LonWorks 네트워크 상의 ANSI/EIA709.1 패킷해석을 위한 프로토콜 분석기의 설계 및 구현

Implementation of Protocol Analyzer for ANSI/EIA709.1 Packet on LonWorks Network

임 일 영*, 최 기 상**, 최 기 홍***
Il Young Im, Gi Sang Choi, Gi Heung Choi

Abstract - Use of intelligent devices that work on the ANSI/EIA 709.1 protocol is increasing. In this study an ANSI/EIA 709.1 protocol analyzer that can monitor and analyze the packets on LonWorks network is designed and developed. The device is based on TMS320LF2407A processor for decoding data packets, and uses XScale processor for sending data to the application program on PC. The application program has various analysis features as well as basic monitoring function. The developed device can be used for debugging purposes in development of any kind of LonWorks devices, and also it is useful in maintenance of LonWorks network or LonWorks devices.

Key words : LonWorks, 프로토콜 분석기

1. 서 론

LonWorks 디바이스들은 빌딩 자동화 분야 및 공장/시설 관리 분야 그리고 철도 차량 분야와 유틸리티 및 홈네트워크 분야에 설치, 운영되고 있다. 일반적으로 네트워크에서 디바이스간 전송되는 패킷에 대하여 실시간으로 분석하여 네트워크상의 여러 상황을 확인할 필요성이 있으며, 네트워크 관리자는 네트워크상에서 전송되는 패킷을 모니터링 하여 보다 쉽게 관리할 수 있도록 하여야 한다. 더 나아가 IP 네트워크와의 연결을 위한 기반 조성을 위하여 임베디드 시스템을 이용한 ANSI/EIA 709.1 프로토콜 분석기의 설계 및 구현이 중요하다.

본 연구에서는 ANSI/EIA709.1 프로토콜을 분석하기 위한 하드웨어 및 응용 프로그램 구현하였다. 구체적으로 네트워크 신호를 디코팅하기 위하여 TMS320LF2407A 프로세서를 사용하였고, PC 응용 프로그램으로 데이터를 전송하기 위해 PXA250 프로세서에 기반한 XScale을 이용하였다. PC 응용 프로그램에서는 기본적인 모니터링 기능 뿐 아니라 사용자의 각종 편의기능을 구현하였다.

2. Protocol Analyzer 하드웨어의 설계 및 제작

ANSI/EIA709.1 프로토콜을 분석하기 위한 하드웨어는 TMS320LF2407A 프로세서를 사용하여 네트워크에서의 물리계층의 신호를 디지털 데이터 신호로 디코팅한다. 디코딩된 신호를 일반 PC상에서 볼 수 있도록 하기 위한 PC와의 데이터 통신은 PXA250 프로세서를 사용한 WINDOWS CE .NET 기반의 XScale을 이용한다. 이때 PXA250 프로세서는 마스터 프로세서가 되고, TMS320LF2407A는 슬레이브 프로세서가 되어, 네트워크에서 protocol이 감지되면 TMS320LF2407A에서 디코팅한 후 PXA250로 보내고 다시 PXA250에서 PC 응용 프로그램으로 데이터를 보내게 한다. PC 응용 프로그램에서는 사용자의 편의를 위하여, 프로토콜을 분석하여 화면에 나타낼 뿐만 아니라 분석된 패킷저장, 필터기능 및 통계치를 구할 수 있도록 프로그램 한다.

실제 보내진 데이터와 구현된 프로토콜 분석기를 통한 데이터 간을 비교 분석하여 실제 LonWorks 네트워크에서 프로토콜 분석기를 사용할 수 있도록 하여 디바이스 개발에 용이하게 하고 네트워크관리를 편리하게 수행할 수 있도록 한다.

2.1 물리 계층

본 연구에서는 TP/FT-10 채널 방식, twisted pair 채널을 사용하는 물리 계층을 고려한다. FT-10 Twisted pair 채널은 트랜시버를 거쳐서 뉴런 CPU로 데이터를 전달할 때 single ended differential Manchester code (Bi-phase space code, BPS)를 이용하여 신호를 전달한다. BPS 방식은 입력된 신호가 다음 클럭에서 전환이 발생하면 0, 전환이 발생하지 않으면 1로 인식하는 방식이다.

