

# ZigBee와 CAN 통신을 이용한 자동차 배기가스 검출 및 자기진단 시스템

## Car Exhaust Gas Detection and Self-Diagnosis System using ZigBee and CAN Communications

천 종 훈\*  
(Jong-Hun Chun)

김 국 세\*\*  
(Kuk-Se Kim)

박 종 안\*\*\*  
(Jong-An Park)

### 요 약

본 논문은 대기오염의 주범인 자동차 배기 가스량을 체크하고 차량내의 고장 유무 진단 시스템 개발이다. 시스템 엔진 정보 추출을 위해 차량 CAN 통신을 이용하고 정보 전송을 위해 ZigBee를 통해 데이터 전송을 한다. 차량 CAN을 위해 차량에서 자체 제공되는 OBD-II 프로토콜을 사용하여 차량의 각종 센서 정보 및 O2 센서 값을 통해 차량 상태 정보 및 배기 가스량을 계산한다. 주행 중인 자동차 엔진 및 내부 고장에 잘 알지 못하는 일반 사용자를 위해 운행 중 실시간 차량의 자기진단 시스템 구축을 목적으로 하고 고장진단 프로토콜 전송을 위한 무선통신 인터페이스로 저 전력 저비용 ZigBee 통신 인터페이스를 구축한다. 자동차 그리고 진단 시스템의 통신을 위해 ZigBee 시스템을 통하여 효율적 저비용 통신 인터페이스를 구성하여 차량내의 엔진 및 각종 센서 정보 네트워크를 지원한다. 차량에서 전송되어 온 각종 센서 정보는 ZigBee 기반을 통해 ZigBee 메인 컨트롤 시스템에 전송된다. 차량에 이상이 생겼을 때 트러블 코드를 저장하고 자동차가 정비소에 갔을 때 정확한 판단을 하여 신속하게 처리 할 수 있게 해 주며 자동차에 대해서 잘 알지 못하는 운전자에게 정확한 정보를 제공한다. 또한 멀티미디어 시스템 기능을 추가하고 주행 중 무선 인터넷이 가능하도록 시스템을 확장한다. 마지막으로 주행 중 차량 자기진단을 위해 저 전력 임베디드 리눅스 시스템을 구축하고 실 실험을 통하여 구현하고 검증한다.

### Abstract

This study provides to car driver with car exhaust gas and sensor information which are car trouble code in engine and many sensors when the car has some problems. This is to provide car manager with many information of car sensors when we go to vehicle maintenance. For example, information of engine RPM, fuel system, intake air temperature, air flow sensors and oxygen sensors can provide to owner or garage, and also add to multimedia system for mp3 files and video files. This system consists of embedded linux system of low power while driving the car which uses OBD-II protocols and zigbee communication interface from CAN communication of car system to self-diagnosis embedded system of car. Finally, low power embedded system has a lot of application and OBD-II protocols for embedded linux system and CAN communication which get sensor informations of car control sensor system while driving the car.

\* 주저자 : 전남도립대학 실버케어과 교수

\*\* 공저자 : 조선대학교 컴퓨터공학과 박사과정 수료

\*\*\* 공저자 : 조선대학교 전자정보공과대학 교수

† 논문접수일 : 2008년 11월 12일

† 논문심사일 : 2008년 12월 8일

† 게재확정일 : 2008년 12월 10일

**Key words:** Car, exhaust gas, self-diagnosis, ZigBee, embedded, embedded linux

## I. 서 론

최근 사람들은 통신에 대한 범위를 단지 컴퓨터 또는 휴대폰으로만 생각하지 않고 모든 사물 간, 사람 간 그리고 사람과 사물간의 커뮤니케이션으로 확대 해석 하고 있는 추세이다 [1]. 또한 유비쿼터스 시대에 접어들어 광대역 통신의 급속한 발전으로 유무선 인터넷 환경에 적용되는 다양한 응용 분야가 급속히 확산되고 있는 추세이다 [2]. 특히 PDA, 모바일 단말기 등은 소형의 내장형 기기로 이동 중에도 데이터의 접근을 가능케 함으로 최근 인기를 끌고 있다. 이와 같은 내장형 시스템은 미리 정해진 특정 기능을 수행하기 위해 컴퓨터의 하드웨어와 소프트웨어가 조합된 전자 제어 시스템을 말하며, 필요에 따라서는 일부 기계가 포함될 수도 있다 [3].

