

산업용 CRDI 엔진을 위한 OBD-II 프로토콜 및 실시간 진단 모니터링 시스템 개발

장성진* · 장중욱**

*동의대학교

Development of Real-time Diagnostic Monitoring System and the OBD-II Protocol for Industrial CRDI Engine

Sung-jin Jang* · Jong-wook Jang**

*Dong Eui University

E-mail : ch99jin@hanmail.net*, jwjang@deu.ac.kr**

요 약

산업용 CRDI 엔진에서 센서의 정보를 분석하여 최적의 조건으로 엔진이 동작하도록 하기 위해 산업용 CRDI 시스템에 적합한 OBD-II 프로토콜을 설계하고 ECU의 센서 정보를 전문가 아닌 일반 사용자가 내용을 이해하기 쉽도록 사용자 중심의 진단기의 개발이 필요하다. 따라서, 본 연구에서는 자동차 고장진단 신호 및 센서 출력 신호를 유선시스템과 무선 시스템인 블루투스 모듈을 이용하여 실시간 통신이 제공 될 수 있는 OBD-II 진단기 S/W를 설계 및 구현하였다. 엔진이상으로 인한 사고의 예방이 가능하고, 최적의 조건으로 엔진이 동작하므로 과도한 배기가스 배출이나 불완전 연소 가스 배출과 같은 대기환경오염을 예방할 수 있다.

키워드

CRDI(common rail direct injection), OBD-II, PID, DTC(Diagnostic Trouble Code), Bluetooth,

I. 서 론

OBD-II 규격은 현재 차량 내 다양한 전자제어 장치 및 센서들의 정보 모니터링 및 고장코드들(Diagnostic Trouble Code: DTC)을 차량 내 온-보드 시스템을 통해 읽어올 수 있도록 하기 위한 진단기술(On-Board Diagnostics : OBD) 표준으로서 최근 차량의 진단 네트워크에 널리 적용되고 있다. 그리고 최근 차량에 탑재되는 개별 전자제어장치(Electronic Control Unit: ECU)는 엔진 및 변속장치와 같은 파워트레인계를 비롯하여, 공압 장치나 브레이크와 같은 새시계 및 바디계의 편의 장치들에 이르기까지 그 적용범위가 늘어나고 있으며, 각각의 제어장치에 차량의 상태를 보다 정밀하게 측정하거나 진단하기 위해 구성되는 센서들의 수가 증가되고 있다[1]. 따라서, OBD-II 표준은 범규화된 배기가스 OBD 관련 표준 기능뿐만 아니라 차량 제조사별로 자사의 차량 모델별로 증가되고 있는 탑재 ECU 및 편의/안전 전장품들에 특화된 진단 기능들까지 포함하여야 함에 따라 범규화된 OBD 표준과

EOBD (EnhancedOBD)로 구분되어 지속적으로 업데이트되고 있다. OBD 표준은 자동차에 부착된 센서들로부터 ECU로 전달된 자동차의 주요 계통에 대한 정보나 고장 등의 정보를 직렬 통신기능을 이용하여 자동차의 콘솔이나 외부장치에서 볼 수 있도록 한 기능이다. 이러한 표준의 제정으로 인하여 자동차의 진단이 손쉽게 되었으며, 향후 자동차의 자동화된 진단 시스템이라는 새로운 시장을 창출할 것으로 예상된다[2].

본 연구에서는 OBD-II 표준을 사용하여 차량 위주의 진단기를 개발하여 운전자 중심의 진단 서비스를 제공하며, 자동차 고장진단 신호 및 센서 출력 신호를 유선시스템과 무선 시스템인 블루투스 모듈을 이용하여 실시간 통신이 제공될 수 있도록 하였다.

II. 관련 연구

현재 OBD-II 규격은 현재 차량 내 다양한 전자 제어장치 및 센서들의 정보 모니터링 및 고장 코드들(Diagnostic Trouble Code: DTC)을 차량 내

온-보드 시스템을 통해 읽어올 수 있도록 하기 위한 진단기술 표준으로서 최근 차량의 진단 네트워크에 널리 적용되고 있다.[1]

2.1 기존 OBD-II 프로토콜 분석

OBD-II 메시지는 차량 진단 툴을 사용하여 차량 ECU로부터 취득할 수 있다. OBD-II 프로토콜 구조는 표1에서 보는 바와 같이 Header, Data, CRC로 구성되며, 총 11 바이트의 데이터가 저장된다[3].

