

블루투스를 이용한 안드로이드 차량관리 시스템 구현

원동옥*, 김주현**, 안준**, 최종필*, 최진구*

한국산업기술대학교 컴퓨터공학과

*e-mail: {wonner, jpchoi, jkchey}@kpu.ac.kr

**e-mail: {shullaa, dkswnsanjun}@naver.com

Implementation of vehicle management System Using Bluetooth

Dong-Ok Won, Ju-Hyouen Kim, Jun An, Jongpil Choi Jingu Choi

Dept of Computer Engineering, Korea Polytechnic University

요 약

본 논문에서는 안드로이드 기반의 스마트폰에서 운전자의 운전 패턴에 따른 연료 소비량을 감소하기 위한 운전 습관을 유도하는 방법을 제안하였다. 이 시스템은 차량의 정확한 내부 상태 정보를 바탕으로 운전 상태를 스마트폰으로 제공한다. 차량의 상태정보를 추출하기 위해서 차량 네트워크인 CAN 버스를 사용하였으며, 이를 하드웨어로 구현하였다. 효율적인 ECO_Driving을 위한 차량 데이터들을 추출하여 분석하였으며, 이들 데이터를 바탕으로 운전자에게 예코여부를 제공하는 애플리케이션을 구현하였다. 이 시스템으로 경제운전으로 배기가스 감축, 에너지절약과 사고 예방을 유도할 수 있다.

1. 서론

우리나라 스마트폰 가입자 수가 전체 이동통신 사용자의 1/3 수준인 1500만 이상이 스마트폰을 사용 중이고, 올 연말경에는 절반인 2500만을 넘어서 최근에는 길거리나 대중교통 등에서 한 손에 스마트폰을 들고 있지 않은 시민들을 찾아보기 힘들 정도로 스마트폰의 대중화가 빠르게 진행되고 있다.

특히 스마트폰 보급이 확대되면서 교통, 금융, 교육, 엔터테인먼트 등 다양한 형태의 모바일 애플리케이션들이 넘쳐나면서 웹 정보검색 또는 웹서핑 활용률도 88%에 달할 정도로 보편화되고 있다. 스마트폰이 이동통신 도구나 단순 웹 검색기기를 넘어서 생활전반에 없어서는 안 될 툴로 자리 잡은 것이다.

반면에 우리나라의 자동차 등록대수는 2011년 6월만 기준으로 1800만대를 넘어섰다. 자동차 1대당 주민등록인구수는 2.77명으로 나타났다. 이에 차량의 에너지의 효율성을 개선하기 위해서 많은 노력들로 연비는 점점 좋아지고 있다. 또한 전기차가 보급되면서 연비가 급격히 좋아지고 있는 추세다. 이러한 상황에서 운전자의 운전 습관에 따라 연비의 차이가 발생하게 되는 것에 주목했고, 그에 따라 ECO-Driving이라는 최적의 운전 방법을 유도한다.

본 논문에서 제안하는 안드로이드 기반의 스마트폰에 ECO 애플리케이션으로 운전자의 운전 패턴을 기록하는 방법으로 연비 개선과 차량 수명연장을 유도하는 것으로 차량에서 정보를 수집하기 위해서 차량내의 장착되어 있는 CAN 통신을 통해서 데이터를 수신 받는다.

CAN BUS를 사용하여 수집된 정보 및 데이터는 블루투

스를 통해 ECO-Driving을 위한 운전 유도 애플리케이션과 연동되어 데이터를 분석하고 자동차 상태를 파악할 수 있는 시스템을 개발하였다.

2. 관련 연구

2.1 CAN 메시지 프레임

CAN 통신에서 데이터는 메시지 프레임을 사용하여 송수신하여 메시지 프레임은 하나 또는 그 이상의 송신 노드로부터 데이터를 수신노드로 이동한다. CAN 통신은 2.0A(표준)와 2.0B(확장)으로 지원되며 대부분의 2.0A CAN Controller는 표준 CAN 방식을 사용하나 2.0B CAN Controller는 표준 또는 확장 방식 모두 사용하여 데이터를 송수신 할 수 있다.

