전동차 차량 네트워크 성능 모니터링 시스템에 관한 연구

A Study of monitoring system for train communication networks

이대은* 손수국** 손강호*** 전성준****

Dae-Eun Lee **Su-goog Shon** **Kang-Ho Shon** **Seong-joon jeon**

ABSTRACT

A few years ago, The trains had little control equipment so the trains had little data transmission between control equipment. Recently, there is a lot of control equipment in a train as traction control, air conditioners and even internet access. for this reason, vehicle network must allow for the big amount of transmission data and must ensure the high reliability. In this paper we present monitering system for verify high reliability of data transmission of MVB of TCN which is an international standard of IEC 61375.

1. 서 론

1.1 연구배경

기존 전동차 내 디바이스간의 프로토콜은 중앙 제어기에 의존하는 중앙 집중 방식으로 단순 제어나 감시 기능만을 지원하였고 전동차내의 프로토콜에 대한 신뢰성이 그리 중요시 되지 않았다. 그러나 최근에 와서는 전동차 내 디바이스간의 제어기술이 발달하면서 분산 환경의 실시간 제어, 온라인 감시, 자가진단 및 승객 정보 서비스 등 다양한 서비스를 제공하면서 디바이스간의 빠르고 정확한 데이터 교환을 보장하기 위한 네트워크 신뢰성 향상에 많은 연구가 진행되어왔다. 또한 하나의 중앙 제어기가 다양한 디바이스를 관리하는 중앙 집중식 프로토콜을 사용하기 때문에 시스템의 제어가 간단하다는 장점은 있지만 소수의 중앙 제어기가 다수의 디바이스를 관리해야하기 때문에 중앙 제어기에 대한 의존도가 높고, 네트워크의 신뢰성을 보장하기 어렵다는 단점이 있다. 본 논문에서는 이러한 네트워크 문제점을 해결하기 위하여 MVB 상의 네트워크 트래픽을 모니터링 함으로써, 네트워크의 신뢰성을 검증하기위한 MVB 프로토콜 분석기를 개발하였다. 이러한 네트워크 프로토콜의 성능분석은 네트워크 시스템 구축뿐만 아니라 응용 프로세스 개발에도 필수적이다.[1][2][3]

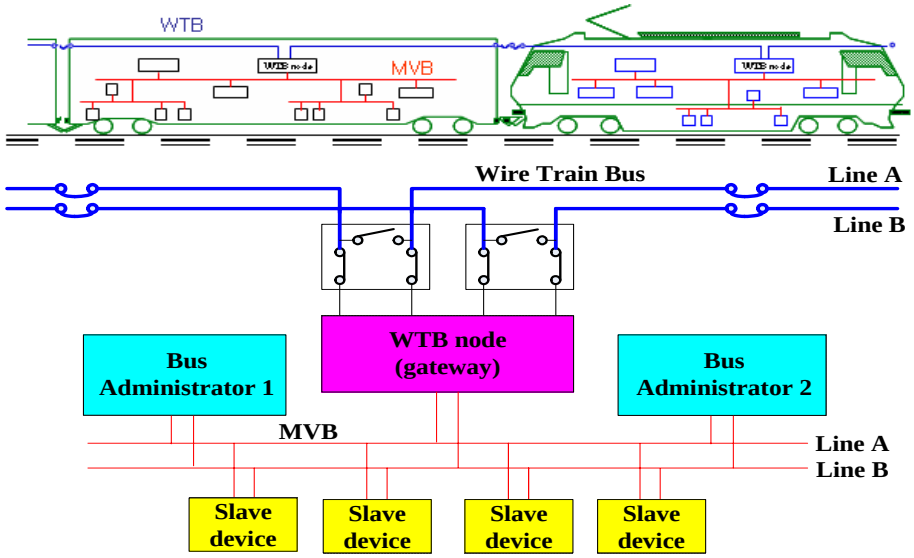
2. 전동차 통신 네트워크 (TCN)

2.1 TCN의 소개.

오늘날의 전동차 시스템은 차량제어, 공기조절장치, 고객정보 서비스 심지어 TV나 인터넷 접속 같은 다양한 임무를 수행하는 디바이스들을 포함하게 되었다. 또한 전동차 통신 네트워크의 특성상 하나의 버스를 여러 개의 디바이스들이 공유하여야 하고, 차량 제어같이 시간 제약성이 강한 time-critical data와 시간 제약성은 작지만 크기가 큰 메시지 데이터를 동시에 처리할 수 있어야 한다. 또한 차량 운행모드 변경에 따른 네트워크 디바이스의 재구성을 만족시키기 위해 디바이스의 초기화 및 디바이스 자동인식 등을 보장해야 하며 이러한 전동차 통신 네트워크의 특성상 발생하는 전송지연에 대한 신뢰성을 보장해야 한다. 이러한 전동차 네트워크의 특성 때문에 차량내의 분산되어있는 다양한 디바이스