물리계층에서는 single ended bps모드를 주로 사용하지만 신호의 신뢰성을 높이기 위해서 differential code를 이용하기도 한다.

물리 계층의 신호는 bi-phase space code로서 전달되므로 FTT-10A의 경우 전송 속도는 최대 78.125 kbps이고, 실제 클럭은 최대 156.25 kHz로 동작하게 된다. Preamble은 데이터 전송 시 수진 측과 입력 측의 동기를 맞춰주기 위한 부분으로서 ANSI/EIA 709.1에서는 differential Manchester '1'이 연속적으로 나타나고 하나의 differential Manchester '0' 비트로 preamble의 종료를 알린다. Preamble의 길이는 최소 4비트 이상이고 데이터 전송이 시작됨을 알린다. 이 코드를 다시 NRZ 데이터로 변경해야 CPU를 통해서 읽어 들일 수 있다.

프로토콜 분석기는 마스터 프로세스, 슬레이브 프로세스 그리고 기타 하드웨어로 구성되며, 슬레이브 프로세서는 론웍스 네트워크 상에서 Manchester 인코딩 되어 돌아다니는 패킷을 디코딩하여 raw 패킷을 얻고 마스터 프로세스는 슬레이브 프로세스가 디코딩하여 얻은 raw 패킷을 받아서 프로토콜을 해석할 준비를 한다.

2.2 LonNetwork 디코딩

론네트워크 신호는 대략 2.5V를 중심으로 +-0.5V 스윙을 한다. 즉, LonWorks 네트워크 신호는 2-3V의 신호이다. 이 신호는 슬레이브 프로세서 TMS320LF2407A가 바로 읽을 수 있도록 신호 범위를 바꿔줘야 한다. 슬레이브 프로세서로는 TMS320LF2407A을 사용한다.

먼저 2-3 V의 신호를 비교기를 이용하여 0-5 V의 신호로

* 임일영 : 서울市立大學校 電子電氣컴퓨터工學部 碩士課程
** 최기상 : 서울市立大學校 電子電氣컴퓨터工學部 敎授
*** 최기홍 : 漢城大學校 機械시스템工學科 敎授

바꿔준다. LonWorks 네트워크 신호의 펄스 주기는 78.125 kHz로 하나의 펄스는 10 ns정도의 길이이다. 그러므로 비교기의 응답 시간도 빨라야 하고, rising time과 falling time이 10 ns보다 짧아야 한다.

비교기의 비교 전압은 2.2 V 정도를 입력하여 IDLE 상태(네트워크 신호가 없을 때)에서는 LOW를 유지하도록 한다. LOW를 유지하는 이유는 슬레이브 프로세서에서 rising edge 인터럽트를 쓰기 때문이다. 비교기와 인버터를 통과한 네트워크 신호는 TMS320LF2407A가 읽기에 좋은 신호가 되며, 비교기에 의해서 반전되었던 신호가 인버터를 통과하면서 원래 신호와 같은 상을 유지하게 된다. 비교기와 인버터의 전원은 3.3V를 공급하여 TMS320LF2407A의 전원과 맞춰 주었다.

2.3 PXA250의 설정

임베디드 시스템은 어떠한 장치가 다른 시스템에 의존하지 않고 독립적으로 기능을 수행하는 것으로써 각종 전자기기, 전자제품, 제어장치 등이 해당된다. 임베디드 시스템의 전형적인 모델은 마이크로프로세서가 내장되어 있고 [4], 특정한 기능을 수행하도록 프로그램이 내장되게 된다. 임베디드 시스템은 시간이 흐를수록 기능과 요구사항이 다양해지고 시스템의 크기는 날로 커져서 임베디드 시스템을 운용하기 위해서 운영체제가 필요한 경우가 많아지고 운영체제의 성능에 따라 시스템의 성능 및 확장성 등이 영향을 받고 있다.