- 2.0A CAN 메시지 구조 (표준)

프레임 시작 필드(SOF Start Of Frame) 필드 메시지 프레임의 시작을 표시하며 메시지 프레임의 최우선에 위치하며 디폴트 "0" 값을 가진다.

중재 필드(Arbitration Field) 11 비트의 식별자와 원격 전송 요구(RTR) 비트를 가지며 디폴트 "0"을 가지는 RTR 비트는 비트 값이 "0" 일 때 CAN 메시지가 데이터 프레임이라는 것을 나타낸다. 역으로 RTR 비트 값이 "1" 이면 CAN 메시지가 원격전송요청(RTR : Remote Transmission Request)을 의미한다. 다시 말해 CAN 메시지가 데이터 프레임이 아닌 원격프레임(Remote Frame) 상태임을 나타냅니다. 원격 프레임은 데이터 버스상의 어떤 한 노드로부터 다른 노드로 데이터를 전송 요청할 때

사용된다.

데이터를 보내기 전에 사용되는 메시지 프레임이기에 데이터 필드를 포함하지 않는다.

제어 필드(Control Field) 6 비트로 구성되며 향후에 사용되기 위해 예약된 두 개의 "0"의 값을 가지는 R0, R1과 데이터 필드의 바이트 수를 가리키는 4 비트의 데이터 길이 코드(DLC:Data Length Code)로 구성된다.

데이터 필드 (Data Field) 한 노드로부터 다른 노드로 전하고자 하는 데이터를 포함하며 0~8 바이트로 구성됩니다. CRC 필드(CRC:Cyclic Redundancy Check) 17 비트의 주기적 중복확인(CRC) 코드를 가지며 데이터필드의 끝을 알리는 "1"의 값을 가지는 비트가 이어진다.

ACK 필드(ACKnowledge Field) 2 비트로 구성되며 첫 번째 비트는 "0" 값을 가지는 Slot 비트이다. 그러나 메시지를 성공적으로 수신한 다른 노드로부터 전송된 "1"의 값으로 기록될 수 있다. 두 번째 비트는 "1"의 값을 가진다. 프레임 종료 필드(EOF:End Of Frame Filed) 7 비트로 구성되며 모두 "1"의 값을 가진다. EFO 뒤이어 모두 "1"의 값을 가진 3 비트의 프레임 중단 필드(INTermission Field)가 이어진다. 3 비트의 INT 주기 이후에 CAN 버斯拉인은 자유 상태로 인식된다.

이후 버스 Idle Time 은 "0"을 포함한 임의의 길이이다.

- 2.0B(확장) CAN 메시지 구조

2.0A 와 구분되기 위해 29 비트의 메시지 프레임 식별자를 가집니다. 기존에 사용 중인 2.0A 와 호환되는 2.0B 로 발전되었다.

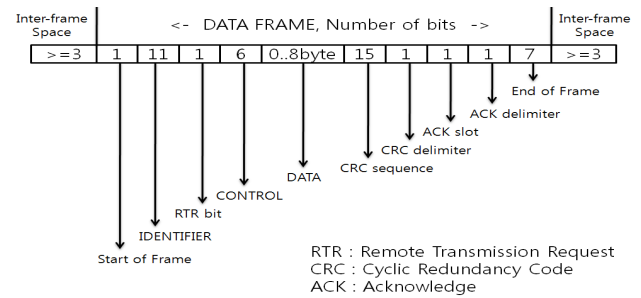
2.0A와 차이점은 중재 필드가 두 개의 CAN 메시지 식별자로 구분되어 포함되며 첫 번째 (기본 ID)는 11 비트 길이로 2.0A 와 호환되게 하며 두 번째 필드 (확장 ID)는 18 비트 길이로 ID 는 총 29 비트로 구성됩니다. 두 개의 ID 필드 사이에 ID 확장자(IDE Identifier Extension)가 있어 두 개의 ID 필드를 구분한다.