* 수원대학교, 대학원생, 비회원
E-mail : choki99@naver.com
TEL : 010-4664-3652
** 수원대학교, 교수, 비회원
*** 인터콘시스템스(주), 정회원
**** 인터콘시스템스(주), 비회원

사이의 빠르고 정확한 정보교환을 보장하기 위한 많은 노력들이 이루어져왔다. 이러한 노력들 중에 하나로 IEC(International Electrotechnical Commission)에서는 차량 간 혹은 차량 내 디바이스들의 상호운영성(Interoperability)을 목적으로 TCN(Train Communication Network)을 통신 프로토콜 표준으로 채택하였다. 차량제어 통신 필드 버스 방식의 국제 표준인 TCN은 그림 1에서 보는 것과 같이 차량간 디바이스의 통신을 연결하는 WTB (Wired Train Bus)와 차량내의 디바이스간 통신을 연결하는 MVB(Multifunction Vehicle Bus)의 이중 계층 구조로 이루어져 있다.



< 그림 1. TCN의 이중 계층 구조 >

2.1.1 차량 간 통신 버스 (WTB)

서로 다른 차량에 있는 디바이스들간의 통신방식인 WTB (Wired Train Bus)는 기존의 UI leaflet 556과의 호환성을 위하여 중계 장치 없이 최대 860m에 22개의 차량과 32개의 노드를 지원하도록 설계되었으며, 차량의 구성 변화에 따른 디바이스의 자동 인식 (plug & play)기능과 디바이스 고장으로 인한 피해를 최소화하기 위한 디바이스 초기화 및 네트워크의 재구성에 중점을 두고 있다. [8]

2.1.2 차량 내 통신 버스 (MVB)

차량내의 디바이스들을 연결하기위한 통신방식인 MVB (Multifunction Vehicle Bus)는 일반적으로 차량내의 디바이스들의 환경이 자주 변경되지 않는다는 특성과 디바이스간 데이터 통신의 높은 신뢰성을 보장해야 한다는 특성을 위해 설계되었다. MVB의 연결 구조는 RS-485표준을 따르는 ESD(Electrical Short Distance) bus segment와 twisted wire pair를 사용하는 EMD (Electrical Middle Distance) bus segment, 광섬유를 사용하는 OGF (Optical Glass Fiber) bus segment를 제공한다. 표1에서는 WTB와 MVB를 구성하는 구성요소와 서로 다른 특성들을 보여준다.

< 도표1. 차량내 버스와 차량간 버스의 특성 비교 >

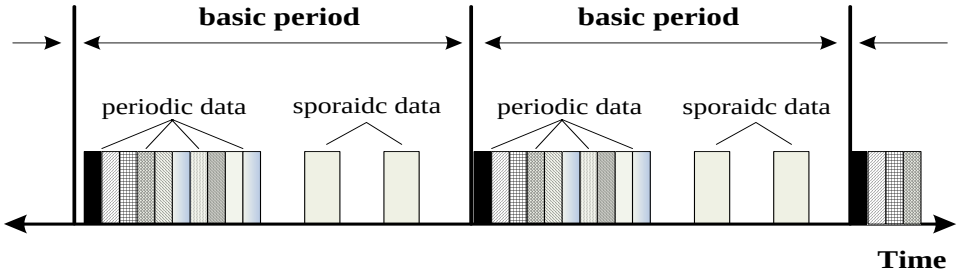
특 성	WTB	MVB	특 성	WTB	MVB
구 조	차량 구성이 변함. self-configuration.	고정된 네트워크 구성과 디바이스 어드레스	Number	차량 내 32개 이하	차량 내 256개 이하
전송매체	shielded twisted wire pair	▪ ESD ▪ EMD ▪ OGF	구 성	connectable in the field	fixed, pre-connected
전송속도	1.0Mbit/s	1.5Mbit/s	Medium Access	25ms	1ms
주소영역	8bit address	12bit address	Link Layer Services	▪ Process_Data. ▪ Supervisory_Data.	▪ Message_Data.

