

CAN 통신을 이용한 차량 내 자동 온도조절 시스템

김장주* · 장종욱**

*동의대학교 컴퓨터공학과

In-Vehicle Auto temperature control System by CAN Network

Jang-ju Kim* · Jong-wook Jang**

*Department of Computer Engineering, Dong-Eui University

E-mail : jangju@deu.ac.kr* jwjang@deu.ac.kr**

요 약

최근 차량용 네트워크 시스템으로 사용되고 있는 CAN(Controller Area Network)은 많은 ECU들이 필요한 미래형 스마트차량에 적합한 네트워크 프로토콜로서 안정성과 신뢰성을 보장해주며, 많은 ECU들의 장착으로 Wiring Harness의 공간과 중량이 늘어남으로 인해 발생하는 에너지 소비와 비용의 증가를 대폭 줄일 수 있는 것으로 나타났다.

본 논문에서는 CAN프로토콜을 이용하여 미래형 스마트 자동차에 요구되는 편의주행, 쾌적주행을 위해 Air conditioner 와 Heater를 제어하여 차량 내부 온도를 운전자의 요구에 맞도록 자동으로 제어할 수 있는 시스템을 구현하고자 한다.

ABSTRACT

Recently, CAN(Controller Area Network) being used in vehicle network system is suitable Network Protocol for smart vehicles with a future that need many ECUs, and it guarantees stability and reliability. It is revealed that being equipped many ECU could reduce the increasing of energy consumption and energy cost from the increasing of Wiring Harness's space and weight.

In this paper, future smart vehicle control Air conditioner and heater for convenient and comfortable driving as using CAN protocol and implement auto control system According to driver's requirement using temperature in the vehicle.

키워드

CAN(Controller Area Network), ECU(Electronic Control Unit), In-vehicle network

1. 서 론

CAN(Controller Area Network)은 차량용 네트워크 시스템으로 현재 많은 연구와 개발이 이루어지고 있으며, 일부 차량에 도입되어 사용되고 있다. 미래형 스마트카에는 고객의 요구를 만족시키기 위하여 차량 내에 여러 가지 전기, 전자적 장치들을 필요로 하게 되어 많은 수의 ECU(Electronic Control Unit)가 장착되게 되면서 여러 가지 문제가 발생 되었다. 차량 내에 분산되어 있는 ECU들을 모두 연결하기 위한 통신 선로가 크게 증가하여 설치비용이 늘어날 뿐만 아니라 시스템이 복잡하게 되어 유지, 정비 및 확장이

어렵고 Wiring Harness가 상당히 복잡하게 되었다. Wiring Harness가 차지하는 공간과 중량의 증가로 인해 차량의 성능을 저하 시키고 에너지 소비를 증가시키는 악영향을 가져오게 된다. 이로 인하여 중량과 공간을 덜 차지하고, ECU간에 빠르고 안정적인 통신방법이 필요 하게 되었으며, 이를 충족 시켜주기 위해 CAN시스템이 개발되었다. 이로 인해 전선사용의 감소뿐 아니라, 제어 및 고장의 진단을 용이하게 하고, 성능과 안정성의 획기적인 개선을 가져 왔다.[1] 이제 차량은 고객의 요구를 충족시키면서 성능의 저하는 최소화 할 수 있게 되었다. 미래형 차량은 운전자에게 편안하고 쾌적한 환경을 제공함과 동시에 안정성

도 보장해주어야 한다. 운전외의 다른 시스템들은 자동으로 처리 해주어 운전자가 운전에만 집중할 수 있도록 다른 불필요한 행동을 최소화함으로써 안정성을 보장하고, 다른 시스템들은 자동으로 처리함으로써 편리성도 제공해 준다. 본 논문에서는 운전자의 편안하고 쾌적한 주행을 위하여 차량의 Heater와 Air conditioner를 온도센서와 CAN통신을 통해 연결하여 자동으로 컨트롤 차량 내 온도를 일정수준으로 유지 시켜주는 시스템을 실험하였다.

