스터 노드 설정 툴의 창으로써, 각 디바이스에 리소스블록, AI블록, AO블록, PID블록과 디바이스 정보가 탑재되었음을 보여준다.

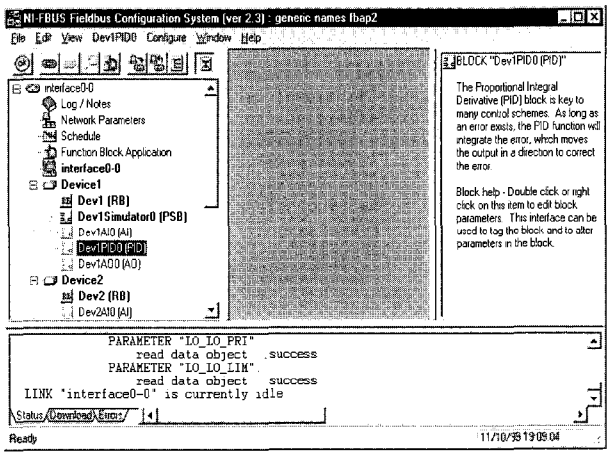


그림 6. 각 디바이스에 기능블록 탑재.

Fig. 6. Leading function block in each device.

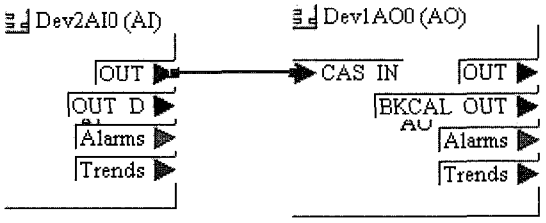


그림 7. 기능블록간의 연결.

Fig. 7. Connection between function blocks.

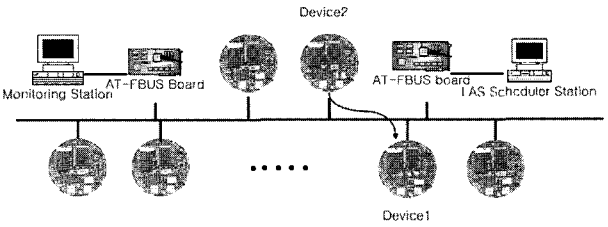


그림 8. 발생된 스케듈 데이터의 이동경로.

Fig. 8. Path of schedule data.

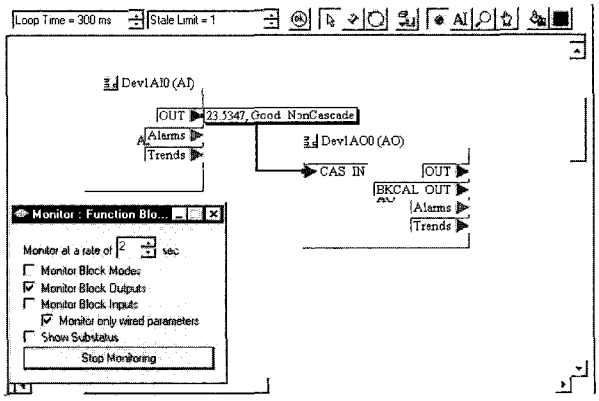


그림 9. 비주기적 데이터의 발생.

Fig. 9. Generation of aperiodic data.

FF 실험 모델의 사용자 계층에서 주기적 데이터를 발생시키기 위해서는 그림 7에서 보는 바와 같이 한 디바이스 내에서 기능블록간의 연결이 아닌 서로 다른 디바이스간에 기능블록이 연결되어야 한다.

