

신호제어시스템 시뮬레이터용 소프트웨어 설계

Software Design for Railway Signaling System Simulator

황종규*, 이종우*, 정의진*, 김태진*
Jong-Gyu Hwang Jong-Woo Lee, Eui-Jin Joung, Tae-Jin Kim

ABSTRACT

The railway signaling system consists of microcomputerized vital devices on board and ground, which are connected to one another by track circuits, and interlocking equipment for route control. Therefore it is important to validate the required functions of developed system and interface between developed signaling systems. To verify the conditions and functions of signaling logic, the laboratory prototype test bench, which consists of personal computers with software module and Ethernet LAN, will be developed. In this paper general design of signaling system test bench for high-speed rail is described and developed software module is presented.

1. 서론

철도 신호제어시스템은 열차운행의 안전성을 보장하는 최종적인 시스템으로서, 기존에는 안전성을 고려하여 릴레이를 이용한 하드웨어 시스템으로 구축되어져 왔으며, 열차제어의 본래의 기능만을 수행하였다. 따라서 열차운행의 고밀도화와 여객수송이 다양화되고 있는 등 사회적 요구에 대응하기 위해서, 이러한 요구되는 서비스 기능들을 실현할 수 있는 마이크로일렉트로닉의 신기술을 이용하여 새로운 시스템을 신호제어시스템에 도입하는 경향이 급속히 확산되고 있다.

이와 같은 소프트웨어를 기반으로 한 새로운 기술의 도입은 시스템의 기능향상을 시킬 수 있지만, 기존의 릴레이를 기반으로 한 Fail-safe 특성에 관한 안전확보 방법이 다르게 되어, 안전성에 관련된 충분한 평가 및 예측을 행할 수 없기 때문에 중대한 사고를 초래할 수 있는 위험성이 있다. 이러한 이유로 시스템의 실용화에 앞서, 개발단계에 있는 시제품들의 특성을 실험에 의해 파악함과 동시에, 새로운 방식의 운전보안 시스템 안전성, 신뢰성을 검증하는 시뮬레이터의 개발이 필요하다[1]-[4].

본 논문에서는 신호제어시스템을 실험할 수 있도록 하는 모의 시험장치의 개발 방안과 주요 소프트웨어 모듈에 대해 살펴보고자 한다. 시뮬레이터는 각 시제품의 기능 검증 및 각 장치간의 인터페이스 확인을 주목적으로 개발이 진행되고 있다[4].

* 한국철도기술연구원, 정회원
* 한국철도기술연구원, 비회원

2. 시뮬레이터의 구성

시뮬레이터는 폐색구간을 바탕으로 하는 열차들간의 간격제어, 진로제어, 연동기능 처리 등 핵심적인 논리들의 검증과 각 제어장치들간의 인터페이스 성능검증을 목적으로 한다. 시뮬레이터 구성을 위한 기본적인 조건 및 방법을 다음과 같이 설정하였다.

- CTC나 지역제어조작반 등을 통하여 진로제어 등의 명령을 인가하는 부분을 PC를 기반으로 하여 별도의 프로그램을 작성하여 구현하고, 각각 진로를 입력할 수도 있고 또한 열차 다이아 데이터를 입력하여 전체적으로 동작될 수 있도록 한다.
- 궤도회로 등 실제 노선과 선로변의 신호장치들을 별도의 소프트웨어를 작성하여 모의한다. 또한 궤도회로나 신호기 등을 편집할 수 있는 소프트웨어를 통해 실제 노선을 소프트웨어로 구현한다. 이 소프트웨어를 이용하여 다양한 형태의 노선을 편집하여 신호로직을 검증할 수 있도록 한다.
- 지상제어장치와 차상제어장치간의 데이터 전송기능을 위하여, 시뮬레이터에서는 이들 각 장치들을 별도의 장치로 구성한다. 그리고 실제로 궤도회로 및 차상 센서를 통해 지상에서 차상제어장치로 데이터 전송이 이루어지나 시뮬레이터에서는 데이터전송 수단을 이에 대응하는 Ethernet LAN을 이용하여 각 폐색구간의 정보를 전송한다.
- 각 장치들간의 인터페이스는 Ethernet으로 하고, 각 장치들의 핵심 모듈들이 구현될 하드웨어는 VME Bus를 기반으로 한 사용 하드웨어를 사용한다.
- 시스템 규모가 커지는 것을 막기 위해 이미 일반화되어진 신호시스템의 기능들을 가능한 한 최소화하고, 핵심이 되는 모듈과 바이탈한 모듈들에 대해서 주로 시뮬레이션 하도록 한다.