본 논문의 구성은 서론에 이어 2장에서는 CAN의 개요와 구조, CAN의 통신방법에 대해 기술하고 3장에서는 온도제어 시스템의 구성을 설명하고 마지막 4장에 결론을 맺는다.

II. C A N(Controller Area Network)

2-1. CAN개요

CAN은 차량내의 ECU들의 통신을 제공하기 위하여 개발되었다. CAN은 다른 필드버스 프로토콜에 비하여 가격 대 성능비가 매우 우수하고, 여러 반도체 제조회사에서 CAN프로토콜이 포함된 일체형 마이크로 컨트롤러를 개발하여 널리 사용되고 있다. CAN은 기본적으로 단일 버스마스터가 따로 없으며, 모든 장치가 마스터가 되는 멀티마스터 구조를 가지고 있다. 브로드캐스팅 방식의 통신을 이용하므로 어떤 장치에서 보내는 메시지는 모든 장치가 동시에 수신하는 것이 가능하다. CSMA/CD+AMP 방식을 사용하며, 최고 1Mbps의 속도로 데이터 전송이 가능하다.[2]

2-2. CAN 구조

CAN시스템에서 전송은 Message Frame을 통하여 이루어지며, 5개의 프레임(frame)으로 구성되어 있다. 5개의 프레임은, 데이터 프레임, 원격 프레임, 에러 프레임, 오버로드 프레임, 인터 프레임 스페이스이며 그 기능은 아래 (표 1)과 같다.

표 1. CAN 프레임의 종류와 기능

종류	기능
Data Frame	일반적인 data를 전송하기 위해 구성된 Frame.
Remote Frame	다른 노드에게 Data전송을 요청할 때 사용.
Error Frame	에러가 감지될 경우 에러발생을 알리기 위한 Frame.
Overload Frame	Data 전송 시 각 Frame 사이에 지연이 필요한 경우 사용.
Inter Frame Space	Frame간의 구분의 위해 사용.

CAN 프로토콜은 표준 CAN(2.0A)와 확장 CAN(2.0B)의 두 가지 형태의 Message Format을 지원한다. 두 종류의 Message는 ID의 길이로 구

분하며, 표준 CAN은 11bit의 ID를 확장 CAN은 29Bit의 ID길이를 가지게 되며, 표준 CAN은 확장 CAN의 인식이 불가능 하나 확장 CAN은 표준 CAN Message의 인식이 가능하다.

CAN message의 data frame은 7개의 서로 다른 필드로 구성되어 있다.

표 2. CAN Message Format

SOF	Arbitration Field	Control Field	Data Field	CRC Field	ACK Field	EOF
-----	-------------------	---------------	------------	-----------	-----------	-----

SOF(Start Of Frame)는 프레임의 시작을 표시하며, Arbitration Field는 표준에선 11Bit의 식별자를 확장 CAN에서는 29Bit의 식별자를 가지며, 이 필드의 RTR Bit로 리모트 프레임과 Data 프레임 구분할 수 있다. Control Field는 4Bit의 DLC와 2Bit의 Reserved로 구성되고, 다음으로 8Byte의 Data가 오게 된다. CRC Field는 CRC Sequence와 CRC Delimiter로 구성되며 Frame의 CRC code를 전송한다. 다음으로 성공적인 전송을 알리는 ACK Field와 마지막으로 Frame의 끝을 알리는 EOF(End Of Frame)가 오게 된다.

