Web-based Measurement of ECU Signals on Vehicle using Embedded Linux

Kwang-Hun Choi*, Hyun-Ho Lee*, Young-Choon Lee**, Tae-Kyu Kwon***,
and Seong-Cheol Lee#

* Dept. of Mechanical Engineering, Chonbuk National University, Jeonbuk, Korea
(Tel : +82-63-270-2320; E-mail: apollo_choi@hotmail.com)
* Dept. of Mechanical Engineering, Chonbuk National University, Jeonbuk, Korea
(Tel : +82-63-270-2320; E-mail: tazanboy@hanmail.net)
** Dept. of Mechanical Engineering, Chonbuk National University, Jeonbuk, Korea
(Tel : +82-63-270-2320; E-mail: leeyc66@hotmail.com)
*** Dept. of Bionics & Bioinformatics Engineering, Chonbuk National University, Jeonbuk, Korea
(Tel : +82-63-270-4066; E-mail: kwon10@chonbuk.ac.kr)
#Dept. of Mechanical Engineering, Chonbuk National University, Jeonbuk, Korea
(Tel : +82-63-270-2320; E-mail: meconlee@chonbuk.ac.kr)

Abstract: In this paper, we present a new method for monitoring of ECU's sensor signals of vehicle. In order to measure the ECU's sensor signals, the interfaced circuit is designed to communicate ECU and the Embedded Linux is used to monitor communication result through Web the Embedded Linux system and this system is said "ECU Interface Part". In ECU Interface Part the interface circuit is designed to match voltage level between ECU and SA-1110 micro controller and interface circuit to communicate ECU according to the ISO, SAE communication protocol standard.

Because Embedded Linux does not allow to access hardware directly in application level, anyone who wants to modify any low level hardware must develop device driver. To monitor ECU's sensor signals the most important thing is to match serial level between ECU and ECU Interface Part. It means to communicate correctly between two hardware we need to match voltage and signal level, and need to match baudrate. The voltage of SA-1110 is 0 ~ +3.3V and ECU is 0 ~ +12V and , ECU's communication Line K does multiple operation so, the interface circuit is used to match voltage and signal level. In Addition to ECU's baudrate is 10400bps, it's not standard baudrate in computer environment. So, we need to develop a device driver to control the interface circuit, and change baudrate. To monitor ECU's sensor signals through web there's a network socket program is working in Embedded Linux. It works as server program and manages user's connections and commands. Anyone who wants to monitor ECU's sensor signals he just only connect to Embedded Linux system with web browser then, Embedded Linux webserver will return the ActiveX webbased measurement software. It works in web browser and inits ECU, as a result it returns sensor signals through web. All the programs are developed with GCC(GNU C Compiler) and, webbased measurement software is developed with Borland C++ Builder.

Keywords: Electronic Control Unit(ECU), Web-based measurement, Embedded Linux, Intel-SA1110 processor, ActiveX control program, RF module

1. INTRODUCTION

Automobile is a product of mechanics and electronics for the basic function of transportation, but nowadays the safety and performance is strongly studied at the vehicle maker for the purpose of customer satisfaction. Recently, Convenience of the compact car has become one of the pursuing items to keep up with the telematics market accompanied by IT. Telematics is a new compound word of telecommunication and information, and it comprises the navigation system like a security service and real time traffic information service by the communication of between each sensor and ECU(electronic control unit), which is controlling all kinds of engine functions and power train parts. In case of the engine failure, ECU has the information of the fault value of embedded engine parts, and this fault information is used in the auto repair shop by the scanner[1]-[3] with the wire cable which is connected to ECU ALDL line.

Recently development of wireless scanner by the RF & Bluetooth module has been studied[4][5], but scanner with web based measurement has not been tried yet. For the purpose of the development of the web based scanner, this study was concentrated on the programming of the embedded Linux platform which can communicate with ECU by designing the interfacing electronic circuit in pursuit of the improvement of complicated users manual of scanner. ECU sensor signal has been graphed at the web based measurement by simultaneous -ly operating the ECU and internet web site, and experimental results show a good agreement to the scanner at the condition of idle mode and acceleration mode.