임베디드 운영체제 중 에서 임베디드 리눅스와 Windows CE가 가장 널리 쓰이고 있는데, 특히 Windows CE는 폭 넓은 분야의 기기에 적용되도록 설계되었다. Windows CE는 시리얼 통신에서부터 TCP/IP에 이르기까지 다양한 통신 환경을 제공한다. 다양한 통신 환경은 모니터링 기능에 있어서 매우 중요하므로 프로토콜 분석기의 마스터 부분의 운영체제로 사용하게 되었다. XScale Board는 Intrinsyc사의 Cerf-cube 보드를 이용하였다.

2.4 SSP/SPI 통신

시리얼 방식의 통신 방법인 SPI는 TMS320LF2407A에서는 SPI(serial peripheral interface)라고 하고 PXA250에서는 SSP(synchronous serial port)라고 한다. 일반적으로는 SPI라고 부른다 SPI 통신에는 마스터와 슬레이브가 존재한다. 본 연구에서는 TMS320LF2407A의 SPI를 슬레이브로 설정하고 PXA250의 SSP를 마스터로 설정한다. SPI/SSP의 동작 클럭은 2 MHz로 설정하였다. 마스터와 슬레이브가 통신할 때 슬레이브가 데이터를 보내기 위해서는 마스터가 발생하는 클럭에 맞추어 데이터를 보내야 한다. 마스터가 rising edge에서 데이터를전송하고 슬레이브는 falling edge에서 데이터를 전송하는 것이다. 그러므로 슬레이브인 TMS320LF2407A에서 raw 패킷이 준비되면 PXA250에 외부 인터럽트를 걸어 통신 시작을 알리고 마스터에서 클럭이 발생되면 raw 패킷을 보내게 된다. 인터럽트와 빠른 SPI 통신을 위해 LonWorks 네트워크의 raw 패킷을 얻는 블록은 디바이스 드라이브로 구현하였다.

2.5 TCP/IP socket 구현

네트워크에서 읽어 들인 raw 패킷을 pc로 전달하기 위하여, PXA250에서 디바이스드라이브 응용 프로그램으로 TCP/IP를 사용하였다. Embedded Visual C++ 4.2를 이용하여 구현하였다.

2.6 Protocol Analyzer 흐름도

전체적인 raw 패킷 전달과정을 살펴보면, 먼저 TMS320LF2407A을 이용하여 LonWorks 네트워크로부터 raw 패킷을 디지털 신호로 디코딩한 후, SSP/SPI 통신을 사용하여 PXA250로 전달하고, PXA250에서는 TCP/IP를 이용하여 PC로 전송하게 된다. 그 흐름은 아래의 그림과 같다.

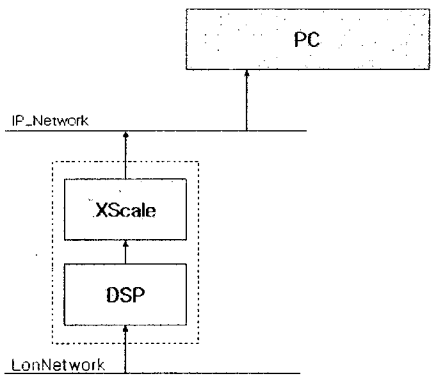

그림 1 protocol analyzer의 raw 패킷 전송 흐름도

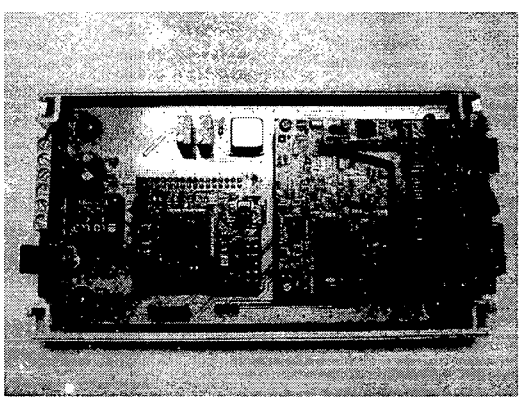

그림 2 구현된 Protocol Analyzer

그림 2는 구현된 프로토콜 분석기의 모습이다. LonWorks 네트워크 연결 포트와 전원부가 있고, 중간에는 TMS320LF2407A를 탑재한 DSP 모듈이 있다. 다른 반대편에 XScale이 있으며, IP 네트워크와 연결할 포트가 있다.