리모트 프레임은 Data Frame에서 Data Field만 없는 것으로 RTR Bit가 1일 경우이다. 에러 프레임은 에러를 감지한 모든 노드에서 생성할 수 있으며 Error Flag와 Error Delimiter로 구성된다. 에러의 상태에 따라 Flag에 채워지는 값은 달라진다. 오버로드 프레임은 노드 내부 상태에 의해 더 이상 Frame을 받을 수 없는 경우 발생시키는 프레임으로 다른 노드들이 Bus를 사용 못하게 하여 일시적으로 Bus를 안정화 시킨다. 인터 프레임 스페이스의 역할은 Frame과 Frame사이를 구분시켜 준다.[3]

2-3. CAN 통신 방법

CAN은 다중통신망(Multi Master Network)이며, CSMA/CD+AMP(Carrier Sense Multiple Access/Collision Detection with Arbitration on Message Priority) 방식을 이용한다. 통신을 위해서는 먼저 CAN 노드(ECU)들은 Bus가 사용 중인지 파악하고, Bus가 비어있을시 Data를 전송한다. 메시지의 충돌이 일어났을 경우, 우선순위에 의해 점유가 결정되는데 이것은 Arbitration Field의 식별자가 낮은 것이 우선순위를 가지게 된다. 우선순위가 낮은 노드들은 수신모드가 되고, Bus가 다시 사용이 가능할 때 까지 전송을 재시도하지 않는다. (그림 1)은 일반적인 Bus중재의 예를 나타낸 것이다.[4] CAN Bus에서는 '0'을 Dominant Bit로 사용하기 때문에 Bus상에서 '0'과 '1'이 동시에 존재 할 때는 Recessive Bit인 '1'은 무시된다. 따라서 (그림 1)에서 Node2가 다른 Node들 보다 우선순위가 높기 때문에 Node2가 송신한 Data Frame만 Bus상에 남아 전송을 하게 되고, 나머지 Node들은 전송을 중단하고 수신모

드로 들어가게 된다.

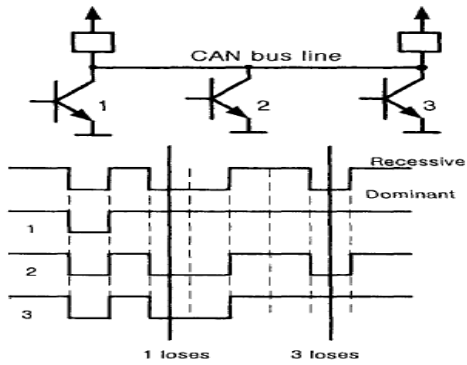


그림 1. CAN의 일반적인 Bus중재

III. 온도제어 시스템의 구성

3-1. Hardware 구성

본 논문에서는 Integrated CAN Controller 타입인 Atmel사의 AT90CAN128을 사용하였다. AT90CAN128은 RISC와 파이프라인 처리기술을 기반으로 프로그램 메모리와 메모리를 접근하는 버스를 독립적으로 사용하는 하버드 구조의 8비트 Micro Controller인 ATmega128 제품에 CAN Controller를 부가한 형태의 Micro Controller이다.[3] 온도감지 ECU는 eMAX_Kt2212_S1 CAN Control Board에 AT90CAN128모듈과 LCD, 온도 센서를 보드에 장착하여 그림 2와 같이 구성하였다.

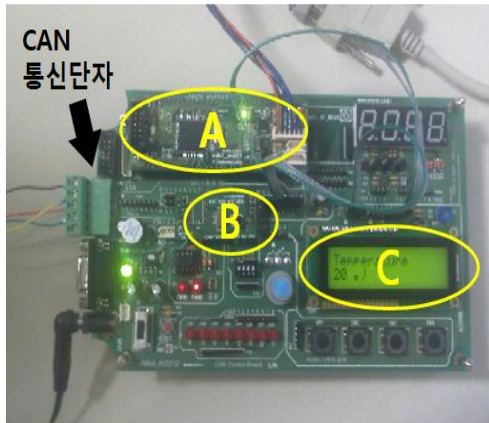


그림 2. 온도제어 ECU 구성도

A는 AT90CAN128 모듈이며, B는 온도센서, C는 LCD로 구성되어 있으며, 왼쪽에 보이는 CAN 통신용 단자를 통해 CANoe와 통신을 하게 된다. eMAX_Kt2212개발보드는 국내 기업인 맥산에서 생산하여 판매하는 보드로 AT90CAN128의 확장 핀과 FND, CAN통신 포트, LED등이 내장되어