2. ELECTRONIC CONTROL UNIT

2.1 ECU Configuration

The information and signals from engine related sensors required to control one vehicle, are very much the same as the others. Particularly, the ignition and fuel management are main function of the ECM(electronic control module), and both are interrelated when it comes to fuel economy, emissions and performance. This is why most management systems now in use combine ignition and fuel control. This particular engine management system is controlled by an electronics control, which is responsible for managing injection and fuelling requirements of the engine.

The ECU also provides an output to the diagnostic system in the form of both warning light and a serial diagnostic data. On board diagnostics are becoming essential for the longer term operation of systems if clean exhausts are to be maintained. In the USA and EC a very comprehensive diagnosis of all components that affect the exhaust is required. Nowadays, all new cars after the year 2000 will have to comply with the KWP2000[6]-[8] system. Digital electronics allow both sensors and actuators to be monitored. This is done by allocating values to all operating states of the sensors and actuators. If a deviation from these figures is detected this is stored in memory and can be output in the workshop to assist with fault detection using wire scanner. ECU can communicate with a scanner through the K-line or L-line for the purpose of monitoring of a vehicle's state.

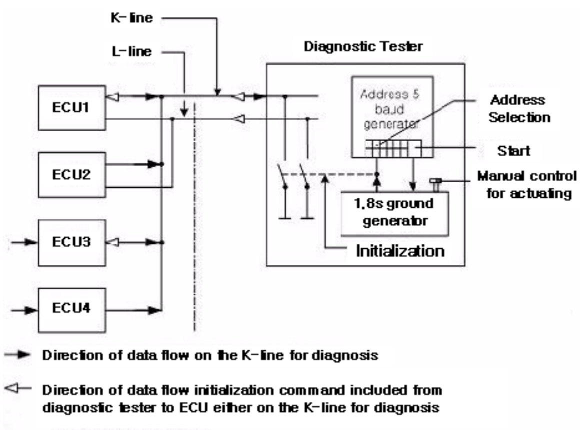
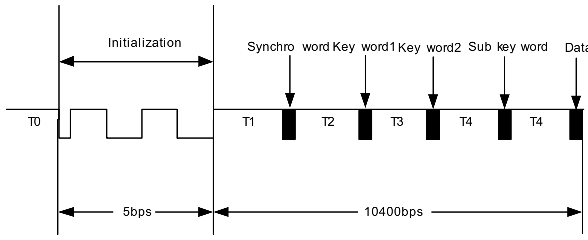


Fig.1 Bus system of ECU's various types

K-line supports the serial type data that are bi-directional bus-line to the external system. Also, K-line used for the initialization of communication. On the other hand, L-line is usually used for the initialization of communication, which is one-direction. This is also categorized with several ways varying with the makers. The system type classify by L-line is or is not, and the direction of K-line. When the several ECUs connect to the same bus line, A ECU only initialize through the fixed address.

2.2 ECU Communication Protocol

ECU has to be initialized before data communication with scanners by serial port. There are two kinds of methods to complete ECU initialization. One is to make the L-line zero state and grounded when key is placed at ignition position.



$$2\text{ms} \leq T_0 \leq \infty$$
$$2\text{ms} \leq T_1 \leq 2\text{s}$$
$$2\text{ms} \leq T_2 \leq 1.2\text{s}$$
$$0.2\text{ms} \leq T_3 \leq 1.2\text{s}$$
$$2\text{ms} \leq T_4 \leq 1.2\text{s}$$

Fig. 2 Communication Initialization

Then, driver's panel gets the red sign and send the error codes in the Free-Running Mode. The other is more realistic way for ECU initialization, first scanner sends the address code at the communication speed of 5 bps to K-line and L-line simultaneously, the ECU sends the serial communication data at the speed of 10.4 Kbps. Fig. 2 shows the process of ECU communication initialization.

After the ECU initialization, faults information for the ECU can be received on the web based environment. The data header is received first which is consisted of synchronization word and keyword. The keyword configuration is different according to the automobile makers. When makers want to use another new type of keyword configuration, they has to summit this to FAKRA for the purpose of keyword homologation.

