

Foundation Fieldbus에서 토큰-패싱 전송 방식의 실험적 성능 평가

Experimental Performance Evaluation of Token-Passing Mechanism in Foundation Fieldbus

배진운*, 홍승호**
(Jin Woon Bae, Seung Ho Hong)

Abstract - FOUNDATION fieldbus provides scheduling and token-passing services for cyclic and sporadic data respectively. In this paper, we evaluate the delay performance of token-passing mechanism in FOUNDATION fieldbus network system using an experimental model. This paper introduces a method of developing an experimental model which consists of 10 nodes of FOUNDATION fieldbus communication device. Using the experimental model, we evaluate the delay performance of time-critical and time-available data with respect to change of TTRT parameter.

Key Words : FOUNDATION fieldbus, token-passing, delay, experimental model

1. 서론

필드버스는 자동화 및 공정제어 시스템의 필드디바이스와 이 디바이스를 제어 및 관리하는 컴퓨터 기반의 자동화 기기에서 생성되는 모든 데이터를 실시간으로 전송하는 통신망이다. Foundation Fieldbus(FF)는 사용자계층에서 요구하는 주기적 및 비주기적 데이터의 전송 서비스를 모두 지원하도록 고안된 프로토콜이며, FF의 데이터링크 계층은 이러한 사용자계층에서 요구하는 통신서비스를 수행하기 위하여 스케줄링과 토큰-패싱에 의한 데이터 전송 방식을 지원한다. 이러한 FF 시스템은 FF 네트워크 성능에 따라 짐속되는 장치들의 동작에 직접적이 영향을 주므로, 네트워크의 성능 해석은 필드버스의 효율적인 엔지니어링에 필수적이라 하겠다. 그럼에도 불구하고 현재 FF의 네트워크 성능해석을 위한 구체적 방법과, 성능을 해석한 결과가 제시된 예가 없다. 본 논문에서는 FF의 데이터 전송 성능을 분석을 위한 실험모델 개발에 대하여 소개하고, 토큰-패싱 방식의 데이터 전송에 있어서 time-critical 데이터와 time-available 데이터의 전송지연시간을 실험모델을 통해 측정하고 분석하였다.

2. FF 데이터 링크 계층의 토큰-패싱 전송 방식

FF의 네트워크 계층 구조는 물리 계층, 데이터 링크 계층, FAS 계층, FMS 계층과 같이 4개의 계층 구조에 어플리케이션 계층을 그 상위 계층에 둔 형태를 가진다[1]. 이

중 데이터 링크 계층은 LAS(Link Active Scheduler)라는 버스 스케줄러를 통해 필드버스로의 메시지 접근을 관리하며, 기본적으로 중앙 제어 방식의 프로토콜을 사용하고 있으나 토큰-패싱과 스케줄링에 의한 데이터 전송을 모두 지원한다.

토큰-패싱 방식의 데이터 전송[3][4]은 LAS가 토큰을 정해진 순서에 따라서 일반 노드에 전송하고 토큰을 수신한 노드는 전송할 데이터가 있으면 데이터를 전송한 후 토큰을 LAS에 반납하는 방식이다. 이 경우 노드가 토큰을 수신하면 토큰의 우선순위보다 같거나 높은 우선순위를 갖는 데이터가 있는지 확인하고, 데이터가 존재하면 우선순위가 높은 데이터부터 전송한다. 토큰이 모든 노드를 방문하고 LAS 노드로 다시 돌아오면 미리 설정된 TTRT(Target Token Rotation Time)와 실제 토큰이 모든 노드에 전송되는데 걸린 시간인 ATRT(Actual Token Rotation Time)를 비교해서 TTRT보다 ATRT가 크면 토큰의 우선순위를 하나 증가시키고 작거나 같으면 토큰의 우선순위를 감소시킨다. 토큰의 우선순위가 결정되면 정해진 순서에 따라 다시 토큰을 전송한다.