있고, 기본적인 CAN통신을 실험 할 수 있다. 장착시킨 온도센서에 인위로 온도를 변화시켜 그 값을 CAN Message형태로 만들어 CAN Bus에 뿌려지게 되고, 이 값을 기반으로 CANoe의 가상 Heater와 Air conditioner를 동작시키게 된다. ECU와 CANoe의 정상적인 통신을 위해서는 CAN_High와 CAN_Low 및 GND, Vcc를 맞추어 주고 저항을 달아줘야 CAN통신이 가능하다.[5]

3-2. Software 구성

본 논문에서 실제로 구현하지 못한 CAN ECU 들은 Simulation Tool중 하나인 CANoe를 사용하여 가상으로 만들어 테스트 하였다. CANoe는 ECU개발하거나, 개발된 ECU를 테스트 할 때 사용할 수 있다. CANoe는 Real모드(실제 ECU들을 연결하여 통신 정보를 모니터링), Simulation 모드(PC상에 가상으로 ECU를 제작하여 테스트), Real/Simulation모드(실제 ECU와 가상의 ECU의 연결 통신 테스트)를 지원해 준다. 이번 실험에서는 Real/Simulation모드를 사용하였고, 온도감지 ECU는 실제구현하고, Heater와 Air conditioner는 (그림 3)과 같이 가상으로 만들어 테스트 하였다.

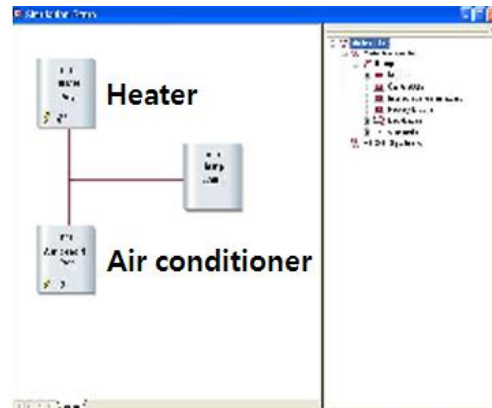


그림 3. CANoe상에 구현된 가상ECU

온도제어 ECU는 현재 온도를 LCD창에 표시해주는 것과 동시에 온도가 적정온도보다 낮거나 높으면, 가상의 ECU를 제어하는 Message를 보내게 된다.

온도제어ECU에서 현재온도 상태에 따라 가상의 Heater와 Air conditioner의 ID를 기억하고 있다가 On/Off 명령을 Data Filed를 통해서 보내게 된다. 가상의 Heater와 Air conditioner는 온도제어 ECU에서 나오는 Data중에 자신의 ID에 맞는 값을 받아들여 On/Off동작을 하게 된다. CANoe 상에 Heater의 ID값은 300으로 설정 하였고, Air conditioner의 ID는 400으로 설정 하였다.

3-3. 온도제어 시스템 Simulation

테스트 방법은 실제로 구현한 온도ECU에 인위

적으로 온도 변화주면 이를 감지한 온도센서가 온도가 낮으면 Heater를 작동 시켜 온도를 높이고, 온도가 적정온도 보다 높으면 Air conditioner를 작동시켜 온도를 낮추어 주어서 적정온도에 맞게 조절한다. 이 시스템의 흐름도는 (그림 4)와 같다.