컴퓨터와 통신을 하고 주행 중인 차량내의 시스템과 통신을 효과적으로 연계해 언제, 어디서나 각종 데이터의 검색이나 수정 등을 가능토록 하는 관리가 필요하며, 제한적인 플랫폼에서 원활한 동작을 위해 장비를 적절히 제어할 수 있는 소규모의 운영체제와 응용 프로그램을 구현할 수 있는 파일시스템, 또한 이동 중인 차량에 맞게 시스템이 구축되어야 한다 [4, 5]. Wibro시대를 맞아 현대인의 이동이 많고 방대해지면서 가정 보다 외부나 자동차에서 생활하는 시간이 늘어나면서 누구나 언제, 어디서나 인터넷이 가능하고 GPS, MP3 그리고 멀티미디어 동영상이 가능한 시스템을 원하고 이동 중 차량에 문제가 생겼을 경우 적절한 대응책을 찾고 진단 수리하고 싶어 한다.

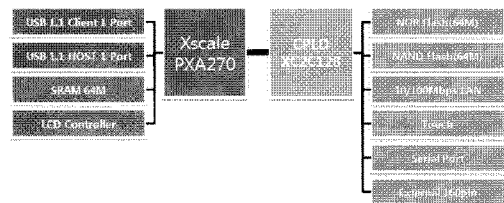
본 논문에서 자동차의 안정성 확보와 Car Control System은 주행 중인 자동차에서 실시간으로 자신의 차량에 대한 이상 유무 및 엔진 상태를 체크 하여 안전한 운행을 도와주는 자가진단 시스템 및 동영상과 MP3 등의 멀티미디어 콘텐츠를 즐길 수 있는 미디어 시스템 그리고 차량 센서

시스템으로부터 단말기 또는 정비소로 효율적 저비용 데이터를 전송하기 위해 802.15.4 프로토콜 ZigBee 통신 인터페이스를 구성한다.

## II. ZigBee기반 자동차 자가진단 임베디드 리눅스 시스템

임베디드 시스템이란 일반적으로 특별한 업무를 수행하기 위한 하드웨어와 소프트웨어를 포함하는 특정한 응용 시스템이라고 할 수 있으며, 이는 보다 구체적으로 마이크로프로세서 혹은 마이크로컨트롤러를 내장(Embedded)하여 원래 제작자가 지정한 기능만을 수행하는 장치를 말한다 [3].

본 논문의 자동차 시스템을 위한 자가 진단용 임베디드 리눅스 시스템 구성은 임베디드 리눅스 소프트웨어 시스템 부분과 저전력 ARM 11 CPU를 사용한 내장형 하드웨어 시스템으로 구성되어 있다. <그림 1>에 임베디드 시스템 하드웨어 구성 블록도를 나타낸다. Intel XScale PXA270 Processor를 탑재하고 Embedded Linux 2.6.xx 및 WinCE 5.0를 지원하고 저 전력, 소형으로 다양한 분야에 응용 가능하도록 설계되어 있으며 Intel wireless MMXTM 기술이 탑재된 시스템이다.

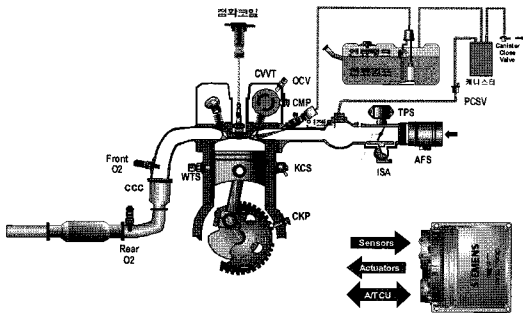


<그림 1> PXA270 기반 저전력 임베디드 시스템  
<Fig. 1> Low power embedded system base on PXA270

### 1. 자동차 자가진단 CAN OBD-II 통신 인터페이스

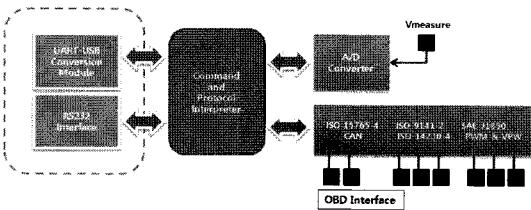
차량용 자가진단 CAN 통신 인터페이스를 위해 OBD-II 프로토콜을 사용한다. <그림 2>는 차량내의 센서 정보 중에 O2 센서를 포함한 입출력 시스템을 대표적으로 보여주고 있다. <그림 3>은 자동차내의 센서 컨트롤러와 CAN 통신을 할 수 있는 차량 센서 정보 전송장치 블록도를 보여주고 있다.