그림 7은 Starter Kit의 configuration tool을 이용하여 디바이스_2의 AI블록과 디바이스_1의 AO블록을 연결함으로써, 이들 간에 스케듈 데이터를 발생시키는 절차를 보여주는 예이다. 그림 7과 같이 사용자계층에서 기능블록을 연결하여 주면 실제로는 아래 그림 8에서 볼 수 있듯이 8개의 디바이스 중 디바이스_2 에서 있는 AI 블록에서 디바이스_1 에 있는 AO 블록으로 주기적 데이터가 발생하게 되는 것이다. FF 프로토콜의 규격서에 의하면 데이터의 종류는 크게 주기적 데이터와 비주기적 데이터로 구분되며, 비주기적 데이터는 긴급(urgent), 보통(normal), 시간허용(time-available)의 세가지 우선순위를 갖는다. 그러나 본 연구에서 사용한 Starter Kit 장비에는 사용자가 임의로 비주기적 데이터를 생성할 수 있도록 하는 기능이 지원되지 않고 있다. 따라서 본 실험에서는 그림 9에서 보는 바와 같이 기능블록 AI블록과 AO블록이 연결된 상태에서 Function Block Monitor라는 프로그램을 동작시켜 비주기적 데이터를 발생시켰다. 즉, LAS의 기능을 수행하는 마스터 노드가 허브를 통하여 연결된 각각의 디바이스에 탑재되어 있는 기능블록의 입출력 값을 모니터링하는 기능을 수행하도록 함으로써 비주기적 데이터가 발생되도록 한 것이다. 이러한 비주기적 데이터는 PT(Pass Token) DLPDU에 의해서 전달된다.

2. 기능블록의 설정

본 절에서는 실험 모델에서 기능블록의 설정 방법에 대하여 기술한다.

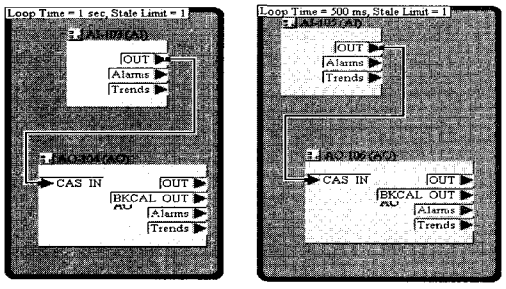


그림 10. 기능블록의 설정 방법.
Fig. 10. Creation of function block.

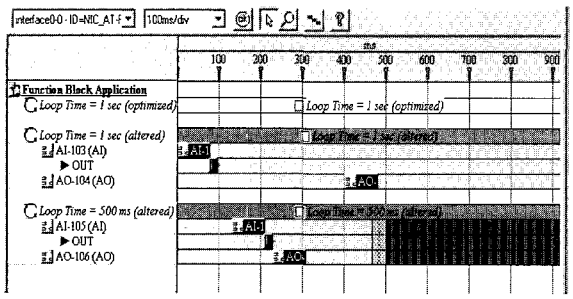


그림 11. LAS 노드에서 스케듈링 방법.
Fig. 11. Scheduling in the LAS node.

그림 10을 보면 현재 라운드 카드 4개에 각각 AI 블록과 AO블록이 탑재되어 있음을 볼 수 있다. 사용자는 원하는 디바이스의 블록을 설정 창에 배치시킨 후 이들을 선으로 연결하면 내부적으로 기능블록간에 연결이 자동적으로 이루어진다. 같은 디바이스가 아닌 다른 디바이스 사이에서의 기능블록 연결만이 주기적 데이터를 발생시킬 수 있으며, 기능블록을 결선하지 않으면 데이터는 생성되지 않는다. 기능블록에서 주기적 데이터의 생성 주기는 사용자가 임의로 설정할 수 있다. 그림 10의 구성에서 보는 바와 같이 왼편의 AI블록과 AO블록은 1sec 간격의 주기로 데이터가 발생하고, 오른편은 500msec의 주기로 데이터가 발생한다.

3. 스케듈링 방법

LAS 기능을 수행하는 마스터 노드는 각 노드에 CD DLPDU를 전송하기 위하여 스케듈 시작시간과 스케듈 주기를 설정하는 기능을 가진다. 그림 11은 스케줄 리스트를 설정하는 창이며, LAS 노드에서 스케듈링은 주기설정과 시작점을 1ms 단위까지 정확하게 설정할 수 있게 되어 있다.

V. 실험 결과 분석

본 장에서는 III장에서 제시한 스케듈링 기법을 실험모델의 사용자계층에서 구현하고 이에 대한 실험 결과를 제시한다. 본 실험에서 전송속도는 31.25Kbps이다. 스케듈 데이터의 길이는 39 바이트로 고정되었으며, 비주기적 데이터의 길이는 가변하나 최대 길이는 110 바이트이다. 본 연구에서 사용된 Fieldbus Device Starter Kit에서는 앞서 언급한 바와 같이 사용자가 비주기적 데이터를 임의로 생성시킬 수 있는 기능이 제공되지 않다. 따라서 본 연구에서는 주기적 데이터의 전송시간 L_a는 비주기적 구간 τ_a를 초과하지 않는 범위에서 수행되었다. 실제 전송되는 데이터의 전송시간은 실험으로 측정한 결과 주기적 데이터는 9.984ms이고, 비실시간 데이터는 최대 28.16ms의 전송시간을 가진다. 각 노드에서 토큰 오버헤드는 5ms로 측정되었다.

