(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년09월07일
(11) 등록번호 10-1654720
(24) 등록일자 2016년08월31일

(51) 국제특허분류(Int. Cl.)
H04L 12/40 (2006.01)
(52) CPC특허분류
H04L 12/40006 (2013.01)
H04L 2012/40215 (2013.01)
(21) 출원번호 10-2015-0126745
(22) 출원일자 2015년09월08일
심사청구일자 2015년09월08일
(56) 선행기술조사문헌
KR1020110041319 A

(73) 특허권자
인천대학교 산학협력단
인천광역시 연수구 아카데미로 119 (송도동)
(72) 발명자
황광일
인천광역시 연수구 아카데미로 119, 7호관 임베디드시스템공학과 (송도동, 인천대학교)
김승희
인천광역시 부평구 주부토로65번길 19, 702호 (부평동, 대경이즈뷰아파트)
(74) 대리인
특허법인충정

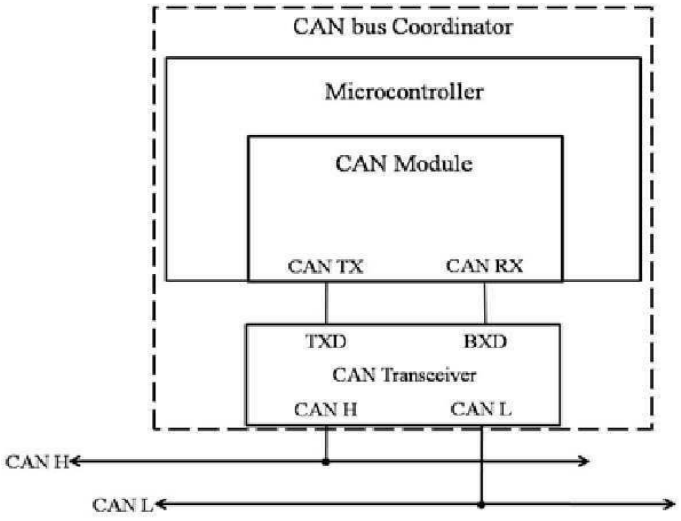
전체 청구항 수 : 총 13 항 심사관 : 김대성

(54) 발명의 명칭 **CAN 코디네이터에 의해 수행되는 CAN 제어 방법**

(57) 요 약

CAN 시스템의 전송 지연을 개선하고 송신과 수신이 모두 가능한 신규 노드에도, 플러그 앤 플레이(Plug and play) 기능을 구현할 수 있는 CAN 코디네이터의 CAN 제어 방법이 개시된다. 상기 방법은 등록 모드의 진입에 응답하여, CAN 내의 복수의 노드들로 등록 모드 시작 메시지를 송신하고 상기 복수의 노드들 중 적어도 하나로부터, 등록 모드의 준비 완료를 나타내고, 해당 노드의 기능별 제어 데이터의 개수 정보를 포함하는 응답 메시지를 수신한다. 응답 메시지를 기반으로, 복수의 노드들 중 응답한 노드 개수 및 응답한 노드 각각의 기능별 제어 데이터 개수를 카운팅하여 기능별 제어 데이터 전송 권한 메시지를 송신하고, 해당 노드의 기능 및 동작 조건에 관한 요청 정보를 포함하는 기능별 제어 데이터 정보 메시지를 수신한다. 또한, 기능별 제어 데이터 정보 메시지를 기반으로 응답한 노드 각각의 기능 및 동작 조건을 설정함으로써, 노드 설정 메시지를 송신한다.

대 표 도 - 도12



(52) CPC특허분류
H04L 2012/40273 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 C02496350100425923
부처명 산업통상자원부
연구관리전문기관 중소기업청
연구사업명 산학연공동기술개발지원사업
연구과제명 Multi-Functional 초저전력 IoT Connectivity (Plug and Connect) 시스템 개발
기 여 율 1/1
주관기관 인천대학교
연구기간 2014.12.01 ~ 2015.11.30
공지예외적용 : 있음

명 세 서

청구범위

청구항 1

CAN (Controller Area Network) 코디네이터에 의해 수행되는 CAN 제어 방법으로서,

등록 모드의 진입에 응답하여, 상기 CAN 내의 복수의 노드들로 등록 모드 시작 메시지를 송신하는 단계;

상기 복수의 노드들 중 적어도 하나로부터, 등록 모드의 준비 완료를 나타내고, 등록 모드의 준비가 완료된 노드의 기능별 제어 데이터의 개수 정보를 포함하는 응답 메시지를 수신하는 단계;

상기 응답 메시지를 기반으로, 상기 복수의 노드들 중 응답한 노드 개수 및 응답한 노드 각각의 기능별 제어 데이터 개수를 카운팅하는 단계;

상기 복수의 노드들 중 응답한 노드로, 상기 카운팅 결과를 기반으로 하는 기능별 제어 데이터 전송 권한 메시지를 송신하는 단계;

상기 복수의 노드들 중 응답한 노드로부터, 상기 응답한 노드의 기능 및 동작 조건에 관한 요청 정보를 포함하는 기능별 제어 데이터 정보 메시지를 수신하는 단계; 및

상기 기능별 제어 데이터 정보 메시지를 기반으로 상기 복수의 노드들 중 응답한 노드 각각의 기능 및 동작 조건을 설정함으로써, 상기 복수의 노드들 중 응답한 노드로 노드 설정 메시지를 송신하는 단계를 포함하는, CAN 제어 방법.

청구항 2

제 1 항에 있어서,

상기 응답한 노드의 기능 및 동작 조건에 관한 요청 정보는 ID 배치 요청을 포함하고,

상기 응답 메시지는 등록 모드 데이터 구간에 속하는 ID 를 가지며,

상기 노드 설정 메시지는 상기 응답한 노드의 기능에 대해 일반 메시지 데이터 구간에 속하는 ID 를 할당하는, CAN 제어 방법.

청구항 3

제 2 항에 있어서,

상기 응답한 노드의 기능 및 동작 조건에 관한 요청 정보는 긴급 메시지 전송 권한 요청을 더 포함하고,

상기 노드 설정 메시지는 상기 응답한 노드의 기능에 대해 긴급 메시지 데이터 구간에 속하는 ID 를 추가로 할당하며, 상기 긴급 메시지 데이터 구간의 ID 는 상기 일반 메시지 데이터 구간의 ID 보다 우선순위를 가지는, CAN 제어 방법.

청구항 4

제 1 항에 있어서,

상기 응답한 노드의 기능 및 동작 조건에 관한 요청 정보는 데이터의 기능, 송신 또는 수신 가능 여부, 기능의 제어 방법 및 정상 모드 운행시의 데이터 획득 위치 중 적어도 하나에 대한 요청을 포함하는, CAN 제어 방법.

청구항 5

제 2 항에 있어서,

상기 ID 배치 요청은 자동 배치, 최우선 배치, 특정 기능 앞에 배치 및 특정 기능 뒤에 배치 중 어느 하나에 관한 요청인, CAN 제어 방법.

청구항 6

제 1 항에 있어서,

상기 기능별 제어 데이터 전송 권한 메시지는 권한 부여의 대상이 되는 노드 번호를 포함하는, CAN 제어 방법.

청구항 7

제 1 항에 있어서,

상기 기능별 제어 데이터 전송 권한 메시지는 상기 복수의 노드들 중 응답한 노드의 기능별 제어 데이터 개수를 기반으로 권한 부여 시간을 설정하는, CAN 제어 방법.

청구항 8

제 1 항에 있어서,

상기 기능별 제어 데이터 전송 권한 메시지는 상기 복수의 노드들 중 응답한 노드가 복수일 경우, 상기 복수의 응답한 노드들로 하여금 순차적으로 기능별 제어 데이터 전송 권한을 가지도록 하는, CAN 제어 방법.

청구항 9

제 1 항에 있어서,

상기 응답 메시지를 수신하는 단계가 상기 복수의 노드들로부터의 응답 메시지 수신에 실패하는 경우, 등록 모드를 종료하는 단계를 더 포함하는, CAN 제어 방법.

청구항 10

제 1 항에 있어서,

상기 CAN 코디네이터에 미리 등록된 노드들 중 어느 하나의 노드로부터의 응답 메시지 수신에 실패하는 경우, 등록 모드를 종료하는 단계를 더 포함하는, CAN 제어 방법.

청구항 11

제 1 항에 있어서,

상기 복수의 노드들 중 응답한 노드로부터, 상기 노드 설정 메시지에 따른 상기 응답한 노드의 기능 및 동작 조건의 설정이 완료되었음을 나타내는 등록 모드 완료 메시지를 수신하는 단계; 및

상기 등록 모드 완료 메시지에 응답하여 등록 모드를 종료하는 단계를 더 포함하는, CAN 제어 방법.

청구항 12

제 9 항 또는 제 10 항에 있어서,

상기 등록 모드의 종료는 상기 CAN 에 대한 오류 발생을 나타내는, CAN 제어 방법.

청구항 13

제 11 항에 있어서,

상기 등록 모드의 종료에 응답하여, 상기 CAN 코디네이터는 작동을 중지하고, 상기 복수의 노드들은 정상 운행 모드로 동작하는, CAN 제어 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 CAN (Controller Area Network) 을 제어하기 위한 방법에 관한 것으로서, 보다 구체적으로는 CAN 코디네이터에 의해 수행되는 CAN 제어 방법에 관한 것이다.

배 경 기 술

[0002] 편리한 운송을 목적으로 등장한 자동차는 현재에 이르기까지 차량 관련 기술의 많은 발전이 있었으며, 차량은 최신의 전기, 전자, 정보 통신 기술이 융합되어 운전자에게 보다 높은 안전과 편의를 제공하며, 운송수단만이 목적이 아닌 사람이 생활하는 공간 즉, 업무 및 휴식 공간의 역할을 할 수 있는 곳으로 진화되고 있다.

[0003] 이렇게 차량이 진화하기 위해서는 다수의 전자제어 장비들이 탑재되게 되고, 차량관련 기술 수준이 높아짐에 따라 전자제어장비들의 수도 급속도로 증가 될 것이다. 특히 완성차 업체에서 자동차를 제작하는데 드는 비용 중에 전자제어장비가 차지하는 비중은 점점 늘어나는 추세이다. 또한, 전문가들은 향후의 혁신적인 기술의 80%는 전자기술에 관련될 것이라고 예측한다(비특허문헌 1 참조). 많아지는 전자제어장비들을 제어하기 위해서는 배선이 증가하게 되는데, 이러한 배선의 증가는 차량 설계 시 어려움이 있고, 차량의 중량을 증가시켜 성능 및 연비의 효율성이 안 좋아지며, 제조 비용 또한 증가하게 된다 (비특허문헌 2 참조). 이에 전자제어장비들 간의 유기적인 제어를 가능하게 해주며, 여러 문제를 해결해 줄 수 있는 네트워크 기반의 통신시스템이 필요하게 되었고, 이러한 요청에 의해 1980년대 독일 보쉬(Bosch)사에 의해 CAN(Controller Area Network) 시스템이 개발되었다 (비특허문헌 3 참조). 이러한 통신 네트워크가 차량에 적용되었다는 것은 기계식 방식의 시스템을 전자식으로 제어하는 것을 의미하고, 이는 보다 정밀한 제어가 가능하며, 차량의 배선 감소로 차량의 경량화를 실현하여 차량의 성능 향상 및 연비 개선 등의 여러 가지 이점을 갖게 된다.

[0004] 하지만, 이러한 네트워크 시스템은 무엇보다 신뢰성 보장에 관한 것이 중요한 것이 된다. 신뢰성이란 여러 부분에 걸쳐서 확보되어야 하지만 특히 네트워크 시스템에서의 신뢰성은 네트워크에 연결되어 있는 디바이스들이 오류 없이 적절하게 제어될 수 있도록 송신, 수신에 대한 부분이 보장되어 있어야 한다. 이러한 네트워크에서 구성하는 부품들에 대한 신뢰성이 확보가 되어 있다고 하더라도 정작 주된 기능인 통신에 대한 신뢰성이 확보되지 않아, 필요한 정보를 적절하게 전달하지 못하게 되면, 오류가 나게 되고 네트워크 시스템으로서 부적합하게 될 것이다.

[0005] 이러한 전송관련 등의 신뢰성이 보장되는 CAN 시스템은 통신을 위한 다양한 메커니즘으로 데이터에 대한 전송관련 안정성, 에러처리 등의 여러 가지 보완기능이 갖추어진 시스템으로 ISO 11898 및 ISO 11519-1 표준안으로 지정되어 있고, 그 신뢰성을 인정받아 현재 양산되는 거의 모든 차종에 적용되어 있으며, 이러한 많은 이점을 갖고 있는 CAN 시스템은 그 적용 범위가 확대되어 기계, 의료 등 많은 산업분야에서 사용되고 있다.

선행기술문헌

비특허문헌

[0006] (비특허문헌 0001) D. Marsh, "Network Protocols Compete for highway supremacy," EDN Eourope, June 2003.

(비특허문헌 0002) 이석, 김만호, 이경창, "차량용 네트워크 기술 연구 동향", 한국정밀공학회 논문집, 제23권, 제9호, 2006.

(비특허문헌 0003) Bosch, CAN specification version 2.0. Published by Bosch GmbH, September 1991.

(비특허문헌 0004) K. N. Ha, M. H. Kim, K. C. Lee, S. Lee "Performance Evaluation of Network Protocol for Automated Transfer Crane System," Journal of Control, Automation, and Systems Engineering Vol. 11, No. 8, August, 2005

(비특허문헌 0005) 박장식, 윤병우, AT90CAN128 을 이용한 CAN 통신 실무, 홍릉과학출판사, 2009

(비특허문헌 0006) Robert Bosch Gmbh, "CAN Specification Version 2.0" 1991

(비특허문헌 0007) K. Etschberger, "Controller Area Network-Basics, Protocols, Chips and Applications", IXXAT Press, 2001

(비특허문헌 0008) 김경섭, “동적 사용자 식별과 ISG 기능을 융합한 스마트키 시스템의 설계 및 구현,” 박사학위논문, 인하대학교 대학원, 2013.

(비특허문헌 0009)

(비특허문헌 0010) http://ww1.microchip.com/downloads/en/DeviceDoc/21667f.pdf - Microchip

(비특허문헌 0011) Bosch, CAN specification version 2.0. Published by Robert Bosch GmbH, September 1991.

(비특허문헌 0012) International Standard 11898 : Road Vehicles-Interchange of Digital Information Controller Area Network(CAN) for High-Speed Communication, ISO, 1993

(비특허문헌 0013) http://www.can-cia.org - CAN in Automation

(비특허문헌 0014) K. Etschberger, "Controller Area Network -Basics, Protocols, Chips and Applications", IXXZT press, 2001

(비특허문헌 0015) William E. Seitz. "Controller Area Network in Embedded Machine Control", Controlin Automation, North America, 2004

(비특허문헌 0016) 도미니크 파레, CAN, LIN, FlexRay 를 활용한 차량용 네트워크, (주)갑우문화사, 2011

(비특허문헌 0017) G. C. Buttazzo, "Rate Monotonic Vs, EDF:Judgment Day," Journal of Real-Time System, 29, 2005

(비특허문헌 0018) P. Pedreiras, L.Almeida, "EDF message scheduling on controller area network", Computer & Control Engineering Journal, 2002

(비특허문헌 0019) M. D. Natale, "Scheduling the CAN BUS with Earliest Deadline Techniques", IEEE Proc. of Real-Time Symposium, 2000

(비특허문헌 0020) J. P. Lehoezky and L. Sha, "Performance of Rear-Time Bus Scheduling Algorithm," ACM Performance evaluation Review, 1986

(비특허문헌 0021) M. H. Kim, K. N. Ha, K. C. Lee, and S. Lee, “Traffic prediction of CAN network system with dual communication channels,” International Conference on Control, Automation and System, 2008

(비특허문헌 0022) M. H. Kim, J. G. Lee, S. Lee, and K. C. Lee, "A Study on Distributed Message Allocation Method of CAN System with Dual Communication Channels," Journal of Institute of Control, Robotics and Systems, 2010

(비특허문헌 0023)

(비특허문헌 0024) http//ww1.microchip.com/downloads/en/DeviceDoc/39977f.pdf - microchip

발명의 내용

해결하려는 과제

[0007] 차량 네트워크 시스템으로 가장 많이 적용되고 있는 CAN (Controller Area Network) 시스템의 메시지를 분석해 보면 주기적으로 지속적인 메시지를 알려주는 주기성 있는 신호와 순간 상황에 따라 일어나는 다량의 이벤트성 메시지로 이루어져 있다. 이러한 데이터 메시지들이 통신을 하는 가운데 버스 상의 트래픽이 상대적으로 낮은 시스템에서 CAN 버스에 연결되어 있는 여러 개의 노드에서 동시에 메시지를 보내게 되면 버스 상에서의 메시지가 충돌을 하게 되더라도, CAN 시스템은 비교적 효율적으로 메시지 충돌에 대한 중재를 할 수 있다. 하지만, 버스 상의 트래픽이 상대적으로 높은 시스템에서는 메시지가 충돌하게 되면 중재 과정을 통하여 충돌에 대한 중재를 하게 되는데 이때 버스의 트래픽이 높기 때문에 보내고자 하는 데이터 메시지의 식별자 보다 우선순위가 높은 데이터 메시지가 많을 경우 전송지연이 발생하여 원하는 시간 내에 데이터를 전송하지 못하거나 전송기회를 놓치게 되는 문제점이 발생 할 수 있다. 즉, 버스의 트래픽이 증가함에 따라 평균적인 전송지연이 증가 하는 특징을 볼 수 있다 (비특허문헌 4 참조).

[0008] 따라서, 상기와 같은 문제점을 해결하기 위한 본 발명의 제 1 목적은, 긴급메시지 건에 대한 전송지연을 개선하고, 송신 및 수신이 가능한 신규 CAN 디바이스를 추가할 때, CAN 코디네이터가 데이터 메시지에 대한 관리를 진행하여, 기존 CAN 디바이스의 업그레이드 없이 CAN 시스템에 연결되어 유기적으로 동작할 수 있는 플러그 앤 플

레이(Plug and Play) 기능을 구현할 수 있는, CAN (Controller Area Network) 코디네이터에 의해 수행되는 CAN 제어 방법을 제공하는 것이다.

[0009] 다만, 본 발명의 해결하고자 하는 과제는 이에 한정되는 것이 아니며, 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위에서 다양하게 확장될 수 있을 것이다.

과제의 해결 수단

[0010] 본 발명의 일 실시예에 따른 CAN (Controller Area Network) 코디네이터에 의해 수행되는 CAN 제어 방법은, 등록 모드의 진입에 응답하여, 상기 CAN 내의 복수의 노드들로 등록 모드 시작 메시지를 송신하는 단계; 상기 복수의 노드들 중 적어도 하나로부터, 등록 모드의 준비 완료를 나타내고, 해당 노드의 기능별 제어 데이터의 개수 정보를 포함하는 응답 메시지를 수신하는 단계; 상기 응답 메시지를 기반으로, 상기 복수의 노드들 중 응답한 노드 개수 및 응답한 노드 각각의 기능별 제어 데이터 개수를 카운팅하는 단계; 상기 복수의 노드들 중 응답한 노드로, 상기 카운팅 결과를 기반으로 하는 기능별 제어 데이터 전송 권한 메시지를 송신하는 단계; 상기 복수의 노드들 중 응답한 노드로부터, 해당 노드의 기능 및 동작 조건에 관한 요청 정보를 포함하는 기능별 제어 데이터 정보 메시지를 수신하는 단계; 및 상기 기능별 제어 데이터 정보 메시지를 기반으로 상기 복수의 노드들 중 응답한 노드 각각의 기능 및 동작 조건을 설정함으로써, 상기 복수의 노드들 중 응답한 노드로 노드 설정 메시지를 송신하는 단계를 포함할 수 있다.

