

OBD 차량진단 코드 발생 시뮬레이터 개발에 관한 연구

하광호, 이종주, 허윤영, *최상열, 신명철
 성균관대학교, *인덕대학

A Study of a Simulator Development Generating OBD Diagnostic Code

Ha kwang ho, Lee jong joo, Heo yoon young, *Choi sang yeol, Shin myong chul
 Sungkyunkwan University, *Induk college

Abstract - 자동차, 항공기, 철도 및 선박 등과 같은 각종 교통수단에 발생하는 이상현상에 대한 사용자의 정확한 복구 조치 능력 향상을 위하여, 발생한 고장코드에 대한 신속하고 정확한 해석은 매우 중요하다. 이에 따라 본 논문에서는 차량의 고장 진단 프로토콜 중 SAE(미국 자동차 기술자 협회) J1979[1]의 방식을 사용하여 차량의 통신방식을 정의하고 이에 따라 발생하는 ECU 정보들을 수집 분석하여 각각의 고장 코드를 해석하였고 배기가스뿐만 아니라 차량에서 발생하는 총체적인 문제점들을 GUI(Graphic User Interface) 기반의 응용 프로그램을 이용하여 차량의 단계별, 부품별 고장코드를 실시간으로 발생시킬 수 있는 시뮬레이터를 개발하였다

1. 서 론

자동차 배출가스의 규제가 더욱 엄격해짐에 따라, 일반 도로에서 주행하는 차량들에 대해서 배출가스를 검사하기 위해서는 더욱 더 정밀한 장비가 필요하고 또한 많은 시간이 소요하게 된다. 따라서 세계적으로 정의된 OBD-II 코드는 자동차 로 하여금 배출가스 배출량이 규제치 이상이 되면 이를 운전자에게 알릴 수 있도록 법제화 한 것으로, 차량에 내장된 컴퓨터(On-Board Computer)로 운행 중 배출가스 제어부품이나 시스템을 감시하여 고장이 진단되면 운전자에게 이를 알려주는 시스템을 장착하도록 한 규정이다.[1] 또한, OBD-II에 관련된 세부항목 중 고장코드를 나타내는 DTC(Diagnostic trouble code)는 자동차 전자제어 시스템의 자기진단으로 고장이 발생하면 고장상태에 대한 내용을 코드로 정리한 것으로 OBD-I에서는 각 자동차 생산회사에서 별도의 고장진단 내용과 코드를 사용하였으나, 대기환경 보호를 목적으로 자동차 배출가스의 규제가 엄격해 지면서 정비 편의성과, 진단기기의 표준화, 자동차 사용자의 정비 불만으로 인한 불만 등을 제거하기 위하여 OBD-II를 법으로 제정하면서 표준화된 코드를 DTC라고 한다. 그리고 차량의 현재 상태를 알 수 있는 중요한 정보인 Freeze Frame은 차량의 ECU에서 올라오는 각종 데이터들 가운데, OBD-II고장 지시등이 점등이 될 때 그 때의 조건을 이해할 수 있도록 한 것으로 고장을 수리하기 위해 필요한 각종 입출력 계통의 변수를 기억시켜 볼 수 있도록 한 것이다. 본 연구에서는 실 차량의 OBD-II코드를 범용화된 차량 진단틀(Diagnostic Tool)을 사용하여 데이터를 취득 하였고, GUI응용프로그램으로 시뮬레이션하여 실 OBD-II코드와 동일한 코드를 발생시켰으며, 정비소에서 고장을 수리하기 위해 필요한 정보를 알 수 있도록 차량상태와 관련 센서측정값 등 고장 발생 항목을 구체적이고 개별적으로 DATABASE화 시켜 GST(Gnentic Scan Tool : 일반적 스캐너)와 통신이 가능하도록 구현 하였다.

