2002년도 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집(2002.11.30)

Profibus Master FDL 보드 구현과 파라미터 변경에 따른 성능평가

박 정 민 , 홍 승 호
한양대학교 전자 전기 제어계측공학과
전화 : 031-400-4084 / 핸드폰 : 017-508-7792

Experimental Performance Evaluation of Profibus-FDL and Implementation of Profibus Master Board

Jong Min Park, Seung Ho Hong
Dept. of Electronic, Electrical, Control and Instrumentation Engineering, Hanyang University
E-mail : susu69@empal.com

Abstract

Profibus is a open industrial communication system for a wide range of applications in manufacturing and process automation. In this study presented in this article, implementation of Profibus master board is implemented. Using an experimental model, this study investigates the relationship between network parameter in the Profibus data link layer and the network performance. Base on the results, this study suggests some factors that should be considered when a Profibus-based automation system is designed and implemented.

I. 서론

생산 자동화 기술의 발달에 따라 자동화 현장에 많은 컴퓨터가 사용되고 있고 그에 따라 각각의 자동화 요소들을 네트워크로 연결하여 모든 공정의 통합화를 추구하는 기술이 필요하게 되었다. 필드버스는 이러한 각종 필드장비를 네트워크로 연동 하여 데이터를 전송하기 위해 사용되는데, 이런 데이터들은 네트워크의 특성으로 인해 하나의 미디엄을 공유한다. 이로 인해 필드버스는 여러 가지 네트워크 파라미터를 정의하고 사용하게되고, 이러한 파라미터의 적절한 선정은 네트워크 시스템 전체의 성능에 커다란 영향을 미치게 된다.

본 논문은 여러 가지 필드버스 중 공정제어 시스템 등에서 널리 사용되는 Profibus[1]의 마스터 보드를 구현하고 이를 이용하여 네트워크 파라미터 변경에 따른 성능을 분석한다. 이를 위해 Profibus의 계층 중 FDL, PHY, FMA 1/2 계층을 PC 인터페이스 보드로 구현하고 FDL 계층에 정의되어 있는 파라미터와 데이터 전송 속도를 변경하여, 그에 따른 전송지연시간을 실험을 통해 측정하여 네트워크 성능을 평가한다. 측정된 실험 결과를 토대로 다양한 환경에 따라 가장 적절한 네트워크 파라미터 선정 기준을 제시한다.

본 논문은 총 5장으로 구성되어있다. 2장에서는 구현된 Profibus 마스터 보드의 H/W와 S/W의 구조에 대해 설명한다. 3장에서는 실험모델 구성 및 개발 환경에 대해 설명하고, 4장에서는 실험 방법과 실험결과를 설명한다. 마지막으로 5장에서 결론과 향후 연구방향을 제시한다.

II. Profibus 마스터 보드 구현

본 논문에서는 상용화된 Profibus 마스터용 ASIC칩인 SINEMSE사의 ASPC2[2]를 이용하여 FDL계층의 일부를 구현하였다. ASPC2가 지원하지 않는 인터페이스 기능과 FDL계층의 관리기능은 Linux(kernel 2.4.5)가 포팅된 INTEL사의 SA1110[3]에 의해 운용된다. 구현된 Profibus 마스터 보드의 오퍼레이팅 시스템으로 Linux를 사용하였으므로 멀티태스킹을 통한 빠른 FDL 서비스가 가능하다. 보드에는 네트워크 관리기능을 담당하는 FMA1/2, ASPC2와 통신 메모리의 관리를 담당하는 FDL, 그리고 상위계층과의 인터페이스를 위한 LLI로 구성된 3개의 태스크가 각각 Firmware로 구현되어 탑재되어있으며 부팅과 동시에 운영된다. PC에 탑재되는 상위계층의 통신서비스들이 Profibus 마스터 보드를 통하여 Profibus 네트워크에 접속할 수 있도록 하기 위해 PC인터페이스는 PIC bus를 사용하였으며, PIC 컨트롤러는 PLX사의 PCI9050을 사용하였다. MAU부분은 최대 12Mbps까지 전송이 가능하도록 설계하였다. WIN98 기반으로 운용되는 상위 드라이버는 JUNGO사의 WinDriver를 이용하여 제작하였다.

2.1 Profibus 마스터 보드의 Hardware 구현

구현된 Profibus 마스터 보드는 크게 SA-1110, ASPC2, MAU, PCI-9050, DP-RAM, 통신-RAM의 6개의 부분으로 구분된다. 다음 그림1은 구현된 보드의 블록도이다.