실험시스템은 신호로직의 확인을 위하여 지상제어장치, 차상제어장치, 연동장치 등으로 구성한다. 각 장치는 PC와 VME 시스템을 이용한 각 제어시스템으로 구성하고, 이들 장치간에는 Ethernet LAN을 이용하여 서로 데이터를 주고받을 수 있도록 한다.

현장 선로변의 신호설비들은 실험실 레벨에서 구현할 수 있도록 소프트웨어로 구성하였다. 즉, 현장 신호설비들인 궤도회로, 전철기, 신호기 등을 임의로 편집하여 이들 상호간을 링크시킴으로서 실제 현장판별의 제작 없이도 현장신호설비를 시뮬레이션 할 수 있도록 하였다. 현장 신호설비들은 소프트웨어화합으로서 시뮬레이터 개발비용을 줄이 수 있고, 또한 선로 프로파일이 변경되어도 쉽게 시뮬레이션 할 수 있는 장점이 있다. 즉, 어떠한 선로프로파일에 대해서도 쉽게 에디팅 하여 시뮬레이션 할 수 있도록 함으로서 고속철도 이외에도 다양하게 활용될 수 있을 것이다. 시뮬레이터에서 열차도 열차성능시뮬레이션 프로그램(TPS)을 바탕으로 한 소프트웨어로 처리하였다.

시뮬레이터의 기본적인 전체구성은 그림1과 같다. TrackBuilder를 이용하여 시뮬레이션 할 운행노선을 편집하고, 구배, 커브 등의 선로 데이터 DB와 링크 시키게 된다. 현장 시뮬레이터에서는 TrackBuilder에서 편집한 정보 바탕으로 시뮬레이션 할 수 있도록 열차 정보를 입력한다. 그리고 진로명령을 내리면 이 정보가 전자연동장치 시제품으로 전송되고, 처리 결과가 다시 이 시뮬레이션 프로그램 전송되어진다. 또한 다중 열차 시뮬레이션도 가능하도록 ATC 시제품과 통신하도록 되어 있다.

즉, 전자연동장치와 ATC 시제품의 처리결과들이 Ethernet을 통해 시뮬레이터로 전송되어 그 처리결과를 확인할 수 있다. 이를 이용함으로써 신호제어장치 주요장치 시제품들의 기능 검증이 가능하고 또한 각 장치간 인터페이스도 확인할 수 있을 것으로 예상된다.

현재 TrackBuilder은 거의 완성되었으며, 시뮬레이션 프로그램은 TPS 기능과 통신기능이 마무리되었다. 다음 장에서는 이들 주요 소프트웨어 모듈들을 설명한다.