2. OBD MESSAGE STRUCTURE

차량 ECU와 진단틀의 통신 관련된 프로토콜[1]은 국제 표준에 근거하여 제작된다 또한 차량의 오작동 관련내용의 진단틀의 표시 방법과 관련된 프로토콜은 대표적으로 SAE J1978, SAE J1979, ISO 14230방식이 있는데, 본 논문에서 소개하는 OBD MESSAGE STRUCTURE는 SAE(미국 자동차 기술자 협회) J1979 PWM 방식을 사용하였다.

2.1 HEADER BYTE

OBD-II메세지는 차량진단틀을 사용하여 차량 ECU(Engine Control Unit)로부터 데이터를 취득할 수 있다. OBD-II 메시지의 표준형태는 <표 1>과 같이 header, data, crc로 구분되며 총 11byte의 데이터가 저장된다.

header1	header2	header3	data1	data2	data3	data4	data5	data6	data7	crc
---------	---------	---------	-------	-------	-------	-------	-------	-------	-------	-----

<그림 1> OBD-II MESSAGE STRUCTURE

header 바이트들이 의미하는 바는 진단틀의 Request와 Response 간의 고유한 값으로써, 각각의 프로토콜마다 정의방식이 다르며, SAE(미국 자동차 기술자 협회)에서 제정한 방식이 대표적으로 사용되고 있다.

2.2 DATA BYTE

Data Byte에는 차량의 상태를 나타내어주는 여러 가지 데이터들을 표시한다. 위의 <그림 1>의 data1은 차량진단모드를 나타내는 것으로써 다음 <표 1>과 같이 9가지의 모드가 있다.

<표 1> OBD-II 진단 모드

구 분	내 용
MODE 1	차량의 속도, 온도, 각종 센서들에 대한 데이터를 얻기 위한 가장 대표적인 모드
MODE 2	모드 1과 유사하며, Freeze Frame 데이터를 표현
MODE 3	차량의 고장코드를 표현
MODE 4	고장코드와 저장된 데이터들을 정리(삭제)
MODE 5	산소센서의 테스트 결과 값
MODE 6	비 지속적인 계측값의 테스트 결과 값
MODE 7	mode3의 형태와 유사하며 미결상태인 고장코드를 표현
MODE 8	제조사사의 특수 제어 모드
MODE 9	ECU에 저장된 사용자의 차량 정보 요청 모드

data 2는 Parameter ID(PID)를 나타내는데, 진단틀이 차량의 ECU로부터 받은 각종 정보 및 데이터들을 사용자가 확인 할 수 있도록 지원 가능 여부를 결정하고, data 3~7까지는 차량의 상태 data로써 고장여부, 엔진의 상태, 냉각수 상태, 속도, 배터리 상태 등을 1byte씩 저장하여 나타낸다. 그리고 마지막의 CRC는 통신 중 오류의 발생여부를 알려준다.

2.3 FREEZE FRAME

OBD-II 고장 지시등이 점등이 될 때 그 때의 조건을 이해할 수 있도록 한 것으로 수리하기 위해 필요한 각종 입출력 계통의 변수를 기억시켜 고장(fault) 최초 발생 시 엔진제어 관련 스캐너를 통해 볼 수 있도록 한다. 연료계통(Fuel system)에 이상이 있다고 감지된 경우나 실화코드(misfire fault)가 발생한 경우에는 다른 고장코드의 프리즈 프레임에 우선하여 기록하며, 고장코드는 삭제 조건(MODE 4)에서 스캐너를 통해 삭제 가능하다. <표 3>은 FREEZE FRAME이 나타내는 항목들에 대하여 나열한 것이다.