[0011] 일 실시예에 따르면, 상기 해당 노드의 기능 및 동작 조건에 관한 요청 정보는 ID 배치 요청을 포함하고, 상기 응답 메시지는 등록 모드 데이터 구간에 속하는 ID 를 가지며, 상기 노드 설정 메시지는 상기 해당 노드의 기능에 대해 일반 메시지 데이터 구간에 속하는 ID 를 할당할 수 있다.

[0012] 일 실시예에 따르면, 상기 해당 노드의 기능 및 동작 조건에 관한 요청 정보는 긴급 메시지 전송 권한 요청을 더 포함하고, 상기 노드 설정 메시지는 상기 해당 노드의 기능에 대해 긴급 메시지 데이터 구간에 속하는 ID 를 추가로 할당하며, 상기 긴급 메시지 데이터 구간의 ID 는 상기 일반 메시지 데이터 구간의 ID 보다 우선순위를 가질 수 있다.

[0013] 일 실시예에 따르면, 상기 해당 노드의 기능 및 동작 조건에 관한 요청 정보는 데이터의 기능, 송신 또는 수신 가능 여부, 기능의 제어 방법 및 정상 모드 운행시의 데이터 획득 위치 중 적어도 하나에 대한 요청을 포함할 수 있다.

[0014] 일 실시예에 따르면, 상기 ID 배치 요청은 자동 배치, 최우선 배치, 특정 기능 앞에 배치 및 특정 기능 뒤에 배치 중 어느 하나에 관한 요청일 수 있다.

[0015] 일 실시예에 따르면, 상기 기능별 제어 데이터 전송 권한 메시지는 권한 부여의 대상이 되는 노드 번호를 포함할 수 있다.

[0016] 일 실시예에 따르면, 상기 기능별 제어 데이터 전송 권한 메시지는 상기 복수의 노드들 중 응답한 노드의 기능별 제어 데이터 개수를 기반으로 권한 부여 시간을 설정할 수도 있다.

[0017] 일 실시예에 따르면, 상기 기능별 제어 데이터 전송 권한 메시지는 상기 복수의 노드들 중 응답한 노드가 복수일 경우, 상기 복수의 응답한 노드들로 하여금 순차적으로 기능별 제어 데이터 전송 권한을 가지도록 할 수 있다.

[0018] 일 실시예에 따르면, 상기 응답 메시지를 수신하는 단계가 상기 복수의 노드들로부터의 응답 메시지 수신에 실패하는 경우, 등록 모드를 종료하는 단계를 더 포함할 수 있다.

[0019] 일 실시예에 따르면, 상기 CAN 코디네이터에 미리 등록된 노드들 중 어느 하나의 노드로부터의 응답 메시지 수신에 실패하는 경우, 등록 모드를 종료하는 단계를 더 포함할 수 있다.

[0020] 일 실시예에 따르면, 상기 복수의 노드들 중 응답한 노드로부터, 상기 노드 설정 메시지에 따른 해당 노드의 기능 및 동작 조건의 설정이 완료되었음을 나타내는 등록 모드 완료 메시지를 수신하는 단계; 및 상기 등록 모드 완료 메시지에 응답하여 등록 모드를 종료하는 단계를 더 포함할 수 있다.

[0021] 일 실시예에 따르면, 상기 등록 모드의 종료는 상기 CAN 에 대한 오류 발생을 나타낼 수 있다. 또는, 일 실시예에 따르면, 상기 등록 모드의 종료에 응답하여, 상기 CAN 코디네이터는 작동을 중지하고, 상기 복수의 노드들은 정상 운행 모드로 동작할 수 있다.

발명의 효과

[0022] 전술한 본 발명의 일 실시예에 따른 CAN 제어 방법에 따르면, CAN 코디네이터의 사용을 통해 기존의 CAN 시스템의 취약점인 전송 지연을 개선할 수 있으며, 송신과 수신이 모두 가능한 신규 노드를 적용하는 경우에도, 기존 노드들의 업그레이드 없이 바로 시스템에서 구동 가능한 플러그 앤 플레이(Plug and play) 기능을 구현할 수 있다.

[0023] 또한, CAN 코디네이터가 등록 모드에서만 동작을 하며 등록 모드의 종료 후에는 CAN 코디네이터의 기능을 정지하도록 함으로써, 현재까지 많은 시행착오와 변화를 거쳐 안정적인 시스템으로 안착된 기존의 CAN 시스템에 영향을 미치지 않도록 할 수 있다. 따라서, 다른 CAN 디바이스들은 기존의 CAN 시스템과 동일하게 유지 시켜 안정성까지 확보되는 시스템을 제공할 수 있다.

도면의 간단한 설명

[0024] 도 1 은 차량의 CAN 통신의 개념도이다.

도 2 는 CAN 적용의 비교 도면이다.

도 3 은 CAN 표준을 나타낸다.

도 4 는 차량에 적용되는 CAN 을 나타낸다.

도 5 는 CAN 시스템의 통신망을 나타낸다.

도 6 은 CAN 버스 신호를 나타낸다.

도 7 CAN 2.0A의 버스에서의 중재과정을 나타낸다.

도 8 CAN 프로토콜의 계층 구조를 나타낸다.

도 9 는 CAN 계층별 데이터 교환을 나타낸다.

도 10 은 CAN의 데이터 프레임 구조를 나타낸다.

도 11 은 CAN 2.0A와 CAN 2.0B의 송수신 관계를 나타낸다.

도 12 는 코디네이터의 구성을 나타내는 블록도이다.

도 13 은 본 발명의 일 실시예에 따른 CAN (Controller Area Network) 코디네이터에 의해 수행되는 CAN 제어 방법의 흐름도이다.

도 14 는 상기 CAN 코디네이터에 의해 제어되는 CAN 디바이스의 작동에 관한 흐름도이다.

도 15 는 CAN 데이터의 기본 형식을 나타낸다.

도 16 은 코디네이터의 메시지 전송형식1 을 나타낸다.

도 17 은 코디네이터의 메시지 전송형식2 를 나타낸다.

도 18 은 코디네이터의 메시지 전송형식3 을 나타낸다.

도 19 는 노드 메시지 전송형식1 을 나타낸다.

도 20 은 노드 메시지 전송형식2 를 나타낸다.

도 21 은 노드 메시지 전송형식3 을 나타낸다.

도 22 는 CAN 테스트 모듈의 구성도이다.

도 23 은 CAN 테스트 노드(Node) 를 나타낸다.

도 24 는 CAN 데이터 메시지 측정 장비를 나타낸다.

도 25 는 실험 환경 구성을 나타낸다.

도 26 은 CAN 데이터 전송지연을 나타낸다.

도 27 은 데이터 메시지 증가에 따른 전송지연 시간을 나타낸다.

도 28 은 긴급 메시지 건에 대한 전송지연 개선을 나타낸다.

도 29 는 특정 식별자의 긴급메시지 전송에 관련된 도면이다.

도 30 은 긴급메시지 전송지연 개선 시스템 적용과 미적용 비교를 나타낸다.

도 31 은 CAN 코디네이터의 CAN 노드 배치 과정을 나타낸다.

도 32 는 CAN 코디네이터의 CAN 노드 배치 과정2 를 나타낸다.

도 33 은 Plug and Play를 위한 통신과정을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0025] 본문에 개시되어 있는 본 발명의 실시예들에 대해서, 특정한 구조적 내지 기능적 설명들은 단지 본 발명의 실시예를 설명하기 위한 목적으로 예시된 것으로, 본 발명의 실시예들은 다양한 형태로 실시될 수 있으며 본문에 설명된 실시예들에 한정되는 것으로 해석되어서는 아니 된다.

[0026] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는바, 특정 실시예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 개시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0027] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 설시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0028] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미이다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미인 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0029] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[0031] ### 1. 개요

[0032] 본 발명에서는 CAN 시스템에 연결되어 있는 각 노드(Node)들의 데이터에 관한 사항이나 정보들을 관리하는 CAN 코디네이터(Coordinator)라는 신규 모듈을 추가하여, 전송지연 부분에 대한 이중 ID 배치를 통한 개선 방안과 CAN 시스템에 수신뿐만이 아닌 송신을 해야 하는 신규 노드(Node) 추가 시 하드웨어적인 수정이 필요한 경우가 아닌 이상 CAN 네트워크에 연결만 하면 CAN 코디네이터에 의한 정보 관리로 기존의 노드들과 통신하여 동작할 수 있는 플러그 앤 플레이(Plug and Play) 기능을 제안하고자 한다. 제시한 시스템의 성능 평가를 위해 테스트 보드를 제작하였고, 실험을 통해 긴급메시지에 대한 전송지연 개선 성능 향상 입증 및 플러그 앤 플레이 기능에 대한 구현을 하였다.

[0033] 이하, 6장에 걸쳐 본 발명을 설명하며, 2장에서는 CAN 시스템의 개요 및 데이터 구조에 대해 소개하고, 3장에 전송지연을 개선하기 위해 노력하였던 기존연구에 대해 소개한다. 4장에서는 본 발명에서 제안하는 시스템에 대하여 소개하고, 5장에서는 성능평가 및 입증을 위한 테스트 모듈을 이용한 실험 및 고찰을 한다. 마지막으로 6장에서는 본 연구를 통해서 얻은 결론에 대해 기술한다.

[0035] ### 2. CAN (Controller Area Network)

[0036] #### 2.1. CAN의 개요

[0037] CAN 프로토콜은 고수준의 보안 기능을 갖춘 실시간, 직렬, 브로드캐스팅 프로토콜이고, 고속 전송을 위한 ISO11898과 저속 전송을 위한 ISO11519-2에 정의되어 있는 국제표준이다. CAN 은 초기에 자동차에 적용하기 위하여 고안된 직렬 네트워크 통신방식이며, 최근에는 다양한 산업분야에 폭넓게 적용되고 있다 (비특허문헌 5 참

조).

[0038] CAN 은 차량 내의 서로 다른 ECU (Electronic Control Unit) 간의 통신을 위한 시스템을 자동차 업체인 벤츠의 요구에 의하여 1933년 독일 보쉬에 의해 개발된 차량용 네트워크 시스템으로, 1986년 미국 디트로이트에서 열린 SAE (Society of Automotive Engineers) 에서 공식적으로 발표되었고, 1993년 ISO 국제표준 규격으로 제정되었다 (비특허문헌 6 참조).

[0039] 도 1 은 차량의 CAN 통신의 개념도이고 도 2 는 CAN 적용의 비교 도면이다. 오늘날 차량에서는 도 1 에서 볼 수 있듯이 차량을 제어하기 위한 각종 제어모듈 및 센서 모듈의 증가로 연결하는 배선 (Harness) 이 증가 되었고, 이로 인하여 결선 및 무게에 대한 문제점이 발생하였다. 이는 도 2 에서 볼 수 있듯이 제어 모듈이 증가함에 따라 배선이 증가되어 제조 비용이 증가할 뿐만 아니라 신뢰성도 떨어지고, 제어 모듈 추가가 어려워지는 문제점이 발생 할 수 있다.

[0040] 이에 CAN 시스템은 두 개의 꼬인 선을 이용하여 제어모듈 및 센서 등을 연결하여 배선 증가로 인한 문제점을 어느 정도 해결하였고, 다른 통신 프로토콜에 비해 가격대비 성능이 우수할 뿐 아니라, 외부 전자파나 노이즈에 강해 자동차 업계에서는 CAN 시스템을 경쟁적으로 도입하게 되었다. 뿐만 아니라 산업자동화, 건물 자동화, 의료 장비, 열차, 중장비 등 다양한 분야에서 활용되고 있다 (비특허문헌 7 참조).

[0041] 차량에는 다양한 네트워크 프로토콜이 적용되고 있으며, 데이터 전송속도에 따라 Class A, Class B, Class C로 구분된다. Class A는 10kbps 이하로 저속이고, Class B는 100kbps 이하로 중속이며, Class C는 100kbps 이상으로 고속으로 분류된다. 여기에서 CAN 통신은 Class C에 속하는 대표적인 고속통신 네트워크로서 파워트레인, 멀티미디어, 바디전장 등의 실시간으로 정확한 제어를 하기위한 네트워크로 사용되고 있다. 도 3은 CAN 표준을 나타낸다 (비특허문헌 8 참조).

[0042] 앞서 언급한 차량에 쓰이는 CAN은 적용하는 제작사 마다 차이가 있지만 일반적으로 CAN의 속도 및 용도에 따라 파워트레인, 멀티미디어, 바디전장 등으로 그룹화 되어 있다.

[0043] 도 4 는 차량에 적용되는 CAN 을 나타낸다. 도 4 를 기준으로 설명하면, 파워트레인 등을 담당하는 샤시 CAN은 500kbps의 통신 속도를 갖으며 ECU(Engine Controller Unit), MDPS(Motor Driven Power Steering), ABS(Anti lock Brake System) 등의 차량이 달리고, 조향하고, 정지하는 제어 모듈을 그룹화 하여 연결 되어 있으며, 바디전장 등을 담당하는 바디 CAN은 100kbps의 통신 속도를 갖고 DDM(Driver Door Module), ADM(Assist Door Module), PSM(Power Seat Module), SWRC(Steering Wheel Remote Controller) 등의 차량의 도어 Lock/Unlock, 시트조절, 핸들 스위치와 같은 부가 적인 편의 사항 등을 그룹화 하여 연결되어 있고, 멀티미디어를 담당하는 멀티미디어 CAN은 바디 CAN과 동일하게 100kbps의 속도를 갖으며, Monitor, HU(Head Unit) 등의 내비게이션이나 오디오 같은 것을 그룹화 하여 담당한다.

[0044] 이렇게 CAN 시스템은 적용되는 분야 또는 제품에 따라 속도별 용도별 그룹화 하여 적용을 하고 있다.

[0046] **2.2. CAN의 특징**

[0047] CAN은 몇 가지의 특징이 있는데, 일단 하드웨어적인 부분과 ISO 규격에 따라 ISO 11898-2(High-speed : 50kbps에서 1Mbps)와 ISO 11898-3(Low-speed : 50kbps에서 125kbps)로 구분이 되고, 비파괴 비트별 중재(Non-Destructive Bitwise Arbitration), 모든 ECU들은 모든 메시지를 수신할 수 있는 브로드캐스팅(Broadcasting) 등이 있다.

[0048] 먼저, 하드웨어적인 특징을 보면, CAN 시스템에서 CAN 노드(Node) 간의 통신망을 형성하게 되는데, 이 노드들은 두 가닥의 꼬임선(Twist Pair Wire)으로 연결되고, 통신은 전기적 차등을 이용하여 하며 이것은 반사파나 노이즈에 매우 강하다. 도 5 를 참조하면, CAN 버스는 양단의 끝에 종단 저항을 연결하며, 종단저항은 대략 120Ω을 연결한다 (비특허문헌 9 참조).

[0049] 도 6 은 CAN 버스 신호를 나타내며, 도 6 을 참조하면 CAN 버스 신호는 전기적 차등을 이용하는데, 즉 버스에서의 전압 레벨의 차이로 '0'과 '1'을 나타내고, 이를 마이크로 컨트롤러에서 인식하여 '0'인 비트를 우성(Dominant), '1'인 비트를 열성(Recessive) 이라고 정의한다 (비특허문헌 10 참조).

[0050] CAN High가 3.5V, CAN Low가 1.5V이면 이 상태는 우성으로 '0'을 나타내고, CAN High와 CAN Low가 2.5V라면 이 때는 열성 상태로 '1'을 나타낸다.

[0051] 도 7 CAN 2.0A의 버스에서의 중재과정을 나타낸다. 이러한 CAN 시스템의 버스에서 우성과 열성은 버스 상의 데

이터를 중재하는데 사용되는데, 도 7은 식별자 11비트를 갖고 있는 CAN 2.0A 방식에서 3개의 데이터가 버스 상에 동시에 충돌이 되었을 때 중재 하는 과정이다.

[0052] 도 7 의 중재 과정을 설명하면, CAN 2.0A의 식별자 11비트를 사용하고, Data 1은 ID 2A0, Data 2는 ID 5A0, Data 3은 ID 220을 갖고 있다. CAN 버스 상에 이러한 3개의 데이터가 충돌이 되면, 중재를 하게 되는데, 먼저 10번 비트에서 우성인 '0'과 열성인 '1'이 만나게 되고, 열성을 갖고 있는 Data2 는 중재에서 탈락을 하게 된다. 그럼 CAN 버스에는 Data 1과 Data 3이 남게 되는데, 7번 비트에서 중재에 의해 열성인 Data 1 이 탈락을 하게 되고, CAN 버스에는 Data 3이 남게 된다. 이러한 중재 과정을 거쳐 CAN 버스에 남게 되는 Data 3는 Data 1과 Data 2에 비하여 우선권을 갖고 먼저 송신을 하게 된다.

[0053] 두 번째는 ISO 규격에 따른 분류로 CAN 데이터 버스 Class C에 해당하는 ISO 11898-2은 1Mbps의 최대 속도를 갖고, 우성 상태의 CAN_H는 3.5V이고 CAN_L은 1.5V이며, 열성 상태의 CAN_H와 CAN_L는 2.5V이다.

[0054] CAN 데이터 버스 Class B에 해당하는 ISO 11898-3은 Fault Tolerant Low Speed를 줄여서 L-CAN이라 부르며, 125kbps의 최대 속도를 갖고, 우성 상태의 CAN_H는 3.6V이고 CAN_L은 1.4V이며, 열성 상태의 CAN_H는 0V이고, CAN_L는 5V이다 (비특허문헌 11 참조).

[0055] 세 번째로 비파괴 비트별 중재(Non-Destructive Bitwise Arbitration)인데, CAN은 CSMA/CD-NDBA(Carrier Sense Multiple Access with Non-Destructive Bitwise Arbitration)라는 메시지 전송 메커니즘을 가지고 있고, CSMA/CD는 이더넷 전송 프로토콜로서 IEEE 802.3 표준에 정의되어 있다. 이는 데이터를 전송하려는 노드가 네트워크상에 다른 노드가 통신하고 있는지를 모니터링하며 버스에 신호가 송출되고 있는지를 확인한 다음 버스가 송출되고 있지 않으면 데이터를 전송하는 방식이다. CSMA/CD는 여러 개의 노드들이 동일한 네트워크에 연결되는 것이 가능하며, 충돌되는 패킷들을 전송하는데 소요되는 대역폭을 줄일 수 있다. 이 방법은 버스에 많은 노드가 연결되어 한 번에 데이터를 전송하는데서 부하가 발생되며 지연이 일어나는 문제점이 있다. 이러한 지연을 피하기 위해, CAN 시스템은 버스에서는 두 개 이상의 노드가 동시에 데이터를 전송할 때, 하나의 데이터에 우선순위를 갖게 하는 방법을 사용하는데, 이를 비파괴 비트별 중재(Non-Destructive Bitwise Arbitration) 기법이라 한다.

[0056] 마지막으로 브로드캐스팅(Broadcasting)은 CAN 시스템이 메시지 지향성 프로토콜(Message oriented protocol)이라는 특징을 갖고 있기 때문에 가능한데, CAN 시스템은 노드의 주소에 의해서 데이터가 통신하는 것이 아니라 메시지의 우선순위에 의해 할당된 아이디에 의해 데이터의 교환이 이루어진다. 따라서 노드는 CAN 버스에 데이터를 전송하게 되고, CAN 버스에 연결되어 있는 노드들은 다른 노드가 버스에 전송하는 데이터를 모두 수신할 수 있게 된다. 이러한 과정 속에 멀티캐스팅(Multicasting), 브로드캐스팅(Broadcasting)이 가능해 지는 것이다. 또한, 이러한 특징이 있다는 것을 생각해 보면 단순하게 수신만 가능한 노드라면 하드웨어나 소프트웨어의 변경 없이 기존의 CAN 시스템에 추가가 가능 하다는 것을 알 수 있다.