SRR(Substitute Remote Request) 비트는 중재 필드에 속해 있으며 표준 데이터 프레임과 확장 데이터 프레임을 중재해야 하는 경우에 대비하기 위해 항상 "1"의 값을 전송한다.

만약 표준 데이터 프레임과 확장 데이터 프레임이 같은 기본 ID (11 비트)를 가지고 있으면 표준 데이터 프레임이 우선순위를 가진다. 그리고 2.0B 메시지 프레임에 있는 다른 필드들은 표준 메시지 포맷으로 식별된다. 2.0B Controller 입장에서 2.0A와 송수신에 있어 완벽하게 호환이 된다. 2.0A Controller 는 두 가지 경우가 있다. 1) 첫째는 2.0A Format 의 메시지만 송수신이 가능한 경우이며 2.0B 의 메시지는 에러를 발생시킨다. 2) 두 번째는 2.0B Passive 로 알려져 있으며 2.0A 의 메시지는 송수신 가능하고 더불어 2.0B 의 메시지는 식별을 하여 무시해버린다. 그 결과 2.0A 와 2.0B 는 하나의 CAN 네트워크상에서 함께 사용할 수 있다.

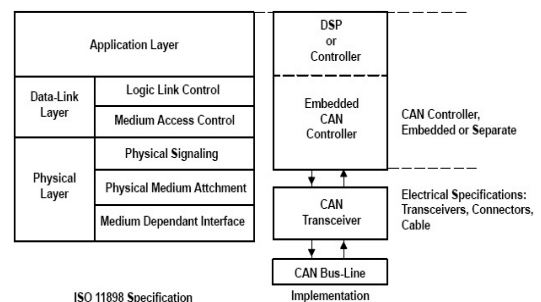
2.2 CAN의 계층 구조

CAN은 표준 네트워크 모델인 OSI 7 Layer중 최하위 2계층인 물리계층(Physical Layer)과 데이터 링크 계층(Data Link Layer)만으로 구성되어 있다.



(그림 1) CAN 메시지 프레임

그림 2는 전체 시스템으로 물리계층, 데이터 링크 계층과 응용계층(Application Layer)으로 구성되며 CAN을 이용한 시스템 설계 시 적용 대상을 고려하여 적절한 응용계층을 함께 설계해야 하는 특징을 가지고 있다. CAN에 의해 데이터가 교환될 때에는 주소화되지 않고 통신에 유일하게 존재하는 메시지 식별자(Identifier)에 의해서 메시지의 데이터를 교환된다. 즉 데이터의 수신지를 정하지 않고 메시지 ID를 이용하여 그 데이터의 사용처를 정해지며, 물리계층은 노드간의 직접적인 통신을 담당하는 부분으로 CAN_low, CAN_high 두 가닥의 선로만으로 통신을 하므로 복잡한 네트워크 선로에 대한 부담을 줄일 수 있다. 데이터 링크 계층은 전달 계층과 객체 계층으로 구분되는데 전달 계층은 CAN BUS상의 메시지 전송률 비트타이밍, 메시지 캡슐화 및 중재, 에러 감지 등의 역할을 수행하며, 객체 계층은 전달 계층으로부터 받은 메시지의 식별자(Identifier)를 통해 필터링 하고 핸들링 한다.



(그림 2) ISO 11898 표준 구조

2.3. 개발환경

본 논문의 애플리케이션은 윈도우 7환경에서 개발되었다. Java SE Development Kit (JDK6) 설치, Eclipse IDE 설치, Eclipse IDE 에 Android Plug-in 설치, Android SDK 2.2 설치 후 실제 단말기와 에뮬레이터에서 개발되었다. 개발에 사용된 단말기는 HTC에서 개발한 'Nexusone'이다. Wi-Fi, 멀티터치 등 본 애플리케이션이 작동하기에 충분한 기능을 갖추고 있기에 선택하였다.