이와 같이 네트워크 트래픽 부하나 TTRT 등과 같은 파라미터의 설정 값에 따라 그 데이터의 전송 성능이 변하게 되며, 그 결과 또한 예측하기가 어렵다. 정해진 스케줄에 의해 주기적 메시지를 전달하는 스케줄링 방식의 데이터 전송에 있어서는 그 전송 성능을 실험적 요소 없이 예측 가능한 반면, 이러한 토큰-패싱 방식의 데이터 전송은 실제 FF 시스템으로 구성된 실험 모델을 통해 데이터의 전송 딜레이를 측정하여 네트워크의 성능을 평가하고 분석할 필요가 있다.

지자 소개

* 배진운 : 한양大學 전자전기제어계측공학科 碩士課程

** 홍승호 : 한양大學 전자컴퓨터 工學部 教授 · 工博

3. 실험 모델 개발

FF를 통한 데이터 전송 성능을 측정하여 네트워크 성능을 해석하기 위해서 FF의 물리계층부터 상위 어플리케이션 계층까지의 기능을 하는 필드 디바이스 갖춘 실험모델을 구축하여야 한다. 이를 위해 FF 통신 디바이스의 하드웨어와 펌웨어를 개발하였고, 이 디바이스들을 이용하여 FF 시스템의 데이터 전송 성능을 평가 할 수 있는 실험 모델을 구축하였다.

3.1 하드웨어 개발

FF 디바이스 하드웨어는 그림 1.과 같이 CPU 모듈, MAU 모듈, Base Board로 구성되며, Base Board에 CPU 모듈 및 MAU 모듈을 탈착될 수 있는 형태로 제작하여 각 기능별로 기능 시험 및 추가확장이 가능하도록 개발하였다. 각 기능별로 보조 전원 공급을 위한 자체 전원회로와, 필드 디바이스의 응용 기능을 하는 하드웨어 연계 시험을 위한 I/O Extension 부와 시리얼 통신으로 하드웨어의 동작 시험을 하기 위한 Communication 부로 나눌 수 있다.

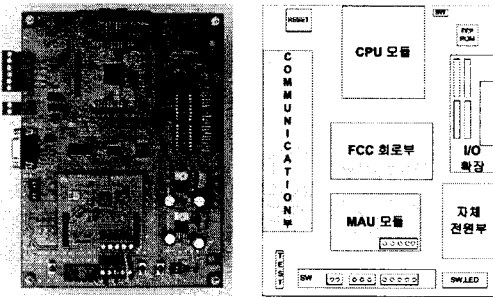


그림 1. 하드웨어 모습과 그 구성도

FF의 물리 계층을 구현한 MAU(Medium Attachment Unit) 모듈은 진송매체를 통해 들어오는 전원과 데이터 신호를 분리하여 레귤레이팅 된 +3V와 +5V의 전원을 공급하고 31.25k/bit의 voltage-mode의 데이터율로 그 신호를 FCC(Fieldbus Communication Controller)로 전달하는 기능을 한다[2]. FF 디바이스 내의 데이터 처리를 구현한 FCC 회로부는 필드버스 표준 규격[8]을 그대로 구현한 YAMAHA 사의 YTZ-420(FIND-1+) 칩을 사용하여 설계하였고, 이는 물리계층의 MAU와 완벽 하게 호환되도록 설계하였다. 또한 저전력 사양의 FF 규격을 따르는 RAM(TC55V100) 과 CPU(M16C)를 채택하여 FF 필드 디바이스 규격에 일치하는 하드웨어를 개발 하였다.

3.2 펌웨어 개발

본 연구를 통해 개발된 FF 디바이스의 펌웨어의 기본적인 개발 환경은 그 과정에서 표준 ANSI-C와 FF 규격에 제시된 Device Description(DD) 등을 통해 그 기능과 동작을 수정, 변경 및 확장할 수 있도록 하였다. 다중 작업환경이 가능한 실시간 운영체제(RTOS)로는 Segger사의 M16C

CPU 전용 RTOS인 embOS를 채택하였고, 프로토콜 스택은 Softing 사의 Foundation Fieldbus Link Master Stack Library를 채택하였으며, 인터페이스 소프트웨어 및 응용 프로그램은 embOS의 다중 작업 환경과 Foundation Fieldbus Stack Library 상에서 표준 ANSI-C 언어와 DD를 통해 구현되었다. DD는 Fieldbus 표준 협회에서 제공하는 Standard DD에 추가 장치 정보를 FF 기반[1][10]의 DDL(Device Description Language) 형태로 추가 작성하였다.