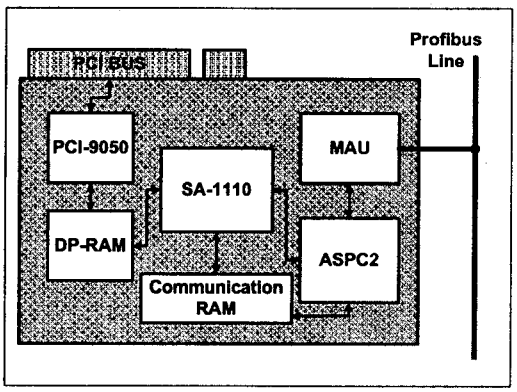

그림 1. Profibus 마스터 보드 블록도

(1) SA-1110

INTEL사의 범용 CPU SA-1110A를 사용하였으며 블록도에는 표시하지 않았지만 기본적인 메모리와 리셋회로 등을 포함하여 나타낸 블록이다. Profibus 마스터 보드의 전체 동작 및 파라미터 초기화 파일을 구동하는 기능을 한다. 리눅스 커널이 롬에 저장되어 있으며 전원투입 시 커널 로딩 및 FDL보드 세팅을 한다. ASPC2에서 지원하지 않는 기능인 상위계층과의 인터페이스 및, FDL 관리기능, 그리고 FMA1/2 프로토콜이 Firmware로 구현되어 저장되어 있다. 구현된 Firmware는 다음절에서 설명한다.

(2) ASPC2

SINEMSE사의 상용화된 Profibus 마스터용 ASIC칩으로 PROFIBUS-DP, PROFIBUS-FMS 및 PROFIBUS-PA에서 모두 사용될 수 있다. ASPC2는 다음과 같은 서비스를 제공한다.

* Request-FDL Status
* Send Data with no Acknowledge, SDN
* Send Data with Acknowledge, SDA
* Send and Request Data with Reply, SRD

(3) MAU

Profibus는 통신 방식으로 RS-485를 사용한다. 전송 데이터의 신뢰성을 위해 MAU부분은 전원을 분리하여 구현하였으며, 이를 위해 사용한 포토커플러는 ASPC2에서 지원하는 최대 속도인 12Mbps의 속도까지 전송이 가능하도록 25Mb/s의 Data Rate를 가지는 포토커플러를 사용하였다. MAU의 전원은 통신 허브에서 공통전원을 공급받도록 설계하였다.

(4) PCI-9050

PC에서 동작하는 사용자계층 프로토콜을 위해서 구현된 Profibus 마스터 보드와 PC의 인터페이스 버스로 PCI-버스를 사용하였다. PCI-버스 컨트롤을 위해 PLX사의 PCI9050을 이용하였으며 PCI9050은 PCI Local Bus Spec. Revision 2.1을 따르는 PCI adapter이고 16-bit Data Path로 동작하도록 설계하였다.

(5) DP-RAM

Profibus 마스터 보드와 상위계층과의 인터페이스를 위하여 4K BYTE의 공간을 가진 16 bit DP-RAM을 이용하였다. 사용된 DP-RAM은 busy, semaphore, interrupt 기능이 지원된다. 보드는 이러한 기능을 이용하여 고속의 데이터 입출력이 가능하도록 설계되었으며, 환형 QUEUE 구조로 사용하였다.

(6) Communication-RAM

Communication-RAM은 SA1110과 ASPC2사이에 존재하는데 Hold-in과 Hold-out의 신호를 이용하여 버

스 억세스 권한을 공유하면서 하나의 RAM을 Communication-RAM으로 사용한다. 본 논문에서는 1Mbit SRAM 2개를 이용하여 16bit의 데이터 폭을 가지는 공유 메모리형태로 구현하였다.

2.2 Profibus 마스터 보드의 Firmware 구현

Profibus 마스터 보드는 Linux(kernel 2.4.5)을 오퍼레이팅 시스템으로 사용하였다. 커널과 응용 프로그램의 구동은 SA-1110과 메모리에서 운용되며 커널 이외에 독립적인 3개의 태스크가 구현되어서 탑재되어있다. 다음 그림 2는 Profibus 마스터 보드의 내부 태스크와 외부 인터페이스의 관계를 보이는 블록도이다. 상위계층과의 인터페이스를 담당하는 LLI()와 ASPC2와 FDL계층의 관리를 담당하는 FDL()이 주 태스크이다. 위의 태스크와 별도로 FMA 1/2 프로토콜을 담당하는 FMA()도 다른 2개의 태스크와 동시에 운전된다. 태스크간의 데이터 교환은 GNU 표준 라이브러리인 GLIBC에서 제공하는 message queue서비스를 이용하여 이루어진다.