OBD-II 프로토콜은 12개 이상의 차량내의 CAN 통신 프로토콜들을 지원하여 각기 다른 차량들의 프로토콜들을 자동으로 세팅하며 AT 커맨드를 지원하여 사용자 중심의 환경을 구축하였다. 또한 차량 배터리 모니터링을 위하여 전압 입력과 저전력 CMOS 디자인을 구현하였다. 통신 인터페이스로는 RS232, USB를 지원한다.

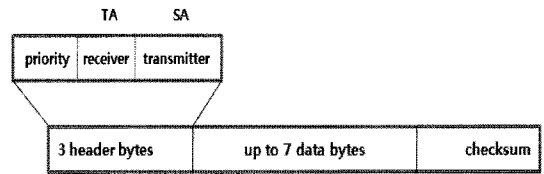


<그림 2> 자동차 O2센서를 포함한 입출력요소의 구성

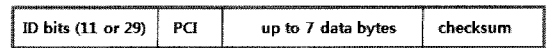
<Fig. 2> Architecture of input and output include O2 Sensor



<그림 3> CAN 통신 차량 센서 정보 전송장치 블록도  
<Fig. 3> Block diagram of car sensor information translations of CAN communication



<그림 4> An OBD message  
<Fig. 4> An OBD message



<그림 5> A CAN OBD Message  
<Fig. 5> A CAN OBD message

OBD-II 메시지 포맷은 각 3Byte의 헤더로 구성되어 Priority, TA Receiver, SA Transmitter와 7Byte의 데이터, 그리고 체크섬으로 이루어져 있으며, CAN OBD 메시지 포맷은 ID Bits (11 or 29), PCI, 7 Data Bytes와 체크섬으로 이루어져 있다. 일반적인 OBD Data Bytes들은 암호화 되어 전송된다. J1850, ISO 9141-2, 그리고 ISO 14230-4 프로토콜은 기본적으로 같은 구조를 가지며 <그림 4>, <그림 5>에 OBD 메시지와 CAN OBD 메시지 구조를 보여 준다.

현재 커맨드 그리고 프로토콜 인터프리터를 통해 차량의 각종 센서 정보를 OBD 인터페이스를 통해 CAN 통신을 한다. 수집된 센서 정보들은 자동차 매니저 인터페이스를 구성하여 차량내의 진단 내용을 사용자가 점검 할 수 있도록 GUI를 구성하였다. 효율적 저비용 통신 인터페이스를 위해 802.15.4 프로토콜 지원을 통한 ZigBee 통신 인터페이스를 구성한다.

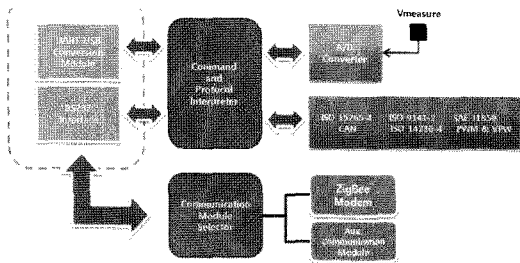
### 2. 자동차 센서 정보 전송을 위한 ZigBee 인터페이스 구성

차량내 센서들에서 획득한 차량의 센서 정보는 차량의 이동성을 고려할 때 근거리 무선통신을 이용하여 정보 저장, 관리, 이용을 하는 것이 효과적이다 [4, 5]. 근거리 무선통신은 블루투스, 무선랜, ZigBee가 있다 [6].