2.2 TCN의 데이터 서비스

TCN을 구성하는 차량간 통신 버스(WTB)와 차량내 통신 버스(MVB)는 설계되어진 목적과 물리적인 연결 구조는 다르지만 동일한 통신 구조를 가지고 있다. WTB와 MVB는 모두 프로세스 데이터, 메시지 데이터, 관리용 데이터 이렇게 3가지 형태의 데이터 구조를 갖는다. 이러한 통신 버스 데이터 구조중 하나인 프로세스 데이터는 전동차의 속도, 모터의 전류 및 전동차의 제어 명령 같은 데이터 크기는 작지만 시간 제약성이 강한 데이터를 포함한다. 이러한 프로세스 데이터들은 주기적으로 전송되어지는 마스터 프레임에 의한 응답으로서 제공 되어 진다. 프로세스 데이터의 주기는 기본 주기(basic period)의 정수배를 가기고 데이터 소스에서 브로드케스팅으로 전달되어진다. [10]

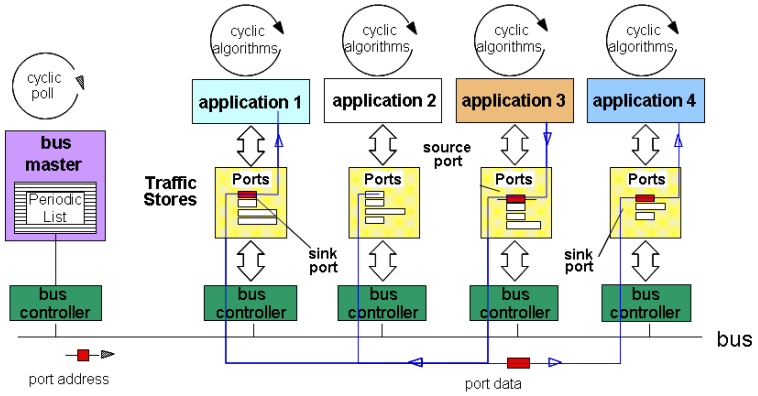
또 다른 데이터 구조중 하나인 메시지 데이터는 프로세스 데이터에 비해 시간 제약성은 약하지만 비교적 긴 데이터(승객정보, 진단정보 등)를 갖는 이벤트성 데이터를 포함한다. 이러한 메시지 데이터들은 비주기적이며 브로드케스팅뿐만 아니라 지정된 어드레스에만 전달할 수도 있다. 메시지 데이터는 그 특성상 이벤트가 발생할 때마다 비주기적으로 데이터 갱신이 이루어지며 각 노드는 데이터 갱신이 일어나면 마스터 노드에게 메시지 데이터 전송요구를 하게 된다.

프로세스 데이터 및 메시지 데이터와는 별개로 네트워크의 유지보수의 목적으로 이벤트나, 마스터 권한 이양, 디바이스의 상태정보를 전달하기 위한 관리용 데이터 (Supervisory Data)는 각 데이터의 특성에 따라 프로세스 데이터 서비스 혹은 메시지 데이터 서비스와 같은 방식으로 사용되어진다. 또한 TCN은 이러한 데이터 타입들의 신뢰성을 보장하기 위해 그림 2에서 보는 바와 같이 basic period내에서 주기적 전송구간(Periodic phase)과 비주기적 전송구간 (Sporadic phase)으로 구분하는 시분할 방식을 채택하고 있다.