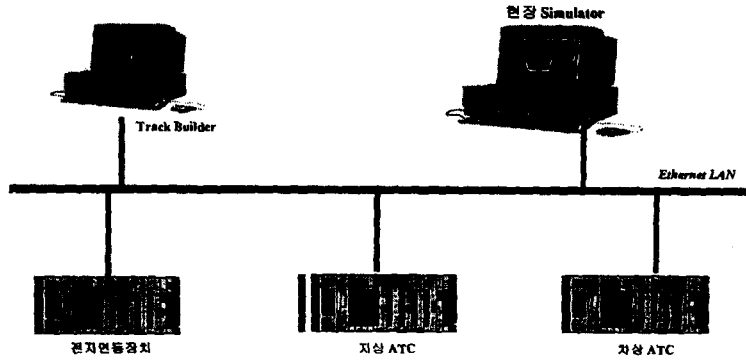


그림1. 시뮬레이터 구성

3. 소프트웨어 구성

3.1 궤도회로 편집 프로그램

이 프로그램은 시뮬레이션 전 단계에서 실행되어야 하는데, 먼저 시뮬레이션을 하기 위한 선로의 구간 및 신호기의 위치, 선로구간 길이 등의 속성들이 결정되어야 한다. 이렇게 결정되어진 정보를 가지고 선로를 편집하게 되는데 편집된 선로는 속성이 부여됨으로서 비로소 궤도회로의 특성을 가지게 된다.

선로의 각 회로는 여러 가지 다른 모양으로 구성되어지는데 본 프로그램에서는 그림2에서 처럼 14 가지로 구성하였다. 그림의 좌측 Pane에 나타난 것처럼 각각의 다른 종류의 회로들을 택해서 사용자가 자유로이 선로를 구성할 수 있도록 하였다.

이때 임의의 회로들이 앞뒤의 궤도회로들과 각각 연결되어야만 시뮬레이션이 가능하다. 즉, 완전한 회로의 연결 하에서만 시뮬레이션이 가능하기 때문에 이 기능은 매우 중요하다고 할 수 있다.

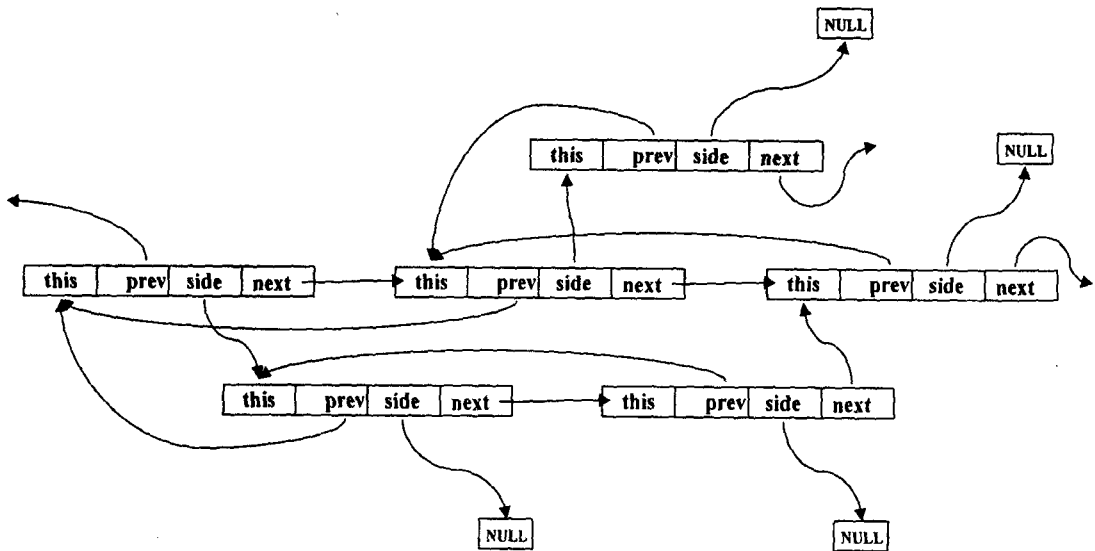


그림2. 궤도회로 연결 원리

회로들의 링크는 그림2 처럼 연결되어진다. 각 회로의 형식에 따라 달라지지만 기본적으로 오른쪽에 위치한 회로는 'next' 포인터로 연결되고, 왼쪽은 'prev' 포인터로 연결되고 측면에 위치하게 되면 'side' 포인터에 연결되게 된다. 이렇게 완벽하게 링크가 마무리되고 난 후, 그림 1과 같은 윈도우에서 마우스로 궤도회로를 그룹핑 하면 선택되어진 몇 개의 회로들이 하나의 그룹으로 묶이어 하나의 궤도회로들을 이루게 된다. 이렇게 되면 새로운 포인터를 생성하도록 하였다.