3. 프로토콜 분석기 소프트웨어

3.1 분석기 프로그램 인터페이스

프로토콜 분석기에서 전송 되어진 raw 패킷은 PC 상의 응용 프로그램으로 전달되어 분석된다. 프로그램의 기본 인터페이스는 그림 3과 같다.

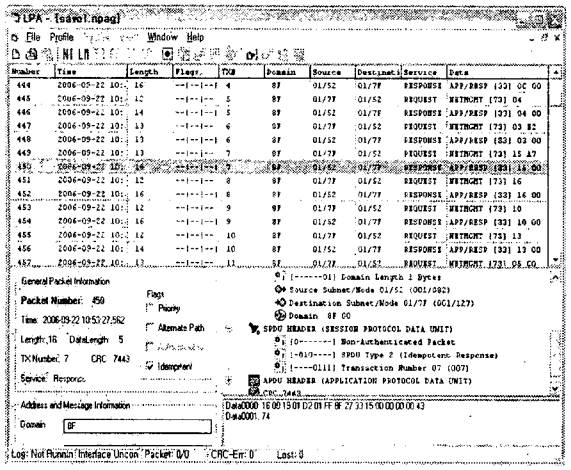

그림 3 PC에서의 응용 프로그램 인터페이스

시간대 별로 앞서 전송 되어진 패킷부터 메인 그리드 화면의 위에서 아래서 순차적으로 패킷을 업데이트를 한다. 메인 그리드 화면 아래 왼쪽은 일반적인 패킷 정보를 문서로 표시하고, 오른쪽에는 각 layer별 format에 맞는 트리 구조로 보다 쉽게 확인 할 수 있도록 비트 정도 및 자세한 분석 정보를 표시한다. 패킷 detail 정보나 protocol detail 정보를 원치 않을 경우 화면상에서 없애고, 메인 그리드 화면만 볼 수 있도록 하였다. 이 부분들은 언제든지 사용자가 원하는 만큼 화면에서 다시 보이도록 구현하였다.

메인 그리드에는 source 주소와 destination 주소 그리고 data 정보를 기본으로 갖고, 추가적으로 각 패킷의 번호, 전송된 시간, 패킷 길이, flag 정보, 전송 번호와 domain 주소 그리고 각 패킷의 service 종류가 나타난다.

3.2 프로그램 기능

응용 프로그램은 다음과 같은 다양한 기능을 수행 할 수 있도록 하였다.

1) 네트워크 연결 : 프로토콜 분석기와의 TCP/IP 통신을 위한 주소 입력과 port 번호 입력 창을 이용하여 프로토콜 분석기의 XScale과 통신을 할 수 있게 한다.

연결된 이 후에도 업데이트일시 정지 및 연결 종료 기능을 두어 언제라도 모니터링을 중지하거나 종료할 수 있도록 하였다. TCP/IP를 통해 프로토콜 분석기와 연결되기 때문에 네트워크 관리자가 원거리에서 모니터링이 가능하다.

2) 필터링 : 프로토콜 분석기로부터 전송된 패킷 중에서 사용자가 보고자 하는 내용의 패킷을 중점적으로 분석할 수 있도록 각 layer 단계별 필터 기능을 수행한다. 필터링 기능으로는 분석할 패킷을 저장할 capture 필터와 화면상으로 볼 수 있도록 할 display 필터가 있다.

