

## 블루투스와 임베디드 시스템을 이용한 자동차 감시 및 제어시스템 구현

양승현\*, 김춘식\*\*, 이관호\*\*, 이석원\*  
 \* 호서대학교 정보제어공학과, \*\* (주)태성전장

### An Implementation of Vehicle's Monitoring and Control system using BlueTooth and Embedded system

Seung-Hyun Yang\*, Choon-Sik Kim\*\*, Kwan-Ho Lee\*\*, Suk-Won Lee\*

\* Department of Information & Control Eng. Hoseo Univ. \*\*TaeSung Electro-Circuit System Co.Ltd..

**Abstract** - 지능형 자동차에 대한 관심이 높아지면서 AICC(Automatic Intelligent Cruise Control)나 무인 자동차가 개발되고 있다. 이러한 기술을 개발하기 위해서는 자동차의 환경과 운항에 관련된 상황을 실시간 감시하는 것이 중요하기 때문에 본 연구에서는 임베디드 시스템을 구성 후 웹 서버를 구현하여 원격지에 존재하는 자동차의 감시 시스템을 개발하고, 자동차 ECU와 인터페이스 하여 엔진 및 자동차의 내부 상황을 웹으로 확인할 있는 임베디드 시스템을 구현하였다. 또한, ECU의 부담과 자동차 Wireless Harness(W/H)에 의한 중량을 줄이기 위해 블루투스 방식으로 분산제어 방식을 이용 각 부분별 제어할 수 있도록 하였다.



〈그림 1〉 분산제어용 블루투스 보드

〈그림 1〉은 자동차의 분산제어용으로 개발한 블루투스와 임베디드 접점 A/D 변환기가 내장된 마이크로 컨트롤러 보드이다. 자동차의 상태를 제어하거나 데이터를 추출할 때 사용하도록 하였다. 〈그림 2〉는 블루투스 모듈

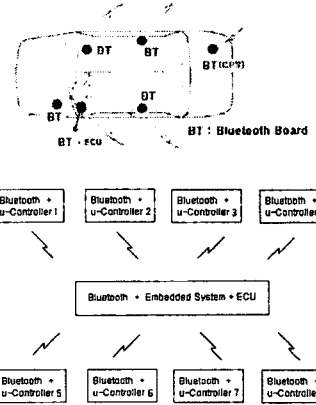
## 1. 서 론

TCP/IP의 발전과 더불어 인터넷의 다양한 기술은 여러 분야에 걸쳐 이용되고 있으며, 이와 더불어 특정한 기능을 독립적으로 수행하던 가전제품 및 산업용 기기를 인터넷과 결합하여 그 기능을 확장하고 있다. 기존의 장비들은 웹 서비스를 하기 위해서는 실시간 운영체제(Real Time O/S (RTOS))를 이용하거나 직접적인 TCP/IP를 작성하여 사용한다. 하지만 이러한 방법들은 경제성과 신뢰성이 다소 떨어지는 단점을 가지고 있기 때문에 최근에는 운영체제가 내장된 형태의 장비들이 이용되고 있다. 이러한 임베디드 시스템을 이용해 웹 서버를 구축하고, 기기를 연결하면 특정한 PC나 서버가 없이 제어가 가능하다. 이렇게 서버 기능이 갖추어지면 운행 중인 자동차를 데이터를 웹으로 파악할 수 있으며, PDA와 같은 모바일 기기나 자동차를 제어할 수 있다. 본 연구에서는 자동차의 ECU에 임베디드 시스템을 인터페이스 시켜 자동차의 주요 데이터를 인터넷으로 실시간 감시하도록 하였다. 또한, 자동차 산업의 비약적인 발전으로 인해 운전의 편리성이나 안전성, 자동차 내부의 자동화 등의 고급화, 고기능화에 대한 연구개발이 이루어지고 있다. 이러한 자동차의 사양들이 다양해지고 고기능화 되어지기 때문에 자동차의 집적적인 통합제어 방식보다는 분산제어 방식에 대한 많은 연구들이 진행 중이다. 차량의 전기전자 부품을 전기적으로 연결하는 Wire Harness는 차량에 대한 중요한 신호 전달 매개체로 신경과 같은 역할을 한다. 하지만 제어대상의 증가와 향상에 따라 전선의 양이 많아져 부피가 커지고 무게 또한 중량이기 때문에 차량에 효율적 배치나 구성이 어려워진다. 이러한 문제점을 해결할 수 있는 차량배선 방법 중 국제적으로 표준화가 진행되고 있는 Controller Area Network (CAN)은 대표적인 표준화 프로토콜이다. CAN 통신은 입력력과 출력 선으로만 구성되어 장치간의 연결선을 최소화 할 수 있으며, 데이터 연계중의 에러 검사와 수정 기능으로 데이터 신뢰도가 높아 자동차를 비롯한 항공기, 철도, 로봇을 비롯한 자동화 시스템에 이용되고 있다. 또한 기존의 Wiring Harness(W/H)에 의한 제어 구성의 낮은 신뢰성과 비효율적인 시스템 구성을 개선 할 수 있어 자동차의 경우 현재 집적 적이고 통합적인 제어방식을 대체하며 새로운 전자제어 장치로 사용되어 지기도 한다. 하지만, 미래의 자동차는 내부의 제어대상이 많아지기 때문에 CAN 통신 보다는 무선통신이 효율성이 크게 증대 될 것이다. 따라서 본 연구에서는 자동차의 분산제어 방식으로 블루투스통신을 이용하고 이러한 통신내용을 임베디드 시스템과 연결시켜 자동차의 전반적인 감시가 가능하도록 한다.