본 논문에서 사용된 주요 기술은 ZigBee 기술로 노드의 확장성, 저전력, 소용량의 데이터 처리에 적합한 기술이다. ZigBee의 가장 큰 특징은 저전력 소모로 ZigBee 칩은 동작 시 전력 소모가 20mA 내외로 저전력 기술인 (UWB의 전력소모가 200mW, 무선랜의 전력 소모가 1W) 사용하고 있고 ZigBee는 저전력을 위해 DSSS(Direct Sequence Spread Spectrum) 방식을 사용하며, ZigBee의 프로토콜 스택 크기는 32Kbyte를 넘지 않는다 [6].

ZigBee 스택은 여러 개의 계층으로 구성되는데 각 계층은 상위 및 하위 계층에 특정 서비스를 제공하며, 데이터를 주고받게 되고 IEEE 802.15.4 표준에서는 물리계층(PHY Layer)과 링크 계층(MAC Sub-layer)의 두 개의 계층에 대해 정의하고 있으며, ZigBee 얼라이언스에서는 네트워크 계층과 응용 지원 부 계층(Application Support Sub-layer), ZDO(ZigBee Device Object), 응용 객체를 포함하는 응용계층(Application Layer)에 대한 프레임워크에 대해 정의한다 [7].

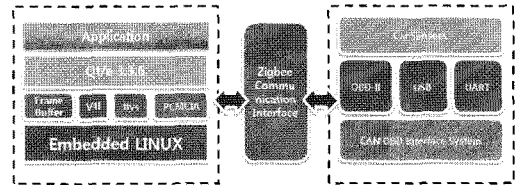
<그림 6>은 자동차 센서 정보 전송을 위한 ZigBee 통신 인터페이스 모듈을 보여주고 있다. 차량내의 각각의 센서 데이터들은 커맨드 프로토콜 인터프리터를 통해 통신 인터페이스 모듈을 통해 ZigBee 통신 모듈을 지나 암호화된 프로토콜을 통해 ZigBee 모듈로 전달되어 호스트로 전송된다.



<그림 6> 자동차 센서 정보 전송을 위한 Zigbee 통신 인터페이스 모듈  
<Fig. 6> Zigbee communication interface module for car sensor information translation

### 3. ZigBee 기반 자동차 임베디드 시스템 소프트웨어 플랫폼

자동차 임베디드 리눅스 시스템의 소프트웨어 플랫폼 스펙을 보면 다음 <그림 7>과 같다.



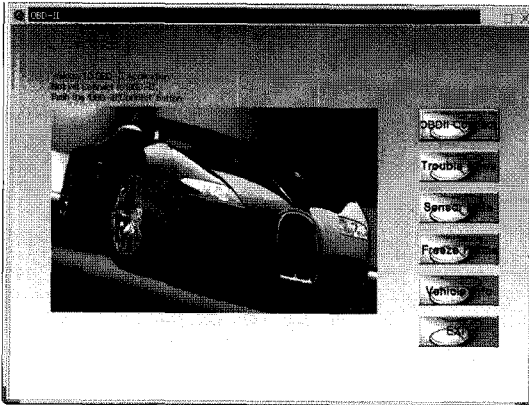
<그림 7> 임베디드 소프트웨어 플랫폼  
<Fig. 7> Embedded software platform

메인 시스템은 임베디드 리눅스 운영체제를 사용하였으며 Qt3.3.6 라이브러리를 이용하여 유저 GUI를 구성하여 운전자가 쉽게 차량의 상태를 파악 할 수 있게 하였다. <그림 8>은 차량내의 CAN OBD 프로토콜 구조를 보여주고 있다.

자동차 내의 통신은 CAN 통신을 하였으며 자동차 센서 정보를 단말 시스템으로 전송하기 위해서 802.15.4 프로토콜을 사용하였다. 그리고 멀티미디어 파일을 다운로드 하기위해 무선 통신 인터페이스를 확장 구성하였다. <그림 9>는 실제 Qt3.3.6 라이브러리를 사용하여 차량 내의 센서 정보를 위한 유저 GUI 인터페이스를 구성한 메인 화면이다.