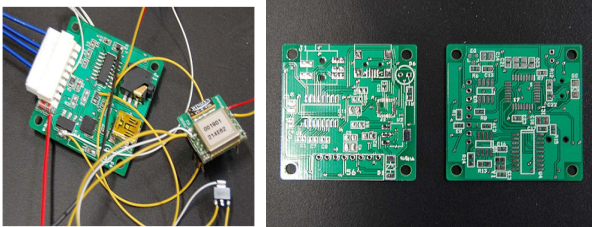
3. 세부 설계 및 구현

ECO-Driving 시스템의 설계의 핵심 기법은 크게 3가지로 구분된다. 첫 번째로는 차량 네트워크를 통해 각각의

제어장치로부터 데이터를 수신 받는 모듈 구성이며 두 번째로는 수집된 정보를 가공하여 무선통신(Bluetooth)을 통해 운전자의 스마트폰으로 데이터 전송이고 세 번째로 운전자의 운전습관과 패턴 등을 인식하여 최적화 운전기법을 유도한다.

3.1 CAN 수신 모듈

차량 정보를 수집을 하기 위해서는 OBD-II 커넥터를 이용하여 CAN 신호를 받아서 처리한다. OBD-II 커넥터는 총 16개의 핀으로 구성되어 있으며, 실제로 사용되는 핀은 9개 정도이며, 나머지 7개의 핀은 나중에 위해서 만들어둔 것이다. 본 시스템에서는 CAN 통신을 하기 위한 핀 6번 CAN_H와 14번 CAN_L을 사용하여 데이터는 전송 받는다. CAN 통신을 위해서 사용되는 CAN Interface는 Atmel사의 AT90CAN128을 사용하였으며, 내부에 CAN Controller가 들어가 있으며, CAN 모듈은 메시지에 관련된 모든 정보를 저장 하는 메시지 객체라는 것을 포함하고 있다.



(그림 3) 개발된 CAN 수신 모듈

우선 CAN 통신을 하기 위해서 CAN BUS에서 OBD-II 커넥터를 이용하여 CAN Transceiver인 PCA82C250T칩으로 보내진다. CAN Transceiver는 CAN BUS의 디지털 전압 레벨을 AT90CAN128에 맞게 바꿔주는 역할을 한다. AT90CAN128에서는 받은 데이터 중 필요한 데이터만 받아서 가공하여 스마트폰으로 전송하는 역할을 담당한다. 그림 3은 실제 제작하여 PCB된 CAN 수신 모듈을 보여준다.

3.2 핵심 정보

CAN 수신 모듈에서 필요한 데이터만을 가공하여 단말장치인 스마트폰으로 전송하게 된다. 필요한 데이터만을 CAN 수신 모듈로부터 받으며, 받은 데이터는 2차적으로 가공하여 운전자에게 출력한다. 다음의 데이터들을 CAN 수신 모듈로부터 받아서 가공한다.

CAN 수신 모듈에서 들어오는 데이터는 엔진의 속도 즉 RPM, 차량의 속도, ASP(Acceleration Pedal Position), 100ms당 연료 소비량, Turbine 속도, 브레이크 상태, Vehicle Road Load Constant, Engine Output Torque, Tubing Torque, Vehicle Weight의 데이터를 받아 2차 가공을 하여 운전자에게 필요한 데이터로 표현하며, 가공된 데이터들의 정보는 다음과 같다.

- Vehicle Acceleration은 초당 차량의 가속도를 의미하며 차량 정지력, 주행저항을 구하기 위해 사용된다. 차량

가속도를 Veh_{Accel} 라고 할 때 다음 식을 정의 할 수 있다.

$$Veh_{Accel} = (CarSPD_{Cur} - CarSPD_{Pre}) / dTM$$

이때 $CarSPD_{Cur}$ 는 현재의 차량 속도이고 차량의 네트워크를 통해서 받은 데이터이다. $CarSPD_{Pre}$ 는 dTM 시간 이전에 읽혀진 차량의 속도이며, dTM 은 100mSec를 의미한다.