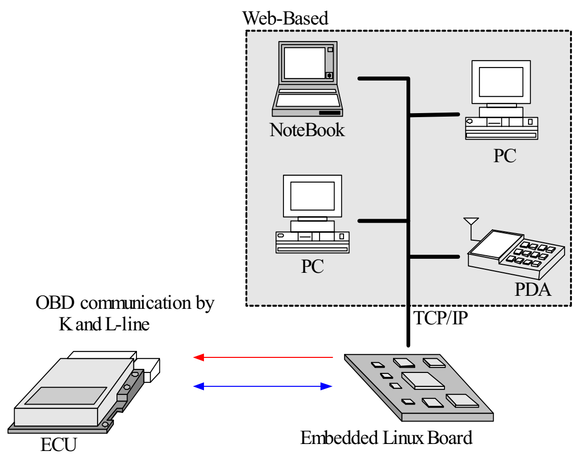


Fig. 3 Configuration of web-based measurement system

Fig. 3 shows the entire system configuration for this study. ECU interface circuit is designed for the communication between the ECU and embedded LINUX system. Socket program mainly used in real time faults detection and communication with embedded web server has been coded on embedded Linux system to enable the ECU to communicate at the condition of PC web browser environment.

3. WEB-BASED MEASUREMENT SYSTEM.

3.1 H/W Interface

Table 1 describes the H/W specification of embedded LINUX system. Intel SA-1110 MCU is loaded to the embedded Linux system used in this paper, and this system has 16M flash ROM, 32M SDRAM and CS9800A network controller.

Table 1 H/W specification of embedded Linux system

Part	Specification	Model
MCU	206MHz Intel Strong ARM RISC Processor	SA-1110
RAM	32Mbytes SDRAM	K4S281632C
ROM	16Mbytes Intel Strata Flash	Intel E28F128J3A
Ethernet	CS8900 10Mbps	CS8900
Serial	RS-232C 1Port	MAX3221E
	RS-232C 1Port	ECU Com
USB	USB Client	Power
GPIO	General Purpose I/O	I/O

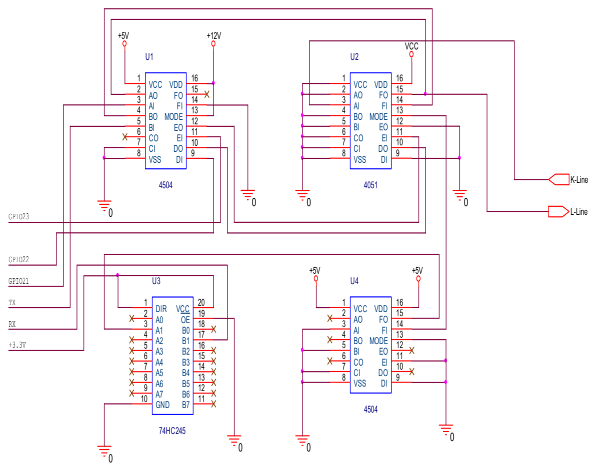


Fig. 4 ECU Interface circuit

ECU interface circuit is needed to communicate with the embedded LINUX system. Fig. 4 explains the ECU interface circuit. ECU interface circuit has the multiplexer for the exact distribution of signals that are overlapped on ECU K-line and level shifter to fit the ECU voltage level to the LINUX module operating voltage. Three GPIO lines have been designed to control these IC circuits. Fig. 5 explains the block diagram of web-based communication between the ECU and LINUX module.