한 궤도회로에 소속된 회로의 포인터는 각 궤도회로의 맨 앞에 위치하는 회로가 Vertex_rail[0], 맨 뒤에 위치하는 회로가 Vertex_rail[1] 그리고 측면 회로가 존재하는 경우 Vertex_rail[2]에 저장된다. 이렇게 함으로서 Vertex_rail[] 의 배열에 들어 있는 포인터에 의해 시뮬레이션이 될 수 있도록 하였다. 시뮬레이션시 마커는 Vertex_rail[]의 배열에 들어 있는 포인터와 상호 연결되어져 있기 때문에 열차의 진입이 가능한지, 다음 궤도회로로 진입이 가능한지 등을 확인할 수 있다. 시뮬레이션 시 각 궤도회로의 내부에서는 회로의 포인터로 열차가 진행하지만 다음 궤도회로로 넘어갈 때는 Vertex 자료구조에 의해 정해진 다음 궤도회로로 진행하게 된다. 이렇게 해서 다음 궤도회로로 연결되어지면 또 다시 회로의 포인터로 연결되는 과정을 반복 수행한다. 즉 이러한 알고리즘에 의해 시뮬레이션 시 열차의 진행이 이루어지도록 하였다.

궤도회로가 연결되는 구조를 살펴보면 다음 그림 왼쪽의 경우처럼 각각 별개의 궤도회로로 구성되어 있다가 하나의 선로 프로파일로 링크 시키게 되면 오른쪽 그림처럼 각각 궤도회로의 맨 앞과 맨 뒤, 그리고 측면 회로의 포인터와 연결되는 점들은 같은 번호를 가지게 된다. 이렇게 하여 임의로 편집된 선로 프로파일 열차가 달릴 수 있는 실제 선로가 되게 된다.

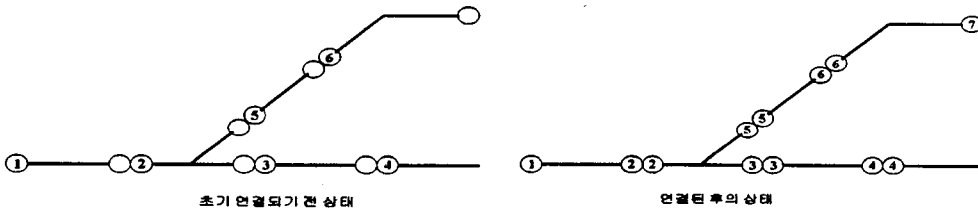


그림3. Vertex 자료구조에 의한 궤도회로 상호간 연결

또한 TrackBuilder에서는 시뮬레이션시 열차성능 시뮬레이션 등을 위해 필요한 선로 구배, 곡선, 속도 등의 데이터를 별도로 Windows NT의 ODBC를 이용하여 MS_ACCESS로 연결시켰다.

선로변의 신호기, 전철기 등을 포함한 궤도회로를 편집할 수 있는 TrackBuilder는 주요 클래스들은 궤도회로 및 폐색구간 등을 에디팅하기 위한 클래스들과 선로 고속선, 구배 등의 정보를 담은 데이터베이스 연동 클래스들 그리고 폐색구간, 신호기 등의 속성을 정의 및 변경할 수 있도록 하는 대화클래스로 구성하였다.

그림4는 이 TrackBuilder를 이용한 선로변의 궤도회로를 생성하는 예를 보인 것이다. 그림에서처럼 여러 종류의 선로 회로 컴포넌트나 상단 Pane의 매크로를 이용해 시뮬레이션 할 임의의 선로 프로파일을 에디팅 한다. 그리고 이들을 블록화 시켜 궤도회로(Track Circuit) 단위로 만들고 이들을 앞에서 설명한 알고리즘에 의해 상호 링크 시켜 실제 열차가 운행할 수 있는 선로 프로파일이 되도록 하였다.