<표 2> FREEZE FRAME

구 분	내 용
흡입 공기량 계산값	CLV = 0.01 * ((256 * A) + B)
엔진회전수(Engine rpm)	RPM = 0.25 * (A * 256 + B)
연료계통 :산소센서 피이드백량	Fuel trim = 0.7812 * (A - 128)
연료압력: 연료압력 센서가 장착된 경우	Fuel pressure[KPa] = 3 * a
차량 주행속도(Vehicle speed)	Vehicle speed = A
냉각수온(Coolant temperature)	Coolant temperature[°C] = A - 40
MAP센서 시그널	가능한 경우 해당
산소센서 피이드백 상태 (Close loop ON/OFF 상태)	oxygen srnsor = 0.005 * A
고장코드	MODE 3

2.4 DTC(Diagnostic trouble code)

이 모드는 차량에 저장되는 고장코드를 검색하는 유일한 기능이며, 데이터 바이트 2개(DATA A, DATA B)로 구성되며 알파벳을 포함하여 5자리로 구성되고 문자나 숫자의 순서에 따라 정의된다. 첫 번째 자리의 숫자의 의미는

- P = power train
- C = chassis
- B = body
- U = network communication을 나타낸다.

두 번째 자리의 숫자의 의미는

- 0 = SAE에서 표준으로 정의한 코드
- 1 = 자동차 메이커에 따라서 정의(SAE에서 표현하지 못하는 경우)
- 2 = SAE에서 표준으로 정의한 코드
- 3 = 자동차 메이커와 SAE가 같이 사용하는 코드를 나타내며,

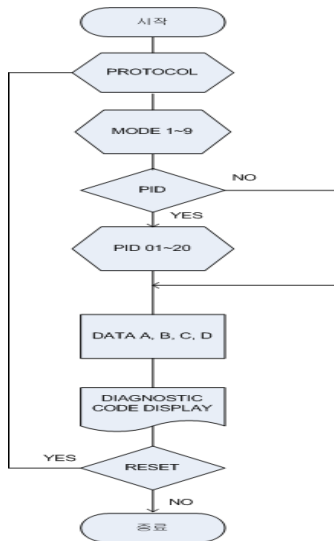
세 번째 오는 정수 또한 시스템별로 구분되며 다음과 같다.

- p00xx = fuel and air metering and auxiliary emission control
- p01xx = fuel and air metering
- p02xx = fuel and air metering
- p03xx = Ignition system or misfire
- p04xx = auxiliary emission control
- p05xx = vehicle speed, idle control and auxiliary inputs
- p06xx = computer and auxiliary inputs
- p07xx = transmission
- p08xx = transmission
- p09xx = transmission 계통으로 이루어진다.

이러한 구성은 data B에도 동일하게 적용되는데, data B의 07~04까지의 내용이 4번째 문자를 구성하고 03~00까지의 내용은 5번째 문자를 구성하게 된다. 그리고 네 번째와 다섯 번째의 숫자는 시스템이나 부품의 고장내용으로 구성된다.

3. OBD-II 차량진단 코드 발생 SIMULATOR

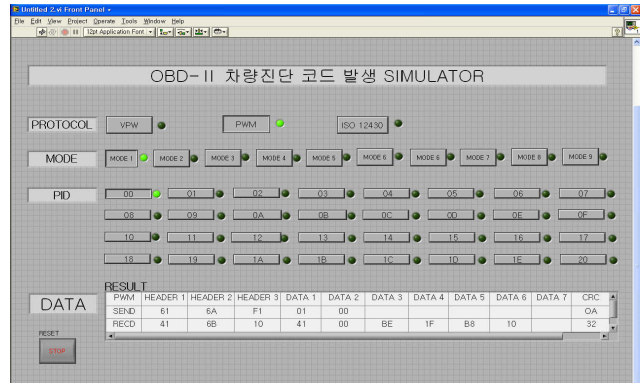
본 논문의 시뮬레이터는 실 차량의 OBD-II코드를 범용화된 차량 진단틀을 사용하여 GUI응용프로그램으로 분석하고, 실제 OBD-II코드와 동일한 코드를 발생시키며 정비소에서 필요한 정보를 알 수 있도록 차량상태와 관련 센서측정값 등 고장 발생 항목과 이를 수리하기 위해 필요한 내용을 구체적이고 개별적으로 DATABASE화 시켜 GST(Gnestic Scan Tool : 일반적 스캐너)와 통신이 가능하도록 구현 하였고, OBD-II 코드의 해석방법과 동일하게 차량의 프로토콜부터 각종 모드 및 PID를 선택하여 임의의 OBD-II코드를 생성하는 시뮬레이터를 구현하였다. 다음의 <그림 2>는 코드생성을 위하여 본 논문에서 구현한 시뮬레이터의 FLOW CHART이다.