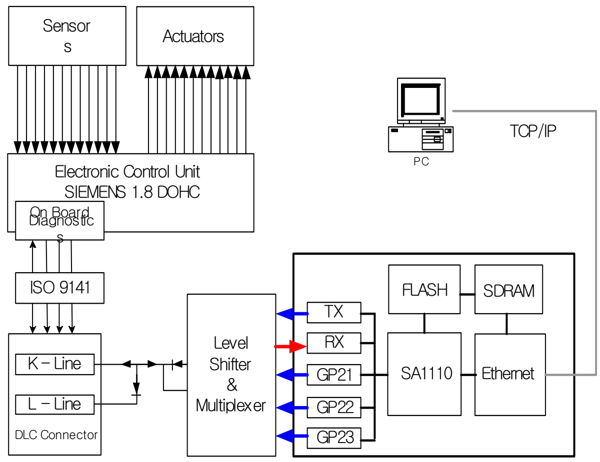


Fig. 5 Block diagram of web-based measurement system

3.2 System S/W

3.2.1 ECU Interface Device Driver

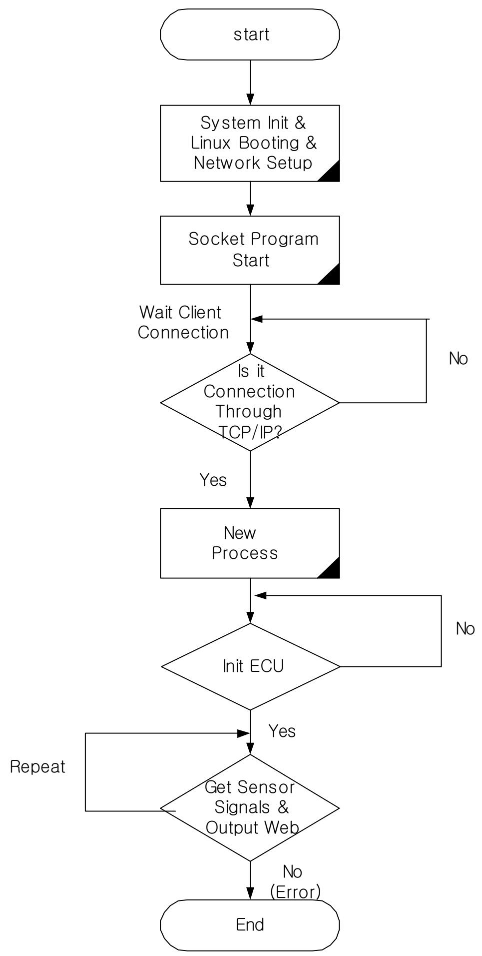


Fig. 6 Flowchart of network socket program

Device Driver of the ECU has to be programmed first to communicate with the embedded LINUX. In this study, The ECU device driver is coded like a module and embedded into LINUX module. The main functions of the ECU device driver are composed of the transmission of ECU initialization signal, the generation of ECU communication speed 10400bps and the management of overlapped I/O K-line signals. The roles of the main functions of ECU device driver are as follows.

□ ***ecu_open( )***

Using ***open( )*** function to control the ECU interface circuit in application program, this one is called for the preparation of communication. This takes management of GPIO port and adaptable initialization. This stops at the error signal.

□ ***ecu_close( )***

Using ***close( )*** function to complete the program that are related to interface control in application, this one is called and returns the resources related to GPIO. This has the watchdog function.

□ ***ecu_ioctl( )***

In application program the ECU initialization is needed to communicate with other fault detecting devices. For performing this work, ***ioctl***(Input/Output Control) function is called to manage the ECU communication properly. In this case, this function is used to manage the ECU initialization signal transmission and configure the communication speed at the 10400bps

3.2.2 Network Socket Program

The programmed device driver is used in the application program, the network socket program takes this role. The network socket program has several functions, which are waiting the client connection and returning the sensor signals to the client by communicating with the ECU. Fig.6 shows the algorithm of the network socket program.

When the connection from the new client is requested, the socket program is forked to child process to deal with the client request and wait the new internet connection. The newly made child process will initialize the ECU to connect with interfaced client and transmit the ECU sensor signals to the client.

3.3 Experiment System

To reduce the developing time to the web based automobile faults detection, an experiment ECU measuring system has been configured which is the same as the common ECU installed in the actual vehicle and performed the measurement of faulted values. Fig.7 shows the actual photo of experiment ECU measuring module. After completing this experimental test successfully, the actual ECU fault detection can be tested.