RF base Protocol 7byte		DATA Protocol 11byte		RF base Protocol 5byte	
Byte	Field	Value	Descript		
1	Packet frame sync	0:7E	Always 0:7E		
1	Packet type	0:42	P_PACKET_NO_ACK (0:42) - User packet with no ACK required		
1	3, 4, 5, 6	0:01	Sensor ID		
1	7	0:02	Command Message Length		
1	8	0:01	Command		
1	9	0:02	Command		
1	10, 11	0:00, 0:00	Illumination Raw Data low, high		
1	12, 13	0:00, 0:00	Temp Raw Data low, high		
1	14, 15	0:00, 0:00	Humidity Raw Data low, high		
1	16, 17	0:00, 0:00	Battery Raw Data low, high		
1	18, 19	0:35, 0:35	Always 0:35		
1	20, 21, 22				
1	Sync	23 0:7E	Always 0:7E		

<그림 8> CAN OBD 프로토콜 구조  
<Fig. 8> CAN OBD protocol architecture



<그림 9> 유저 GUI 인터페이스  
<Fig. 9> User GUI interface

### III. 실험 및 구현

#### 1. 시스템 구성 및 사용자 인터페이스

본 논문에서는 자동차 자가 진단용 시스템을 위해 직접 임베디드 하드웨어를 제작 실험 하였다. 시스템 하드웨어 구성은 <표 1>과 같이 ARM11 Intel PXA270 CPU를 사용 구성하였다. 차량 자가 진단용 임베디드 시스템은 차세대 X-SCALE 기반으로 개발된 시스템으로서 시스템내에 무선 이터넷, UART, USB, LCD 컨트롤러 등이 내장되어 있으며, Embedded Linux, WinCE OS를 포팅하여 여러 가지 응용 및 시스템 호환에 적합하도록 디자인 하여 멀티미디어 미치 여러 가지 응용 프로그램에 기능을 구현 할 수 있도록 하였다. 또한 확장 80x2 포트를 구성하여 확장가능하도록 디자인하였다. <표 2>는 해당 시스템 하드웨어 스펙이다.

자동차 내의 CAN 통신을 위해 OBD-II 프로토콜 구성하여 RS232로 컨버트하여 자동차에서 사용되는 9가지 프로토콜을 제공하고 인터페이스 부는 차량에 사용되는 프로토콜과 동기화를 이루고 F/W 수준의 커맨드 프로토콜 인터페이스를 제공한다. 그리고 커맨드 프로토콜 인터프리터는 차량 내부에서 나오는 데이터 프로토콜을 받아 hex, 데시멀 디지털 스트링 값으로 변환한다.

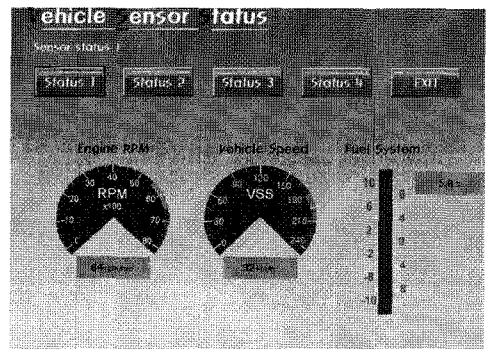
<표 1> 자가진단용 임베디드 시스템 스펙  
<Table 1> The spec of self-diagnosis embedded system

Main CPU : PXA270(500Mhz)
Memory : 64Mbyte Memory (SDRAM)
BOOT Flash : NOR
(Strata Flash 16~32Mbyte)
NAND Flash Memory : 16~ 128Mbyte
ETHERNET : CS8900(10M)
USB : Host, Slave INTERFACE
UART : FPUART, STUART(OPTION)
JTAG : Down Loading port
RTC : PORT 내장
LCD : 640x480 콘넥터 내장
Touch : MK712, ADS7843
Camera Input : Camera Input port 내장
Console Port : STUART, FPUART

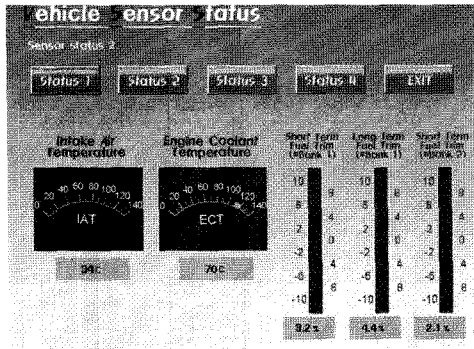
변환된 데이터는 RS232를 통해 자동차 자가 진단 시스템으로 전송된다. 테스트용 차량은 현대 EF-Sonata 2004년식 ISO1914-2 프로토콜을 사용하였다. <그림 10, 11, 12> 그리고 <그림 13>은 실험 구현 유저 GUI를 나타낸다.