[0057] CAN 시스템 중 추가적으로 특징을 몇 가지 더 설명을 하면, CAN 시스템은 꼬임선을 사용한 전기적 차등 통신을 하여 차량과 같은 전기적으로 불안정한 시스템에서 전기적 노이즈에 대한 매우 강한 내구성을 갖고 있고, 하드웨어적인 오류 보정이 가능하기 때문에 하드웨어적으로 수신필터를 어떻게 설정하는가에 따라 원하는 아이디 및 갖고 있는 데이터와 특정 그룹 및 전체 수신을 할 수 있다. 또한. 한번에 8개의 데이터를 전송하는 하드웨어 패킷을 제공하기 때문에 UART(RS232, RS485) 통신에서처럼 패킷 통신을 하기 위해 패킷의 형식을 사용자가 일일이 만들어 주고, 데이터를 수신받을 때에도 상대방이 어떻게 보냈는가에 대해 분석이 가능하다.

[0058] 그러나 CAN 시스템은 8바이트 데이터를 담는 하드웨어 패킷 통신을 하기 때문에 데이터 버퍼에 데이터를 넣고 전송하거나 수신된 데이터 버퍼를 읽어 오기만 하면 되는 구조를 가지고 있기에 특별하게 제어할 필요 없이 하드웨어가 모든 처리를 한다. 이러한 이유를 바탕으로 현재 많은 완성차 제조업체 및 다양한 분야에서 여러 방면으로 사용하고 있고 그 수요에 부합하기 위해 많은 마이크로컨트롤러 칩(Chip) 등의 반도체 제조업체들이 다양한 CAN 컨트롤러와 트랜시버를 개발, 판매하고 있어 다양한 기능과 안정성을 겸비한 제품의 수급이 용이하고 가격 또한 칩 제조사의 경쟁 하에 낮아지고 있어 CAN 시스템 도입이 가능한 제품을 만드는 제조회사들은 CAN 시스템을 많이 적용해 나가고 있다.

[0060] **2.3. CAN의 계층 구조**

[0061] 도 8 은 CAN 프로토콜의 계층 구조를 나타낸다. CAN 통신 프로토콜의 데이터 정보교환 방식을 ISO의 OSI 7 Layer에 비교하면, 도 8 과 같다. CAN 프로토콜 또한 OSI 7 Layer를 기반으로 구동이 되고 CAN 프로토콜은 OSI 7 Layer 중 하위 두 계층인 데이터 링크 계층(Data Link Layer)과 물리 계층(Physical Layer)을 사용하고

있다.

[0062] 먼저, 물리 계층(Physical Layer)은 데이터 전송을 위한 물리적 링크를 설정, 유지, 절단하는데 관련된 최하위 계층이고, '0'과 '1'을 나타내는 신호의 형식, 연결하는 커넥터의 규격, 전압이나 전류의 허용치, 각종 제어신호의 명칭과 역할 등으로서 전송하는 매체를 통해 데이터 정보를 실질적으로 전기적 신호를 전송하는 역할을 한다. 그 다음으로 데이터 링크 계층(Data Link Layer)은 CAN 프로토콜을 구현하기 위해 가장 중요한 부분이라고 할 수 있는데, 물리 계층(Physical Layer)에서 만든 0, 1의 두 가지 신호를 실질적 데이터로 만드는 인코딩(Encoding)을 통해 만들어진 데이터를 가지고 프레임을 구성하는 프레이밍(Framing), 오류 발생을 막기 위한 에러검출 (Error Detection), 각 노드와 노드 간의 속도차를 보정하기 위한 흐름 제어 (Flow Control) 등의 역할을 수행한다 (비특허문헌 12 참조).

[0063] 도 9 는 CAN 계층별 데이터 교환을 나타낸다. CAN 시스템의 계층(Layer)에 따른 메시지 송, 수신 구조를 도 9에서 볼 수 있듯이 물리계층(Physical Layer)에서는 CAN 버스의 전압 레벨(데이터의 우성과 열성)로 정보를 전달하고, 데이터 링크 계층(Data Link Layer)에서는 전달 받은 전압 레벨을 CAN 컨트롤러가 '0'과 '1'의 로직 레벨로 변환하여 식별자(Identifier)와 데이터(Data)로 구분하여 정보를 인식한다. 또한, 어플리케이션 계층(Application layer)에서는 각 노드의 시스템에 맞게 데이터 정보를 구분 한다 (비특허문헌 13 참조).

[0064] 또한, CAN 프로토콜은 다중 마스터 통신을 하고 있기 때문에 CAN 버스에 연결되어 있는 CAN 노드들은 모두가 마스터 역할을 할 수 있고 언제든지 버스를 사용하고 싶을 때 사용할 수 있다.

[0066] **2.4. CAN의 프레임 구조**

[0067] CAN의 프레임은 데이터 프레임(Data Frame), 원격 프레임(Remote Frame), 에러 프레임(Error Frame), 오버로드 프레임(Overload Frame)으로 4개의 타입으로 구성되어 있고, 인터 프레임(Inter Frame)을 포함하여 5 개의 타입으로 분류하기도 하며, 그 종류와 기능은 [표 2-1]에서 설명하는 바와 같다.

[0069] [표 2-1] CAN 프레임의 종류와 기능

[0070]

종류	기능
Data Frame	송신 노드(Node)에서 수신 노드(Node)로 일반적인 데이터를 전송하는 프레임
Remote Frame	수신 노드(Node)가 다른 노드(Node)로부터 데이터 전송을 요청할 때 사용되는 프레임
Error Frame	수신 노드(Node)에서 에러가 감지되는 경우 다른 노드(Node)에게 전송하기 위해 사용하는 프레임
Overload Frame	내부적인 과부하 상태로 데이터 프레임(Data Frame) 또는 원격 프레임(Remote Frame)을 지연 시킬 필요가 있거나 에러가 있을 때 전송하는 프레임
Inter Frame	이전 프레임과 이후에 들어올 Frame 간의 구분을 하기위해 사용하는 프레임

[0072] 본 발명에서 제안하는 시스템은 데이터 프레임을 활용하여 진행하기 때문에 관련된 부분을 중점적으로 살펴보고자 한다.

[0073] 도 10 은 CAN의 데이터 프레임 구조를 나타낸다. CAN 시스템의 데이터 프레임은 도 10 에서 볼 수 있듯이 CAN 2.0A (Standard Format)와 CAN 2.0B(Extended Format) 두 가지가 있고, 데이터 프레임 안에서는 SoF(Start of Frame), Arbitration Field, RTR(Remote Transmission Request), Control Field, Data Field, CRC Field, ACK Field, EoF(End of Frame)의 7개의 서로 다른 필드로 구성되어 있으며, 데이터 필드에서는 최대 64비트의 데이터를 전송할 수 있다 (비특허문헌 14 참조).

[0074] 앞서 언급한 바와 같이 CAN 프로토콜 버전은 CAN 2.0A(Standard Format)과 CAN 2.0B(Extended Format) 두 가지가 있는데, 가장 큰 차이점은 중재 필드(Arbitration Field)에 있다. CAN 2.0A는 11비트로 구성이 되어 있고, CAN 2.0B는 기본(Base) ID 11비트와 확장형(Extended) ID 18비트를 포함한 29비트로 구성이 되어 있다. 여기에서 CAN 2.0B는 기본(Base) ID 11비트를 갖고 있어 CAN 2.0A의 중재 필드 11비트와 호환이 가능하다.

[0075] 즉, CAN 2.0A와 CAN 2.0B의 송수신 관계를 나타내는 도 11 에서 볼 수 있듯이 CAN 2.0A에서 기본ID 11비트로 송신한 메시지를 CAN 2.0B에서는 수신가능 하지만, CAN 2.0B에서 확장형ID 29비트로 송신한 메시지는 CAN 2.0A에서 수신이 안 된다 (비특허문헌 15 참조).

[0076] CAN 2.0A를 기준으로 데이터 필드를 살펴보면, SoF(Start of Frame)은 메시지 프레임의 시작을 표시하고, 중재 필드(Arbitration Field)는 11비트의 식별자와 원격 전송을 요구하는 RTR(Remote Transmission Request) 비트를 가지며, RTR 비트가 '0' 일 때는 데이터 프레임을 '1' 일 때는 원격전송을 의미한다. 제어 필드(Control Field)는 6비트로 구성되며, 2비트의 예비 필드와 4비트의 데이터 길이 코드(Data Length Code)로 구성된다. 데이터 필드는 전송하고자 하는 데이터를 포함하며 64비트로 구성된다. CRC(Cyclic Redundancy Check)필드는 15비트의 주기적으로 중복을 확인하는 CRC 코드와 1비트의 딜리미터로 구성되며, 메시지에서 에러가 있는지를 체크하게 된다. ACK 필드는 2비트로 구성이 되며, 1비트의 ACK 슬롯과 1비트의 ACK 딜리미터로 구성된다. EoF(End of Frame)은 7비트로 구성되며 모두 '1'의 값을 갖고 이는 메시지의 끝을 알려 준다.

[0077] 제어 필드는 6비트로 구성되며, 2비트의 예비 비트와 4비트의 데이터 길이 코드로 구성되며, 데이터 필드는 전송하고자 하는 데이터를 0~8비트까지 전송할 수 있다. 즉, 최대 64비트 까지 전송 가능하다. 그리고 CRC 필드는 15비트의 주기적으로 중복을 확인하는 CRC 코드와 1비트의 딜리미터로 구성되며 이러한 구성으로 메시지의 에러 유무를 검사하게 된다. ACK 필드는 2비트로 구성되며 1비트의 ACK 슬롯과 1비트의 ACK 딜리미터로 구성된다. 마지막으로 EoF는 7비트로 구성되며 모두 1의 값을 갖고 있다.

[0079] ## 3. 관련연구

[0080] ### 3.1. 기존의 CAN 실시간 보장(전송지연 개선)에 관한 연구

[0081] 제품의 시스템을 구성하는 많은 전자제어 장치들 간의 제어를 하기 위해서는 각각의 전자제어 장치에서 송, 수신하는 제어신호를 적절한 타이밍, 즉, 실시간으로 통신을 하기 위하여 어떠한 방법으로 신호를 보낼 것인가에 대해 많은 이슈가 되어 왔다. 즉, 여러 개의 데이터 신호가 버스로 구성되어 있는 네트워크를 통해 송, 수신이 된다면 동시간 때 발생하는 데이터 전송에 대하여 충돌이 발생할 것이고 충돌에 대한 중재과정으로 인하여 전송지연 및 전송손실이 발생할 것이다. 이러한 전송지연 및 전송손실은 기기가 오작동 하거나 동작이 안 되는 등의 오류를 유발시킬 수 있다.

[0082] 특히 본 발명에서 다루고자 하는 CAN(Controller Area Network) 시스템의 경우 차량 안에서 수만은 전자제어장치들 간의 제어를 위한 통신으로 시작을 하였고, 적용되는 분야도 차량의 경우에서 많이 사용되기 때문에 적절한 타이밍에 제어 신호를 송신, 수신하는 것은 사람의 안전과 연계되어 더욱 중요하다고 할 수 있다.

[0083] 예를 들어 차량과 같은 경우를 들어 보면, 차량의 추돌 사고가 발생하였을 때 에어백(AIR-BAG)이 전계되지 않는 경우, 또는 브레이크를 밟았는데 그와 관련된 정보가 적절한 시기에 전송되지 않는 등의 문제가 발생하면 이러한 전송지연으로 인하여 각종 디바이스들이 유기적으로 동작하지 않게 되어, 제 기능을 하지 못 하는 결과가 발생하게 되고, 이는 엄청난 손실로 이어지게 된다.

[0084] 2장에서 설명한 바와 같이, CAN에서는 여러 이벤트가 동시에 발생하여 각각의 전자제어모듈이 한 번에 다량의 데이터를 버스로 전송하게 되면, 버스에서 여러 데이터 메시지가 충돌하게 되고, 데이터 충돌을 중재하는 과정에서 데이터 메시지의 식별자 우선순위 전송 특성에 따라 빠르게 전송되어야 하는 데이터 메시지가 있음에도 불구하고, 다른 데이터 메시지의 식별자에 비해 우선순위에 밀려, 전송주기 내에 전송을 완료하지 못하는 문제점이 발생할 수 있다.

[0085] 이에 이러한 문제들을 개선하기 위해 CAN 시스템에서 발생하는 전송지연과 관련된 여러 가지 연구가 선행되었고, 이러한 기존의 연구들을 검토해 보면 데이터 메시지들이 전송되는 순서를 적절하게 스케줄링하여 전송지연이 되는 문제를 개선하고자 연구한 것을 볼 수 있다.

[0086] 전송지연이 되는 문제를 개선하고자 하는 스케줄링 방법에는 정적(Static) 또는 동적(Dynamic) 스케줄링, 선점(Preemptive) 또는 비 선점(Non-preemptive) 스케줄링으로 구분된다 (비특허문헌 19 참조).

[0087] 정적 (Static) 스케줄링이라는 것은 기존 CAN 시스템에 많이 적용되어 있는 스케줄링 방법으로, 네트워크를 구축하는 초기에 데이터 메시지의 중요도에 따라 우선순위를 부여하고 그에 따른 ID를 배치한다. 이렇게 배치된 ID에 따라 데이터 메시지가 전송되는 것을 의미한다. 이에 비하여 동적 (Dynamic) 스케줄링은 시스템이 정상적인 동작을 하고 있는 가운데 상황 또는 필요에 따라 전송되는 데이터 메시지의 ID가 달라지는 것 즉 우선순위의 변화가 있다는 것을 의미한다.

[0088] 선점 (Preemptive) 스케줄링은 그 의미 그대로 버스 상에 임의의 메시지가 버스를 선점하여 데이터 메시지를 전송하고 있다면 그 데이터 메시지가 전송이 완료 될 때까지 대기하는 것을 의미하고, 비 선점 (Non-preemptive) 스케줄링은 임의의 메시지가 버스 상에 있더라도 우선순위에 의해 메시지가 전송될 수 있게 하는 것을 말한다.

여기에서 CAN 시스템은 임의의 메시지가 버스 상에 있을 경우 다른 노드들은 데이터 메시지를 전송하지 않고, 임의의 메시지가 전송이 완료될 때까지 기다리는 선점 스케줄링 특성을 가지고있다.

[0089] 그래서 CAN 시스템의 전송지연 개선에 관한 기존 연구를 보면, 정적 또는 동적 스케줄링 기법 등의 스케줄링을 통한 방법이 있고 이 기법들의 대표적인 것은 정적 스케줄링 기법의 경우에 DMS (Deadline Monotonic Scheduling) 이고, 동적 스케줄링 기법의 경우 EDF (Earliest Deadline First) 가 대표적이다. 또한, 전송 지연 개선에 관한 방법 중 분산 메시지 할당 방법도 있다. 본 발명에서는 전송지연을 개선하기 위한 기존연구에서 스케줄링 기법과 분산 메시지 할당방법을 소개하고자 한다.

[0091] **3.1.1. 스케줄링(Scheduling) 기법**

[0092] 정적인 스케줄링 기법의 대표적인 방법인 DMS (deadline Monotonic Scheduling) 은 RMS (Rate Monotonic Scheduling) 의 기법을 CAN 시스템에 적용하기 위하여 적합한 형태로 변경시킨 스케줄링 기법이다. 최적의 정적 스케줄링 기법으로 알려진 RMS에서는 주기의 길이로 우선순위를 결정하는데, 짧은 주기를 갖는 메시지가 긴 주기를 갖는 메시지보다 우선순위에서 앞서게 된다. 이러한 방식에 의하여 부여된 우선순위대로 메시지들은 동기화된 시간 선상에 차례대로 배치되어 전송된다. RMS에서는 일반적으로 전송주기를 데드라인(Deadline)과 동일하다고 가정하지만 항상 그런 것은 아니다. 이에 주기가 아닌 데이터 메시지의 전송이 완료되었을 때까지의 데드라인의 시간이 짧은 데이터 메시지에게 우선순위를 높게 부여하도록 변화 시킨것이 DMS이다. 이러한 고정된 스케줄링 기법들은 구현이 간단하다는 장점은 있지만 이것을 위해서 네트워크를 구성하는 노드들 간의 동기화가 필요한 문제가 있고, 다량의 데이터 메시지가 충돌하여 높은 트래픽이 발생하는 경우 우선순위가 상대적으로 낮은 데이터 메시지들은 전송 지연 또는, 전송이 불가능할 수도 있다.

[0093] 동적인 스케줄링 기법은 EDF (Earliest Deadline First) 방법이 대표적인데, 이 스케줄링 기법에서는 데드라인이 가까운 메시지를 먼저 전송하게 된다. 이러한 방법은 데이터 메시지의 우선순위를 정하는 것이 아니라 한 번의 데이터 메시지 전송이 끝나게 되면 바로 모든 노드들은 자신이 가지고 있는 데이터 메시지들의 데드라인을 확인하여 데드라인까지 시간이 적게 남은 메시지를 먼저 전송하게 된다. 동적인 EDF 기법은 정적인 DMS에서 발생할 수 있는 문제를 개선할 수 있고 버스 사용율을 높일 수 있다.

[0094] EDF와 같은 동적 스케줄링 기법은 많은 실시간 프로서서 스케줄링 알고리즘이 기반이 되지만 통신시스템에서 메시지를 위한 스케줄링 기법으로는 많이 사용되지 않는다. 이러한 이유는 통신을 하기 위해 버스를 차지하는 과정에서 매번 과도한 오버헤드가 발생하고, EDF를 수행하기 위해서는 모든 노드들이 동기화 되어야 한다는 것이다. 또한, 네트워크에 연결되어 있는 모든 노드들은 자신이 전송하고자 하는 데이터 메시지가 데드라인에 얼마나 가까이 있는지와 다른 데이터 메시지들과 얼마나 차이가 나는지를 계속해서 모니터링 해야 하기 때문에 실질적으로 구현하기도 힘들고, 그것을 처리하는 프로세서에도 많은 부하를 주게 된다 (비특허문헌 16 내지 비특허문헌 18 참조).

[0095] 기존의 스케줄링 기법에 대한 연구 부분을 보면, 정적인 스케줄링과 동적인 스케줄링에 대하여 각각의 장단점이 있지만 정적인 스케줄링 기법은 동적인 스케줄링 기법에 비해 효율적이지 못함을 나타내고 있다 (비특허문헌 19 참조).

[0097] **3.1.2. 분산메시지 할당 기법**

[0098] 분산메시지 할당 기법은 CAN 시스템의 네트워크 전송 용량을 증가시켜 데이터 메시지의 실시간 QoS 성능 (real-time Quality-of-Service performance) 을 보장하는 방법으로서 두 개 이상의 CAN 컨트롤러를 갖고 있는 마이크로컨트롤러 (Microcontroller) 와 이에 맞게 두 개의 CAN 트랜시버 (Transceiver) 를 이용하여 중복 채널 (Dual channel) 을 운영할 수 있는 시스템이다. 그러나 두 개의 채널을 이용하여 메시지를 전송하는 경우 적절하게 채널 선택이 되지 않는다면, 네트워크 트래픽이 한쪽으로 급격하게 기우는 현상이 발생할 수 있다.

[0099] 그것을 개선하기 위해 연구된 방법 중에는 CAN 노드에서 두 개의 채널을 이용하여 메시지를 전송하는 경우에 어떤 채널로 메시지를 전송할지를 결정하는 메시지 할당 방법이 제안되었다 (비특허문헌 20 참조). 그러나 이 방법은 트래픽을 모니터링 하는 트래픽 예측 모드 (Traffic prediction mode) 에서 에러가 발생했을 때는 데이터 메시지를 적절하게 할당하지 못하는 단점이 있다.

[0100] 분산 메시지 할당 기법에 대한 다른 연구에서는 모든 노드가 트래픽을 예측하고 적절하게 메시지를 할당할 수 있는 분산메시지 할당 방법이 제안되기도 하였다 (비특허문헌 21 참조).