8개의 노드로 구성된 실험 모델에서 각 노드에서 생성되는 스케듈 데이터의 최대 허용지연시간은 Φ=[200, 200, 500, 500, 1000, 1000, 2000, 2000]msec로 설정되었다. 이러한 성능 요구사항에 대하여 III장에서 제시한 대역폭 할당 기법을 적용하여 각 노드에서 스케듈 데이터의 주기를 구하면 다음과 같다.

$$S_i = [200, 200, 400, 400, 800, 800, 1600, 1600]\text{ms}$$

또한 각 노드에서 스케듈 데이터의 첫 번째 스케듈 시작시간 s_i는 다음과 같이 구해진다.

$$s_i = [0, 50, 100, 150, 300, 350, 700, 750]\text{ms}$$

다음 그림 12는 주어진 S_i와 s_i에 대하여 각 노드에 대한 LAS의 스테듈 패턴을 나타낸 그림이다. 그림에서 보는 바와 같이 γ=4개의 윈도우가 T_1의 주기내에 균등하게 분산되었다. 이러한 스케듈 패턴은 T_{N_P}의 주기로 반복된다.

그림 13은 사용자계층에서 주기적 데이터와 비주기적 데이터의 지연시간을 측정한 결과이다. 그림 13에서 가로축은 미디엄을 통하여 전송되는 데이터에 부여한 일련번호이고, 세로축은 각 데이터의 전송 지연시간이다.

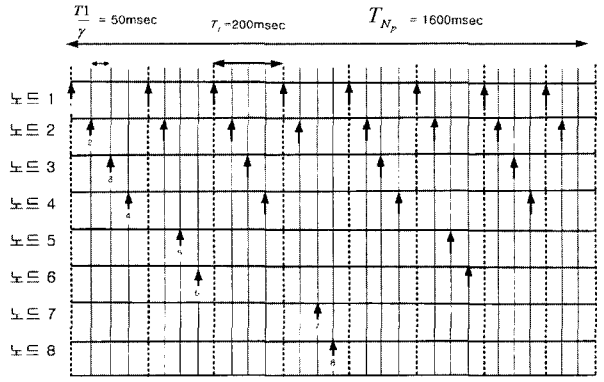


그림 12. 스케줄링기법을 적용한 경우 주기적데이터의 스케줄 패턴.

Fig. 12. Schedule pattern of periodic data.

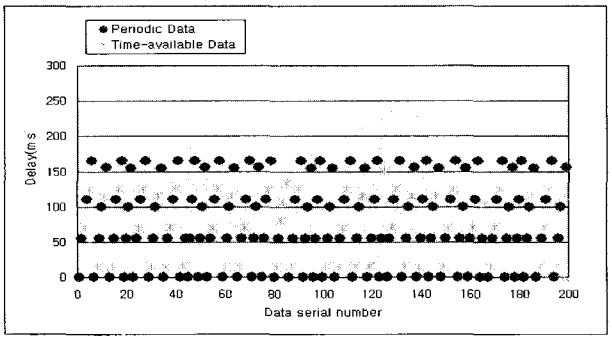


그림 13. 스케줄링기법을 적용한 경우 데이터 지연시간 측정.

Fig. 13. Delay in case when a scheduling method is applied.

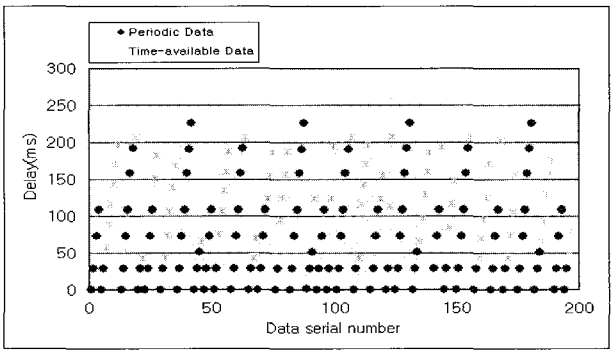


그림 14. 사용자 임의 스케줄을 적용한 경우 데이터 지연시간측정.