<그림 10>은 자동차 내의 센서 상태 1을 나타낸다. 엔진 센서 RPM, SPEED 그리고 연료 시스템 상태와 각 센서들의 상태를 나타낸다.

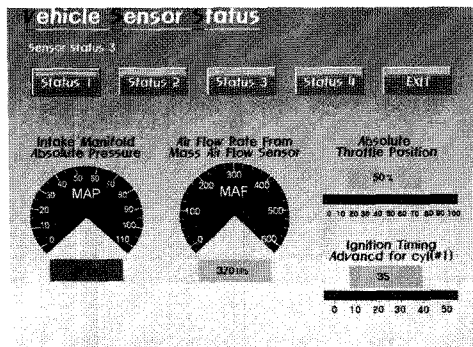
<그림 11>의 센서 상태 2는 Air Temperature, Engine Coolant Temperature 그리고 Short Term Fuel Trim, Long Term Fuel Trim 그리고 Short Term Trim bank 2 센서 상태를 나타내고 있다.



<그림 10> 센서 상태 1  
<Fig. 10> The first sensor state

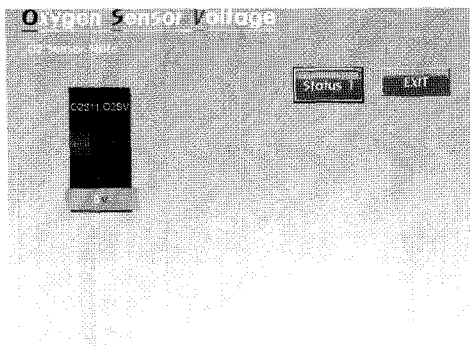


<그림 11> 센서 상태 2  
<Fig. 11> The second sensor state



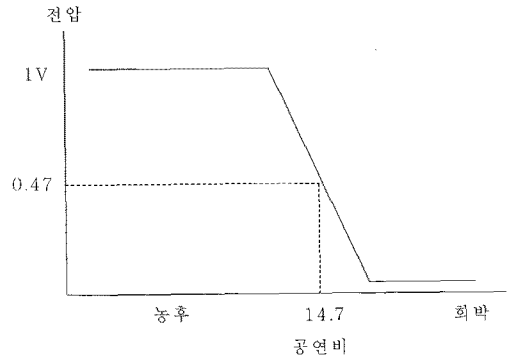
<그림 12> 센서 상태 3  
<Fig. 12> The third sensor state

<그림 12>는 Intake Manifold Absolute Pressure, Air Flow Rate From Mass Air Flow Sensor 그리고 Absolute Throttle Position과 Ignition Timing Advance for Cyl 센서들의 값들을 보여준다.



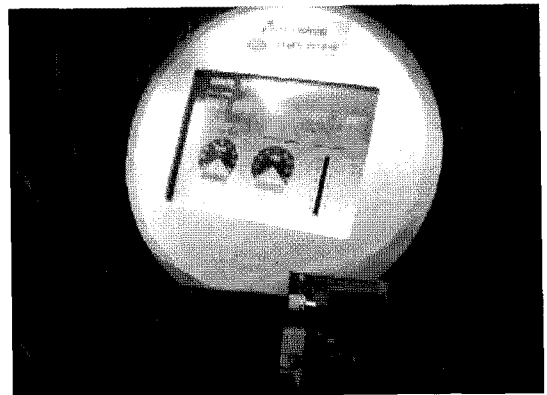
<그림 13> O2 센서 상태  
<Fig. 13> O2 sensor state

<그림 13>는 Oxygen Sensor 값을 보여주고 있으며 <그림 14>는 O2 센서 산소 농도차에 의한 출력 전압의 변화를 보여주고 있다.