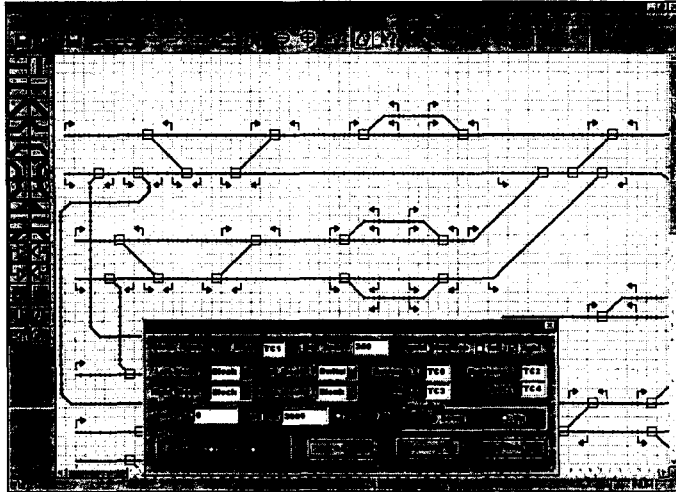


그림4. TrackBuilder를 이용한 궤도회로 생성

3.2 신호 시뮬레이션 프로그램

앞에서 설명한 TrackBuilder를 이용하여 시뮬레이션 하고자 하는 선로 프로파일을 생성하여 시뮬레이션 프로그램에서 각 신호제어장치 시제품들과 통신을 하면서 시뮬레이션을 수행하게 된다. 이러한 시뮬레이션 절차가 그림5와 같다.

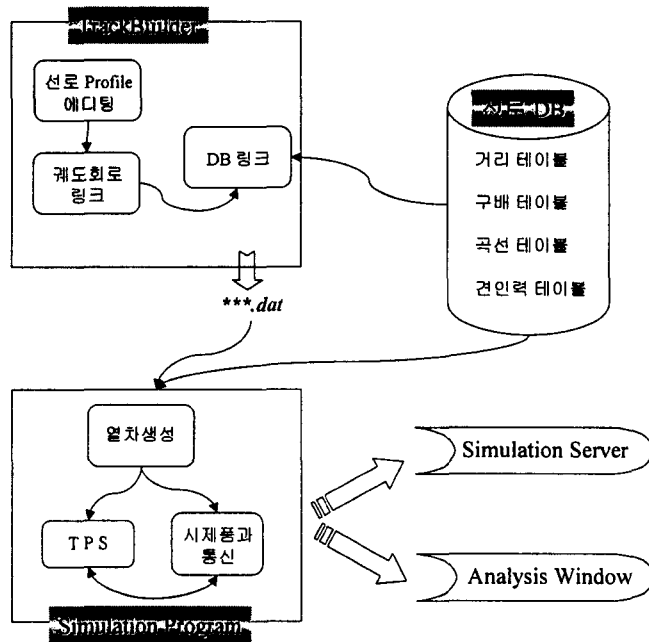


그림5. 시뮬레이션 흐름도

TrackBuilder를 이용하여 궤도회로를 생성시키고, 이 선로 프로파일에 맞는 선로 DB와 링크시

켜 실제 열차가 운행될 수 있도록 데이터 파일을 생성한다. 이 데이터 파일을 시뮬레이션 프로그램에서 사용하게 되는데 이 소프트웨어 모듈에서는 궤도회로의 편집을 할 수 없도록 되어 있다. 이 모듈에서 시뮬레이션 할 열차를 생성하게 되는데 열차 중량, 길이, 열차 형식, 열차 위치 등 시뮬레이션을 위한 열차의 정보들을 입력하게 되며, 여러 열차의 생성이 가능하도록 하였다.