표 1. OBD-II Message Structure

헤더1	헤더2	헤더3	Data1	Data2	Data3	Data4	Data5	Data6	Data7	CRC
-----	-----	-----	-------	-------	-------	-------	-------	-------	-------	-----

데이터1은 차량 상태를 나타내며 여러 가지 데이터를 표시한다. 아래 표2에서 보이는 것과 같이 9 가지 모드가 있다.

표 2. OBD-II Data Byte

구분	내용
MODE1	차량속도, 온도, 각종 센서들에 대한 데이터를 표현
MODE2	모드1과 유사, Freeze Frame 데이터 표현
MODE3	차량의 고장코드를 표현
MODE4	고장코드와 저장된 데이터들을 정리(삭제)
MODE5	산소센서의 테스트 결과 값
MODE6	비 지속적인 계측 값의 테스트 결과 값
MODE7	모드3과 유사, 미결상태인 고장코드 표현
MODE8	제조사만의 특수 제어 모드
MODE9	ECU에 저장된 사용자의 차량 정보 요청 모드

데이터2는 PID(Parameter ID)를 나타내며, 차량의 ECU로부터 받은 정보 및 데이터들을 사용자가 확인 할 수 있도록 지원가능 여부를 결정한다.

데이터3~7은 차량 상태 데이터로써 고장여부, 엔진의 상태, 냉각수의 상태, 속도, 배터리 상태 등의 1Byte씩 저장하여 나타낸다. 그리고 마지막의 CRC는 통신 중 오류의 발생여부를 알린다[4].

전자 제어 엔진이 이상이 있는 경우나 배출 가스 관련 부품에 이상이 발생하는 경우 고장 내용에 따라 고장 코드(DTC)를 정하여 ECU 내의 RAM 메모리에 자동으로 기록되도록 하고 경고등(MIL:Malfunction Indicator Lamp)을 점등 시켜 운전자에게 전자 제어 엔진에 이상이 있다는 것을 알려 주는 기능이 있다. 이것은 차량 이상시 점검 위치를 쉽게 확인 할 수 있도록 OBD를 규정하여 놓은 것이었지만 컴퓨터의 빠른 발전으로 ECU(컴퓨터)의 입·출력 이상시 freeze frame 기능(DTC 발생시 ECU에 기록할 수 있는 기능) 뿐만 아니라 ready test 기능(배출가스 장치의 모니터링)을 하는 기능을 가지게 되었다[5]. 따라서 전자 제어 엔진 차량의 이상시 기본 점검으로 가장 우선 점검하여야 하는 것이 자기 진단(diagnosis)기능이다.

본 연구에서는 CAN 통신과 ISO 방식을 수행하여 차량의 상태정보를 실시간으로 확인하여 수집, 저장하고 이 수치들을 블루투스를 통해 윈도우

프로그램 및 스마트 폰으로 전달되어 운전자에게 보여 줄 수 있게 개발한다.

III. 시스템 설계 및 구현

OBD-II 진단기를 통해 OBD-II 네트워크의 프로토콜을 분석하여 자동차의 현재 운전 상태정보를 수집하고, 수집된 정보를 가공하여 운전자가 보기 쉽게 윈도우용 프로그램으로 개발하고, 또한 블루투스 네트워크를 이용해 언제 어디서나 운전자가 자동차 현재 상태를 스마트폰에서 볼 수 있게 제공하는 서비스 플랫폼에 대해 개발하고자 한다. 그림1은 OBD-II 및 블루투스 네트워크를 이용한 차량 진단시스템의 구성도이다[6].