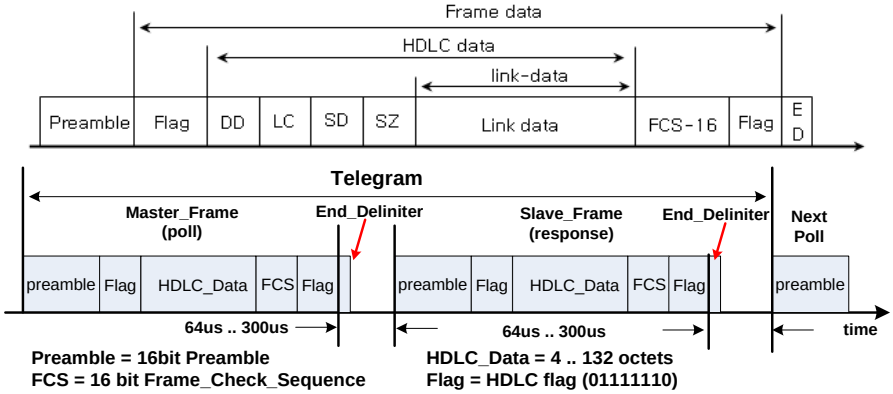
< 그림 2. TCN의 Basic Period >

주기적 전송 구간에서 데이터는 포트 단위로 전송되기 때문에 각 포트의 고유 전송 주기인 개별 주기(Individaul Period)를 근거해 스케줄링하게 되며, 그러 인해 실시간 통신을 보장하고 각 노드에 대한 전송 지연 시간을 보장하게 된다. 또한 마스터 프레임과 슬레이브 프레임 사이의 시간 간격이 4ms 보장해야 하고 이러한 짧은 시간 간격을 만족시키기 위하여 슬레이브 프레임에 보낼 데이터들을 미리 포트(Port)라 불리는 레지스터에 저장한다. 그리고 각 디바이스들은 각 포트에 대한 임의의 숫자를 가지고 있으며, F_code에 의해 소스 포트, 또는 싱크 포트로 구분되어 전송된다. 이러한 이유로 각 디바이스들에게는 가상 데이터베이스와 같이 논리적 포트가 배정되어 있어 특정 데이터를 다른 디바이스들과 공유할 수 있다. 데이터의 주기적 전송을 위하여 버스 마스터는 전송되어야 할 논리적 포트 리스트를 생성하고 제공한다. 따라서 각 디바이스들은 각 프로세스 데이터를 정해진 주기로 전송하고, 이를 다른 디바이스들이 받을 수 있게 된다. 그림 3은 주기적 전송구간에서 프로세스 데이터를 데이터 세트 형식으로 그 주기에 따라 데이터 서비스하는 것을 보여주고 있다. 주기적 전송은 데이터의 유효성과는 무관하게 전송되어 지기 때문에 낭비적인 요소가 많고, 이벤트 발생과 같은 기능을 구현 할 때 발생하는 전송 지연을 피할 수 없다는 단점이 있다. [1][3]



<그림 3. Process Data Exchange>

비주기적 전송구간에서는 디바이스의 전송 요구에 의하여 마스터가 순차적인 폴링에 의한 전송을 사용한다. 차량 내 버스에서는 어드레스에 대한 탐색 트리구조를 바탕으로 매체 접근을 제어하는 반면 차량 간 버스에서는 HDLC 프레임 형식을 사용하여 요청에 의한 순차적인 폴링에 의한 전송을 사용한다. 모든 프레임은 HDLC(ISO 3309) 형태로 그림 4처럼 되어 있다. 각 프레임은 preamble에 의해서 시작되고, 8bit의 flag, 8bit의 Destination Device(DD), 8bit의 Link Control(LC), Link Data Size, Link Data, 그리고 End Delimiter로 구성되어 있다.[10]



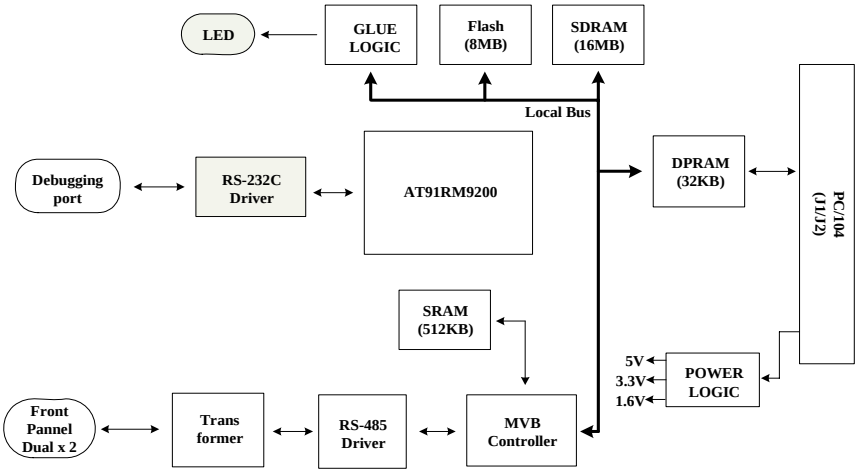
<그림 4. 비 주기적 전송구간에서의 데이터 프레임>

3. MVB 프로토콜 분석기

오늘날 과학의 발전과 이를 접목한 산업의 발전은 전동 차량의 주요 기기들에게도 영향을 주고 있다. 전동 차량의 신뢰성 있는 제어를 위하여 마이크로 프로세스를 장착하고 제어를 위하여 기기의 상태를 얻어오거나, 원인 규명이 어려운 부분에 마이크로 프로세스 기술을 결합하여 진단하고 있다. 그로 인하여 축적된 차량 제어 기술과 모니터링 기술을 접목, 발전시켜 네트워크의 상태를 진단하고 문제점을 발견하여 이를 해결 할 수 있는 방안 수립이 가능해 졌다. 본 논문에서는 네트워크의 상태를 진단할 수 있는 시스템을 만들기 위해 MVB 통신 보드, MVB 프로토콜 분석기, PC환경의 모니터링 프로그램을 제작하였다.

3.1 MVB 통신 보드

본 논문에서 제작한 MVB통신 보드는 TCN(Train Communication Network, IEC 61375-1)통신 규약 중 MVB (Multifuntion Vehicle Bus)통신을 만족하는 독립적인 보드로써, 전동차량에 적용될 TCN의 MVB 통신을 담당하는 Local Device에 장착되어 MVB 통신 라인으로 연결된 시스템의 MVB 통신을 담당한다. 외부 Data Interface를 위하여 PC104 Bus 규격의 인터페이스를 지원하며, Local Device의 LED와 Runtime Debugging을 위한 RS-232C 통신 포트를 가지고 있다. 그림 5는 MVB 통신 보드 블록도를 보여주고 있다.