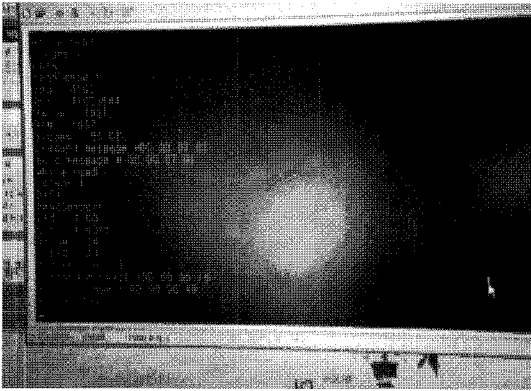
<그림 14> O2센서 산소 농도차에 의한 출력전압의 변화  
<Fig. 14> The difference of output voltage between oxygen concentration of O2 sensor

<그림 15>는 실제 운행 중 차량내의 CAN 통신 데이터를 ZigBee를 통하여 전송하는 실험 화면을 보여주고 있다.



<그림 15> 운행 중 차량내 CAN 데이터 ZigBee 전송 화면  
<Fig. 15> Display of the vehicle CAN data transfer operation ZigBee

<그림 16, 17>은 운행 중 전송되어 온 ZigBee 데이터 모니터링 화면을 보여 주고 있다.



<그림 16> 운행중 차량 CAN 데이터 ZigBee 모니터링 1

<Fig. 16> The first display of the vehicle CAN data ZigBee monitoring



<그림 17> 운행중 차량 CAN 데이터 ZigBee 모니터링 2

<Fig. 17> The second display of the vehicle CAN data ZigBee monitoring

본 논문의 차량 자가 진단 임베디드 시스템을 통해 대부분의 운전자가 자동차의 이상 현상이 발생했을 때 피부로 느끼지 못하여 자동차 내부 이상 현상을 인지하지 못한 채 운행을 하거나 차량 정비소에서 정비를 받아야 알 수 있는 정보를 실시간으로 눈으로 확인하여 DTC 정보를 확인하고 트러블 코드에 대한 구체적인 정보를 제공해 줌으로써 차량 정비를 정상적으로 유도하고 차량의 안전성을 보장할 수 있다.

## 2. 저전력 시스템 구성 및 결과

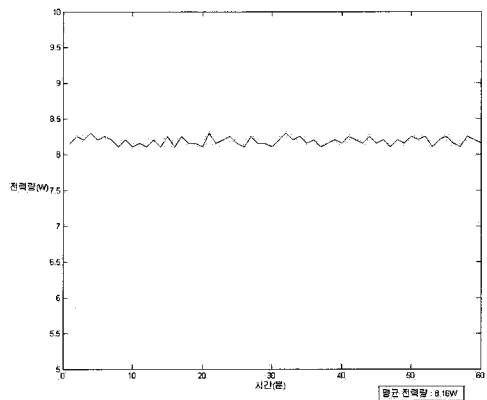
PXA270 ARM11 시스템으로 구성하여 내장형 프로그램을 통해 불필요한 전력 소모를 최대한 줄였다. 논문에서는 시스템 평균 전류 사용량 체크를 위해 전력 측정 스위칭 장치에서의 안전 작동 구역(SOA : Safe Operating Area)에서 테스트 하였다. 다음 식 1은 사용 전력 값 계산식에 해당한다.

$$P_n = V_n \times I_n \quad (1)$$

식 1에서  $P_n$ 은 순간 전력 값을 나타내고  $V_n$ 은 전압 값, 그리고  $I_n$ 은 전류 값을 나타낸다.  $n$ 은 특정 지점에서의 샘플을 나타낸다. 본 시스템에서는 식 1을 통해 식 2에서 보이는 것과 같은 평균 전력량을 유도하여 계산하였다. (단,  $N$ 은 플롯에서 동일한 값을 가진 샘플 수)

$$P_{Avg} = \frac{1}{N} \sum_{n=0}^{n=N} V_n I_n \quad (2)$$

<그림 18>에서 평균 전력 사용량을 그래프로 보여주고 있다. 차량용 임베디드 시스템들의 평균 전력 사용량이 보통 10W를 훨씬 넘어간다는 점에서 본 시스템에서 평균 전력량은 8.16W를 보이고 있다.