이렇게 선로 프로파일과 열차의 생성이 마무리되면 진로명령을 통해 TPS 모듈에 의해 열차의 주행 시뮬레이션을 수행하게 되며, 동시에 전자연동장치와 자동열차제어장치 시제품들과 TCP/IP 소켓 프로그램을 통해 통신을 하게된다.

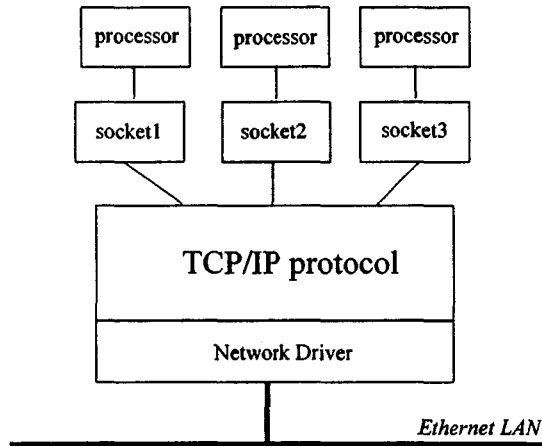


그림6. Socket Program의 기본 원리

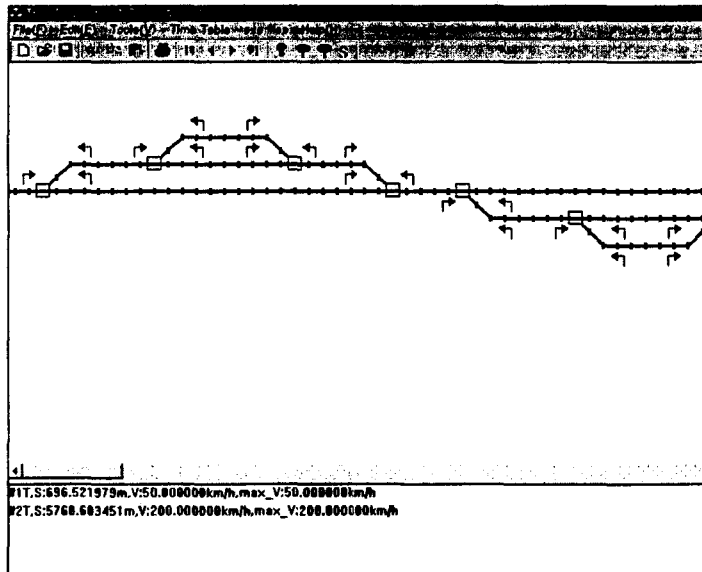


그림7. 열차 시뮬레이션 윈도우

소켓프로그램의 원리는 그림6과 같은데 소켓은 실행되고 있는 응용프로그램과 연결하기 위한

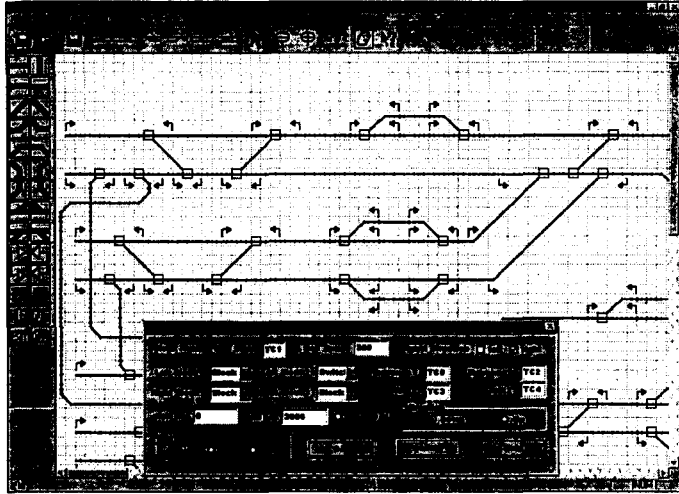


그림4. TrackBuilder를 이용한 궤도회로 생성

3.2 신호 시뮬레이션 프로그램

앞에서 설명한 TrackBuilder를 이용하여 시뮬레이션 하고자 하는 선로 프로파일을 생성하여 시뮬레이션 프로그램에서 각 신호제어장치 시제품들과 통신을 하면서 시뮬레이션을 수행하게 된다. 이러한 시뮬레이션 절차가 그림5와 같다.