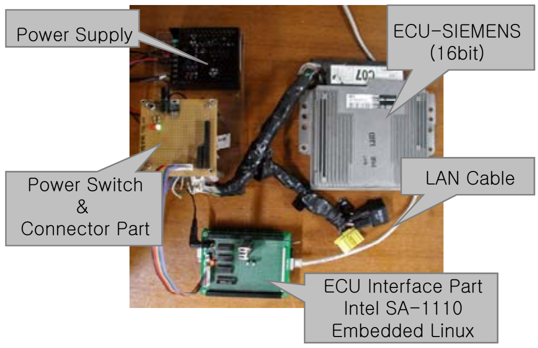


Fig. 7 Configuration of experiment system in the Lab.

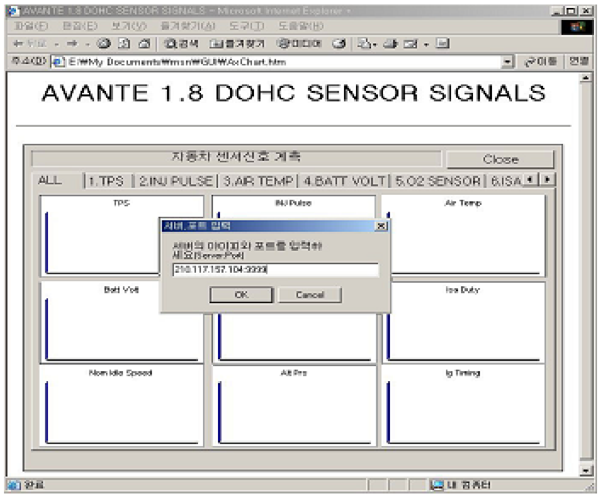


Fig. 8 Display panel of the Embedded Linux System

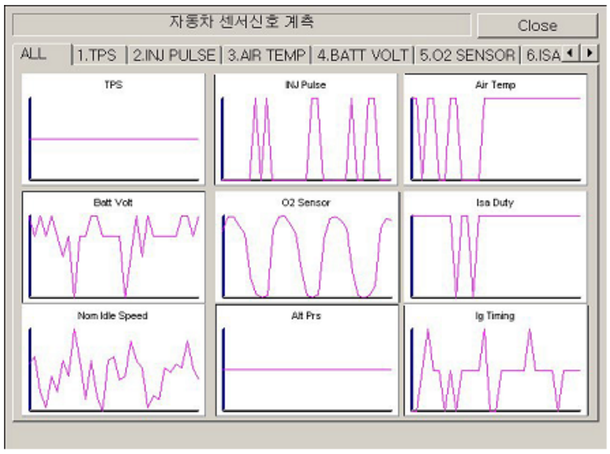


Fig. 9 Results from the real ECU in idle state

4. EXPERIMENTAL RESULTS

The client program that can receive faults values and communicate with embedded LINUX system additional to the above explained interface circuit, device driver and socket program for the web based remote measurement. In this study the active X control program is coded that can execute in internet web browser.

The active X control program is saved in the embedded system normally and when the client tries to connect to the web browser, this is transmitted to the web site automatically. For this communication, there is an embedded web server execution so called Boa in the embedded LINUX system.

Fig.8 is the display panel that is trying to connect to the socket program after the active X program is installed in the embedded web server. After the completion of the connection to the web browser, ECU initialization is done and the sensor output is detected on the web environment.

The display panel rendering the outputs of the sensor at the idle state mode is shown in Fig.9. The graph has the auto scaling x-axis and y-axis to compare the results of the ECU sensor outputs

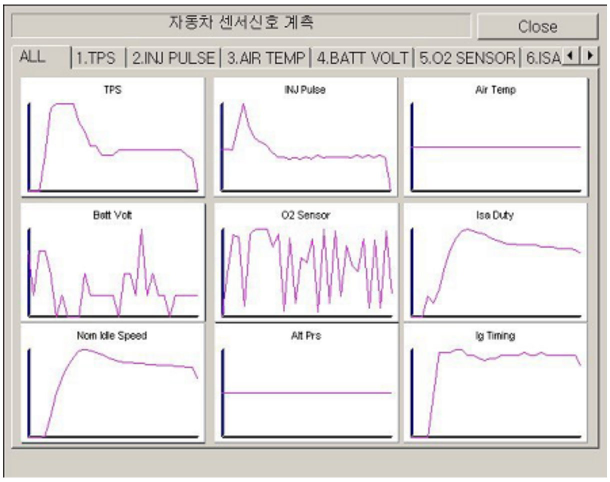


Fig. 10 Results of the ECU in accelerated state

Fig.10 shows the test results from sensors to be mounted on the engine at the acceleration mode. The characteristics of the test results are quite different compared to the idle mode.