<그림 18> 자동차 자가 진단 임베디드 시스템 평균 전력 사용량

<Fig. 18> The average power consumption of car self-diagnosis embedded system

## V. 결 론

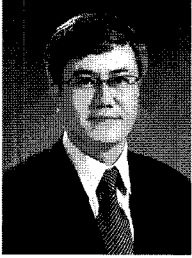
본 논문은 주행 중인 차량에 대해서 언제 어디서든 멀티미디어 시스템을 이용하면서 차량내의 엔진 센서로부터 차량의 정보를 받아 고장 진단 및 트러블 코드를 운전자에게 직접 알려줌으로써 비전문가도 차량의 문제점을 파악하고 안전한 운행을 할 수 있도록 하였다. 그리고 ZigBee 기반 저전력 차량용 임베디드 시스템으로 전력 사용량을 최대한으로 줄여 배터리 부하의 해미지를 막고 ZigBee 통신을 통한 차량의 내부 상태 모니터링을 할 수 있는 시스템을 구성하였고 임베디드 리눅스 시스템을 차량내의 센서 정보를 통해 고장 진단용 시스템을 구성하는 것이다. 하지만 아직 전 차량에 대한 확장성과 호환성에 문제가 있고, 사용자가 손쉽게 연결을 위한 인터페이스가 부족한 실정이다 이에 앞으로 개선해 가야 할 문제점이라고 볼 수 있다.

## 참고문헌

- [1] 신종욱 “임베디드 시스템을 이용한 홈 오토메이션 시스템 설계 및 구현”, 경성대학교 대학원 전기전자공학과 학위논문(석사), 2005
- [2] 윤중호 “인터넷 가전제품용 임베디드 이더넷 보드의 구현”, 한국항공대학교 학술지, 제38권, pp. 161-166, 2000.
- [3] 오세만, 이양선, 고광만, “임베디드 시스템을 위한 가상기계의 설계 및 구현”, 한국멀티미디어 학회, 제8권, 제9호, pp. 1282-1291, 2005. 9.
- [4] G. J. Pappas, G. Lafferiere, and S. Sastry. “Hierarchically consistent control systems,” *IEEE Trans. Automatic Control*, vol. 45, no. 6, pp. 1144- 1160, June 2000.
- [5] A. B. Kurzhanski and P. Varaiya, “Ellipsoidal techniques for reachability analysis,” *Hybrid Systems : Computation and Control*, Lecture Notes in Computer Science, LNCS 1790, pp. 202-214, 2000.
- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communication Magazine*, vol. 40, no. 8, pp. 102-114, Aug. 2002.
- [7] ETRI, “저속, 저가, 저전력, 무선 PAN 표준 개발 동향,” *ETRI 전자통신동향분석*, 제18권, 제2호, pp. 37-44, 2003. 4.

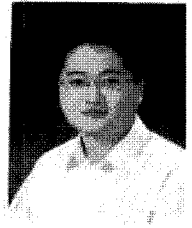


저작소개



**천 종 훈 (Chun, Jong-Hun)**

1999년 ~ 현재 : 전남도립 남도대학 컴퓨터정보통신과 교수  
1997년 ~ 1999년 : (주)한화 정보통신연구소 책임연구원  
1996년 : 조선대학교 전기공학과 공학박사  
1994년 ~ 1997년 : (주)현대전자 정보통신연구소 주임연구원  
1991년 ~ 1994년 : (주)맥슨전자 기술연구소 주임연구원  
1990년 : 조선대학교 전자공학과 공학석사  
1988년 : 조선대학교 전자공학과 공학사



**김 국 세 (Kim, Kuk-Se)**

2008년 2월 : 조선대학교 컴퓨터공학과 박사과정 수료  
2001년 2월 : 조선대학교 컴퓨터공학과 공학석사  
1999년 2월 : 조선대학교 컴퓨터공학과 공학사



**박 종 안 (Park, Jong-An)**

1975년 ~ 현재 : 조선대학교 정보통신공학부 교수  
1990년 ~ 1991년 : 영국 Surrey 주립대학 전기 & 전자공학과 객원교수  
1983년 ~ 1984년 : 미국 Massachus sette주립대학 전기&전자공학과 객원교수  
1986년 2월 : 조선대학교 전기공학과 공학박사  
1978년 2월 : 조선대학교 전기공학과 공학석사  
1975년 2월 : 조선대학교 전자공학과 공학사