[0101] 이러한 분산 메시지 할당 기법 같은 경우에는 상기 언급한 바와 같이 두 개의 CAN 컨트롤러를 갖고 있는 마이크

로컨트롤러(Microcontroller)와 두 개의 트랜시버(Transceiver)가 사용되어야 하는데, 이러한 특징은 분산 메시지 할당 기법을 구현하는 데 있어서는 여러 가지 장점이 있겠지만, CAN 디바이스를 만드는 데 있어서 비용의 추가가 발생할 수 있다. 실질적으로 칩을 제조하는 업체에서 제공하는 가격에 대한 부분을 보면 같은 스펙의 칩임에도 불구하고 CAN 컨트롤러가 추가되어 있는 마이크로컨트롤러 같은 경우에는 두 배 이상의 가격으로 형성되어 있는 것을 확인할 수 있고 그에 따른 CAN 트랜시버 또한 모듈 당 두 개가 사용되어야 한다.

[0102] CAN 시스템의 전송지연 개선에 대한 많은 연구가 있었지만, 대표적인 두 개의 연구 결과를 소개하였고, 이러한 연구들이 제시하는 결과물을 바탕으로 각각의 개선 방안에 대한 장단점을 참고하였다.

[0103] 본 발명에서는 CAN 시스템의 개선점을 보완하기 위해 CAN 코디네이터(Coordinator)라는 신규 노드를 추가할 수 있다. 이는 CAN 시스템이 1986년 미국 디트로이트에서 열린 SAE(Society of Automotive Engineers)에서 공식적으로 발표, 1993년 ISO 국제표준 규격으로 제정된 이후로 보완점에 대하여 개선되어 왔고, 현재의 CAN은 대부분의 양산 차량에 적용되어 있는 안정된 시스템이라 할 수 있다. 이러한 안정된 시스템에 정상적인 운행 중 변형을 가하는 것은 실질적인 양산 제품에서는 또다시 시스템에 대한 신뢰성이 낮아지는 결과를 가져와 다시 시스템에 대한 신뢰성을 증명하는 시간이 필요할 수 있기 때문에, CAN 코디네이터라는 모듈을 추가하여, CAN 코디네이터가 CAN 시스템에 간섭하는 주기를 한정하여 등록 모드라는 신규 모드를 제시하고, 이 등록 모드에서는 긴급메시지 건에 대한 전송지연 개선 및 송신 수신이 가능한 신규 CAN 디바이스 추가 시 CAN 코디네이터가 데이터 메시지에 대한 관리를 진행하여, 기존 CAN 디바이스의 업그레이드 없이 CAN 시스템에 연결되어 유기적으로 동작할 수 있는 플러그 앤 플레이(Plug and Play)기능을 구현하며, 등록모드가 끝나면 일반적인 CAN 시스템과 동일하게 동작하여 본 발명에서 제시하는 기능을 구현하고, 안정성까지 보장할 수 있다.

[0105] ## 4. 제안시스템

[0106] ### 4.1. 제안시스템의 개요

[0107] CAN 시스템을 포함한 대부분의 통신에 관련된 시스템은 요청에 대한 작업 처리를 함에 있어 신뢰성과 적절한 타이밍에 데이터를 송수신 할 수 있어야 하는데, 특히 실시간으로 전송되어야 하는 정밀 기기, 차량, 의료 등에 적용되는 CAN 시스템의 경우 데이터가 전송됨에 있어서 지연이 되지 않고 적절하게 전송되는 것은 더욱더 중요한 사항이 된다. 또한 기존의 CAN 시스템에서는 노드의 주소에 의해 데이터가 교환 되는 것이 아니라 데이터 메시지의 우선순위에 따라 할당된 ID에 의해 버스 상에서 데이터 교환이 이루어지기 때문에 멀티캐스팅(Multicasting), 브로드캐스팅(Broadcasting)이 가능하다. 이에 단순하게 수신만 가능한 노드의 경우면 CAN bus에 접속되어 있는 기존의 노드들의 하드웨어나 소프트웨어의 변경 없이 추가가 가능하지만, 단순하게 수신만 하는 노드가 아닌 송, 수신을 함께해야 하는 노드의 경우에는 기존 노드들의 업그레이드를 진행함으로써 신규 노드를 추가 적용 할 수 있었다.

[0108] 이러한 점들을 개선하기 위해 본 장에서는 CAN 코디네이터(Coordinator)라는 신규 디바이스 타입을 새롭게 정의하고 이를 기반으로 하여 기존의 CAN 시스템의 취약점인 전송지연에 대한 개선책과 송, 수신이 가능한 신규 노드를 적용 시 기존 노드들의 업그레이드 없이 바로 시스템에서 구동 가능한 플러그 앤 플레이(Plug and play) 기능을 제안 하고 또한 CAN 시스템이 1986년 SAE(Society of Automotive Engineers)에서 공식적으로 발표된 이후로 현재까지 많은 시행착오와 변화를 거쳐 안정적인 시스템으로 안착되었기 때문에 제안하는 시스템은 기존의 CAN 시스템에 영향을 미치지 않는 것을 목표로 하고 있다. 이로 인하여 제안하는 시스템에서는 등록모드라는 개념을 새롭게 도입하고, CAN 코디네이터는 등록모드 안에서만 동작을 하며 등록모드 종료 후에는 CAN 코디네이터는 기능을 정지되고, 다른 CAN 디바이스들은 기존의 CAN 시스템과 동일하게 유지시켜 안정성까지 확보되는 시스템을 제안하고자 한다.

[0110] #### 4.1.1. 시스템의 구성

[0111] 도 12 는 코디네이터의 구성을 나타내는 블록도이다. 도 12 에 도시된 바와 같이, 코디네이터 H/W는 크게 CAN 모듈을 포함하는 마이크로컨트롤러와 CAN 트랜시버로 구성된다. H/W는 CAN 코디네이터도 CAN 시스템에서 하나의 노드와 같은 개념이기 때문에 다른 노드 구성과 기본적인 부분과는 동일하다.

[0112] CAN ID 적용 부분은 CAN 2.0A의 11비트와 CAN 2.0B의 29비트 모두 본 시스템 적용이 가능하지만, 이하에서는 현 차량에 가장 많이 적용되어 있는 CAN 2.0A의 11비트 ID를 적용한 것으로 예를 들어 설명한다. 또한, 예시적인 동작 속도는 차량의 Body Electric System에 많이 사용하는 100kbps를 기준으로 할 수 있다.

[0114] ##### 4.1.1.1. 제안시스템 디바이스 타입정의

[0115] 4.1.1.1 에서는 제안시스템의 구성요소인 CAN 코디네이터와 노드 그리고 전송데이터 메시지 타입에 관하여 정의한다.

[0117] **가) CAN Coordinator**

[0118] 도 13 은 본 발명의 일 실시예에 따른 CAN (Controller Area Network) 코디네이터에 의해 수행되는 CAN 제어 방법의 흐름도이다. 이하, 도 13 을 참조하여 본 발명의 일 실시예에 따른 CAN 제어 방법을 보다 상세히 설명한다. CAN 코디네이터가 처음 파워-온(Power On)이 되면 등록 모드로 진입하게 되고, 등록 모드의 진입에 응답하여, 상기 CAN 내의 복수의 노드들로 등록 모드 시작 메시지를 송신한다. 여기서, CAN 코디네이터는 노드들이 연결되어 있는 CAN bus 에 등록 모드 시작 메시지를 송신할 수 있다.

[0119] 등록 모드 설정 시작 메시지를 보낸 CAN 코디네이터는 일정시간 동안 CAN bus 에 연결되어 있는 각각의 노드들로부터 응답을 기다리게 되는데, 이때 각각의 노드는 등록 모드 데이터 구간, 예를 들어 701~7FF 사이의 ID를 갖고 노드 관련 데이터의 응답 메시지를 송신할 수 있으므로, CAN 코디네이터는 등록 모드에 진입을 하면 예를 들어 701~7FF 사이의 ID를 갖는 데이터 메시지를 모니터링할 수 있다. 문제 발생 및 오류로 인하여 노드로부터 응답이 없으면 CAN 코디네이터는 등록 모드를 종료하고 노드에서 응답이 없음을 알려주어 시스템에 대한 점검을 할 수 있도록 도와준다. 즉, 복수의 노드들로부터의 응답 메시지 수신에 실패하는 경우, 등록 모드를 종료할 수 있고, 등록 모드의 종료는 상기 CAN 에 대한 오류 발생을 나타낼 수 있다.

[0120] 모니터링의 결과, 복수의 노드들 중 적어도 하나로부터, 등록 모드의 준비 완료를 나타내고, 해당 노드의 기능별 제어 데이터의 개수 정보를 포함하는 응답 메시지를 수신할 수 있다. 즉, CAN 코디네이터의 등록모드 시작 메시지에 대하여 각각의 노드가 등록모드 진입에 관련된 첫 번째 메시지를 정상적으로 노드 순서에 맞게 응답을 하면, CAN 코디네이터에서는 CAN Bus에 연결되어 있는 노드의 개수를 카운팅 하고, 그 노드가 몇 개의 기능별 제어 데이터를 가지고 있는지도 카운팅 한다. 다시 말해, CAN 코디네이터는 수신한 응답 메시지를 기반으로, 복수의 노드들 중 응답한 노드 개수 및 응답한 노드 각각의 기능별 제어 데이터 개수를 카운팅할 수 있다.

[0121] 여기에서 시스템을 최초 구동하는 것이 아닌 경우 기존의 CAN 노드 정보를 CAN 코디네이터가 가지고 있기 때문에 CAN 코디네이터에 미리 등록되어 저장되어 있는 노드 중 응답을 안 하는 노드가 발생을 하는 상황이 생기면 등록 모드 진행은 정지가 되며 CAN 코디네이터에서 오류가 생겼음을 알려주어 시스템 점검을 할 수 있도록 유도할 수 있다.

[0122] 노드에서 보낸 첫 번째 메시지가 이상이 없으면 CAN 코디네이터는 연산을 완료하고, 각각의 노드들이 가지고 있는 기능을 제어 및 상태를 알려주는 데이터를 송신 받을 수 있도록 해당 노드에 데이터 전송권한을 부여하는 메시지를 순차적으로 노드에 보낸다. 즉, 복수의 노드들 중 응답한 노드로, 전술한 카운팅 결과를 기반으로 하는 기능별 제어 데이터 전송 권한 메시지를 송신할 수 있다.

[0123] 기능별 제어 데이터 전송 권한 메시지는 권한 부여의 대상이 되는 노드의 노드 번호를 포함할 수 있으며, 복수의 노드들 중 응답한 노드의 기능별 제어 데이터 개수를 기반으로 권한 부여 시간을 설정할 수도 있다. 또한, 복수의 노드들 중 응답한 노드가 복수일 경우, 복수의 응답한 노드들로 하여금 순차적으로 기능별 제어 데이터 전송 권한을 가지도록 할 수도 있다. 보다 구체적으로, 기능별 제어 데이터 전송 권한에 관한 메시지는 예를 들어 노드가 3개가 있다면 1번 노드에 먼저 전송권한을 부여하고 1번 노드가 가지고 있는 데이터 개수만큼 데이터에 관한 정보를 CAN 코디네이터가 받으면 다시 2번 노드에 전송권한을 부여하고, 다시 같은 과정을 거쳐 3번 노드에 전송권한을 부여하는 방식이다.

[0124] 상기와 같은 권한 부여에 따라, CAN 코디네이터는 복수의 노드들 중 응답한 노드로부터, 해당 노드의 기능 및 동작 조건에 관한 요청 정보를 포함하는 기능별 제어 데이터 정보 메시지를 수신할 수 있다. 상기 해당 노드의 기능 및 동작 조건에 관한 요청 정보는 데이터의 기능, 송신 또는 수신 가능 여부, 기능의 제어 방법 및 정상 모드 운행시의 데이터 획득 위치 중 적어도 하나에 대한 요청을 포함할 수 있다.

[0125] 이렇게 각각의 노드들로부터 기능별 데이터 정보가 들어 있는 두 번째 메시지를 받은 CAN 코디네이터는 관련된 역할, 동작, 동작 방법, 동작을 하기 위한 위치, 긴급메시지 여부, ID 배치요청 등의 정보를 분석하여 저장하고, 상기 복수의 노드들의 각각의 기능 및 동작 조건을 설정할 수 있다.

[0126] 이렇게 분석된 정보 중 데이터에 있는 ID 배치 요청 사항에 맞게 CAN 코디네이터는 데이터 메시지의 ID를 일반 메시지 데이터 구간, 예를 들어 100~6FF 사이에 자동으로 배치할 것인가, 최우선으로 배치할 것인가, 아니면 특정한 데이터의 기능에 대한 ID의 앞/뒤에 배치할 것인가 등을 분석한 내용에 맞게 데이터 메시지의 ID를 부여하

게 된다. 또한 기존의 정상 ID 가 배치된 데이터 중에서도 제품의 성능향상이나 개선을 위해 ID 재배치 요청이 있는지 확인하고, 이때 ID 재배치 요청이 있으면 CAN 코디네이터가 해당 데이터의 요청에 맞게 ID를 재배치할 수도 있다.

[0127] 해당 노드의 기능 및 동작 조건에 관한 요청 정보는 긴급 메시지 전송 권한 요청을 더 포함할 수 있으므로, ID 배치에 더해 긴급 메시지 전송 권한 요청이 있는지 여부를 더 확인한다. 긴급 메시지 전송 권한 요청이 있으면 긴급 메시지 데이터 구간, 예를 들어, 000~0FF 사이에서 비어 있는 ID를 CAN 코디네이터가 긴급 정도에 따라 선택하여 해당 데이터 기능에 대해 100~6FF 사이의 ID와는 별도로 긴급메시지 건에 대하여 추가로 ID를 배치하게 된다. 즉, 이벤트 상황이 발생했을 때 평상시와는 다르게 긴급하게 보내야 하는 기능과 관련된 데이터 메시지의 경우에는 평상시 상태 등을 정보를 전송하는 100~6FF 사이의 ID와 긴급한 상황에 빠르게 전송해야 하는 긴급 메시지는 상기 100~6FF 사이의 ID 보다 우선순위를 가지는 000~0FF 사이의 ID가 배치되어 하나의 기능에 대한 데이터에 두 개의 ID가 배치될 수도 있다.

[0128] 이렇게 데이터의 역할 및 동작에 대한 부분과 ID 배치가 끝나면, CAN 코디네이터는 노드 각각의 기능 및 동작 조건에 대한 설정을 완료하고, 이러한 설정에 관한 정보를 포함하여 노드 설정 메시지를 복수의 노드들 중 응답한 노드로 송신할 수 있다. 즉, CAN 코디네이터는 데이터 메시지의 기능에 대한 동작정보, 정상모드일 때 송, 수신 할 수 있는 데이터 ID등을 포함하여 미리 정의되어 있는 기능의 순서에 맞게 차례대로 알려주는 노드 설정(Setup) 메시지를 보내게 된다. 셋업 메시지를 기능별로 모두 전송하게 되면, CAN 코디네이터는 마지막으로 설정 메시지 전송완료 및 등록모드 종료 메시지를 전송하게 된다. CAN 시스템의 특성상 브로드캐스팅이 되기 때문에 해당 데이터에 의하여 제어가 되는 노드를 비롯하여 그 기능을 사용해야 하는 노드들은 기능에 대한 데이터 부분을 선별하여, 그 데이터 메시지에 대한 정보를 셋업 메시지에 맞게 설정하고, 데이터에 대한 정보를 잃어버리지 않기 위해 저장을 한다.

[0129] 셋업 메시지에 맞게 설정을 완료한 노드들은 설정 완료 및 등록모드를 종료 메시지를 보내게 되고, 이러한 절차가 끝나게 되면 등록모드가 종료되게 된다. 즉, CAN 코디네이터는 복수의 노드들 중 응답한 노드로부터, 노드 설정 메시지에 따른 해당 노드의 기능 및 동작 조건의 설정이 완료되었음을 나타내는 등록 모드 완료 메시지를 수신할 수 있으며, 등록 모드 완료 메시지에 응답하여 등록 모드를 종료할 수 있다.

[0130] 등록 모드가 종료되면 CAN 코디네이터는 파워-오프(Power Off) 후 다시 파워-온(Power On)이 될 때까지 정상 CAN 시스템에는 관여 하지 않으며, 각각의 노드들은 CAN bus상에서 CAN 코디네이터가 기능에 맞게 부여한 데이터 ID와 역할, 동작 방법 등을 통하여 통신을 할 수 있다.

[0132] **나) CAN Device**

[0134] 도 14 는 상기 CAN 코디네이터에 의해 제어되는 CAN 디바이스의 작동에 관한 흐름도이다. 이하, 도 14 를 참조하여, CAN 디바이스 (노드) 의 작동과 관련하여 보다 상세히 설명한다. 처음 파워-온이 되면, 각 노드들은 등록 모드로 들어가고, CAN 코디네이터로부터 등록 모드 설정 시작 메시지 수신을 일정 시간동안 기다리게 된다. 이때, CAN 코디네이터의 고장 및 다양한 문제 발생으로 CAN 코디네이터로부터 등록 모드 설정 시작 메시지를 수신하지 못하게 되면 시스템 정지를 방지하기 위하여 정상 운행 모드로 전환하여 수행하게 된다.

[0135] 일정 시간 안에 CAN 코디네이터로부터 등록 모드 설정 시작 메시지를 수신하게 되면, 각 노드들은 미리 정해진 순서대로 노드의 존재 여부 및 그 노드가 가지고 있는 기능을 제어 할 수 있는 기능별 제어 데이터의 개수 정보를 첫 번째 데이터 메시지에 실어 CAN 코디네이터에게 전송하고, CAN 코디네이터로 노드의 기능을 제어 할 수 있는 기능별 제어 데이터 전송 권한 메시지를 수신하여, 기능별 제어 데이터 전송 권한을 부여 받기 위해 기다린다.

[0136] CAN 코디네이터가 첫 번째 메시지 수신 후 처리를 완료하면 노드에서 보낸 기능별 데이터에 대한 정보를 전송받기 위해 노드에 데이터 개수에 맞는 시간 동안 전송 권한을 부여하는 두 번째 메시지를 송신하게 된다.

[0137] 전송 권한을 받은 노드는 두 번째 메시지에 각 노드에서 갖고 있는 기능 데이터의 ID와 역할, 동작 등을 담은 각각의 노드의 기능 및 동작 조건에 관한 요청 정보를 포함하는 기능별 제어 데이터를 CAN 코디네이터에 순차적으로 전송한다.

[0138] CAN 코디네이터는 노드에서 전송된 두 번째 메시지를 갖고 노드에서 요청한 내용에 맞게 처리를 하며, 처리 완료 후 CAN 코디네이터는 처리된 내용을 데이터 형식에 맞게 정보를 담아 노드로 노드 설정 메시지를 전송한다. 그 메시지를 받은 각각의 해당 노드는 CAN 코디네이터가 지정한데로 노드에서 발생하는 데이터에 대한 ID, 필요한 데이터를 수신할 ID와 출력이 필요한 경우 어느 ID로 어떻게 보내야 하는 가를 확인하여 노드의 통신 부분에

대한 설정을 완료하고 저장한다. 물론 설정에 해당하지 않는 노드는 설정에 있어서 제외된다.

[0139] 설정을 완료 하면 각각의 노드들은 설정 완료 및 등록모드 완료 메시지를 보낸 후 정상 운행 모드로 진입하여, CAN 코디네이터에서 부여한 기능별 ID를 정보를 가지고 기존의 CAN 시스템과 동일하게 동작 한다.

[0141] **다) 데이터 형식**

[0143] 본 발명에서는 예시적으로, CAN 2.0A 를 기준으로 하여 시스템 구성을 하고 있다. 이는 본 발명에서 제안하는 시스템의 특성상 CAN 식별자의 영역을 지정 및 배치하여야 하는데, CAN 2.0A는 11비트의 식별자를 갖고 있기 때문에 총 식별자의 범위는 000-7FF 까지가 될 수 있다.