FF 디바이스의 펌웨어는 기본적으로 통신기능을 수행하면서 동시에 여러 개의 상위 어플리케이션 기능을 수행하여야 하므로 다중 작업이 가능한 실시간 운영체제기반에서 프로토콜 스택 및 응용 프로그램이 개발되었다. 또한 프로토콜 스택은 향후 응용기능의 추가 및 확장에 따라서 그 동작이 변경되지 않도록 상위 어플리케이션 프로그램에 상호 독립적으로 적용되었다. 이를 위해 그림 2.의 소프트웨어 개발프로세스에 보이는 바와 같이 각각의 단계별로 개발될 수 있도록 환경을 구축하여 최종 펌웨어를 개발 하였다.

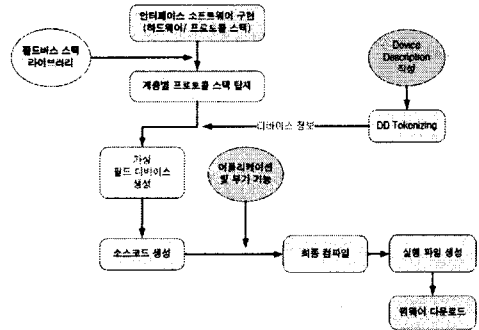


그림 2. 소프트웨어 개발 프로세스

프로토콜 스택은 그림 3.과 같이 각 계층별로 Task를 할당하여 운영체제의 실시간 멀티 테스킹 환경에서 동작되도록 구현하였다. 인터페이스 소프트웨어는 FF의 규격[5][6][7]에 따라 평선 블록(Function Block)을 통해 모든 기능이 동작되도록 다음 그림 4.과 같이 프로토콜 스택에 상위에 사용자 어플리케이션 기능이 탑재되도록 설계되었다. 이는 평선 블록의 동작을 GPIO(General Purpose I/O)를 통해 실제 FF를 이용한 하드웨어 장치에 연결하는 기능을 한다.

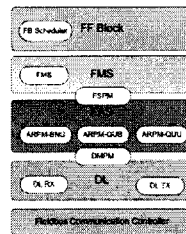


그림 3. 프로토콜 스택의 다중 작업 Task.

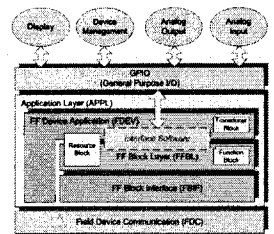


그림 4 인터페이스 소프트웨어 구조

3.3 실험 모델

본 연구에서 개발된 하드웨어에 펌웨어를 탑재한 총 10대의 FF 디바이스를 이용하여 10개의 노드로 이루어진 FF 실험 모델을 구축하였다. 이 실험 모델은 필드 디바이스들을 configuration 하고 monitoring 하기 위한 컴퓨터가 PC interface board를 통하여 필드버스에 연결되며, 이를 위한 소프트웨어와 PC interface board는 National Instrument사가 개발하고 FF 표준 협회에서 보급하는 Foundation Fieldbus Starter Kit을 이용하였다[10].

다음 그림 5.과 같이 FF 디바이스들은 버스 토폴로지로 구성된 필드버스 H1 통신 버스 시스템을 구축하여, 실제 FF 시스템과 그 시스템을 설정하고 모니터링 할 수 있는 가상의 시스템 구축으로 FF 실험 모델을 완성한 것이다.

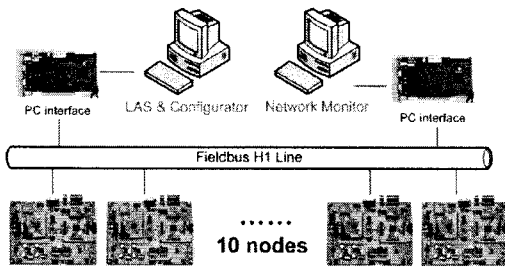


그림 5. 실험모델 구성도

이러한 실험 모델을 통하여 FF 디바이스의 각종 파라미터들의 값을 설정할 수 있으며, 또한 그 값의 설정에 따라 변화하는 데이터의 전송 성능을 모니터링 할 수 있다. 이러한 실험적 측정을 통하여 예측하기 힘든 토큰-패싱 방식의 데이터 전송 성능을 예측 가능하게 하고, 이를 통해 토큰-패싱 방식에서의 데이터 전송 성능 향상에 대한 연구를 가능하게 한다. 그림 6.은 FF 디바이스로 구성된 실험 모델의 실제 실험 환경이다.