<그림 5. MVB 통신 보드 블록도>

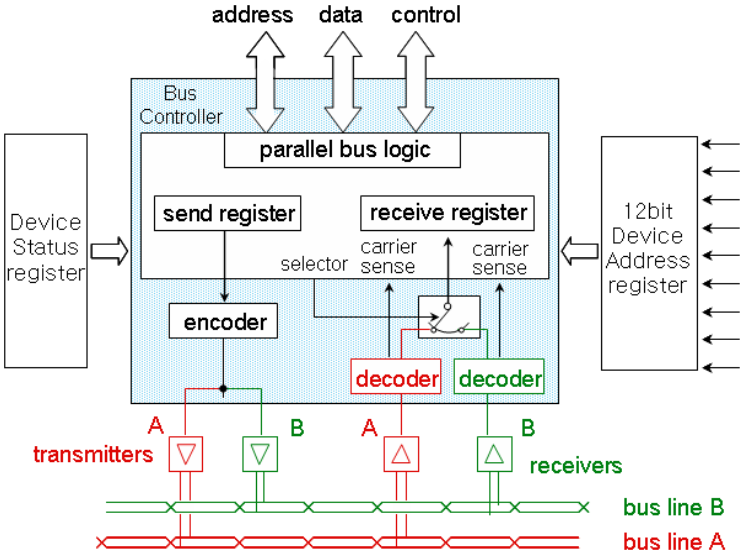
MVB통신 보드의 구조는 크게 Internal Interface, Control Block and External Interface로 구성된 3개의 메인 블록으로 구성되어 있다. 이 블록들은 16bit 데이터 버스 구조를 지원 한다.

3.1.1 Internal Interface

Internal Interface는 Main Processor의 local bus와 연결되는 intenal bus로써 Traffic Memory에 전송받은 데이터나 기타 컨트롤 비트를 저장하는 행위가 일어날 경우 혹은 그 반대의 경우 폴링에 의한 방법으로 들어온 데이터를 Main Processor로 전달하는 역할을 한다.

3.1.2 Control Block

Control Block은 크게 Main Processor와 MVB Controller로 나누어진다. 각 장치들은 각자의 기능에 맞는 작업을 수행한다. MVB Controller는 MVB와의 통신을 담당하는 block으로써 512Kbyte의 전용 Traffic Memory를 갖는다. MVB Controller는 Traffic Memory의 내용을 전송하거나 전송 받은 데이터를 Traffic Memory에 저장하고 이에 따른 에러나 기타 컨트롤 비트를 저장하는 기능을 수행한다. 본 논문에서는 TCN 프로토콜의 내용을 만족시키기 위해 Adtranz사가 개발한 MVB 전용 Controller인 MVBC를 사용하였다. 그림 6은 Adtranz사의 MVBC의 구조 전송부와 수신부 그리고 상태값들을 저장하는 레지스터부로 구성되어 있다.



<그림 6. MVBC 블록 다이어그램>

Main Processor는 논리적인 데이터 처리와 I/O Interface, MVBC와의 통신을 처리하기 위해 high speed microcontroller이다. ARM Thumb Processor인 ARM920T를 사용하였고 프로그래밍과 Processor operation의 모니터링 서비스를 위한 시리얼 인터페이스를 지원한다.

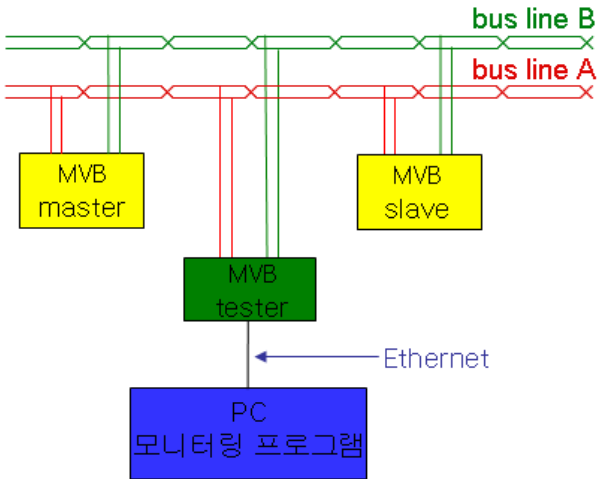
3.1.3 External Interface

External Interface는 외부와의 데이터 교환을 위해 사용되어 진다.