[0144] 이러한 식별자는 긴급메시지 데이터를 배치하기 위한 구간, 일반적인 메시지 데이터를 배치하기 위한 구간, 등록 모드에서 CAN 코디네이터와 노드들 간에 데이터 메시지가 송, 수신하기 위한 구간으로 구분될 수 있다.

[0145] 보다 많은 CAN 데이터가 필요한 경우 CAN 2.0A의 확장형인 CAN 2.0B 를 사용하면 CAN 2.0B는 29 비트의 식별자를 갖고 있기 때문에 11비트의 식별자를 갖고 있는 CAN 2.0A 에 비하여 더 큰 범위로 시스템을 구성 할 수 있다.

[0147] [표 4-1] 데이터 구분에 따른 ID 영역

ID	사용구간
000-0FF	긴급메시지 데이터 구간
100-6FF	일반메시지 데이터 구간
700-7FF	등록모드 데이터 구간

[0148]

[0150] 1) 긴급 메시지 데이터(Urgent message data) : 긴급 메시지의 경우 어떠한 메시지 보다 우선권이 있어야 하며, 이 우선권 획득으로 인하여 전송지연에 대한 자유로움을 가져야 한다.

[0151] 차량의 경우를 예를 들어 보면 추돌이 일어났을 때 그 사고의 충격이 커서 에어백(AIR BAG)의 전개가 필요한 경우 센서들의 값들이 다른 메시지의 우선순위에 밀려 전송지연이 일어나 에어백(AIR BAG)이 전개되지 않는다거나 적절하지 못하게 전개되는 경우 에어백에 대한 본래의 역할을 수행하지 못 할 뿐만 아니라 2차 사고로 이어져 더 큰 위험을 초래할 수 있다.

[0152] 예시로 든 상황은 극단적인 경우일 수도 있지만 실제로 발생할 수 있는 상황이며, 통신 시스템에서 적시성이 보장되지 않으면 예상되지 않은 상황이 발생할 수도 있다는 것을 의미한다. 그래서 긴급하게 전송되어야 하는, 즉, 다른 무엇보다도 적시성이 보장되어야 하는 데이터는 긴급 메시지 데이터 구간 (예들 들어, 000-0FF) 로 배치를 하여 식별자 우선순위에 의한 전송 지연이 발생할 수 있는 부분을 개선할 수 있다.

[0154] 2) 일반 메시지 데이터(Normal message data) : 일반적인 데이터 메시지는 정상적인 CAN 시스템이 운영되는 중에 사용하는 것으로서 이 메시지 데이터는 예를 들어 100-6FF 까지의 ID로 배치할 수 있다.

[0156] 3) 등록 모드 데이터(Registration mode data) : 처음 CAN 시스템이 파워-온 되면 등록 모드에 진입하게 되는데 이때, CAN 코디네이터는 700 의 ID 를 갖고 노드에 데이터 메시지를 전송할 수 있으며, CAN 시스템에 연결되어 있는 각각의 노드 들은 노드의 정보 메시지를 예를 들어 701-7FF 사이의 ID를 갖고 통신하게 될 수 있다.

[0158] 도 15 는 CAN 데이터의 기본 형식을 나타낸다. 이하, 도 15 를 참조하여 CAN Bus로 전송하는 CAN 데이터의 기본 형식, 즉 CAN 코디네이터와 노드들 간의 통신에 관한 데이터형식을 보다 구체적으로 설명한다.

[0159] 본 발명에서 예시적으로 사용하는 CAN 2.0A의 기준에 따라 식별자(ID)는 11비트를 갖고, 데이터 구간은 D0-D7까지 8개로 구성되어 있으며 Dn은 8비트로 구성되어 있어 총 D0-D7까지 8바이트로 구성될 수 있다.

[0160] 먼저, CAN 코디네이터에서 노드로 송신하는 메시지는 등록모드 시작 메시지, 노드의 기능별 제어 관련 데이터 전송 권한 메시지, 노드의 두 번째 메시지를 수신하여 요청에 맞게 설정한 후 보내주는 설정 메시지, 등록 모드 종료를 알려주는 종료 메시지로 총 4 가지이지만 등록 모드 시작 메시지와 등록 모드 종료 메시지는 형식이 동일하기 때문에 메시지들을 그룹화하면 3가지로 나눌 수 있다. 하기에서, 첨부된 도면 및 표와 함께 보다 구체적으로 설명한다.

[0162] 1) 코디네이터 메시지 전송형식1 (등록 모드 시작, 등록 모드 종료)

[0164] 도 16 은 코디네이터의 메시지 전송형식1 을 나타낸다. 여기서, 도 16 의 코디네이터의 메시지 전송형식1은 도 15의 CAN 데이터의 기본 형식을 비트별로 펼쳐서 표현한 것이고, 각 부분별 데이터 전송에 관한 내용을 포함하고 있다. 앞서 언급한 바와 같이 CAN 코디네이터의 데이터 전송 형식은 3가지가 있고 그 첫 번째는 등록 모드 시작, 등록 모드 종료 메시지로서, 2 가지 메시지의 형식이 동일하기 때문에 전송형식1 에서 함께 설명한다.

[0166] [표 4-2] 코디네이터 메시지 전송형식1 중 D0 데이터

D0									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	1	0	0	0	1	0x11	등록 모드 시작
0	0	0	1	0	0	1	0	0x12	노드의 기능 메시지 전송권한
0	0	0	1	0	0	1	1	0x13	노드 설정 메시지
0	0	0	1	0	1	0	0	0x14	등록모드 종료

[0167]

[0169] [표 4-2]의 코디네이터 메시지 전송형식1 중 D0 의 데이터 내용을 살펴보 면, 7-4 까지의 비트는 이 메시지를 보내는 것이 누구인가를 나타내는 송신자식별 부분이며, 구분 방법은 D0의 7-5 까지의 비트가 0이 되고, 4번째 비트가1이 되면 CAN 코디네이터가 노드에게 전송하는 메시지를 의미한다.

[0170] 3-0 까지의 비트는 지금 보내는 메시지가 어떤 메시지인 것인가를 알려주는부분이며 본 발명에서는 4가지의 메시지를 구분하여 표현하고 있다. 이렇게 D0의 3-0까지의 비트를 통해 동일 형식을 유지하는 메시지들을 구분하여 전송할 수 있고, 등록 모드 시작과 등록 모드 종료 메시지들은 단순하게 여부나 응답의 의미만 갖고 있기 때문에 D1-D7의 데이터에는 모두 Null 값('0')을 넣어 전송할 수 있다.

[0172] 2) 코디네이터 메시지 전송형식2 (노드의 기능 메시지 전송권한)

[0173] 코디네이터 메시지 전송형식2 는 노드의 기능을 제어하기 위하여 사용되는 데이터 정보를 CAN 코디네이터가 수신하기 위하여 각각의 노드에게 전송 권한을 부여하는 메시지로서 노드로부터 받은 첫 번째 데이터 메시지 안에 포함되어 있는 기능별 제어 데이터의 개수를 기준으로 전송 권한을 부여한다. 전송권한 부여에 관한 메시지 형식은 도 17과 같다. 도 17 은 코디네이터의 메시지 전송형식2 를 나타낸다. 전송형식의 D0 데이터는 앞서 설명한 [표 4-2]의 '노드의 기능 메시지 전송권한'에 맞게 설정되고 D1의 데이터 [표 4-3]과 같이 미리 지정된 노드의 번호를 나타내며, 각 노드의 전송 권한을 부여할 때 각각의 노드 번호를 나타낸다.

[0175] [표 4-3] 코디네이터 메시지 전송형식2 중 D1 데이터

D1									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	노드번호 1
0	0	0	0	0	0	0	1	0x01	노드번호 2
0	0	0	0	0	0	1	0	0x02	노드번호 3
0	0	0	0	0	0	1	1	0x03	노드번호 4
≈									노드번호 N
1	1	1	1	1	1	1	1	0xFF	노드번호 256

[0176]

[0178] 3) 코디네이터 메시지 전송형식3 (노드 설정 메시지)

[0179] 코디네이터 메시지 전송형식3 은 CAN 코디네이터가 노드에 보내는 메시지 중에서 가장 중요한 부분이기도 한데, 노드로부터 수신한 두 번째 메시지를 요청에 맞게 CAN 코디네이터가 배치 및 설정 과정을 거쳐 노드의 기능별 CAN 데이터를 설정할 수 있는 베이스를 제공하는 메시지이며, 형식은 도 18 과 같다. 도 18 은 코디네이터의 메시지 전송형식3 을 나타낸다.

[0180] D0 데이터의 7-4 비트는 송신자 식별 부분이고, 3-0 비트는 메시지 식별 부분으로 코디네이터의 메시지 전송형식1과 동일하다.

[0182] [표 4-4] 코디네이터 메시지 전송형식3 중 D1 데이터

D1									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	데이터의 기능 1
0	0	0	0	0	0	0	1	0x01	데이터의 기능 2
0	0	0	0	0	0	1	0	0x02	데이터의 기능 3
0	0	0	0	0	0	1	1	0x03	데이터의 기능 4
≈									데이터의 기능 N
1	1	1	1	1	1	1	1	0xFF	데이터의 기능 256

[0183]

[0185] D1 데이터는 [표 4-4]에서 볼 수 있듯이 데이터의 기능을 나타낸다. 데이터의 기능 정의는 8비트를 사용하고 있기 때문에 총 256개의 기능을 정의할 수 있으며, 이 데이터의 기능을 기준으로 CAN 코디네이터에서 송신한 데이터를 인식하고, 각각의 노드는 해당되는 기능에 대하여 ID 설정 및 기능을 제어하기 위한 송, 수신 방법 등의 정보를 획득하여 설정을 수행할 수 있다.

[0187] [표 4-5] 코디네이터 메시지 전송형식3 중 D2 데이터

D2									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	송/수신 가능, 제어(01:on/00:off)
0	0	0	0	0	0	0	1	0x01	송/수신 가능, 제어(00:on/01:off)
0	0	0	0	0	0	1	0	0x02	송/수신 가능, 제어(10:on/01:off)
0	0	0	0	0	0	1	1	0x03	송/수신 가능, 제어(01:on/10:off)
0	≈								송/수신 가능, 제어방법 사용자 지정
1	0	0	0	0	0	0	0	0x80	수신 가능, 수신(01:on/00:off)
1	0	0	0	0	0	0	1	0x81	수신 가능, 수신(카운터 형)
1	≈								수신가능, 수신방법 사용자 지정

[0188]

[0190] D2는 데이터 메시지의 송신, 수신 가능 여부와 그 기능을 제어하기 위한 방법에 대해 나타내고 있으며, 노드에서 수신된 두 번째 메시지의 있는 내용을 다른 노드에도 전달하기 위해 메시지에 담아 보내 주게 된다.

[0191] D2의 7번 비트는 송신 수신 모두 가능 또는 수신만 가능한 것인지를 나타내고 있고, 그 방법은 7번 비트가 '0' 이면 송신 수신 모두 가능, '1' 이면 수신만 가능 하다는 것을 의미한다.

[0192] 6-0번 비트는 제어 방법에 대한 기준을 말하는 데, 제어 방법에 대한 부분은사용자가 사용하고자 하는 용도에 따라 많은 방법이 제시되기 때문에 [표4-5]에 기제 되어 있는 내용은 차량에서 제어 방법으로 사용하는 것 중에서 예를 들면 On을 '1'로 사용할 하고, Off를 '0'으로 사용할 것인가 아니면 On을'0'으로 사용하고 Off를 '1'로 사용할 것인가 하는 부분과 모듈의 상태 정보 제공 시 '1'과 '0'으로 제공할 것인가 아니면, 엔진 RPM 등을 표시하는 것과 같이 카운터 정보를 이용할 것인가 등이 있다.

[0194] [표 4-6] 코디네이터 메시지 전송형식3 중 D3 데이터

D3									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	D0에 데이터 송/수신
0	0	0	0	0	0	1	0	0x02	D0-D2에 데이터 송/수신
1	0	0	0	1	0	0	0	0x88	D8에 데이터 송/수신
1	0	0	0	0	0	1	0	0x82	잘못된 설정

[0195]

[0197] D3 데이터는 CAN 정상 모드로 운행될 때, 송/수신 데이터를 D0-D7 중 어느 곳에서 데이터를 획득하고, 제어해야

하는지에 대하여 정의한다.

[0198] D3의 데이터 중 7-4 번째 비트는 처음 위치를 설정하고, 3-0 번째 비트는 종료되는 위치를 정의한다. [표 4-6]의 예를 들어 설명하면 데이터 제어 위치가 D0로 하나인 경우에는 7-4 번째 비트의 처음 위치를 '0'으로 하고, 3-0 번째 비트를 '0'으로 하여 D0 위치를 설정해 준다.

[0199] 데이터의 제어 위치가 D0-D2로 범위가 넓어지는 경우에는 7-4 번째 비트의 처음 위치를 '0'으로 하고, 3-0 번째 비트를 '2'로 설정하여 범위를 정해 준다.

[0200] 즉, 이러한 약속으로 인하여 7-4 번째 비트 설정 값이 3-0 번째 비트 설정값보다 클 수가 없고, 이를 어기게 되면 잘못된 설정이 된다.

[0201] 마지막으로 D4-D7에 대한 데이터인데, 총 32비트의 범위를 갖고 있고, 이는 CAN 2.0B 의 식별자가 29비트에도 대응할 수 있도록 정의 한 것이다. 이 중에서 D4 영역의 7번째 비트는 CAN 2.0A 또는 CAN 2.0B 사용에 관한 것으로서 CAN 2.0A를 사용하면 '1', CAN 2.0B를 사용하면 '0'을 설정하여 각각의 식별자 형식을 따른다. 본 발명에서는 CAN 2.0A의 11비트 식별자 형식을 사용하기 때문에 D4의 5-0번째 비트는 [표 4-7]상에서 사용 안함으로 표시한다.

[0202] 6번째 비트는 긴급메시지 유, 무에 관한 사항으로 '1'이면 긴급메시지 있음, '0'이면 긴급메시지 없음으로 정의한다.

[0204] [표 4-7] 코디네이터 메시지 전송형식3 중 D4-D7 데이터

[0205]

D4								메시지 설명
7	6	5	4	3	2	1	0	
0	0		0	0	0	0	0	2.0B(4-0번째 비트 사용), 긴급메시지 없음
0	1		0	0	0	0	0	2.0B(4-0번째 비트 사용), 긴급메시지 사용
1	0							2.0A(4-0번째 비트 사용안함), 긴급메시지 없음
1	1							2.0A(4-0번째 비트 사용안함), 긴급메시지 있음

[0206]

D5								메시지 설명
7	6	5	4	3	2	1	0	
								2.0A(7-0번째 비트 사용안함)
0	0	0	0	0	0	0	0	2.0B(7-0번째 비트 사용)

[0207]

D6								메시지 설명
7	6	5	4	3	2	1	0	
					0	0	0	2.0A(7-3번째 비트 사용안함)
10	0	0	0	0	0	0	0	2.0B(7-0번째 비트 사용)

[0208]

D7								메시지 설명
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	2.0A(7-0번째 비트 사용)
10	0	0	0	0	0	0	0	2.0B(7-0번째 비트 사용)

[0210] D4-D7 데이터는 노드에서 보낸 두 번째 메시지 데이터 요청을 기준으로 그 기능에 해당하는 데이터에 대하여 배치된 ID를 알려주는 영역이고, 이 배치된 ID를 가지고 정상 모드에서 노드 간의 제어를 위한 데이터 메시지의 식별자로 사용될 수 있다.

[0212] 그 다음으로 노드에서 CAN 코디네이터로 송신하는 메시지로는 등록 모드 시작 관련 첫 번째 메시지, 각각의 노드를 제어하기 위한 기능별 데이터의 정보가 있는 두 번째 메시지, CAN 코디네이터에 설정이 완료되었고 등록

모드 종료를 알리는 등록모드 종료 메시지로 총 3가지이고, 형식은 각각의 전송하고자 하는 메시지가 다르기 때문에 3가지 형식으로 나누어진다. 자세한 내용은 그림 및 표와 함께 설명하고자 한다.

[0214] 1) 노드 메시지 전송형식1 (등록모드 시작 메시지)

[0215] 노드 메시지 전송형식 1은 CAN 코디네이터로부터 등록 모드 시작이라는 최초 메시지를 받은 후 등록 모드 준비 완료라는 의미와 함께 그 노드가 가지고 있는 기능을 제어할 수 있는 데이터 개수를 CAN 코디네이터에게 알려주는 역할을 하는 메시지이다.

[0216] 도 19 는 노드 메시지 전송형식1 을 나타낸다. 도 19 에서 볼 수 있듯이 이 메시지는 보내는 곳과 메시지 의미 그리고 이 메시지를 보낸 노드 번호, 노드가 가지고 있는 기능을 제어하기 위한 메시지 개수로 구성될 수 있다.

[0218] [표 4-8] 노드 메시지 전송형식1 중 D0 데이터

D0									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	1	0	0	0	0	1	0x21	등록 모드 시작
0	0	1	0	0	0	1	0	0x22	기능별 데이터 정보 메시지
0	0	1	0	0	1	0	0	0x24	설정완료 및 종료 메시지

[0219]

[0221] [표 4-8]의 D0의 데이터 내용을 살펴보면, 7-4 까지의 비트는 이 메시지를 보내는 것이 누구인가를 나타내는 송신자 식별 부분이며, 구분 방법은 D0의 7-4 까지의 비트 중 5번째 비트만 '1' 값을 가지며 나머지 비트가 '0'이 설정되면 노드가 CAN 코디네이터에게 전송하는 메시지를 의미한다.

[0222] 3-0 까지의 비트는 지금 보내는 메시지가 어떤 메시지인 것인가를 알려주는 부분이며 3가지의 메시지를 구분하여 표현하고 있다. 이렇게 D0의 3-0까지의 비트를 통해 각각의 메시지에 따른 형식을 구분하여 전송할 수 있다.

[0224] [표 4-9] 노드 메시지 전송형식1 중 D1 데이터

D1									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	노드1
0	0	0	0	0	0	0	1	0x01	노드2
0	0	0	0	0	0	1	0	0x02	노드3
≈									노드N
1	1	1	1	1	1	1	1	0xFF	노드256

[0225]

[0227] [표 4-8]의 D1은 실질적인 제어 신호에 의하여 동작을 하는 노드의 번호에 관해 정의를 하고 있다. 노드의 수는 256개까지 설정 가능하고, 해당하는 노드 번호의 표시형식에 따라 미리 정의되어 있는 데이터를 D1에 담아 전송하면 CAN 코디네이터에서는 CAN Bus에 어떤 노드가 연결되어 있고, 차후 그 노드의 번호를 기억하여 시스템이 다시 파워-온 될 때 그 노드의 응답 여부에 따라 시스템 이상을 알려 조기에 점검을 받을 수 있도록 하는 기능에서 활용할 수 있다.

[0228] D2 데이터는 노드가 가지고 있는 기능에 대하여 제어할 수 있는 데이터 개수를 말하는 데, 노드의 기능별로 제어하는 데이터가 모두 다르다는 가정을 하면 기능의 개수와 그에 따른 제어 신호의 개수는 동일하다고 할 수 있다.

[0230] [표 4-10] 노드 메시지 전송형식1 중 D2 데이터

D1									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	기능 제어 데이터 없음
0	0	0	0	0	0	0	1	0x01	기능 제어 데이터 1개
0	0	0	0	0	0	1	0	0x02	기능 제어 데이터 2개
≈									기능 제어 데이터 N개
1	1	1	1	1	1	1	1	0xFF	기능 제어 데이터 255개

[0231]

[0233] 이러한 제어 데이터 개수는 [표 4-10]과 같이 표시하며, 초기에 CAN 코디 네이터에게 알려주는 이유는 전송형식 2에 해당하는 데이터 메시지를 송신할 때, 기능의 많고 적음에 따라 CAN 코디네이터에 전송하는 시간이 달라질 수 있기 때문에 데이터를 모두 전송할 때까지의 시간을 확보하기 위해서 알려주게 된다. 그리고 D3-D7의 데이터에는 모두 Null 값('0')을 넣어 전송한다.