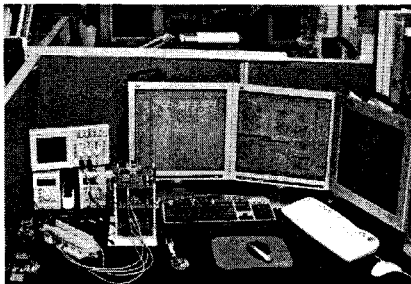


그림 6.. 실험 모델

4장. 전송 지연시간 측정

본 장에서는 3장에서 제시한 실험 모델을 이용하여 FF상에서 토큰-패싱 방식의 데이터 전송시에 발생할 수 있는 전송 지연 시간을 측정하고, 예측하기 어려운 네트워크의 성능이 TTRT값의 변화에 따라 어떻게 달라지는지 실험 결과를

바탕으로 FF 토큰-패싱 방식에서의 데이터 전송 성능을 평가 한다.

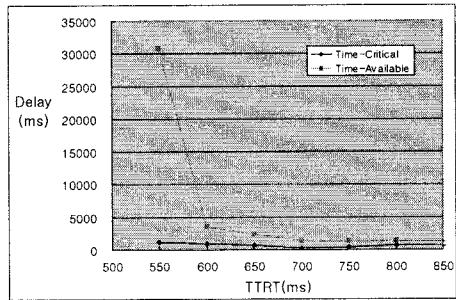


그림 7. TTRT 변화에 따른 데이터 전송 지연 시간의 변화

그림 7.에서 보는 바와 같이 토큰-패싱 전송 방식에서 Time-Critical 데이터와 Time-Available 데이터가 TTRT에 따라 그 전송 지연시간이 변화함을 알 수 있다.

5.장 결론

FF는 토큰-패싱 방식의 데이터 전송에서 그 네트워크의 성능을 예측하기 어렵다. 따라서 본 논문을 통하여 FF의 실험모델을 제시하고 토큰-패싱 방식의 데이터 전송에서의 네트워크 성능을 실험적으로 평가하였다. 그 결과 TTRT의 변화에 따라 네트워크 성능이 변화함을 알 수 있으며, 네트워크 성능의 효율성 극대화 할 수 있는 TTRT의 값을 예측 가능하게 한다. 본 연구에서는 네트워크 트래픽 부하에 따른 네트워크 성능은 고려하지 않았다. 후속 연구에서는 이 실험 모델을 이용하여 실제 FF 디바이스의 모든 네트워크 파라미터에 따른 네트워크 성능을 측정하고, 그 결과를 분석하여 파라미터 설정 방식을 제시할 수 있다.

참 고 문 헌

- [1] FOUNDATIONTM Specification : System Architecture, Fieldbus Foundation 2003.
- [2] FOUNDATIONTM Specification : 31.25kbit/s Physical Layer., Fieldbus Foundation 2003.
- [3] FOUNDATIONTM Specification : Data Link Protocol, Fieldbus Foundation 2003.
- [4] FOUNDATIONTM Specification : Data Link Service, Fieldbus Foundation 2003.
- [5] FOUNDATIONTM Specification : Communication Profile, Fieldbus Foundation 2003.
- [6] FOUNDATIONTM Specification : Function Block Application Process Part 1, Fieldbus Foundation 2003.
- [7] FOUNDATIONTM Specification : Function Block Application Process Part 2, Fieldbus Foundation 2003.
- [8] IEC 61158-4 : Fieldbus for use in industrial control systems - Part 4 Data link protocol specification, IEC, 2000.
- [10] Jonas Berge, Fieldbus for Process Control. ISA Publication, 2004.