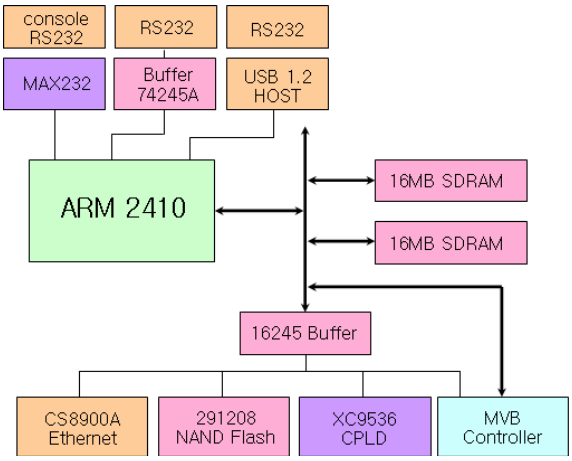
- RS-232 Interface : Processor Operation의 모니터링을 위해 User Interface로써 사용되어 진다.
- RS-485 Interface : TCN 프로토콜 상의 MVB bus에 연결되어있는 디바이스간의 연결을 위해 사용되어지는 Interface이다.

3.2 MVB 프로토콜 분석기

MVB 프로토콜 분석기는 TCN(Train Communication Network, IEC61375-1) 통신규약 중 MVB (Multifunction Vehicle Bus) 상에 연결 되어지는 독립적인 디바이스로써, 그림 7에서와 같이 마스터 디바이스와 슬레이브 디바이스 사이에 연결되어 마스터 디바이스에서 슬레이브 디바이스로 전송되는 마스터 프레임과 마스터 프레임의 응답으로서 전송되어지는 슬레이브 프레임 또는 디바이스 상태정보를 분석하여 원격에 설치되어있는 PC환경의 모니터링 프로그램에서 TCP/IP 프로토콜을 통해 분석되어진 데이터를 전송하는 시스템이다. MVB 프로토콜 분석기의 운용 시스템 구성도는 그림 7과 같다.
MVB 프로토콜 분석기는 Samsung S3C2410A-266을 탑재한 리눅스 기반 시스템이다. MVB 프로토콜 분석기는 커널 2.6.8이 포팅되어 있고, Processor Operation 모니터링을 위한 RS-232C Interface와 PC 환경의 모니터링 시스템과의 인터페이스를 위한 이더넷 포트를 지원한다. 그림 8는 MVB 프로토콜 분석기의 하드웨어 블록도를 보여주고 있다.



<그림 7. MVB 프로토콜 시스템 구성도>

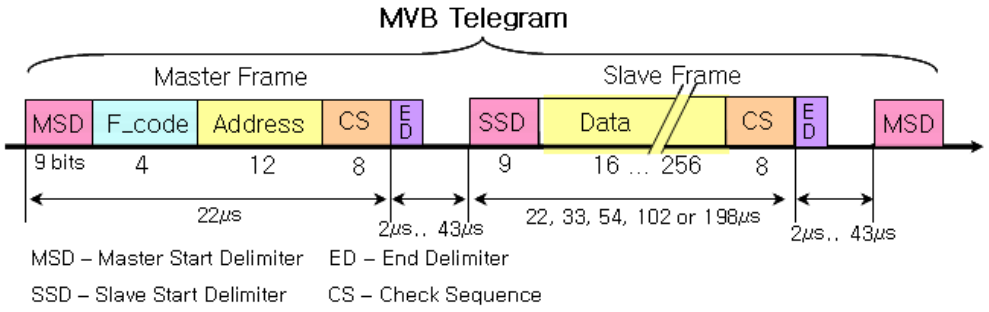


<그림 8. MVB 프로토콜 분석기 블록도>

MVB 프로토콜 분석기는 FALinux(주) 사에서 제공한 이지부트라는 부트로더를 사용하여 하드웨어에 대한 초기화(cpu speed, memory, interrupt, uart(시리얼 관련)를 수행하고 호스트상에 컴파일된 커널이나 Root 이미지를 시리얼 및 이더넷 포트를 이용하여 쉽게 다운로딩 할 수 있다.
TCN 분석기의 전원이 인가되면 부트섹터에 있던 부트로더는 NAND-Flash에 있는 커널과 root 이미지를 압축을 풀고 각각 SDRAM의 0x30000000 번지와 0x30800000번지에 데이터를 로드한다. 로드된 커널은 초기화 후 root 파일 시스템에 저장되어있는 MVB 프로토콜 분석기 운용 펌웨어 프로그램을 실행시키게 된다.