[0235] 2) 노드 메시지 전송형식2(기능별 데이터 정보 메시지)

[0237] 도 20 은 노드 메시지 전송형식2 를 나타낸다. 노드(Node)의 메시지 전송형식2는 본 발명에서 제안하는 시스템에서 가장 중요하고 핵심적인 부분이기도 한데, 노드가 CAN 코디네이터에 전송하는 두 번째 메시지로도 20 에서 볼 수 있듯이 전송형식에 실어서 보내는 내용은 데이터의 기능, 컨트롤 방법, 컨트롤하는 위치, 긴급메시지 전송 권한 요청 여부, ID 배치에 대한 요청사항에 관한 정보를 담고 있다. CAN 코디네이터가 배치 및 설정 과정을 거쳐 노드의 CAN 데이터를 설정할 수 있는 정보를 제공하는 메시지이다. D0-D1 까지의 데이터 메시지는 노드의 전송형식1에서 설명한 바와 동일하다.

[0239] [표 4-11] 노드 메시지 전송형식2 중 D2 데이터

D2									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	데이터의 기능 1
0	0	0	0	0	0	0	1	0x01	데이터의 기능 2
0	0	0	0	0	0	1	0	0x02	데이터의 기능 3
0	0	0	0	0	0	1	1	0x03	데이터의 기능 4
≈									데이터의 기능 N
1	1	1	1	1	1	1	1	0xFF	데이터의 기능 256

[0240]

[0242] D2 데이터는 [표 4-11] 에서 볼 수 있듯이 데이터의 기능을 나타낸다. 데이터의 기능 정의는 8비트를 사용하고 있기 때문에 총 256개의 기능을 정의할 수 있으며, 이 데이터의 기능을 기준으로 CAN 코디네이터는 기능별 데이터를 인식하고, ID 배치 등 여러 기능을 수행해 나간다.

[0244] [표 4-12] 노드 메시지 전송형식2 중 D3 데이터

D3									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	송/수신 가능, 제어(01:on/00:off)
0	0	0	0	0	0	0	1	0x01	송/수신 가능, 제어(00:on/01:off)
0	≈								송/수신 가능, 제어방법 사용자 지정
1	0	0	0	0	0	0	0	0x80	수신 가능, 수신(01:on/00:off)
1	0	0	0	0	0	0	1	0x81	수신 가능, 수신(카운터 형)
1	≈								수신가능, 수신방법 사용자 지정

[0245]

[0247] [표 4-12]의 D3는 데이터 메시지의 송신, 수신 가능 여부와 그 기능을 제 어하기 위한 방법에 대해 나타내고

있으며, D3의 7번 비트는 송신 수신 모두 가 능 또는 수신만 가능한 것인지를 나타내고 있고, 그 방법은 7번 비트가 '0'이면 송신 수신 모두 가능, '1' 이면 수신만 가능하다는 것을 의미한다.

[0248] 6-0번 비트는 제어 방법에 대한 기준을 말하는 데, 이 제어 기준을 CAN 코 디네이터가 기록을 하고 세 번째 메시지인 노드 설정 메시지를 보낼 때 그 기능 을 사용하고자 하는 다른 노드들도 확인을 하여 정상 모드에서 제어 방법으로 활 용을 하게 된다. 제어 방법에 대한 부분은 각각의 시스템을 구성하는 방법 및제작하고자 하는 용도에 따라 많은 방법이 정의될 수 있다.

[0249] D4 데이터는 CAN 정상 모드로 운행될 때, 송/수신 데이터를 D0-D7 중 어느 곳에서 데이터를 획득하고, 언제 제어해야 하는지에 대하여 정의한다.

[0251] [표 4-13] 노드의 메시지 전송형식2 중 D4 데이터

D4									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	D0에 데이터 송/수신
0	0	0	0	0	0	1	0	0x02	D0-D2에 데이터 송/수신
1	0	0	0	1	0	0	0	0x88	D8에 데이터 송/수신
1	0	0	0	0	0	1	0	0x82	잘못된 설정

[0252]

[0254] [표 4-13]의 D4의 데이터 중 7-4 번째 비트는 처음 위치를 설정하고, 3-0 번째 비트는 종료되는 위치를 정의한다. 표 4-11의 예를 들어 설명하면 데이터 제어 위치가 D0로 하나인 경우에는 7-4 번째 비트의 처음 위치를 '0'으로 하고, 3-0 번째 비트를 '0'으로 하여 D0 위치를 설정해 준다.

[0255] 데이터의 제어 위치가 D0-D2로 범위가 넓어지는 경우에는 7-4 번째 비트의 처음위치를 '0'으로 하고, 3-0 번째 비트를 '2'로 설정하여 범위를 정해 준다.

[0256] 즉, 이러한 약속으로 인하여 7-4 번째 비트 설정 값이 3-0 번째 비트 설정값보다 클 수가 없고, 이를 어기게 되면 잘못된 설정이 된다.

[0258] [표 4-14] 노드의 메시지 전송형식2 중 D5 데이터

D5									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	긴급메시지 없음
0	0	0	0	0	0	0	1	0x01	긴급메시지 ID 추가 배치 요청

[0259]

[0261] D5의 메시지는 긴급메시지 전송 권한 요청에 관한 부분으로 평상 시에는 100-6FF의 ID를 사용하다가 긴급한 여러 가지 상황에 의하여 적시성이 더욱 보장 되어야 하는 메시지에 000-0FF의 ID를 추가로 부여한다.

[0262] 처음부터 000-0FF 사이의 메시지를 부여하지 않는 이유는 정말 긴급하게 메시지를 전송해야 할 때, 000-0FF 사이의 ID를 평상시 통신용 ID로 사용하고 있는 데이터가 있으면, 그 안에서 전송지연에 대한 문제점이 개선되지 않을 수 있어, 000-0FF 사이의 ID는 긴급한 경우만 사용함으로써 전송 지연에 대한 부담감을 줄이며 보다 적시성이 보장될 수 있도록 하기 위해 긴급메시지 건은 평상시 통신하는 메시지와는 별도로 추가로 배치하고자 한다.

[0263] [표 4-14]를 보면 D5에서 '0x00' 이면 긴급메시지 없음, '0x01'이면 긴급메 시지 추가 배치 요청이라고 정의하고 있다. 긴급메시지 요청 부분에 8비트의 많 은 경우를 부여한 이유에는 차 후 시스템에서 전송 프로토콜의 변경 없이 보다 더 강화된 기능의 추가 지원을 하기 위함이다.

[0265] [표 4-15] 노드의 메시지 전송형식2 중 D6 데이터

D6									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	자동배치(담당하는 노드의 순서)
0	0	0	0	0	0	0	1	0x01	최우선 배치
0	0	0	0	0	0	1	0	0x02	특정기능 앞에 배치(특정기능 지정필요)
0	0	0	0	0	0	1	1	0x03	특정기능 뒤에 배치(특정기능 지정필요)

[0266]

[0268] [표 4-15]의 D6는 데이터의 ID를 100-6FF 사이에 배치를 하는데, 어떻게 배치를 할 것인가에 대한 요청을 하는 구간이다. 즉, D6에 '0x00'으로 설정을 하면 CAN 코디네이터에 노드 존재 메시지 전송 순서, 다시 말하면 D1의 데이터 순서에 맞게 노드에서 담당하는 데이터를 순서에 맞게 ID 배치를 실행한다.

[0269] '0x01'로 설정을 하면 해당 데이터는 100-6FF의 ID 중 평상 동작 시 사용하 는 ID 구간에서 가장 최상위인 100을 ID로 부여받게 된다.

[0270] '0x02'로 설정을 하면 해당 데이터를 특정 기능 앞에 배치하게 되는데, 차량 의 예를 들어보면 해당 데이터가 차량 혼(Horn)과 관련된 부분이고, 특정 데이터는 차량의 헤드라이트를 담당 하는 부분이라면, 제작자가 판단하기에 헤드라이트 기능 보다 차량의 혼에 관한 데이터가 우선순위에 있어야겠다고 생각이 들면 관련 CAN 디바이스를 제작 시 특정 데이터 위치 앞이라고 지정해줄 수 있고 CAN 코디네이터는 그 요청에 맞게 ID를 배치한다.

[0271] 만약에 특정 기능을 하는 데이터 앞에 배치해 달라고 요청했을 때, 바로 앞 에 다른 데이터가 ID 를 배치받았다면, 특정 기능을 하는 데이터 ID를 한 단계 내리고, 그 ID를 요청한 ID 에 부여하게 된다.

[0273] [표 4-16] 노드의 메시지 전송형식2 중 D7 데이터

D7									메시지 설명
7	6	5	4	3	2	1	0	표시형식	
0	0	0	0	0	0	0	0	0x00	지정 데이터의 기능 1
0	0	0	0	0	0	0	1	0x01	지정 데이터의 기능 2
0	0	0	0	0	0	1	0	0x02	지정 데이터의 기능 3
0	0	0	0	0	0	1	1	0x03	지정 데이터의 기능 4
≈									지정 데이터의 기능 N
1	1	1	1	1	1	1	1	0xFF	지정 데이터의 기능 256

[0274]

[0276] [표 4-16]의 D7은 D6에서 '0x02' 또는 '0x03' 기능을 사용하였을 때 설정 하게 되는데, D2에서 정의된 데이터의 기능을 참고하여, 배치하고자 하는 곳에 데이터 기능을 지정하는 것이 되면 CAN 코디네이터가 참고를 하게 되는 메시지이다.

[0277] 만약, D6에서 '0x02' 또는 '0x03' 기능을 사용하지 않는다면 D7은 Null 값('0')을 넣어 전송한다.

[0279] 3) 노드 메시지 전송형식3 (설정완료 메시지)

[0281] 도 21 은 노드 메시지 전송형식3 을 나타낸다. 노드 메시지 전송형식3의 경우에는 CAN 코디네이터에서 설정메시지를 수신한 후 노드에서 설정을 하게 되고 그 것을 완료하게 되고, CAN 코디네이터로부터 등록모드 종료 메시지를 수신하면 CAN 코디네이터에 설정을 했다는 의미와 함께 등록 모드 완료 메시지를 보내는 것인데, 전송 방법은 도 21에서 제시하는 것과 같다. 즉 실질적으로 노드는 설정을 끝내고, 의미상 본 발명에서 제안하는 시스템에서 등록 모드를 종료하는 메시지로 활용되고 있다.

[0282] 노드 메시지 전송 형식3은 노드 메시지 전송형식1과 거의 유사한데 CAN 코 디네이터에게 노드의 기능을 제어하는 데이터 개수를 나타내는 D2데이터 부분이 빠졌다는 것만 빼면 전송형식 1과 동일하다.

[0283] D0와 D1의 내용은 노드 메시지 형식1과 동일하기 때문에 노드 메시지 형식 1의 내용으로 갈음하고, 노드 전송형식3 의 경우에도 단순히 설정 종료 및 등록 모드 완료에 대한 알림 메시지이기 때문에 특별한 다른 메시지를 담을 필요 없이 나머지 D2-D7 까지의 데이터는 모두 Null 값('0')을 넣어 전송한다.

[0285] **4.2. 실시간 보장(전송지연 개선)**

[0287] CAN 시스템에서 기능 제어 및 정보를 제공하는 데이터 메시지들 중에 적절한 시간에 데이터를 전송해야 하는 적시성과 무관한 데이터는 없을 것이고, 다만 얼마만큼의 적시성이 보장되어야 하는가에 대한 부분이 차이가 있을 것이다.

[0288] 즉, 여러 데이터 중에서도 보다 적시성이 확보되어야 하는 데이터가 있기 마련인데, 예를 들어 차량의 경우 자동차를 운행함에 있어 중요한 가속페달 및 브레이크 페달에 대한 정보 신호라든지, 차량 추돌 사고 시 탑승자의 안전을 보장하기 위해 있는 AIR-BAG 등의 데이터 신호가 있고, 이러한 데이터에 비하여 조금은 우선순위가 떨어지는 창문, 도어 열림 등의 일반적인 신호가 있다.

[0289] 이러한 데이터들을 일반적인 경우와 긴급메시지의 경우를 구분하여 일반적인 경우 100-6FF 까지의 ID 배치하고 긴급 메시지의 경우에는 000-0FF 사이의 ID를 추가 배치함으로써 전송지연을 개선하고자 한다.

[0290] 즉, 평상시에는 모듈 상태나 이상 여부 등의 상태 메시지 같은 경우에는 100-6FF의 ID를 갖고 통신을 하고, 사고발생과 같은 긴급한 상황이 발생하면 추가 배치된 000-0FF의 ID로 데이터를 전송한다.

[0291] 긴급메시지 ID 배치는 본 발명에서 제안한 시스템의 등록 모드 상태에서 노드가 CAN 코디네이터로 전송하는 두 번째 메시지에서 그 정보를 알려주게 되며, 노드에서 보내는 두 번째 메시지(기능별 데이터 정보 메시지)의 내용은 '4.1.1.1. 제안시스템 디바이스 타입정의에서 다) 데이터 형식'에 설명되어 있고, 그 내용에서 알 수 있듯이 노드에서 CAN 코디네이터로 보내는 두 번째 메시지의 D5 데이터 영역에 약속된 데이터 형식에 의해 설정을 하게 되면, CAN 코디네이터는 D2의 데이터에 정의된 기능에 평상시 통신할 수 있는 ID를 100-6FF 사이로 배치하고, 추가로 긴급메시지를 전송할 수 있는 ID를 000-0FF 사이에 부여하게 된다.

[0292] 또한, CAN 코디네이터는 세 번째 메시지인 노드 설정 메시지에서 현 시스템에서 사용하는 CAN 데이터에 대한 기능과 평상시의 ID, 긴급 상황 발생시 ID 등을 모든 노드에 알려주고, 해당 기능과 관련 있는 노드는 초기 시스템 설정 시 그 데이터를 기록하여 설정하고, 초기 시스템 설정이 아닌 경우에는 자신에 설정되어 있는 값과 비교한 후 제품 개선 등의 업그레이드 절차에 의하여 긴급메시지 전송 건에 대한 추가 메시지가 있으면 그에 맞는 설정을 하며, 특별한 것이 없으면 설정이 완료 및 등록 모드 종료 메시지를 CAN 코디네이터에게 보내게 된다.

[0293] CAN 정상 모드에서는 모든 데이터들이 100-6FF 사이의 ID를 갖고 일정한 간격으로 통신을 하게 될 것이고, 여러 상황에 의해 긴급 상황이 발생하면, 긴급메시지 전송 권한 요청에 의해 CAN 코디네이터로부터 추가 배치받은 000-0FF 사이의 ID를 통해 긴급 메시지를 전송하게 된다. 이는 평상시 통신을 하고 있는 데이터에 비해 CAN BUS 상에서 우선권을 갖게 되어 전송지연 개선의 효과가 있게 된다. 실질적으로 개선된 수치나 개선된 사항에 관한 부분은 5장 실험 및 성능 평가 부분에서 다뤄보기로 한다.

[0295] **4.3. Plug and Play**

[0297] 기존의 CAN 시스템에서는 제품의 기획에서 제작 후 출시까지 모든 여러 상황을 고려하여, 그에 맞는 각각의 노드 별 데이터 전송에 관한 ID 및 동작에 관한 사항 등을 고정적인 개념으로 접근해 왔다.

[0298] 이는 하나의 제품에 CAN 시스템이 적용되고 나면, 제품이 단종되기 전까지 CAN 시스템에 변화를 줄 만큼의 상황이 발생 되지 않았고, 특정 기업의 경우 제품의 성능 향상을 위한 업그레이드보다는, 성능이 향상된 신제품을 출시하는 쪽을 택하기 때문에, 어떤 것이 더 효율적 인가는 입장에 따라 달라지겠지만 이러한 이유들 때문에 CAN 시스템에 모듈을 추가하는 경우가 더욱 발생 되지 않는 것이 아닐까라는 생각을 하게 된다.

[0299] 그래서 CAN 시스템의 특성 중 하나인 단순하게 수신만 가능한 노드는 얼마든지 추가가 가능하다. 이것은 앞서 언급한 바와 같이 고정적인 개념이 적용되었기에 가능한 것인데, 단순하게 수신만 가능한 노드는 CAN 시스템의 특징 중 하나인 브로드캐스팅이라는 것 때문에 기존 CAN BUS에 연결되어 있는 노드들에 영향을 미치지 않고 동작이 가능한 것이다.

[0300] 만약에, 제품과 관련된 법률 변경으로 인하여 기존제품까지 소급 적용이 되어 CAN 시스템에 송신, 수신 모두 가능한 신규 CAN 디바이스를 추가하여 보정 및 개선해야 하는 경우 제조사들은 이것에 대응하기 위해 송, 수신이 신규 CAN 디바이스에서 수신 요청하는 데이터 또는 송신 요청하는 데이터 등을 적용시키기 위해 기존의 CAN BUS에 접속되어 있던 노드 중 해당사항이 있는 부분들은 개별적으로 업그레이드를 진행해야 하며, 신규 노드와 관련된 기존 노드 수가 많을수록 시간과 비용에 관한 손실은 커질 수밖에 없다.

[0301] 이에 본 발명에서는 송신, 수신을 모두 요청하는 신규 노드가 추가되더라도 CAN 코디네이터에 의한 관리가 가능

하기 때문에 기존노드에 추가로 업그레이드 없이 송, 수신을 요청하는 신규 노드가 CAN Bus에 연결되는 것으로도 기존 노드들과 유기적 통신이 가능한 CAN 시스템에서의 플러그 앤 플레이(Plug and Play) 기능을 포함한다.

[0302] CAN 시스템에서의 플러그 앤 플레이 기능은 본 발명에서 제안한 시스템의 동작 구간인 등록 모드에서 구현이 가능한데, 노드에서 CAN 코디네이터로 전송하는 두 번째 메시지(기능별 제어 데이터 정보 메시지)에 그 기술적 핵심 내용이 있다.

[0303] 노드에서 CAN 코디네이터로 보내는 두 번째 메시지인 기능별 제어 데이터 정보 메시지의 내용을 보면, 고정적인 ID 개념을 극복하기 위해 데이터의 기능과 동작 (수신 만 하는 데이터 또는, 송신, 수신 모두 가능한 데이터)에 관한 사항 그리고 8바이트의 데이터 중 어느 곳의 신호를 이용하여 해당 데이터의 정보를 획득하고 구동하는지에 대해 나타내고 있다. 이것은 신규 노드가 기존의 CAN 시스템에 접근하였을 때, 그 기능을 어떻게 이용해야 하는가에 대한 지표가 되기 때문에 CAN 코디네이터가 알려주는 메시지를 갖고 스스로 학습하여 CAN 시스템에 접속되는 효과를 볼 수 있다.

[0304] 첫 번째 메시지 구간에 신규 노드는 CAN 코디네이터에게 자신의 존재를 알리는 신호를 송신하게 되고, CAN 코디네이터는 기존의 노드 수와 비교하여, 노드의 개수 변화가 있음을 감지한다.

[0305] 즉, 신규 모듈의 추가 및 기능을 CAN 코디네이터가 인지를 하게 되고, CAN 코디네이터는 신규 노드의 기능에 대한 데이터 정보를 얻기 위해 전송권한을 부여한다. 전송권한을 부여받은 신규 노드는 두 번째 메시지인 기능별 데이터 정보 메시지를 보내게 되고, 그 정보를 수신한 CAN 코디네이터는 기능별 데이터를 관리하기 위하여, 데이터에 요청에 맞게 100-6FF 사이에 있는 ID로 배치시키며, 또한 긴급메시지 여부 등의 다른 요청을 파악하여 그에 맞는 설정 후 저장한다.