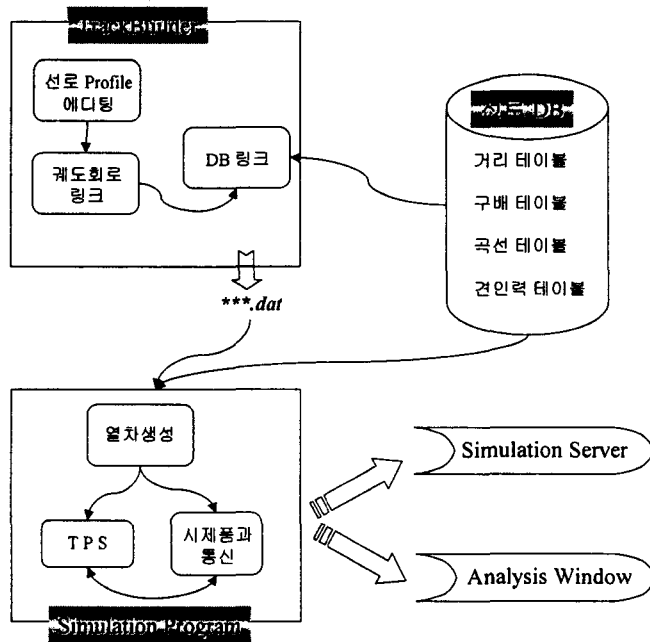


그림5. 시뮬레이션 흐름도

TrackBuilder를 이용하여 궤도회로를 생성시키고, 이 선로 프로파일에 맞는 선로 DB와 링크시

켜 실제 열차가 운행될 수 있도록 데이터 파일을 생성한다. 이 데이터 파일을 시뮬레이션 프로그램에서 사용하게 되는데 이 소프트웨어 모듈에서는 궤도회로의 편집을 할 수 없도록 되어 있다. 이 모듈에서 시뮬레이션 할 열차를 생성하게 되는데 열차 중량, 길이, 열차 형식, 열차 위치 등 시뮬레이션을 위한 열차의 정보들을 입력하게 되며, 여러 열차의 생성이 가능하도록 하였다.

이렇게 선로 프로파일과 열차의 생성이 마무리되면 진로명령을 통해 TPS 모듈에 의해 열차의 주행 시뮬레이션을 수행하게 되며, 동시에 전자연동장치와 자동열차제어장치 시제품들과 TCP/IP 소켓 프로그램을 통해 통신을 하게된다.