3.2.1 MVB 프로토콜 분석기 운용 펌웨어

MVB 프로토콜 분석기 운용 프로그램은 디바이스 드라이버 형태로 root 파일 시스템에 저장되어진다. 이 MVB 프로토콜 운용 프로그램은 MVB line상에 돌아다니는 프레임들을 분석하여 연결되어있는 네트워크의 성능 및 신뢰도를 판단한다. MVB line 상에 돌아다니는 프레임은 마스터 프레임과 슬레이브 프레임으로 구분할 수 있고 각 프레임의 구조는 그림 9과 같다.

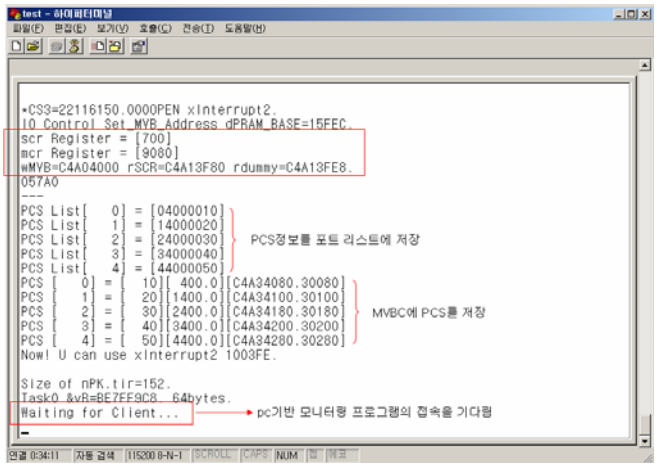


<그림 9. 주기적 전송구간에서의 데이터 프레임>

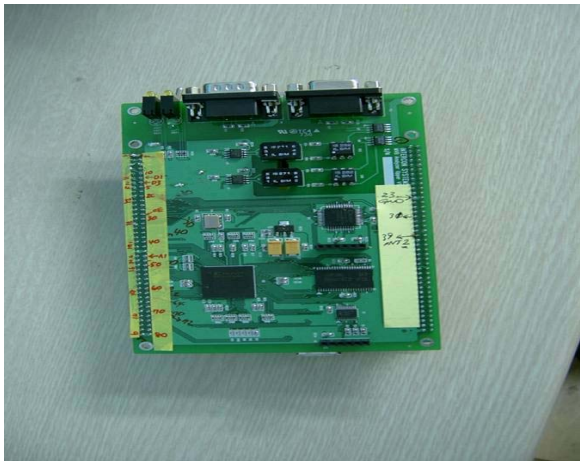
MVB 프로토콜 분석기는 각 프레임의 시작 Delimiter를 분석함으로써 마스터 프레임과 슬레이브 프레임을 구분하고 마스터의 F_Code를 이용하여 이 마스터 프레임의 응답으로 전송되는 Slave Frame의 길이와 data의 속성을 분석한다.

3.2.2 프로토콜 분석기 운용 프로그램의 구조

이 펌웨어는 크게 interrupt 부분과 network 부분으로 나누어진다. interrupt 부분은 각 트래픽 메모리 영역을 초기화 하고 host로부터 받은 PCS정보를 port list에 저장하고 MVBC에 PCS를 등록한다. 또한 Traffic Memory와 DPRAM의 데이터 통신을 담당한다. network 부분은 MVB 프로토콜 분석기의 상태 LED부분을 제어하고 PC 환경의 모니터링 프로그램과의 소켓 통신에 사용된다. MVB 프로토콜 분석기의 소켓은 non-blocking mode로 동작하며 모니터링 프로그램으로부터 접속 요청이 들어올 때까지 대기하고 있다가 접속 요청이 들어오면 소켓을 연결하고 분석되어진 MVB 트래픽 데이터를 소켓을 이용하여 모니터링 프로그램으로 전송한다. 그림 10에서 보는바와 같이 PCS 레지스터 값들을 이용하여 MVB 프로토콜 분석기의 초기화 과정을 보여주고 있다. 그림 11는 MVB에 연결되는 MVB 프로토콜 분석기의 실제 모습이다.



<그림 10. MVB 프로토콜 분석기의 초기화>



<그림 11. MVB 프로토콜 분석기>

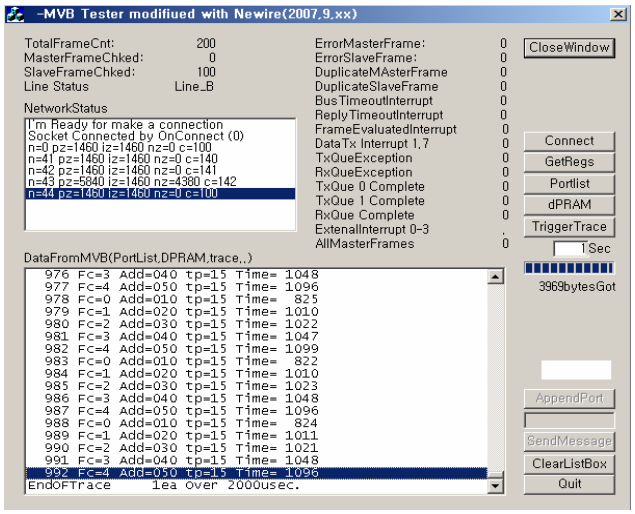
3.3 모니터링 프로그램.