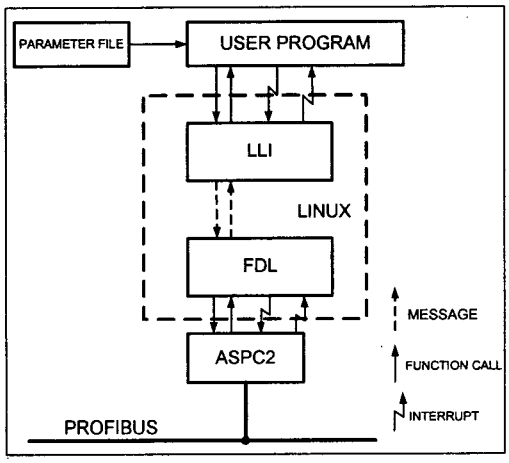

그림 2. Profibus 마스터 보드의 Firmware 블럭도

(1) LLI()

LLI()태스크는 상위계층과의 인터페이스를 담당하며 User 함수, 인터럽트 처리 함수, 데이터 처리 함수, 상태 함수 등 4부분으로 나누어진다. 상위계층에서 송신하고자 하는 데이터가 발생하였는지, 혹은 FDL계층에서 상위계층으로 보낼 데이터가 있는지를 확인하며 환형 큐로 구성된 DP-RAM을 억세스한다.

(2) FDL()

FDL()태스크는 하드웨어로 구현된 FDL 계층의 각종 동작을 설정하는 기능과 ASPC2에서 하지 않는 FDL계층의 역할을 담당하는 기능을 수행한다. FDL User 함수, 인터럽트 처리 함수, 프레임 처리 함수, 상태 천이 함수 등 4부분으로 나누어진다. ASPC2의 관리와 FDL계층의 전체 상태를 전환하는 기능을 하면서 Communication-RAM을 억세스한다.

(3) FMA()

FMA()태스크는 Profibus 1,2 계층의 관리기능을 하는 FMA 1/2 프로토콜이 구현되어 동작한다. FDL 파라미터 변경기능과 상위계층에서 FDL계층의 상태를 변경하거나 확인을 하고자 할 때 사용된다.

III. 실험모델 구성과 실험 환경

실험모델은 크게 모니터링과 파라미터 설정을 위한 모니터 노드와 주기적으로 메시지를 발생시키는 두 개의 노드로 구성되어있다. 모니터링 노드는 PC의 PIC 버스를 통해 인터페이스 되며 PC쪽 모니터 프로그램은 WIN98 시스템에서 운영되는 S/W로써 전송된 메시지의 모니터링 기능 이외에 PCI 인터페이스를 위한 디바이스 드라이버와 네트워크 파라미터 설정기능을 포함하고 있다. 주기적 메시지를 발생하는 노드는 실험시나리오에 따른 트래픽 G에 근거하여 우선 순위별로 구분되어 계산된 발생주기와 정해진 길이의 데이터를 우선 순위 구분된 전송 큐에 저장한다. 토큰을 받아 전송권한을 가지게 되면 전송 큐에 있는 데이터는 모니터링 노드로 전송된다. 다음 그림 3은 구성된 실험 모델을 나타낸다

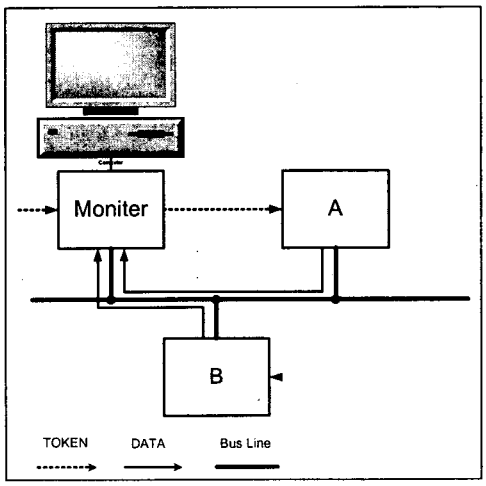

그림 3. 실험모델

IV. 실험 및 결과

4.1 실험방법과 조건

실험은 네트워크 트래픽 G에 따른 두 가지의 환경하에서 다양한 TTRT의 설정을 통한 데이터 평균 전송 지연시간의 변화에 관해 수행하였다. 데이터의 전송 속도는 최대속도인 12Mbps로 고정하여 실험하였다. 다음 식1을 통해 G가 계산된다.

$$G=\frac{LN}{BT} \quad \text{----------} \quad (\text{식 1})$$

L : 발생된 데이터의 길이

N : 데이터를 발생하는 총 노드 수

B : 전송속도

T : 메시지 발생 주기

Profibus에서는 두 가지의 우선 순위를 지원하므로 각 순위별로 발생되는 데이터의 길이를 다르게 하여 전체 G를 계산하였다. 다음 식2를 통해 구해진다.

$$G_T = G_H + G_L = \frac{(L_H + L_L)N}{BT} \quad \text{--(식 2)}$$

생성된 데이터의 길이는 High의 경우 90Byte이고 Low의 경우 180Byte이다. 전송률과 생성된 데이터는 위에서 설정한 길이로 고정하고, 데이터의 발생주기를 변화시키면서 G를 조정하였다. 실험시나리오에 사용된 G는 0.3과 0.6이다. 각각의 G를 만족하는 메시지 발생 주기는 각각 150 μsec와 75 μsec 이다.