- Turbine Efficiency는 자동 변속기 내의 토크 컨버터(Torque Converter)의 동력 전달 효율을 의미하며 수동 변속기 대비 약 10%의 손실이 발생하는데 이는 급가속등으로 더욱 악화된다. 효율 η 는 다음의 식으로 정의 할 수 있다.

$$\eta = T_i \times S_i / T_o \times S_o$$

T_i 는 토크 컨버터 입력토크의미하고, T_o 는 토크 컨버터 출력 토크로 정의되며, S_i 는 토크 컨버터 입력의 속도로 즉 엔진 회전 속도 RPM이고, S_o 는 토크 컨버터 출력 속도로 정의된다.

- Brake Force은 브레이크 조작으로 인해 발생된 차량의 제동력을 의미하며 차량의 제동력 Brk_{Force} 은 다음과 같이 정의 할 수 있다.

$$Brk_{Force} = (Veh_{Accel} - RL_{Accel}) \times Veh_{WET} / 3.6$$

Veh_{Accel} 은 차량의 총 감속도(KPH/s)를 의미하며, RL_{Accel} 은 주행 저항에 의한 차량 감속도(KPH/s)이고, Veh_{WET} 는 차량의 질량(kg)으로 CAN 수신 모듈로 받은 차량 데이터이다.

- Accumulated Fuel Consumption은 차량의 주행으로 소모된 누적 연료량을 의미하면 운전자에게 연비를 눈으로 볼 수 있는 값이며 평균 연비를 계산하기 위해서 필요한 데이터로 Acc_{Fuel} 는 다음과 같이 정의 할 수 있다.

$$Acc_{Fuel} = FuelCon_{pre} + FuelCon$$

$FuelCon_{pre}$ 은 이전부터 계속된 누적 연료 소모량(ℓ)을 의미하며 $FuelCon$ 은 100mSec동안 측정된 구간 연료 소모량(ℓ)이다.

- Average Driving Efficiency는 차량의 평균 연비를 의미하며, 차량의 속도로부터 계산된 거리와 측정된 연료량을 이용하여 아래와 같은 식으로 정의 되었다.

$$AvgDRV_{Effi} = Acc_{Dist} / Acc_{Fuel}$$

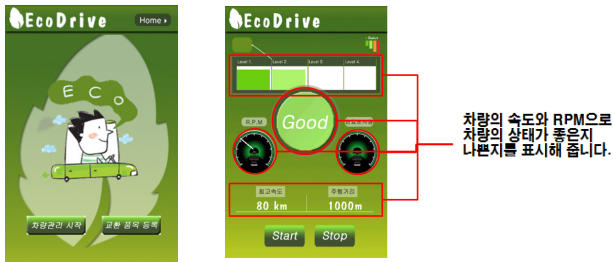
Acc_{Dist} 는 차량의 총 누적 주행 거리를 나타내고, Acc_{Fuel} 는 위에 차량의 연료 소모량을 의미한다.

단말장치에서는 위에 식을 이용하여 운전자에게 필요한 정보를 제공하는 역할을 한다. 실제 미디어장치에서의 구현은 엔진속도, 차량 속도, 연료소모량, 평균연비를 표현하였으며, ECO-Driving 모드를 구하기위해서도 여러 데이

터를 사용된다.