pc환경의 모니터링 프로그램은 total frame count, frame checked, reply timeout interrupt 같은 네트워크 성능 평가에 필요한 데이터 값들을 모니터링 하는 프로그램이다. 이 모니터링 프로그램은

MVB 프로토콜 분석기에서 TCP/IP 프로토콜을 이용해 보내오는 정보를 수집하고 분석하여 모니터링 한다. 이 모니터링 프로그램은 그림 12에서 보는바와 같이 visual c++ 6.0을 이용하여 구현하였고 아래와 같은 기능들을 이용하여 네트워크의 성능 및 상태를 분석할 수 있다.

- frame counter display
- internal registers display
- Portlist display
- dPRAM display
- Trigger Trace.

특히 일반적인 프레임들의 카운터만이 아닌 trigger trace를 통해 정해진 시간동안에 MVB상의 데이터 프레임 들을 display 할 수 있다.



<그림 12. PC환경의 모니터링 프로그램>

4. 결 론

현재의 전동차 제어 시스템은 정보화 시대에 발맞추어 차량에 탑재된 각종 제어 장치로부터 들어오는 여러 가지 다양한 정보를 실시간으로 처리하고 모든 정보를 쉽게 접근할 수 있어야 하며, 또한 위험한 상황이 발생 시 즉시 조치를 취할 수 있어야 한다. 이러한 상황에 따라 정보의 다양화, 업무의 효율화, 운행간격의 단축, 편리성의 향상 등을 목표로 단순한 모니터링이 아닌 차량 전체의 제어와 운행 정보의 통합이 이루어진 통합시스템으로서의 모니터링이 이루어져야 한다. 이러한 이유로 본 논문에서는 차량 전체의 제어와 운행 정보의 통합이 이루어진 통합시스템 구축을 위한 첫 번째 단계로써, 차량제어 통신 필드 버스방식의 국제표준인 TCN의 MVB상의 네트워크 성능을 평가하기 위한 시스템을 구현하였다. 또한 마스터 노드와 슬레이브 노드를 제작하여 실제 MVB 환경을 구축하였고 MVB 프로토콜 분석기를 마스터 노드와 슬레이브 노드 사이에 연결하여 원격지에 있는 서버에게 데이터를 보냄으로서 가상의 MVB 상에 존재하는 데이터들을 분석하였다. 그로 인해 MVB 상의 네트워크 성능 및 네트워크의 상태를 평가할 수 있음을 확인하였다. 본 논문에서는 MVB상의 네트워크 환경을 분석하고 판단하는 연구를 수행하였지만, 향후 TCN 전체의 네트워크 환경을 분석하고 판단하는 시스템을 구현하기 위한 방법도 고려되어져야할 것으로 생각된다.

참고문헌

[1] IEC 61375-1 Standard Train Communication Network : Part (1) General Architecture (2) Real-time Protocol (3) Multifunction Vehicle Bus (4) Wire Train Bus (5) Train Network Management (6) Train Communication Conformance Testing, 1999.

[2] G.Fadin and F.Cabaliere, "IEC TC9 WG22 train communication network dependability and sagety concepts, "World Congress on Railway Research 97, 1997.

[3] H.Kirrmann and P.A. zuber, "IEC/IEEE train communication network." 1996.

[4] UIC 556 Standard, Information Transportation on the Train Bus, 1999.

[5] Chavarria, J.L de Arroyabe, A. Zuloaga, J.Jimenez, J.L. Martin, G. Aranguren, "Slave node architecture for train communication networks," IEEE Conference, 2000.

[6] ABB Daimler-Bensz Transportation(Switzerland) Ltd, "Multifunction vehicle Bus Controller", Adteanz, 1997.

[7] jae-youn you and jae-jyun park, "A Study on the Implementation of the Fault-Injector for the Fault Tolerant Train Communication Network.

[8] Sang-chul Lee, Jea-Hyun Park, Nae-Hyuck Chang, "A Study on the Implementation and the Performance Evaluation of the Train Communication Network.

[9] myeong-ho Choe, "A Study on the Performance Improvement of Message Transmission over MVB.

[10] yeong-gi jo, "A Study on TCN/Node Redundancy Scheme for KHST