그림 1. OBD-II 및 블루투스 네트워크를 이용한 자동차 진단 개략도

3.1 차량용 시뮬레이터

개발 과정에서 필요한 테스트들은 실제 차량에서의 테스트가 어려운 관계로 차량과 동일하게 통신 테스트가 가능한 차량용 시뮬레이터를 사용하여 테스트하였다. 그림 2는 직접 제작한 OBD-II 시뮬레이터로 블루투스 통신을 위한 동글과 OBD-II 커넥터로 구성되어 있다[7,8].

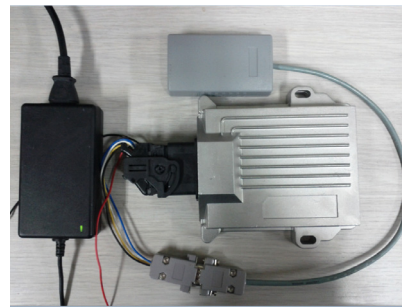


그림 2. OBD-II 커넥터

3.2 블루투스 OBD-II 프로토콜 설계

개발된 OBD-II 프로토콜은 기존 OBD-II 표준을 기반으로 제작되었으나 구조에 차이점이 있다. OBD-II 프로토콜의 표준의 경우 차량에 요청한 하나의 PID 정보에 대해서만 정보를 읽어 응답해 주지만 개발된 산업용 차량 OBD-II 프로토콜의

경우 필요한 차량의 정보를 모두 읽어 들여 한꺼번에 차량정보를 전송해 준다.

(1) 상태 정보 획득을 위한 OBD-II 프로토콜 구조

■ OBD-II 프로토콜 요청 메시지 구조

OBD-II 메시지는 엔진 진단 틀을 사용하여 차량 ECU로부터 취득할 수 있다. 표 3은 제안된 OBD-II 프로토콜 요청 메시지 구조로 Header, Data, Checksum으로 구성되고, 총 12바이트의 데이터가 저장되며 HEX 코드를 사용한다. 표 4는 OBD-II 프로토콜 요청 메시지 세부 코드이다.

표 3. 제안된 OBD-II 프로토콜 요청 메시지 구조

Command STX	Command ID	Info	Opt1	Opt2	Checksum	Command ETX
-------------	------------	------	------	------	----------	-------------

표 4. OBD-II 프로토콜 요청 메시지 세부 코드

내용	BYTE 정보	HEX 코드
Command STX	0x02	0x02
Command ID	0x14	0x31
		0x34
Info	0x00	0x30
		0x30
Opt 1	0x00	0x30
		0x30
Opt 2	0x00	0x30
		0x30
Checksum	0xEC	0x45
		0x43
Command ETX	0x03	0x03

■ OBD-II 프로토콜 응답메시지 구조

제안된 OBD-II 프로토콜의 경우 ECU에게 차량 정보를 요청할 경우 한번의 요청 메시지로 차량의 전체 센서 정보를 읽어오도록 설계되었다. ECU에서 제공하는 센서정보는 31가지이다. 표 5는 제안된 OBD-II 프로토콜 응답 메시지 구조를 보여준다.

표 5. 제안된 OBD-II 프로토콜 응답 메시지 구조

Data STX	Data1	Data31	Checksum	Data DTX
----------	-------	-------	--------	----------	----------

(2) 고장 코드 획득을 위한 OBD-II 프로토콜 구조

전자 제어 엔진에 이상이 있는 경우나 배출 가스 관련 부품에 이상이 발생하는 경우 고장 내용에 따라 고장 코드(DTC)를 나타내어 엔진 이상에 대한 사고 예방이 가능하다. 표 6은 ECU 고장 코드 응답 메시지 구조이다. 표 7은 ECU DTC 응답 메시지 세부 코드이다.

표 6. ECU DTC 코드 응답 메시지 구조

Command STX	Command ID	MODE	DTC 코드	Checksum	Command ETX
-------------	------------	------	--------	----------	-------------

3.3 OBD-II 프로토콜 진단 S/W 알고리즘

ECU의 상태 정보를 수집하기 위해서는 다음 그림 3과 같은 과정을 통해 데이터가 전송되어진

표 7. ECU DTC 응답 메시지 세부 코드

내용	BYTE 정보	HEX 코드
Command STX	0x02	0x02
Command ID	0x1D	0x31
		0x44
MODE 정보	0x01	1 : Mode3 0x30
		0 : Mode7 0x31
DTC 코드 (P201)	0x0201	0x30
		0x32
		0x30
		0x31
Checksum	0xE7	0x45
		0x37
Command ETX	0x03	0x03

다. 올바른 요청 데이터가 전송되었다면 31개의 데이터를 저장한다. 그림 3은 엔진의 상태 정보 수집을 위한 알고리즘이다.