[0306] CAN 코디네이터는 노드 설정 메시지 구간에서 기존의 기능별 데이터 정보 및 신규 노드의 기능별 데이터 정보인 기능, 제어방법, 제어위치, 제어 ID 등의 내용을 담아 CAN Bus를 통해 송신하게 되고, CAN의 브로드캐스팅 특성상 CAN 시스템에 연결되어 있는 노드는 모든 데이터를 확인할 수 있게 된다.

[0307] 이러한 메시지를 수신한 노드들은 그 역할과 필요에 맞게 재설정을 하게 되고, 신규 모듈의 추가로 인하여 특별한 업그레이드 없이 기존의 CAN 노드들과 유기적으로 동작할 수 있게 된다. 이는 각각의 노드에 대한 기능 및 제어 방법을 갖고 있고 그 데이터를 관리하는 CAN 코디네이터라는 존재로 인하여 신규 모듈을 추가하는 상황이 발생해도 간단하게 적용될 수 있다.

[0309] ## 5. 실험 및 성능평가

[0311] ### 5.1. 실험환경구성

[0313] 본 발명에서 제안하는 CAN 시스템의 전송지연 개선에 대한 성능 평가 및 플러그 앤 플레이 기능을 구현하기 위해 CAN 코디네이터와 CAN 노드들을 제작하였다.

[0315] 도 22 는 CAN 테스트 모듈의 구성도이다. 성능 평가 및 구현 용도로 제작된 CAN 코디네이터와 CAN 노드는 하드웨어 상으로는 동일하고, 소프트웨어적인 부분만 각각의 역할에 따라서 다르게 적용된다. 그 이유는 본 발명에서 제안한 CAN 코디네이터의 경우도 CAN 시스템에서의 노드 중 하나에 속하기 때문에 특별한 하드웨어적인 장치 없이 각각의 노드들과 동일한 구성으로 되어 있다. 소프트웨어는 CAN 시스템에 연결되어 있는 노드들을 관리할 수 있도록 하는 CAN 코디네이터와 관련된 것, 그리고 CAN 시스템에 맞는 데이터 통신을 통해 각각의 노드들에 온-오프(On-Off) 동작 제어 및 상태 정보 알림 등을 하는 노드와 관련된 부분으로 나뉜다. 또한, 노드들 각각은 동작에 대한 역할에 따라서 각기 다른 소프트웨어가 적용된다.

[0316] 먼저, 하드웨어의 구성에 대해 살펴보면, 크게 CAN 컨트롤러를 내장하고 있는 마이크로 컨트롤러(Micro Controller), CAN 데이터 신호를 버스에 송신, 수신하는 트랜시버(Transceiver)로 구성되며, CAN 시스템에 연결되어 있는 노드들에게 제어 신호 송신을 하기 위한 여러 개의 스위치 및 제어 신호를 수신받아 구동 확인을 할 수 있는 LED와 부저 (Buzzer) 로 구성되어 있다. 마이크로 컨트롤러는 마이크로칩사의 18F66K80을 사용하였으며, 칩의 CAN 버스 모듈 관련 기능은 [표 5-1]과 같다 (비특허문헌 22 참조).

[0318] [표 5-1] 18F66K80의 CAN Bus Module Features

Device	Features
18F66K80	Conforms to CAN 2.0B Active Specification
	Three Operating Mode
	Message Bit Rates up to 1Mbps
	DeviceNet Data Byte Filter Support
	Six Programmable Receive/Transmit Buffers
	Three Dedicated Transmit Buffers with Prioritization
	Two Dedicated Receive Buffers
	16Full, 29Bit Acceptance Filters with Dynamic Association
	Three Full, 29Bit Acceptance Masks
	Automatic Remote Frame Handling
	Advanced Error Management Features

[0319]

[0321] 먼저, 18F66K80의 CAN 모듈의 기능 중 상대적으로 중요한 기능에 대하여 간단하게 설명하면, CAN 시스템에서 CAN 2.0B 의 사양을 지원한다. 즉 CAN 2.0A 를 포함하여 지원하고, 메시지 비트 레이트 (bit Rate) 는 최대 1Mbps 속도까지 가능하다. 그리고 CAN 트랜시버는 마이크로컨트롤러와 동일한 제작사인 마이크로 칩의 MCP2551 을 사용하였으며, 기능은 [표 5-2]와 같다 (비특허문헌 9 참조).

[0323] [표 5-2] MCP2551의 Features

Device	Features
MCP2551	Supports 1Mb/s operation
	Implements ISO-11898 standard physical layer requirements
	Suitable for 12V and 24V systems
	Externally-controlled slope for reduced RFI emissions
	Detection of ground fault(permanent dominant) on TXD input
	Power-on reset and voltage brown-out protection
	An unpowered node or brown-out event will not disturb the CAN bus
	Low current standby operation
	Protection against damage due to short-circuit conditions
	Protection against high-voltage transients
	Automatic thermal shutdown protection
	Up to 112 nodes can be connected
	High noise immunity due to differential bus implementation
	Temperature ranges (E) : -40℃ to +125℃

[0324]

[0326] CAN 트랜시버에서는 지원되는 동작 속도와 CAN 2.0A 만 지원이 가능한지 아니면 CAN 2.0B까지 지원이 가능한가를 확인할 필요가 있다. 이는 CAN 트랜시버를 선정할 때 마이크로컨트롤러의 CAN 모듈의 성능에 따라서 적절한 선택을 해야 하는데, 예를 들어 마이크로컨트롤러의 CAN 모듈이 1Mbps의 성능을 갖고 있는데, CAN 트랜시버가 1Mbps의 성능을 지원하지 못하는 경우 (SW CAN 트랜시버 같은 경우는 최대 83.3kbps까지 지원) 가 있기 때문에 CAN 시스템 관련 하드웨어를 설계할 때 주의해야 한다.

[0327] 이렇게 제작된 CAN 테스트 모듈은 도 23과 같으며, CAN 코디네이터로 사용되어질 테스트 모듈을 제외한 CAN 테스트 노드 모듈은 총 6개를 제작하였다. 도 23 은 CAN 테스트 노드(Node) 를 나타낸다.

[0328] 또한, CAN 데이터 메시지 관리를 위한 CAN 코디네이터와 CAN 데이터 메시지를 이용하여 온-오프(On-Off) 동작을 하는 CAN 테스트 노드들을 구현하기 위하여, 4장에서 제안한 시스템의 알고리즘에 맞게 프로그래밍을 하였으며, 데이터 메시지들이 알고리즘에 맞게 실제로 정확하게 송신 또는 수신하는 지를 확인하기 위해서 CAN 계측 장비인 Intrepid Control System의 NEO VI 장비와 Vehicle Spy 어플리케이션을 사용하였다.

[0329] 도 24 는 CAN 데이터 메시지 측정 장비를 나타낸다. 즉, CAN 데이터 메시지 계측을 위해 사용된 장비는 도 24와 같고, 이 장비는 CAN 버스 상에 어떤 데이터 메시지가 통신을 하고 있다면 그 메시지의 통신 시간, 식별자, 8바이트(Byte)의 데이터 등을 보여 주며, 이 장비로 각 노드의 기능을 담당하는 데이터가 버스 상에서 정상적으로 동작하는 것인가를 모니터링 할 수 있다.

[0330] 이 모니터링 기능을 이용하여 버스 상의 CAN 메시지들의 통신시간을 측정하여 전송지연이 되는 것을 확인할 수

있으며, CAN 코디네이터가 시스템에 연결되어 있는 각각의 노드들에게 알고리즘에 의한 순서대로 데이터 송수신을 하는 것을 모니터링 하고, 등록 모드 종료 후 각각의 노드의 스위치 등을 이용하여 정상적으로 기능제어 데이터가 송신되는지 확인하여 플러그 앤 플레이 기능이 구현되는 것을 알 수 있다.

[0331] 도 25 는 실험 환경 구성을 나타낸다. 앞에서 설명되었던 CAN 코디네이터로 활용될 테스트 노드를 제외한 CAN 모듈로 활용할 테스트 노드 6개 총 7개의 테스트 모듈을 활용하여, 도 25 와 같이 CAN 버스에 순차적으로 연결하였고, CAN 테스트 노드를 연결하는 데는 통신을 하기 위한 CAN 통신 라인과 5V 전원을 인가하기 위한 전원 라인으로 구성되어 있다.

[0332] 먼저, 전송 지연에 대한 실험을 진행하기 위하여 연결된 각각의 CAN 테스트 노드에서는 같은 시간에 CAN 데이터 메시지를 송신해야 한다. 그래서 전송지연 실험 과정에서는 노드마다 하나의 메시지만을 같은 시간에 송신하도록 설정하고, CAN Bus에 실험 대상 노드의 데이터보다 우선순위의 노드를 추가하며 그에 따른 전송 지연 시간을 CAN 데이터 메시지 측정 장비를 통하여 확인을 하였다. 또한, 긴급 메시지에 대한 부분은 노드에 추가로 구성되어 있는 지정된 스위치 입력을 통해 마이크로컨트롤러에 입력이 들어오면 송신을 하였으며, 긴급메시지가 다른 데이터 메시지에 비하여 우선권을 갖고 전송지연 개선이 가능한지도 측정하였다.

[0333] 그리고 CAN 코디네이터의 추가적인 기능 중 하나인 플러그 앤 플레이 기능에 대한 부분도 CAN 테스트 노드를 순차적으로 하나씩 연결하며, 제안된 시스템의 알고리즘에 맞게 데이터 메시지가 정상적으로 송, 수신 되는 부분 및 등록 모드 종료 후 정상 모드에서 CAN 코디네이터가 배치한 식별자를 이용하여 정상적인 통신을 하는 가를 측정 장비를 이용해 모니터링 하였다. 자세한 실험 방법 및 성능 개선에 대한 부분은 5.2. 절의 성능 평가 부분에서 설명하고자 한다.

[0335] **5.2. 성능평가**

[0337] 본 절에서는 CAN 코디네이터를 이용한 시스템에서 긴급 메시지에 대한 전송지연 개선 및 신규 노드 추가 시 다른 노드들에 신규모듈 추가 부분에 대한 업그레이드 없이 기존의 연결되어 있던 CAN 노드들과 유기적으로 함께 동작을 하게 되는 플러그 앤 플레이(Plug and Play) 기능을 구현하고 제안된 알고리즘에 맞게 개선 및 동작을 하는지에 대한 성능 평가를 하고자 한다.

[0338] 먼저 긴급 메시지에 대한 전송지연 개선에 대한 실험을 진행하였다. CAN 시스템에서 실제로 전송지연이 발생되는지 여부를 확인하기 위하여 전송지연이 일어 날 수 있는 환경을 실제로 제작한 CAN 테스트 노드를 통하여 재현하였다. 재현 방법은 각각의 CAN 테스트 노드에 하나의 메시지만 동일한 시간에 송신하도록 설정하였고 각각의 데이터에는 각기 다른 CAN 식별자를 부여하였으며, 그에 대한 데이터 메시지도 달리하였다. 식별자 및 데이터를 노드 순번에 따라 [표 5-3]과 같이 부여 하였다.

[0340] [표 5-3] 전송지연 재현을 위한 테스트 노드별 데이터 메시지

노드번호	식별자	메시지							
		D0	D1	D2	D3	D4	D5	D6	D7
노드1	100	01	00	00	00	00	00	00	00
노드2	110	01	10	00	00	00	00	00	00
노드3	120	01	20	00	00	00	00	00	00
노드4	130	01	30	00	00	00	00	00	00
노드5	140	01	40	00	00	00	00	00	00
노드6	150	01	50	00	00	00	00	00	00

[0341]

[0343] 각 노드에 순번에 따라 다르게 부여된 CAN 데이터 메시지는 동일하게 약 1초에 한 번씩 송신할 수 있도록 조건을 설정하였다. 하나의 CAN Bus에 1초에 한번 씩 노드들이 동시에 CAN 데이터 메시지를 전송하기 때문에 여러 개의 데이터 메시지가 충돌하게 되고 이 충돌을 중재하는 과정에서 우선순위가 밀리는 데이터는 전송지연이 일어날 것으로 가정하였다. 또한, 제작한 CAN 테스트 노드를 모두 활용하여 1개의 노드에서 6개로 순차적으로 증가시키며, 전송지연 시간이 얼마나 늘어나는지 여부를 측정 장비를 통해 측정하였다.

[0344] 도 26 은 CAN 데이터 전송지연을 나타낸다. 즉, 도 26은 총 4개의 CAN 테스트 노드가 각각 하나의 데이터를 100kbps의 속도로 [표 5-3]의 형식에 맞게 송신한 것을 CAN 데이터 메시지 측정 장비를 통하여 측정한 것이다.

측정된 결과를 보면 최초 1번에 100번 식별자를 갖고 01 00 00 00 00 00 00 00 데이터를 송신한 부분을 볼 수 있고 2-4번까지의 데이터 메시지가 1번과 동일하게 1초 마다 송신하는 것이 아니라 우선순위에서 밀려난 데이터 메시지는 바로 앞의 우선순위를 갖고 있는 데이터 메시지에 비해 약 1.2-1.7ms 전송지연이 발생하는 것을 볼 수 있다. 즉, 데이터 식별자의 우선순위가 1번에서 4번까지 순서로 정해 있고, 전송하고자 하는 데이터보다 바로 앞의 우선순위 데이터에 비해 전송지연이 1.5ms 정도가 전송지연이 발생한다고 가정하면, 최고 우선순위를 갖고 있는 1번 데이터 메시지에 비하여, 최저 우선순위를 갖고 있는 4번 데이터 메시지는 약 4.5ms의 전송 지연을 갖게 된다. [표 5-4]는 도 26 을 기반으로 작성하였다.

[0346] [표 5-4] CAN 데이터메시지의 전송지연 결과

Cycle	TX Node	ID	Data								Time	Delay
			D0	D1	D2	D3	D4	D5	D6	D7		
1	Node1	100	01	00	00	00	00	00	00	00	992.540ms	0ms
	Node2	110	01	10	00	00	00	00	00	00	993.773ms	1.233ms
	Node3	120	01	20	00	00	00	00	00	00	995.006ms	1.233ms
	Node4	130	01	30	00	00	00	00	00	00	996.239ms	1.233ms
2	Node1	100	01	00	00	00	00	00	00	00	990.539ms	0ms
	Node2	110	01	10	00	00	00	00	00	00	991.802ms	1.263ms
	Node3	120	01	20	00	00	00	00	00	00	993.735ms	1.933ms
	Node4	130	01	30	00	00	00	00	00	00	994.97ms	1.235ms
3	Node1	100	01	00	00	00	00	00	00	00	991.823ms	0ms
	Node2	110	01	10	00	00	00	00	00	00	993.057ms	1.234ms
	Node3	120	01	20	00	00	00	00	00	00	994.292ms	1.235ms
	Node4	130	01	30	00	00	00	00	00	00	995.527ms	1.235ms

[0347]

[0349] 도 27 은 데이터 메시지 증가에 따른 전송지연 시간을 나타낸다. 실험은 총 100회 이상을 하였지만 기록은 1-3차로 변경되는 전송지연에 대한 부분만 [표 5-4]에 기록하였다.

[0350] 도 27 에서, 실험에 따른 기록 [표 5-4]를 기준으로 테스트 노드를 2개 더 추가하여 동일한 전송지연 실험을 하였고, 그에 따른 평균값을 구하여 Node6에서 송신하는 데이터보다 우선순위가 높은 데이터 메시지 수가 증가함에 따른 전송지연 시간을 그래프로 표현한 것이다.

[0351] X축은 CAN 버스 상에 각각의 노드들이 하나의 데이터를 한 번에 송신하는 개수의 증가를 나타내고 Y축은 각각의 데이터들이 전송되는 시간을 나타낸다. 위에서 언급되었듯 한 번에 전송되는 데이터 개수가 증가함에 따라 최고 우선순위 데이터와 비교하여 순위가 낮은 데이터의 전송지연이 길게 발생함을 알 수 있다. 이에 본 발명에서 제안하는 긴급 메시지에 대한 전송 지연 개선에 대한 부분을 적용하면, 도 28과 같은 결과를 얻을 수 있다. 도 28은 긴급 메시지 건에 대한 전송지연 개선을 나타낸다.

[0353] [표 5-5] CAN 데이터메시지의 전송지연 개선 결과

Cycle	TX Node	ID	Data								Time	Delay
			D0	D1	D2	D3	D4	D5	D6	D7		
1	Node2	10	01	10	00	00	00	00	00	00	986.670ms	0ms
	Node1	100	01	00	00	00	00	00	00	00	988.709ms	2.039ms
	Node3	120	01	20	00	00	00	00	00	00	995.413ms	6.704ms
	Node4	130	01	30	00	00	00	00	00	00	998.014ms	2.601ms
2	Node2	10	01	10	00	00	00	00	00	00	984.900ms	0ms
	Node1	100	01	00	00	00	00	00	00	00	987.029ms	2.129ms
	Node3	120	01	20	00	00	00	00	00	00	995.028ms	7.999ms
	Node4	130	01	30	00	00	00	00	00	00	998.007ms	2.979ms

[0354]

[0356] 도 28 의 결과를 기반으로 기록한 [표 5-5]를 보면 100번 식별자 다음으로 110, 120, 130 순서로 식별자 중재에 의해 전송지연이 발생해야 하는데, 110번 식별자가 긴급메시지 이벤트가 발생하여(CAN 테스트 노드상의 스위치 입력이 발생), 긴급메시지 건에 대하여 000-0FF 사이에 추가 배치된 식별자인 010을 통하여 기존의 110의 식별자를 가지고 있는 데이터 01 10 00 00 00 00 00 00를 송신하기 때문에 기존의 다른 식별자에 비하여 우선적으로 전송되어 메시지 전송 순서가 010, 100, 120, 130 의 순서로 송신되는 것을 확인할 수 있다.

[0357] 이러한 긴급메시지 전송지연 개선에 대한 사항을 보다 명확하게 입증하기 위하여 한 번에 전송되는 데이터를 순차적으로 증가시키며 추가적인 실험을 하였고 그 결과를 도 29 의 그래프로 나타내었다. 도 29 는 특정 식별자의 긴급메시지 전송에 관련된 도면이다.

[0358] 실험상 전송 권한에서 가장 후순위를 갖고 있는 노드6 에서 발송하는 메시지에 긴급메시지 우선권을 부여 하였고, 노드1 에서 노드5 까지 순차적으로 데이터를 한 번에 버스에 전송하였다. 그 결과 노드6 의 막대를 보면 약 998.24ms 마다 다른 노드에서 전송하는 데이터에 비하여 최우선으로 전송되는 것을 볼 수 있었다. 이렇게 긴급메시지 건에 대하여 본 발명에서 제안한 시스템이 적용된 것과 적용되지 않은 것을 비교하여 보면 도 30 과 같다. 도 30 은 긴급메시지 전송지연 개선 시스템 적용과 미적용 비교를 나타낸다.

[0359] 시스템이 적용되지 않은 경우를 보면 Y축의 TX 데이터가 증가함에 따라 우선순위가 낮은 데이터는 전송지연 증가로 인하여 전송시간이 증가 됨을 볼 수 있다. 이것과 비교하여 긴급메시지 전송지연 개선 시스템이 적용된 경우에는 버스 상에 한 번에 전송되는 TX 데이터 수가 증가해도 긴급하게 전송되어야 하는 메시지는 우선권이 확보되어 다른 식별자에 비하여 우선되기 때문에 전송지연 없이 데이터가 전송되는 시간이 거의 일정한 것을 볼 수 있다.