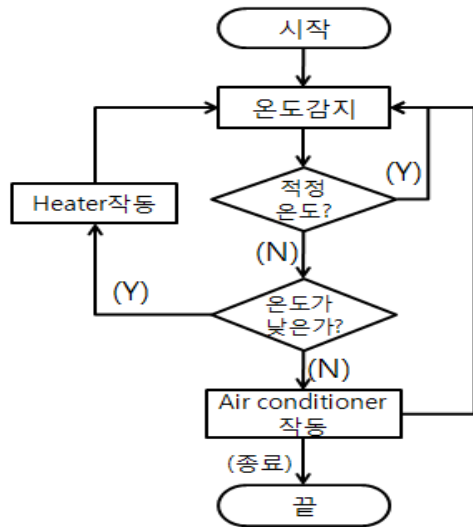


그림 4. 시스템 흐름도

Simulation은 Heater와 Air conditioner를 가상으로 제작하였기 때문에 실제로 동작하여 현재 온도를 올리거나 낮추는 것이 불가능 하다. Heater와 Air conditioner 작동 시에 그에 맞는 동작을 인위적으로 온도센서에 가해 주었다. 예를 들어 Heater가 작동을 하면 드라이기를 이용하여 온도센서 주변의 온도를 올려 주었고, 온도가 적정 온도가 되면서 CANoe상의 Heater가 작동을 멈추면 인위적으로 가했던 동작도 멈추게 했다. 가상의 ECU들의 동작정보는 CANoe상의 Trace창을 통하여 확인이 가능하다.

IV. 결 론

본 논문에서는 AT90CAN128모듈과 온도센서를 사용하여 차량 내의 온도를 자동으로 제어 해주는 하나의 ECU를 제작하여, ECU에서 나오는 신호를 받아서 처리하는 Heater와 Air conditioner를 가상으로 CANoe상에 만들어 테스트 하였다. 이 시스템이 차량에 장착될 경우 운전자가 자신이 원하는 온도를 한번 설정 해놓으면 운전 중에 별다른 조작 없이도 쾌적한 환경을 제공해 줄 것이다. 이와 비슷한 시스템은 현재 차량에도 있지만 CAN통신을 사용하지 않는다. 미래에 차량에 지금보다 많은 수의 환경조절 시스템이 장착된다고 볼 때 CAN통신을 사용하면 비용과 무게 등의

절감 효과를 가져 올 것으로 기대 된다.

향후 실제 CAN모듈이 장착된 ECU를 제작하여 본 온도제어ECU와 테스트 해봐야 할 것이다. 실제 ECU와 연결될 시에는 각 모듈의 통신 신뢰성이 만족되어야 하기 때문에 본 연구에서 진행되었던 실험 보다 더 어려울 것 이라고 판단된다.

온도 센서가 실제 차량에서 사용 될 수 있는 지 여부와 어느 위치에서 온도를 측정해야 원하는 온도 값이 측정이 가능한지의 연구가 필요하다.

또한 온도센서뿐 아니라 습도센서를 장착하여 좀 더 쾌적한 환경을 제공해 줄 수 있도록 연구 할 것이다.

앞으로 차량의 많은 장비에 CAN이 적용 될 것으로 보이고, Heater와 Air conditioner에도 CAN시스템이 적용 될 수 있을 것이다. 실제로 장착하여 동작하도록 하는 연구 및 실험을 진행 할 것이다.

Acknowledgment

본 연구는 지식경제부(정보통신연구진흥원), 부산광역시 및 동의대학교와 중소기업 산학협력 개발 지원 사업의 지원을 받아 수행된 연구결과임.(08-기반-13, IT특화연구소:“부산IT융합부품연구소”설립 및 운영)

참고문헌

- [1] RT(Real-Time) CAN 시스템 구조 연구.
- [2] "CAN System Engineering". Spring
- [3] CAN Specification 2.0 part A and B. R.Bosch, Sept.1991
- [4] ATMEL, AT90CAN128 Datasheet, 9 2005.
- [5] CAN-the technical introduction, CAN in Automation, <http://www.can-cia.de/ican.htm>
- [6] CANoe www.vector.com