Capture 필터와 display 필터는 기본적으로 비슷한 화면과 기능을 갖고 있다. 정보 저장을 위한 capture 필터와 화면상에서 보이도록 하기 위한 display 필터는 각각 다른 설정을 할 수 있지만, capture 필터기능이 먼저 실행되고, 그 이후에 display 필터 기능이 실행되기 때문에 capture 필터에 의해 차단된 패킷에 대해서는 더 이상 모니터링 할 수 없다. 하지만 capture 필터를 통과하여 저장 되었던 패킷은 display 필터에 의해 모니터링 할 수 없더라도 나중에 display 필터 재 설정을 통하여 다시 모니터링 할 수 있다.

3) 파일 저장 및 읽기 : 전송된 패킷을 분석 한 후에도 다음에 다시 볼 수 있도록 데이터 파일 저장 및 읽기를 수행한다.

4) View 변경 : 메인 그리드 화면에서 사용자가 보기 원하는 정보만 볼 수 있도록 일부 칼럼을 화면에서 지우거나 다시 볼 수 있도록 구현하였다. 또한 메인 그리드 화면 아래에 있는 일반적 general 패킷 detail과 protocol detail 정보를 화면상에서 지우거나 다시 생성할 수 있도록 하였다.

5) Display 변경 : 분석된 정보는 decimal 또는 hexadecimal 정보로 볼 수 있도록 data format 변경한다. 또한 각 패킷이 전송된 시간에 대해서도 년,월,일을 포함한 자세한 시간대, 또는 시,분,초만을 표시하거나 초와 밀리초 별로 확인 가능 하도록 time stamp format 설정을 변경한다.

Time stamp mode는 패킷의 전송 시간을 바꿀 수 있다. Absolute는 절대적인 시간을 표시하고, start with 0는 첫 번째 패킷 시간을 0으로 하였을 때 그 뒤에 전송된 패킷의 시간차를 더하여 표시하게 된다. Differential 은 바로 앞에 전송된 패킷과의 시간차만을 표시한다.

6) 통계 기능 : 분석된 모든 패킷은 각 layer별 또는 기타 기능별로 통계를 내어 보다 편하게 패킷을 분석할 수 있도록 한다.

7) 시뮬레이션 기능 : 사용자가 직접 패킷을 만들어 프로그램 자체적으로 raw 패킷을 확인 가능하도록 구현하였다. 이 기능은 단지 프로그램 자체적인기능으로 실제적으로 론네트워크에 데이터를 보내지 않는다.

3.3 프로그램 흐름도

응용 프로그램의 분석 흐름도는 그림 4와 같다.

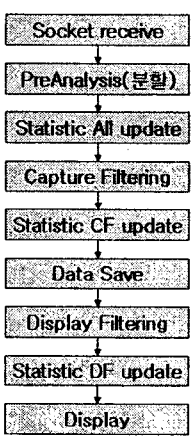

그림 4 응용 프로그램 흐름도

프로토콜 분석기로부터 전송된 패킷은 TCP/IP 소켓 버퍼에서 받아지고, 분석할 수 있도록 각 레이어 별로 패킷을 분할한다. 패킷을 분할 할 때는 프로그램 내에서 각 바이트 단위의 변수에 레이어 별 포맷에 맞도록 저장하고, 분할 된 변수를 이용하여 프로그램 실행 시간을 되도록 단축시킨다. 이

때 정확한 패킷인지 에러 패킷 인지 확인하도록 각 레이어 포맷에 맞는지 체크한다. 포맷에 맞지 않는 에러 패킷일 경우 그에 해당하는 레이어 별 에러 상황을 메인 그리드에 표시하여 에러 패킷임을 알린다. 이 에러 패킷은 더 이상 분석되어지지 않고 다음 패킷 분석을 기다린다. 전송된 모든 패킷에 대하여 통계를 구하고, 사용자가 원하는 패킷만을 저장하기 위하여 capture 필터 기능을 수행하게 된다. Capture 필터를 통과한 패킷은 capture 필터 통계를 업데이트 하고, 프로그램 메모리 내에 저장된다. 저장된 패킷은 사용자가 모니터링을 하기 원하는 패킷인지 알기 위해 display 필터를 거치고 다시 display 필터 통계치를 업데이트 하게 된다. 마지막으로 최종적으로 화면 설정에 맞게 패킷을 분석하고 화면상에 나타내게 된다.