3.3 ECO 기능

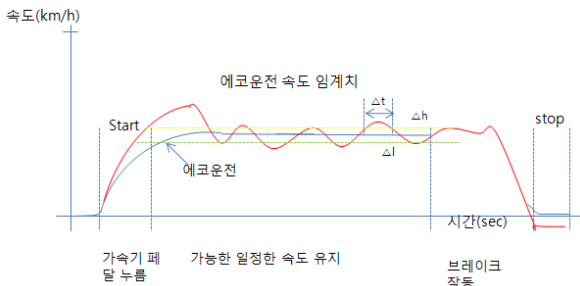
ECO 기능은 운전자가 ECO-Driving을 하는지를 판단하는 것으로 가속도 페달의 량을 조사한다. 가속도 정도는 CAN 데이터에서 수집하며, 일반적으로 가속 페달의 조작량이 차량정보에 포함된 차량속도에 대응되게 미리 설정하고 가속 페달의 조작량이 임계범위에 속하는지 여부를 토대로 에코 운전을 하고 있는지를 판별한다.



(그림 4) ECO 모드 활성화

그림 4는 주행 속도, RPM, 연료 분사량의 관계를 토대로 가속성을 높이기 위해 Warning, Bad, Good, Best로 네 레벨로 나누어 Eco여부 정보를 제공하고 판단하게 했다. 실제 운전에서의 속도와 가속도 페달의 변화는 매우 민감하므로 일정 편차(10%~15%)를 두고 에코모드의 임계치 값으로 판별한다.

에코 모드인지를 판단하는 측정치는 시간에 대한 속도의 변화량이다. 이 속도의 변화량도 마찬가지로 변화에 대해 덜 민감하도록 일정한 시간을 두어 임계치를 이상 또는 이하로 지속되어야 한다. 속도 정도는 CAN 데이터에서 수집하며, 속도에 따라서 에코 모드 표시는 그림 6과 같다.



(그림 5) 시간에 따른 속도

에코모드에 대한 임계치 값을 기준으로 녹색 실선과 노란색 실선으로 구분된다. 빨간색선의 경우 실제 운전하면서 발생할 수 있는 속도의 변화량이며 일정 시간동안 속도가 임계범위 안에 들어와 있으면 에코모드로 판별한다.

3.4 차량 관리 기능

차량 관리 기능은 교환품목을 등록할 수 있고 교환품목 목록에서 품목, 점검주기, 교환주기, 현재사용(교체유무) 등을 확인 할 수 있다.



(그림 6) 차량관리 모드 활성화

3.5 차계부 기능

차계부 기능은 차량을 소유하면서 지출되는 금액을 담을 수 있는 기능이다. 주유비, 차량유지비, 정비내역, 고정비, 범칙금, 기타 등 지출되는 금액의 일정 정보를 기간별로 확인 할 수 있다.



(그림 7) 차계부 모드 활성화

4. 결론 및 향후 연구과제

본 논문에서 제안하는 안드로이드 기반의 스마트폰용 ECO 시스템으로 운전자의 운전 습관을 변경하도록 차량 상태 정보, 연비 정보 등을 제공하는 것을 구현하였다. 이 Eco-Drive 시스템은 차량의 정확한 상태 정보를 바탕으로 운전 상태를 운전자에게 제공하여 경제운전을 하도록 유도하였다. 차량의 상태정보를 추출하기 위해서 차량 네트워크인 CAN 버스를 사용하였으며, 이를 하드웨어로 구현하였다. 효율적인 Eco-Drive를 위한 차량 데이터들을 추출하여 분석하였으며, 이들 데이터를 바탕으로 운전자에게 에코여부를 제공하는 스마트폰용 애플리케이션을 구현하였다. 향후 연구에서는 운전자의 운전습관과 자동적인 차량제어, 실시간 도로교통상황을 고려하여 우회를 유도하는 기능 등을 추가할 계획이다.

참고문헌

- [1] BOSCH, CAN Specification, Part A/B, 1991
- [2] Road vehicles, Interchange of digital information - Controller area network (CAN) for high-speed communication, ISO11898, 1993.
- [3] Road vehicles, Low-speed serial data communication - Part 2: Low-speed Controller area network (CAN), ISO11519-2, 1994.