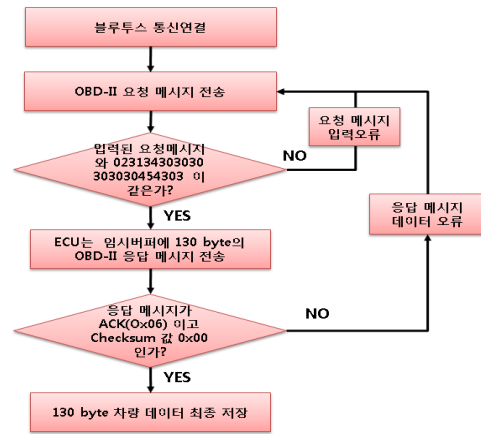


그림 3. 차량 정보 수집 알고리즘 순서도

ECU의 고장코드를 수집하기 위해서는 다음 그림 4과 같은 과정을 통해 고장코드 데이터가 전송되어진다. ECU의 고장코드 수집은 앞서 설명한 차량 정보 수집 알고리즘과 유사하다.

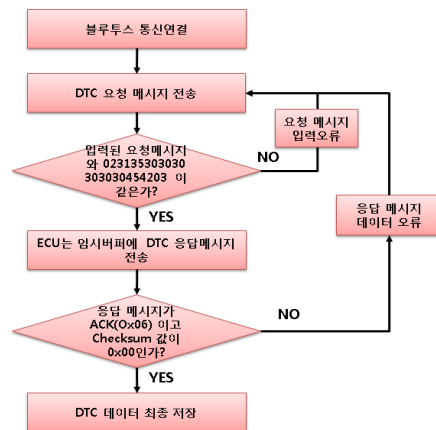


그림 4. 엔진 고장 코드수집 알고리즘 순서도

IV. 실험 및 결과

본 절에서는 구현한 OBD-II 커넥터를 통해 차량 정보를 수집하여 운전자가 보기 쉽게 윈도우 프로그램을 개발하였다. 그림 5는 산업용 차량 진단 시스템의 메인 화면으로 차량정보 입력, 차량정보 스캔, 차량자기진단, 차량정보 스캔 메뉴로 구성되어 있다.



그림 5. 산업용 차량 진단 시스템의 메인 화면

그림 6은 산업용 엔진 진단 시스템에서 엔진 상태 데이터를 전송받아 보여주는 화면이며, 그림 7은 차량의 다양한 센서 데이터를 그래프로 표현하여 조금 더 정확한 진단을 할 수 있도록 구현하였다. 그림 8은 엔진에 고장이 발생되었을 시 고장 코드를 나타내는 화면이다.

InputTime	RPM (rpm)	APS (V)	AFS (V)	ATS (°C)	RPS (Bar)	FTS (°C)	WTS (°C)	EGT
2015-06-02 13:08:58	2335	2.57	0	35	157	-15	-30	520
2015-06-02 13:08:59	2336	2.67	0	35	167	-15	-30	520
2015-06-02 13:09:01	2337	2.77	0	35	177	-15	-30	520
2015-06-02 13:09:03	2338	2.88	0	35	188	-15	-30	520
2015-06-02 13:09:04	2340	2.99	0	35	198	-15	-30	520
2015-06-02 13:09:05	2341	2.68	0	35	158	-15	-30	520
2015-06-02 13:09:06	2342	2.78	0	35	168	-15	-30	520
2015-06-02 13:09:13	2349	2.94	0	35	194	-15	-30	520
2015-06-02 13:10:52	2347	2.76	0	35	176	-15	-30	520
2015-06-02 13:10:53	2348	2.87	0	35	187	-15	-30	520
2015-06-02 13:10:54	2349	2.97	0	35	197	-15	-30	520
2015-06-02 13:10:55	2350	2.67	0	35	167	-15	-30	520
2015-06-02 13:10:56	2351	2.67	0	35	157	-15	-30	520
2015-06-02 13:10:57	2352	2.77	0	35	167	-15	-30	520
2015-06-02 13:10:58	2353	2.87	0	35	177	-15	-30	520
2015-06-02 13:10:59	2354	2.97	0	35	187	-15	-30	520
2015-06-02 13:11:13	2363	2.77	0	35	177	-15	-30	520
2015-06-02 13:11:44	2368	2.87	0	35	187	-15	-30	520
2015-06-02 13:11:45	2369	2.97	0	35	197	-15	-30	520