## 2. 본 론

### 2.1 블루투스

블루투스(Bluetooth) 통신은 최대 데이터 전송속도가 1Mbps이고, 10m의 근거리 무선 데이터통신이지만, 자동차에 내부 데이터 송수신이 충분하며 기기간 상호 연결을 통한 PAN(Personal Area Network)의 구성이 가능하고 전송속도가 2.4Ghz로 다른 기기들과 같이 사용하는데 간섭이 없다. 또한 사용하기에 편리하고 신뢰성도 다른 RF기기를 보다 뛰어나다.

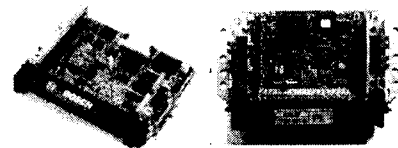


〈그림 2〉 분산되어진 부분별 제어기

을 자동차의 각 부분에 배치시킨 모습이다. BT는 블루투스 모듈이 내장된 마이크로 컨트롤러 보드로 제어 대상인 부분에 센서 취득과 입출력인 On/Off 제어를 하게 된다. BT(GPS)는 자동차의 현재 위치를 위성을 통해 확인하기 위해 연결한 것이고, BT+ECU는 자동차의 ECU에 많은 정보가 있기 때문에 ECU의 데이터를 블루투스를 통해 임베디드 시스템으로 보내기 위한 부분이다. 블루투스는 각각의 보드에 ID를 부여하게 되면 서로의 데이터를 혼란 없이 처리할 수 있기 때문에 유용하게 무선으로 데이터를 송수신 할 수 있다.

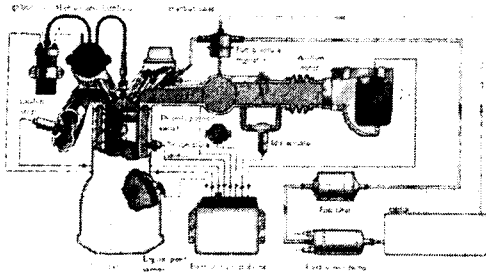
### 2.2 ECU (Electric Control Unit)

자동차에 존재하는 전자제어장치인 ECU에는 자동차에서 사용되고 있는 센서신호들과 스톱/스타트 등 자동차의 전반적인 내용들을 살펴보고 상황에 맞는 신호를 전자적으로 출력해줘 자동차 운행에 많은 영향을 준다. 이러한 ECU의 단자를 이용하면 자동차의 상태를 파악하기 용이하기 때문에 본 연구에서는 ECU의 신호를 블루투스 보드에 연결해 ECU의 정보를 임베디드 시스템에 전송하게 하였다.