Fig. 14. Delay in case when an arbitrary scheduling is applied.

표 2. 스케줄링기법 적용여부에 대한 성능비교(단위 msec).

Table 2. Performance comparison of scheduling scheme.

	스케줄링 기법		사용자 임의	
주기적 데이터 지연시간	최대값	165	최대값	226
	최소값	0.2	최소값	0.2
	평균값	77.5	평균값	63.2
비주기적 데이터 지연시간	최대값	150.2	최대값	208
	최소값	15	최소값	44
	평균값	73.2	평균값	128.4

비주기적 데이터가 불규칙하게 생성되어 전송되는 환경에서도 주기적 데이터의 최대 지연시간은 T_1=200 ms를 초과하지 않으며, 따라서 모든 주기적 데이터 생성 노드에서 주기적 데이터의 지연시간에 대한 성능 요구사항을 만족한다. 비주기적 데이터의 지연시간 역시 주기적 데이터의 최대 지연시간을 초과하지 않으며 평균 지연시간은 평균 73.2msec 측정되었다. 스케줄링 기법의 타당성을 평가하기 위하여 다음에는 사용자가 임의로 주기적 데이터의 생성을 스케줄링 하는 경우에 대하여 실험을 수행하였다. 실험모델에서 8개의 노드에서 생성되는 스케줄 데이터의 최대 허용 지연시간은 이전의 실험과 동일하게 Φ=[200, 200, 500, 500, 1000, 1000, 2000, 2000]msec로 설정하였다. 두번째 실험에서는 스케줄 주기 S_i를 주기적 데이터의 최대 허용지연시간으로 설정하였고, 스케줄 시작점은 s_i=[0, 25, 50, 72, 100, 125, 148, 170]msec의 임의로 값으로 설정하였다. 그림 14에서 보는 바와 같이 주기적 데이터의 전송 패턴은 일정하지 않으며, 따라서 데이터 지연시간 또한 일정치 않음을 알 수 있다. 실험 결과를 보면 주기적 데이터의 지연시간이 첫 번째 노드의 최대 허용지연시간인 ϕ_1=200msec를 초과하는 경우가 발생한다. 따라서 첫 번째 노드에서는 실시간 데이터 전송의 요구사항을 만족시키지 못하는 경우가 발생할 수 있다. 비주기적 데이터의 지연시간 역시 스케줄링 기법을 적용한 경우와 비교하여 커짐을 알 수 있다. 이는 스케줄링 기법에서 주기적 구간을 전 대역폭에 결쳐 골고루 분산시키고, 이러한 주기적 구간의 사이에서 비주기적 데이터가 전송되도록 한 반면에, 스케줄링 기법의 적용되지 않은 경우에서는 토큰-패싱에 의하여 전송되는 비주기적 구간이 시분할 대역폭에 불규칙하게 분포되기 때문이다.

표2에는 스케줄링 기법을 적용한 경우와 적용하지 않은 경우에 데이터 지연시간의 최소값, 최대값 및 평균값이 주어져 있다. 주기적 데이터와 비주기적 데이터의 모든 경우에 대하여 스케줄링 알고리듬을 적용한 경우 데이터 지연시간의 성능이 우수함을 알 수 있다.

VI. 결론

본 연구에서는 FF의 대역폭 할당 기법 가운데 주기적 데이터의 전송을 위한 스케줄링 기법의 타당성을 실제 필드에서 사용되고 있는 장비를 가지고 검증하였다. 또한 FF의 사용자 환경에서 제공하는 스케줄 리스트에 스케줄링 기법을 통하여 도출한 스케줄링 파라미터의 적절한 값만을 설정함으로써 스케줄링 기법이 기존의 FF시스템에 바로 적용됨을 실제 장비를 통하여 확인 할 수 있었다. 실험 결과에 의하면 스케줄링 기법은 주기적 데이터의 성능 요구사항을 충분히 만족하는 것으로 나타났다. 본 연구에서 사용된 FF 장비의 Starter Kit는 아쉽게도 주기적 데이터를 전송하는 제한된 기능만을 가지고 있으며, 토큰-패싱에 의한 비주기적 데이터의 생성 및 전송을 사용자가 임의로 설정하는 기능은 지원하지 않고 있다. 따라서 FF의 대역폭 할당 기법[8] 가운데 비주기적 데이터의 성능 요구사항을 만족하도록 하는 방법은 본 실험 모델을 통하여 검증할 수 없었다. 추후에 토큰-