그림 6. 실시간으로 전송되는 차량 데이터 화면

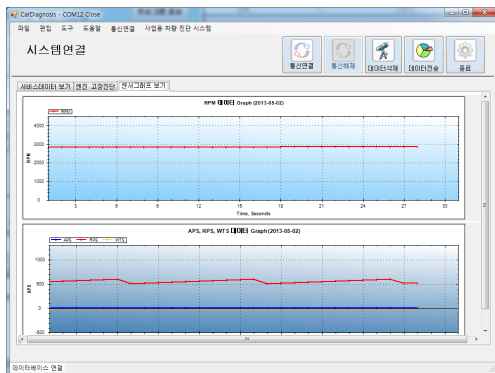


그림 7. 차량 센서 데이터 그래프 출력 메뉴

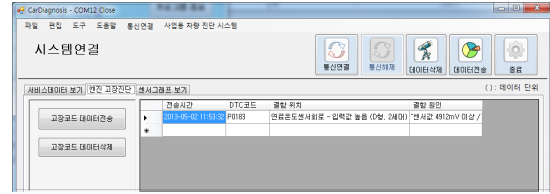


그림 8. 고장진단 메뉴

V. 결론

본 연구에서는 OBD-II 진단기를 통해 OBD-II 네트워크의 프로토콜을 분석하여 자동차의 현재 운전 상태 정보를 수집하고, 수집된 정보를 가공하여 운전자가 보기 쉽게 윈도우용 프로그램으로 개발하였다. 윈도우 프로그램 S/W를 통하여 실시간으로 차량 정보를 수집하고 수집된 정보를 가공 처리하여 운전자가 보기 쉽도록 그래프로 구성하였다. 블루투스 통신을 기반으로 엔진 센서 데이터를 스마트폰으로 전송하여 앱을 통해 자동차 진단이 가능하게 함으로써 진단기를 항상 가지고 다닐 필요 없이 차량의 상태를 확인 할 수 있다. 또한 차량 고장시 차량고장신호(DTC)를 확인하여 운전자가 빠른 조치를 취해 사고를 미연에 방지할 수 있도록 하였다.

향후, 본 연구에서 자동차 정비시 필요한 다양한 엔진 센서 데이터 및 기능을 추가하여 산업용 CRDI 엔진에 직접 적용할 수 있는 최적의 안드로이드 어플리케이션을 개발하고 현재 지원하지 않는 아이폰 어플리케이션을 개발하고자 한다.

감사의 글

이 논문은 2015년도 BB21과 중소기업청에서 지원하는 2015년도 산학협력력 기술개발사업 C0249807의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

- [1] <http://www.obdii.com/>
- [2] <http://en.wikipedia.org/wiki/OBD-II#OBD-II>
- [3] Santini, AI, OBD-II, Cengage Learning, 2010
- [4] OBD-II를 이용하는 오픈소스 프로젝트 (<http://www.opendiag.org/>).
- [5] 이봉우, OBD-II(배기가스), 경영사, p16, 2005
- [6] 허광호, 이종주, 허윤영, 최상렬, 신명철, OBD 차량진단 코드 발생 시뮬레이터 개발에 관한 연구, 대한전기학회, 2007.
- [7] Henderson, Bob/Haynes, John H, The Haynes OBD-II & Electronic Engine Management Systems Manual, Haynes Publishing, 2006.
- [8] 김민복, “전자제어 엔진”, 플드벨