〈그림 3〉 BOSCH 사의 ECU

〈그림 3〉은 보쉬사에서 제작한 자동차용 ECU이며, 우측의 그림은 본 연구에서 사용되어진 현대자동차의 아반떼용 ECU 이다.



〈그림 4〉 ECU 구성도

〈그림 4〉에서 볼 수 있듯이 ECU에는 자동차의 전반적인 상황들이 전기적 신호로 입력되고, 또한 전기적 신호로 출력되어 자동차의 많은 부분을 제어하게 된다. 그렇기 때문에 ECU의 신호를 잘 파악하게 되면 자동차의 전체적인 상황을 모니터링 할 수 있게 된다. 현재 자동차의 계속 장비 또한 ECU에 연결해 자동차의 데이터로 이상 유무를 판별하고 있다.

### 2.3 임베디드 시스템 (Embedded System)

자동차의 ECU와 측정대상의 다양한 센서들을 비롯한 분산 연결된 블루투스 제어기들에서 출력되는 신호들을 입력받아 처리하고 웹(Web)으로 전송하기 위해서는 임베디드 시스템 보드에 무료 배포되는 임베디드 리눅스를 설치하고, 웹 서버(Server)를 포팅 해야 한다.

#### 2.3.1 웹서버구축을 위한 프로그램

웹기반 프로그램이란 클라이언트의 웹 브라우저를 이용해 인터넷에 연결되어 있는 원격지의 서버로 기기를 관리한다는 의미를 가지고 있다. 이러한 웹서버를 구현하고 클라이언트로 서버를 제어하기 위해서는 세 가지 요소를 갖추어야 한다.

- 클라이언트 풀(Client Pull)과 CGI

웹 페이지의 화면을 자동으로 반복표시(refresh) 해주는 방법에는 크게 클라이언트 풀(Client Pull) 방식과 서버 푸쉬(Server Push) 방식이 있다. 원격지에 있는 자동차의 상황을 실시간적으로 모니터링하기 위해 클라이언트 풀 방식을 사용하여 한번 CGI로 구성된 페이지를 열게 되면 반복적으로 페이지가 변경되게 하는 방법이 유용하다. 클라이언트의 요청이 있을 때마다 서버에게 새로운 CGI 프로세스의 생성과 종료로 유발시켜 주기 때문에 서버의 동작에 부하를 주지 않는다.

- 자바애플릿

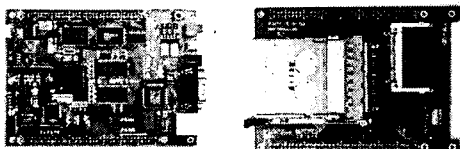
Sun Microsystems에서 개발된 자바언어(Java Language)는 객체지향 언어로 내장형 시스템을 위해 플랫폼에 구애받지 않는 언어로 처음 개발되었다. 자바가 플랫폼에 구애받지 않기 위해서는 동작할 플랫폼에 자바 가상머신(JVM, Java Virtual Machine)이 내장되어 있어야 한다. 하지만 임베디드 시스템에 자바가상머신을 올리려면 많은 리소스를 필요로 하기 때문에 자바 애플릿(Java Applet)을 이용한다. 이는 클라이언트 상에만 자바가상머신만 있으면 된다.

- GoAhead 웹 서버

본 연구의 프로그램 구현시 사용한 웹 서버는 오픈소스인 GoAhead 웹 서버를 사용하였다. 웹 서버는 PXA 255기반의 리눅스 플랫폼상에서 실행될 수 있도록 교차 컴파일하여 사용하였다. 웹 서버를 포팅해야만 임베디드 보드가 하나의 서버가 되기 때문에 원격지에서 클라이언트를 통해 접속할 수 있다.

#### 2.3.2 웹서버 구축을 위한 하드웨어 구성

웹서버용 보드는 리눅스와 임베디드 웹서버를 포팅한 것으로 〈그림 5〉에 보이고 있다.