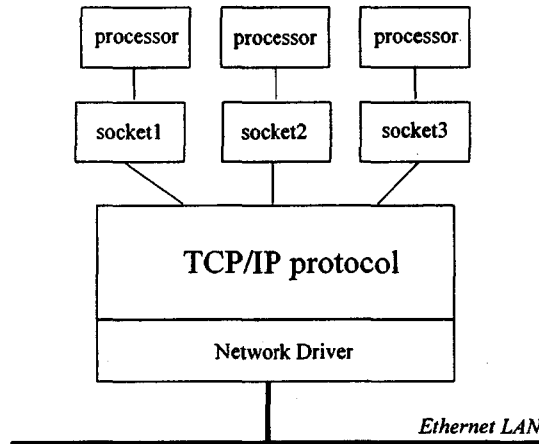


그림6. Socket Program의 기본 원리

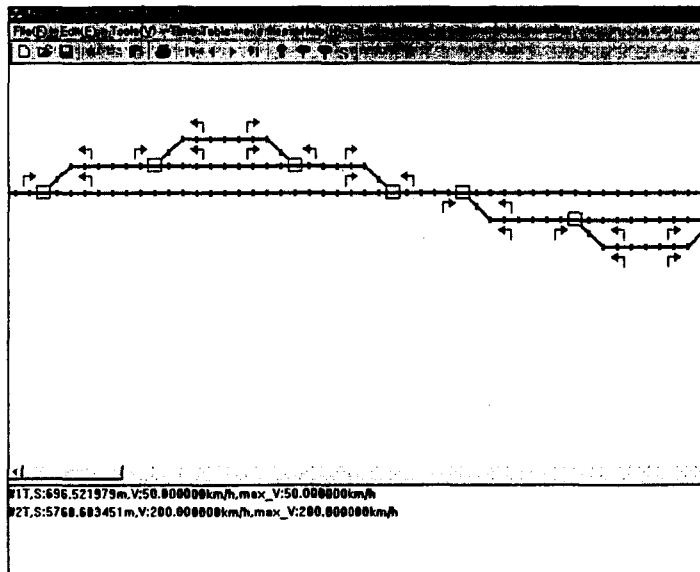


그림7. 열차 시뮬레이션 윈도우

소켓프로그램의 원리는 그림6과 같은데 소켓은 실행되고 있는 응용프로그램과 연결하기 위한

포트를 생성하고 논리적으로 가지고 있는 IP주소와 MAC계층의 네트워크기판의 물리적 주소를 연결시켜줘야만 데이터통신이 가능하게 된다. 이 과정을 'bind()'라고 한다. TCP/IP 데이터통신은 접속 지향형과 비접속 지향형으로 나눌 수 있는데 대표적인 접속 지향형 프로토콜은 TCP 프로토콜이고 비접속 지향형은 UDP 프로토콜이다. 본 프로그램에서는 TCP프로토콜을 이용하여 통신을 하도록 설정하였다. 위에 설정된 기본과정을 통해 데이터송수신이 이루어지고 종료되게 된다.

실제 송·수신되는 데이터 형태는 구조체로 하여 캐드회로 점유상태, 마커 정보, 포인터 정보, 속도 정보 등을 전송하도록 구성하였다.

4. 결론

본 논문에서는 신호제어시스템 시제품의 기능 검증 및 시스템간 인터페이스 확인을 위한 시뮬레이터의 기본 방향과 이 시뮬레이터를 구성하고 있는 주요 소프트웨어 모듈에 대해서 설명하였다. 이 중 TrackBuikder는 현장의 캐드회로를 생성하고 또 이들 사이를 링크시켜 실제 열차가 주행할 수 있도록 해 주는 모듈로서 매우 중요한 부분이다. 현재 이 모듈의 개발은 거의 마무리 단계에 있으며, 이 모듈로부터 발생하는 선로 데이터를 바탕으로 실제로 시제품과 통신을 해가면서 시뮬레이션을 수행하는 모듈은 지속적으로 개발 중에 있다.

이 소프트웨어 모듈이 완성되면 신호제어장치 시제품과 Ethernet LAN으로 연결하여 시제품의 주요 기능을 확인하고, 또한 각 장치간의 인터페이스 정보도 도출, 확인할 예정이다.

<참고문헌>

- [1] 稱毛弘苗, et al., '次世代運轉制システムのシ室内實驗', RTRI Report Vol 5, No. 1, pp. 48-55, 1991.
- [2] 1·2차년도 연구보고서, '열차운용 시뮬레이션 프로그램 개발에 관한 연구', 한국철도기술연구원, 1996·1997.
- [3] G7사업 1단계 연구보고서, '전기신호시스템 엔지니어링 기술개발', 한국철도기술연구원, 1999.
- [4] 이종우, 외, '차세대 고속철도 신호제어시스템의 실험 기본설계', 한국철도학회 춘계학술회의, pp. 315-321, 1999.