패싱 방식에 의한 비주기적 데이터의 전송을 포함하여 FF 규격서에서 제시하는 모든 기능이 구현된 FF 장비가 확보되면 이를 통하여 비주기적 실시간 데이터의 성능 요구사항을 만족하는 설계기법을 실험 모델을 통하여 검증할 계획이다. 본 연구에 대한 후속 연구로 FF에 적용한 대역폭 할당 기법의 성능을 다른 종류의 필드버스에서 제공하는 우선순위 기법과 비교 분석하는 연구를 수행할 예정이며, 또한 FF의 지연시간 성능을 분석하는 수학적 모델을 개발하고, 본 연구의 실험 결과를 수학적 모델에서 도출한 결과와 비교 분석할 계획이다.

참고문헌

[1] *Foundation™ Specification: System Architecture, Fieldbus Foundation,* 1996.

[2] *Foundation™ Specification: System Management, Fieldbus Foundation,* 1996.

[3] *Foundation™ Specification: Network Management, Fieldbus Foundation,* 1996.

[4] *Foundation™ Specification: Fieldbus Message Specification, Fieldbus Foundation,* 1996.

[5] *Foundation™ Specification: Fieldbus Access Sublayer, Fieldbus Foundation,* 1996.

[6] *Foundation™ Specification: Data Link Protocol Specification, Fieldbus Foundation,* 1996.

[7] *Foundation™ Specification: 31.25 kbit/s Physical Layer Profile, Fieldbus Foundation,* 1996.

[8] S. H. Hong, I-H. Choi, "Experimentaql evaluation of a bandwidth allocation scheme for foundation fieldbus", *IEEE Trans. on Instrument and Measurement.* vol. 52, no 6, pp. 1787-1791, 2003. 12.

[9] *IEC 61158-2: Fieldbus Standard for Use in Industrial Control Systems: Physical layer specification and service definition,* IEC, 2003.

[10] *IEC 61158-4: Fieldbus Standard for Use in Industrial Control Systems: Data Link Protocol Specification,* IEC, 2003.

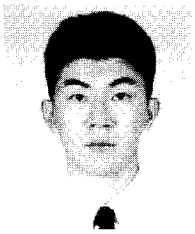
[11] *NI-FBUS™ Configurator User Manual (Part Number 321423B-01),* 1998.

[12] *MC68331-Based Fieldbus Round Card User Manual (Part Number 321 866A-01),* 1998.

[13] *NI-FBUS™ Communications Manager User Manual for windows 95 and Windows NT (Part Number 321287B-01),* 1997.

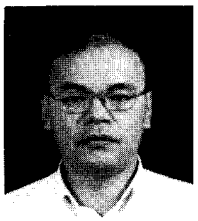
[14] *NI-FBUS™ Monitor User Manual (Part Number 321018 B-01),* 1997.

[15] *NI- FBUS™ Function Block Shell Reference Manual (Part Number 321016C-01),* 1998.



송 승 민

1974년 6월 12일생. 2001년 한양대학교 전자컴퓨터공학부(공학사). 2003년 한양대학교 전자전기제어계측공학과(공학석사). 2003년~현재 LG전자 UMTS 시스템 연구소 비동기 시스템 RAN 소프트웨어 개발. 관심분야는 필드버스, 빌딩자동화통신망, 홈네트워크.



홍 승 호

1956년 5월 31일생. 1982년 연세대학교 기계공학과(공학사). 1985년 Texa Tech University 기계공학과(공학석사). 1989년 Pennsylvania State University 기계공학과(공학박사). 1989~1992년 한국전자통신연구소 선임연구원. 1992년~현재 한양대학교 전자컴퓨터공학부 교수. 관심분야는 필드버스, 빌딩자동화통신망, 홈네트워크.