〈그림 5〉 임베디드 시스템과 PCMCIA 형태의 무선 랜 보드

〈그림 5〉의 좌측 보드는 PXA255 RISC ARM 400[Mhz]인 CPU를 사용하고 있으며, 64[Mbyte] Nand-Flash, 32[Mbyte]의 SDRAM과 부트(Boot) Falsh로 512[Kbyte]의 메모리를 가지고, 인터넷 컨트롤러로 CS8900 Chip을 사용하고 있다. 또한 우측의 PCMCIA 카드를 연결할 수 있도록 확장보드를 설계하여 무선 랜카드 연결이 가능하도록 하였다. 〈그림 7〉은 임베디드 보드에 GoAhead 웹서버를 포팅하고, 외부의 클라이언트로 서버에 접속한 그림이다.

### 2.3.3 자동차 속도 제어

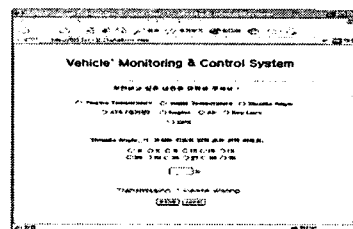
자동차의 데이터는 ECU나 부분별로 설치된 센서에 의해 데이터를 추출할 수 있다. 하지만 자동차를 제어하기 위한 출력 부분은 자동차의 동작 전압이 12V이상이기 때문에 포토커플러나 전압변환기를 사용하여 제어하였다.



〈그림 6〉 전동 실린더를 이용한 속도제어

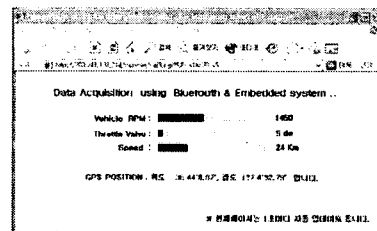
〈그림 6〉은 블루투스를 내장한 제어기 보드가 수직운동을 하는 전동실린더를 제어하는 그림이다. 브레이크와 가속페달을 제어하게 되는데 전동실린더의 사용은 초당 2mm이다. 이 또한, 임베디드 시스템이 블루투스 통신을 이용해 제어하게 된다.

## 3. 실험



〈그림 7〉 전동 실린더를 이용한 속도제어

〈그림 8〉은 인터넷으로 자동차에 설치한 임베디드 웹서버에 접속후 로그인 한 화면이다. 감시하고 싶은 대상과 자동차의 위치 및 자동차의 가속페달에 압력을 가했을 경우 변하는 스로틀밸브의 각도를 설정할 수 있게 하였다. 또한 구체적인 제어대상은 다른 창을 띄워 제어할 수 있다.



〈그림 8〉 웹을 이용한 자동차 데이터 취득화면

인터넷 창에 자동차의 IP를 입력하게 되면 〈그림 7〉과 같이 제어대상과 측정대상을 선택할 수 있는 화면이 나온다. 이때 선택하고 요청을 실행하면 〈그림 8〉과 같이 결과 화면이 나오게 된다.

## 4. 결 론

본 연구에서는 블루투스와 임베디드 시스템을 이용해 웹으로 운행 중인 자동차의 상태를 감시할 수 있는 시스템을 구현 하였다. 블루투스를 사용하여 자동차의 제어와 측정 대상들을 분산시켜 자동차의 ECU 부하를 줄이도록 해보았다. 블루투스를 사용함으로 자동차에 사용되는 Wiring Harness의 양을 줄여 자동차의 중량을 감소시킬 뿐만 아니라, 지능화되고 편리성이 증대되어 전자제어부분이 많이 적용됨에 따라 많은 도움을 줄 것이다. 또한 ECU를 비롯한 분산된 제어 대상들을 웹으로 확인할 수 있도록 하여 자동차의 상태나 위치를 장소에 구애받지 않고 컴퓨터나 모바일 기기를 이용해 확인할 수 있도록 하였다.

### 〔참 고 문 헌〕

- [1] C. D. Leidigh, "Web Based Management od Network Device", ESCC 2001, no 204, Chicago, 2001
- [2] L.Q.kong, J.Malee and T.Korte, "A Simple Architecture for Real-Time Web Based Device Control and Monitoring", ESCC 2001, no. 230, Chicago, 2001.
- [3] C. D. Leidigh, "Web Based Management od Network Device", ESCC 2001, no 204, Chicago, 2001