5. CONCLUSIONS

Development of wireless scanner using the RF and Blue -tooth module has been studied, but scanner with web-based measurement has not been tried yet. For the purpose of the development of the web-based scanner, this study was concentrated on the programming of the embedded Linux platform which is able to communicate with ECU by designing the interfacing electronic circuit in pursuit of the improvement of complicated users manual of scanner. From the results of designing the ECU interface with web browser, we found the follows:

[1] The web-based automobile faults detection was possible. Measured values from the web browser showed the same as the values of the wire scanner.

[2] The test method and process are simplified by inducing the web environment to the users.

[3] When the internet web browser was connected to the faults detection program (program upgrade is automatically done), the convenience of the web based faults detection has been enhanced because all of the related active X control were installed as soon as possible.

[4] The web-based measurement showed the stable and exact values compared to the common wire scanner.

ACKNOWLEDGEMENT

This research was supported by the Mechatronics Research Center(MRC) of Chonbuk National University.

REFERENCES

[1] NEXTECH (http://www.nex-tek.com)

[2] HANATECH (http://www.hanatech.net)

[3] YOOLIM (http://www.nchecker.com)

[4] J. H. Jeong, et al, "*Remote Measurement for ECU Self Diagnostic Signal by RF Module*," in the Proceedings of the KSPE 2001 Spring Conference, pp.231-234, 2001.(in Korean)

[5] Yun Yeo-heung et al, "*PC-PC software development of ECU self-diagnostic signal using Bluetooth*," in the proceedings of the KSPE 2001 Spring Conference, pp. 257-260, 2001.(in Korean)

[6] International Standard, "*Road vehicles (Diagnostic systems) Requirements for interchange of digital information,*" Reference number ISO 9141: 1989.

[7] International Standard, "*OBD-II Protocol*," Reference number ISO 9142.

[8] International Standard, "*Road vehicle diagnostic systems keyword protocol 2000*," 1st/ed. 1999-03-15, Reference number ISO 14230: 1999(E).

[9] Kwang-Hun Choi, "*Remote Measurement of the KWP 2000 ECU Signals using Bluetooth*," Master degree thesis of Chonbuk National University, Feb. 2004, Korea.

[10] Hyun-Ho Lee, "*Webbased Measurement of ECU Sensor signals on Vehicle using Embedded Linux*," Master degree thesis of Chonbuk National University, Feb. 2004, Korea.

[11] SAE Recommended Practice, "*Diagnostic Connector -SAE J1962 FEB98*," Report of the SAE Vehicle of E/E systems Diagnostic Standards Committee approved June 1992 and completely revised February 1998.

[12] SAE Recommended Practice, "*OBDII Scan Tool-SAE J1978 FEB98,*" Report of the SAE Vehicle of E/E Systems Diagnostic Standards Committee approved March 1992 and revised February 1998.

[13] Martin Sachenbacher, Peter Struss, and Reinhard Weber, "*Advances in Design and Implementation of OBD Functions for Diesel Injection based on a Qualitative Approach to Diagnosis,*" SAE 2000 World Congress, Detroit, USA, 2000(SAE SP-1501), pp.23-32, 2000.

[14] R. Frank, M. Baker, L. C. Puhl, E. A. Dabbish, and M. Danielsen, "*Sensing and Systems Aspects of Fault Tolerant Electronics Applied to Vehicle Systems,*" Proc. of the Int. Congress on Transportation Electronics, Vehicle Electronics in the 90's, pp.45-54, 1990.

[15] Gil Shulz, "*Portable On-Board-Diagnostic(OBD)II /CAN scan tool*," Siemens Components, Inc.