<그림 2> OBD-II 차량진단 코드 발생 FLOW CHART

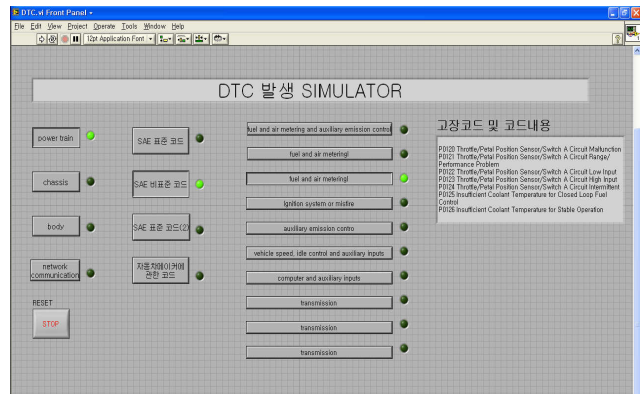
아래의 <그림 3>은 GUI 프로그램인 LABVIEW를 사용하여 구현한 시뮬레이터의 실행 화면으로써, 각각의 제어 가능한 아이콘을 사용하여 사용자가 임의의 OBD-II코드를 생성 할 수 있도록 하였으며, 추가적으

로 SAE 표준 프로토콜을 사용하는 차량은 별도의 장비 없이 통신이 가능하며, 자동으로 모드를 초기화 하는 기능을 포함 시켰다.



<그림 3> OBD-II 차량진단 코드 발생 SIMULATOR

<그림 4>는 <그림 3>의 세부항목 중 하나인 고장코드를 발생하는 기능을 시뮬레이션 한 것이다. MODE3은 차량의 DTC를 발생하는 유일한 기능으로 각종 부품 및 기계적, 전자적 결함 등을 알 수 있도록 구성 하였다.



<그림 4> DTC 발생 SIMULATOR

4. 결 론

본 논문에서는 OBD-II시스템에서의 실시간 고장코드 발생을 위하여 OBD MESSAGE STRUCTURE에 따라 차량 진단 모드를 단계적으로 구현하였다. 그 결과 차량의 단계별, 부품별 고장코드를 실시간으로 생성할수 있으며, 이러한 데이터들은 사용자 및 자동차 제작회사에서 유용하게 사용될 수 있음을 증명하였고, Freeze Frame 발생으로 현재의 차량상태를 명확하게 판단할 수 있다. 또한 본 논문에서는 OBD의 차량진단 코드 발생 시스템을 대표적 GUI프로그램인 LabVIEW를 이용하여 OBD-II 시스템의 차량의 현재 상태 및 고장상태 분석 시뮬레이터 구현하여 이를 기반으로 추후 필요한 기능추가가 용이할 것으로 사료된다.

The research was supported by the Driving Force Project for the Next Generation of Gyeonggi Provincial Government in republic of Korea

[참 고 문 헌]

[1] 이봉우, "OBD-II(배기가스)", 경영사, 2005
 [2] Wolfhard Lawrenz, "CAN System Engineering From Theory to Practical Application", Springer, 1997
 [3] Rami Baroody, Nizar Al-Holou and Asif Rashid, "Dynamic Discovery Service Protocols for Next Generation Vehicle Network", SAEinternational SP-1852, SAEinternational, pp5~12, March 8-11, 2004
 [4] 장종관, "자동차 전자제어 공학", 보성각, 2003