여기에서 전송지연시간은 메시지가 발생한 후부터 모니터링 노드가 메시지를 수신한 시간까지의 지연시간으로 정의하였다. 또한 네트워크 상태가 안정상태로 들어가기 이전의 초기 상태에 수신된 데이터는 전송지연시간에 포함시키지 않았다.

4.2 실험 결과

그림 4는 G=0.3으로 네트워크 트래픽을 생성하여 TTRT(Target Token Rotation Time)를 20 μsec 에서 1500 μsec 까지 변화시켰을 때의 각 우선 순위별 평균 전송지연시간을 나타낸 것이다. 그림5는 G=0.6으로 생성된 트래픽에 대한 평균 전송지연시간에 대한 것이다. 두 가지 실험에서 High와 Low 메시지에 대한 트래픽 비율은 1:2 로 생성하였다.

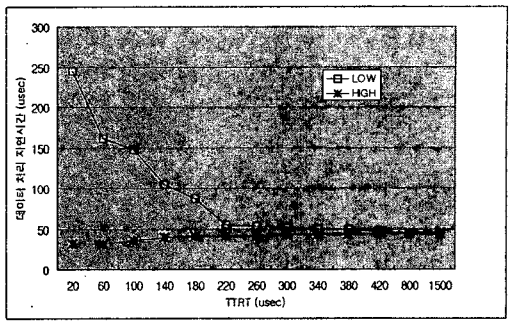

그림 4. TTRT변화에 따른 전송지연시간(G=0.3)

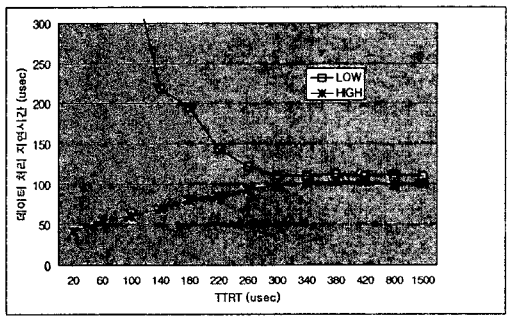

그림 5. TTRT변화에 따른 전송지연시간(G=0.6)

실험 결과를 통해 다음의 사실을 확인할 수 있다.

(1) TTRT가 ATRT(Actual Token Rotation Time)이상 설정된 경우 전송지연시간에 대해 큰 영향을 미치지 않는다.

(2) 그림 4에서 볼 수 있는 것처럼 트래픽이 상대적으로 작은 경우 TTRT는 높은 우선 순위를 가지는 데이터의 전송지연시간에는 큰 영향을 미치지 않고, 오히려 낮은 우선 순위를 가지는 데이터의 전송지연시간을 길게 하는 효과를 가져온다.

(3) 그림 5에서 볼 수 있는 것처럼 높은 트래픽 부하를 가지는 경우 전체적인 전송지연시간은 늘어나지만, 우선 순위별로 구분된 데이터의 평균 전송지연시간은 두 가지 순위를 가지는 전송 데이터에 대해 모두 TTRT의 설정에 많은 영향을 받는 것을 알 수 있다.

(4) 트래픽 부하가 높은 경우 TTRT의 적절한 조정은 높은 우선 순위를 가지는 전송데이터의 전송지연시간을 줄이는데 큰 영향을 미친다.

VI. 결론

본 논문은 Profibus 마스터 보드를 FDL계층까지 포함하는 PC인터페이스 카드형태로 제작하고, 구현된 보

드를 이용하여 다양한 트래픽환경에서 Profibus의 중요한 네트워크 파라미터인 TTRT의 변화에 따른 평균 전송지연시간에 대해 실험하였다.

본 논문의 후속 연구로는 TTRT이외에 Profibus FDL계층의 다른 파라미터들이 시스템의 성능에 미치는 영향을 실험을 통하여 평가해야 할 필요가 있다. 또한 비주기적 메시지가 생성되는 경우와 주기적, 비주기적 메시지가 동시에 생성되는 경우에 대해서 위와 같은 실험을 통해 성능을 평가 할 필요가 있다.

참고문헌

[1] DIN 19 245 Profibus Standard Part 1 and Part2: 1991
[2] SIMETIC NET, ASPC2 User Description, July 1999
[3] Intel StrongARM SA-1110, Advanced Developer's Manual, December 1999