LonWorks 네트워크에서 에러 패킷이 확인되면 다음과 같이 처리하도록 한다..

1) 패킷 길이 확인

패킷의 길이를 확인하여 프로토콜 포맷에 맞는 패킷인가 확인한다. 최소 길이보다 작거나 최대 길이보다 길면, 패킷을 분석하기 전부터 short_length 및 long_length 에러를 표시하고 분석 알고리즘을 빠져 나온다

2) CRC 에러 체크

CRC를 확인하여 에러가 발생하였을 경우 CRC_error를 발생하게 한다.

3) 분석 단계에서의 에러

분석 과정에서 각 레이어별 포맷에 맞는지 확인하여 각 데이터 필드에 맞는 에러를 표시한다. 이러한 에러 패킷이 발생하게 되면, 에러 패킷에 대하여 통계치를 업데이트하고 그리고 분석 단계를 빠져 나와 다음 패킷을 기다리게 된다.

에러 패킷의 발생 원인은 1) 초기에 프로토콜 분석기 동작 시 네트워크에 전송 되고 있는 패킷의 중간 부분부터 디코딩 된 경우, 2) 네트워크에 ANSI/EIA 709.1 프로토콜을 사용하지 않는 다른 디바이스가 연결되어 다른 프로토콜을 전송하는 경우, 3) 마스터 프로세서와 슬레이브 프로세서 사이에서 통신 에러가 발생하는 경우 등이 있을 수 있다.

4. 결 론

본 연구에서는 ANSI/EIA 709.1을 내장한 여러 디바이스들로 구성된 LonWorks 네트워크의 패킷을 분석하고 모니터링 할 수 있도록 프로토콜 분석기를 하드웨어적으로 구현하였고, 일반 PC상에서 그 패킷을 분석할 수 있도록 응용 소프트웨어를 개발하였다.

개발된 분석기는 실제로 LonWorks 네트워크에서 디바이스 간의 데이터 통신을 효과적으로 모니터링 할 수 있었다. 개발된 장비를 이용하면 네트워크 관리자는 보다 용이하게 네트워크를 관리할 수 있을 것이다.

참 고 문 헌

[1] H. Shahnasser and Q. Wang, "Controlling Industrial Devices over TCP/IP by Using LonWorks," Proc. IEEE, pp. 1309-1312, 1998.

[2] Motorola, LonWorks Technology Device Data, 1996.

[3] Echelon, Engineering Bulletin, 1999.

[4] EIA Standard : Control Network Protocol Specification.

[5] Dietmar Loy, Dietmar Dietrich and Hans-joerg Schweinzer, "Open control networks : LonWorks/EIA 709 technology", 2001

[6] G. H. Choi, G. S. Choi, and J. S. Kim, "LonWorks Based Virtual Device Network(VDN) for Predictive Maintenance", ICMIT'01, Yamaguchi Seminar Park, Japan, 2001

[7] H. Shahnasser and Q Wang, "Controlling Industrial Devices over TCP/IP by Using LonWorks" Proc.IEEE, 1998.

[8] O. Vajamaki A. Allen and J. Gaff, "High Speed Peer-to-Peer Communication System for Integrated Protection and Control in Distribution Network", Development in Power System Protection, IEEE, pp.341-347, 1997.

[9] Stefan Soucek, Thilo Sauter and Gerald Koller, "Impact of Qos Parameters on Internet-Based EIA-709.1 Control Applications", IEEE, pp 3176-3181

[10] Marek Miskowicz, "Analysis of the LonTalk/EIA-709.1 Channel Performance under Soft Real-Time Requirements", IEEE, pp.705-708, 2003.

[11] Marek Miskowicz, Maria Sapor, Marcin Zych and Wojciech Latawiec, "PerFormance Analysis of Predictive p-Persistent CSMA Protocol for Control Networks", 4th IEEE International Workshop on Factory Communication Systems, Vasteras, Sweden, August 28-30, 2002.