[0360] 다음으로, 플러그 앤 플레이 기능 구현에 대해 실험방법 및 결과를 설명한다. CAN 코디네이터를 제외한 CAN 테스트 노드 4개를 이용하여, 최초 시스템이 구성되었을 때와 기존의 시스템에 신규 모듈이 추가되었을 때 두 가지 경우를 가정하여 실험하였으며, 구성한 시스템이 4장에서 제안한 알고리즘에 맞게 동작하는가를 CAN 데이터 메시지 측정 장비를 통해 확인하였다.

[0361] 도 31 은 CAN 코디네이터의 CAN 노드 배치 과정을 나타낸다. 먼저, 도 31은 최초 시스템이 구성되었고 CAN 코디네이터와 한 개의 노드가 연결되었을 때를 가정하고 실험한 것이며 CAN 코디네이터와 CAN 노드 간에 등록 모드에서 통신한 것을 CAN 데이터 메시지 측정 장비 상에서 기록된 것이고, 그 기록을 바탕으로 표로 나타낸 것이 [표 5-6]이다.

[0362] [표 5-6]을 보면 CAN 코디네이터는 700번의 식별자 그리고 노드는 701번의 식별자를 가지고 통신을 한다. 최초 CAN 시스템에 파워 온(Power-On)이 되면 CAN 코디네이터가 700번의 식별자를 갖고 11 00 00 00 00 00 00 00 라는 등록모드 시작 메시지를 보내고, 등록모드 시작 메시지를 코디네이터로 부터 받은 노드는 약 15ms의 차이를 두고 첫 번째 메시지를 송신한다.

[0364] [표 5-6] CAN코디네이터의 CAN 노드 배치과정 결과

Line	Sender	TX Time	ID	Data							
				D0	D1	D2	D3	D4	D5	D6	D7
1	Coordinator	39.878ms	700	11	00	00	00	00	00	00	00
2	Node	14.907ms	701	21	00	02	00	00	00	00	00
3	Coordinator	2.0749s	700	12	00	00	00	00	00	00	00
4	Node	11.270ms	701	22	00	00	00	11	01	00	00
5	Node	149.763ms	701	22	00	01	01	11	00	00	00
6	Coordinator	2.199674s	700	13	00	00	11	C0	00	01	11
7	Coordinator	1.045713s	700	13	01	01	11	80	00	01	21
8	Coordinator	2.03076S	700	14	00	00	00	00	00	00	00
9	Node	54.737ms	701	24	00	00	00	00	00	00	00

[0365]

[0367] 첫 번째 메시지에는 노드가 21 00 02 00 00 00 00 00 라는 등록모드 준비 완료 메시지에 D1의 00을 통해 노드의 번호를 알려주고, D2에 02를 통해 노드가 가지고 있는 기능 데이터의 수를 보낸다. 노드의 데이터 메시지를 수신한 CAN 코디네이터는 12 00 00 00 00 00 00 00 메시지를 송신하여, 데이터 메시지 전송권한을 부여하는데, D1에 00 이라는 노드 번호를 알려주어 해당 노드에게 전송 권한을 부여한다. 앞에서 데이터 메시지를 2개 가지고 있다고 코디네이터에게 알려준 노드는 코디네이터로부터 2개의 기능 데이터 형식을 보내는 동안 CAN 버스를 차지할 수 있는 권한을 갖는다. 권한을 부여받은 노드는 CAN 코디네이터에게 두 개의 기능 데이터에 관한 형식을 전송하고, 메시지를 수신받은 CAN 코디네이터는 그에 맞게 식별자를 배치한 후 설정 메시지를 노드에 보내고, 설정메시지 전송이 끝난 코디네이터는 설정메시지 전송완료 및 등록모드 종료 메시지를 보낸다. 설정 메시지를 받은 노드는 설정을 완료하고, 설정완료 및 등록모드 종료 메시지를 코디네이터에게 보내면서 등록 모드는 종료된다. 다음은 CAN 노드의 수를 점차 늘려가며 신규 노드가 추가되었을 때 CAN 코디네이터가 제안된 시스템에 맞게 각각의 노드들을 관리하여 플러그 앤 플레이 기능이 적용되는 것에 대하여 측정 장비를 통해 코디네이터와 노드간의 실질적인 통신 내용을 확인하였고 그 결과는 도 32 와 같다. 도 32 는 CAN 코디네이터의 CAN 노드 배치 과정2 를 나타낸다.

[0368] CAN 시스템에서 플러그 앤 플레이 기능이 동작되기 위하여 CAN 코디네이터가 CAN 노드들을 관리하기 위한 필요한 정보를 CAN 노드와의 통신을 통해 얻게 되는데, 실질적으로 측정된 도 32 를 바탕으로 플러그 앤 플레이 기능을 위한 통신 과정을 도 33 으로 표현하였다. 도 33 은 Plug and Play를 위한 통신과정을 나타낸다.

[0369] 등록 모드에서 CAN 코디네이터와 노드 간에 기본적인 통신과정은 안정성을 위하여 일원화되어 있기 때문에 플러그 앤 플레이 기능이 동작 되기 위하여 특별하게 추가되는 과정은 필요하지 않다. 즉, 초기 시스템, 기존 시스템, 신규노드 추가 등의 모든 기본 과정은 동일하게 동작을 하게 되며 과정 중 신규 노드 진입에 대한 처리만 추가되게 된다. 먼저, 등록 모드에 진입하기 위해서는 코디네이터가 등록 모드 시작이라는 메시지를 기존 노드와 신규에 전송하고 이것을 받은 각각의 노드들은 CAN 코디네이터에게 첫 번째 메시지를 보내게 된다. 이 노드가 보낸 첫 번째 메시지를 받은 CAN 코디네이터는 기록되어 있는 기존의 노드들과는 다른 기록되지 않은 신규노드가 CAN 버스에 진입되었고 또한 어떠한 기능을 몇 개 가지고 있는지를 확인할 수 있다. 이렇게 신규 노드가 감지되면 CAN 코디네이터에서는 노드 번호와 그 기능의 수를 기록하고, 기존의 노드들 및 신규노드에 각각의 기능에 대하여 설명할 수 있는 권한을 부여한다. 전송권한을 부여받은 노드들은 각각이 가지고 있는 기능에 대하여 설명한 메시지를 CAN 코디네이터에게 보내고, CAN 코디네이터는 노드가 보낸 기능 메시지에 대한 설명 및 요청에 맞게 식별자를 배치하여 모든 노드들이 수신할 수 있도록 기능별 설정메시지를 송신한다. 코디네이터가 설정메시지 송신을 완료하면, 설정메시지 송신 완료 및 등록 모드 종료 메시지를 송신하고, 설정메시지를 받은 각각의 노드들은 관련된 기능에 대한 데이터 메시지를 수신하여 그 기능을 사용하기 위해서 설정메시지에 맞게 수신 또는 전송 방법에 대하여 설정을 완료 하고, 설정 완료 및 등록 모드 종료 메시지를 보낸다.

[0370] 이러한 방식으로 여러 노드들은 CAN 코디네이터에게 기능에 관한 데이터 정보를 제공하고 CAN 코디네이터는 이러한 기능 메시지의 요청에 맞게 식별자(ID)를 배치하며, 모든 노드가 수신할 수 있도록 3번째 메시지인 설정메시지를 기능별로 순서에 맞게 송신하여 관련된 기능 제어 및 정보획득이 필요한 노드는 설정메시지를 수신하여 그에 맞는 설정을 할 수 있기 때문에 신규 노드가 기존이 CAN 시스템에 추가 되더라도 각각의 노드에 관한 정보를 재입력하기 위하여 업그레이드하는 불편함 없이 유기적으로 기존의 노드들과 연계되어 운영되는 플러그 앤

플레이 기능이 가능하게 된다.

[0372] **6. 결 론**

[0374] 기존의 CAN 시스템에 CAN 코디네이터(Coordinator)라는 관리자 차원의 모듈을 추가하여, CAN 시스템 상에 부족한 부분을 보완하고, 사용되어지는 데이터 메시지들을 보다 유용하게 관리하고자 하였다. 그 중에 CAN 데이터 메시지가 전송 중 다른 메시지와 충돌로 인한 전송지연 개선과 CAN 시스템에 송신, 수신이 가능한 신규 노드를 추가 하게 될 때, 기존이 CAN 시스템에 연결되어 있는 관련된 다른 노드까지 업그레이드를 진행하였는데, 그럴 필요가 없는 플러그 앤 플레이(Plug and Play) 기능을 구현하였다.

[0375] 먼저 전송지연 개선 부분은 동시에 여러 개의 데이터 메시지를 100kbps로 전송하게 되면, 메시지 우선순위 별로 약 1.5ms의 시간이 전송 지연되는 것을 볼 수 있었다. 이에 이벤트 발생 시 긴급하게 전송되어야 하는 데이터 메시지가 있는 경우 CAN 코디네이터에서 긴급메시지 전송 건에 대하여 추가적으로 ID를 배치하여 다른 메시지의 경우 보다 우선권을 주어 적절한 시간에 데이터 메시지를 전달할 수 있도록 하였다. 하지만 5장의 실험 환경에서와 같이 모든 노드들이 동일한 시간에 데이터 메시지를 계속 전송하고 있는 환경에서는 우선권을 갖고 있는 긴급메시지 전송으로 인하여 다른 메시지들은 우선권에서 밀려나는 현상 즉, 전송지연이 다른 데이터 메시지로 전가되는 현상이 일어나는 것을 확인할 수 있었다. 물론 실제 제품에 적용된 CAN 시스템의 경우에는 모든 데이터 송신이 같은 시간에 일어나는 것이 아니라 필요에 의하여 시간차를 두고 CAN 통신을 하는 것이기 때문에 만일의 상황에 대비하는 차원에서 긴급하게 전달해야 할 데이터 메시지에 대해서 우선권을 부여해도 실험환경과 같이 모든 메시지가 충돌되어 전송지연이 전가되는 상황이 나타날 확률은 매우 낮다.

[0376] 그리고 플러그 앤 플레이(Plug and Play) 기능에 관해서는 데이터의 기능, 동작방법, 동작을 위한 데이터 메시지 위치 등의 정보를 CAN 코디네이터가 모두 관리하기 때문에 기존의 시스템에 새로운 송, 수신을 하는 모듈이 진입을 해도 CAN 코디네이터가 가지고 있는 정보 등을 알려 줌으로써 신규 모듈은 기존에 존재하는 데이터에 관한 정보를 인지하고, 기존의 노드들과 유기적으로 운영될 수 있는 조건을 갖추게 되고 실제로 구현된 결과를 도출하였다.

[0377] 이렇듯 기존의 CAN 시스템에 CAN 코디네이터를 추가하여 CAN 시스템을 보완 사항을 개선하여, 다방면으로 활용할 수 있는 여건이 마련되었으며, 향후 다양한 연구를 통하여 CAN 시스템이 아닌 다른 시스템에서 사용하고 있는 기능 또는 새로운 혁신적인 기능을 추가하여 CAN 코디네이터를 가지고 있는 CAN 시스템이 더욱 진화 될 수 있을 것으로 기대된다.

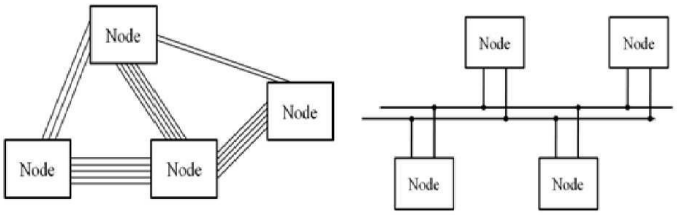
[0379] 상기에서는 본 발명을 바람직한 실시예를 참조하여 설명하였지만, 해당 기술분야의 숙련된 당업자는 하기의 특허청구범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 것이다.

도면

도면1

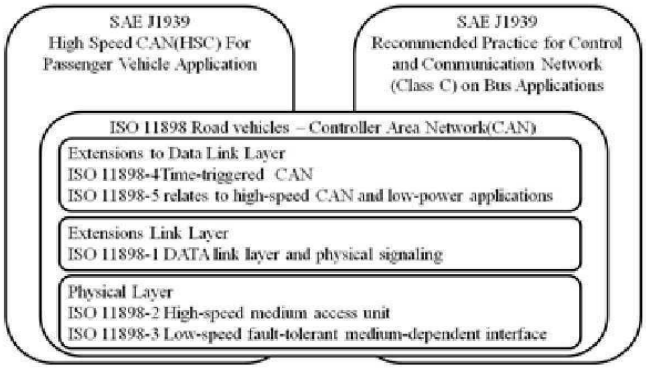


도면2

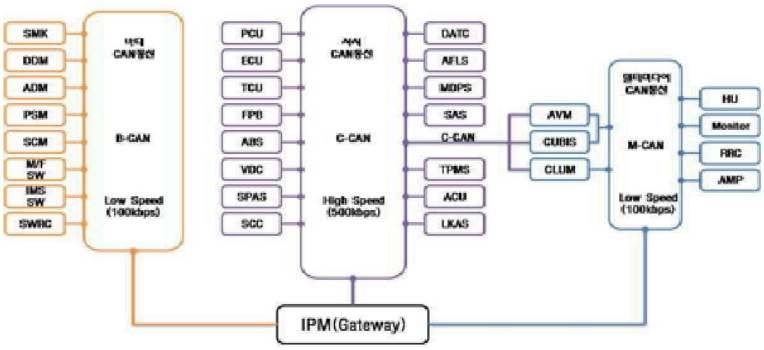


(a) CAN 시스템 미적용 (b) CAN 시스템 적용

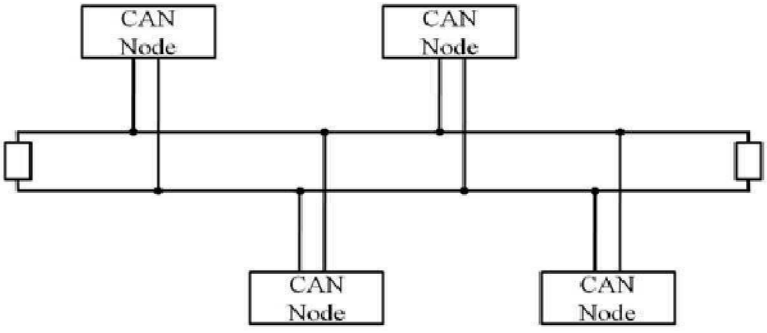
도면3



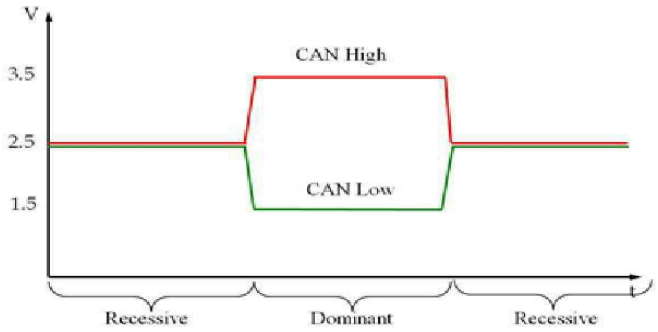
도면4



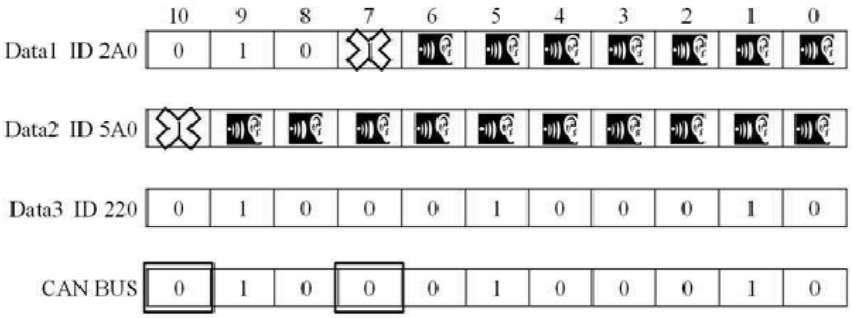
도면5



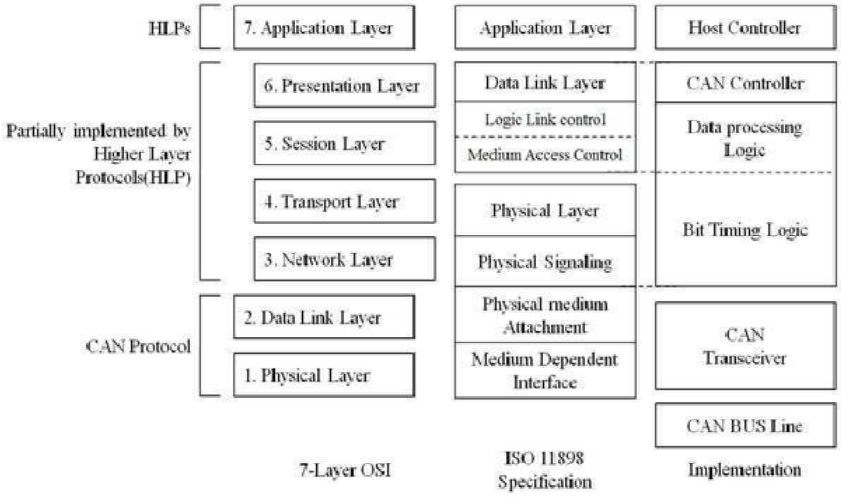
도면6


V
3.5
2.5
1.5
CAN High
CAN Low
Recessive
Dominant
Recessive

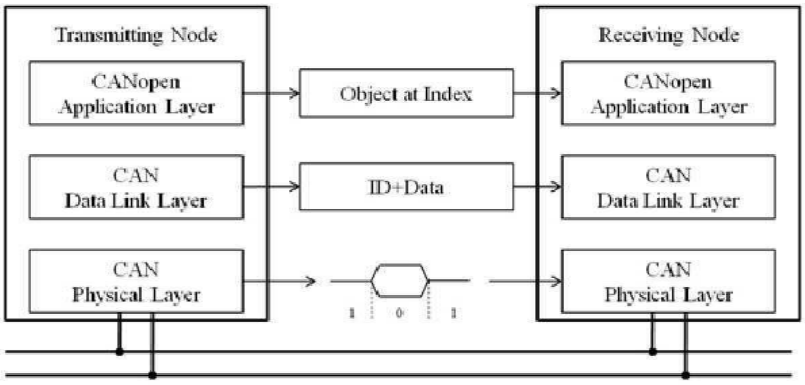
도면7


10 9 8 7 6 5 4 3 2 1 0
Data1 ID 2A0 0 1 0
Data2 ID 5A0
Data3 ID 220 0 1 0 0 0 1 0 0 0 1 0
CAN BUS 0 1 0 0 0 1 0 0 0 1 0

도면8

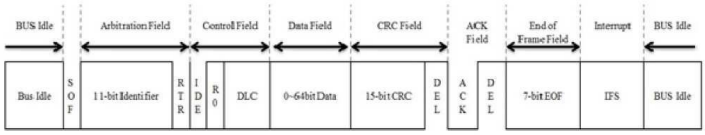

HLPs
7. Application Layer
Application Layer
Host Controller
6. Presentation Layer
Data Link Layer
CAN Controller
Partially implemented by Higher Layer Protocols(HLP)
5. Session Layer
Logic Link control
Medium Access Control
Data processing Logic
4. Transport Layer
Physical Layer
Bit Timing Logic
3. Network Layer
Physical Signaling
CAN Protocol
2. Data Link Layer
Physical medium Attachment
CAN Transceiver
1. Physical Layer
Medium Dependent Interface
CAN BUS Line
7-Layer OSI
ISO 11898 Specification
Implementation

도면9

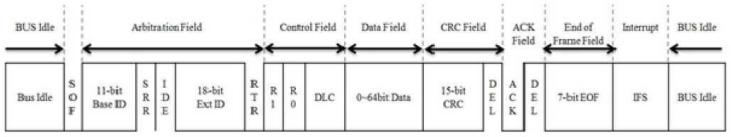


도면10

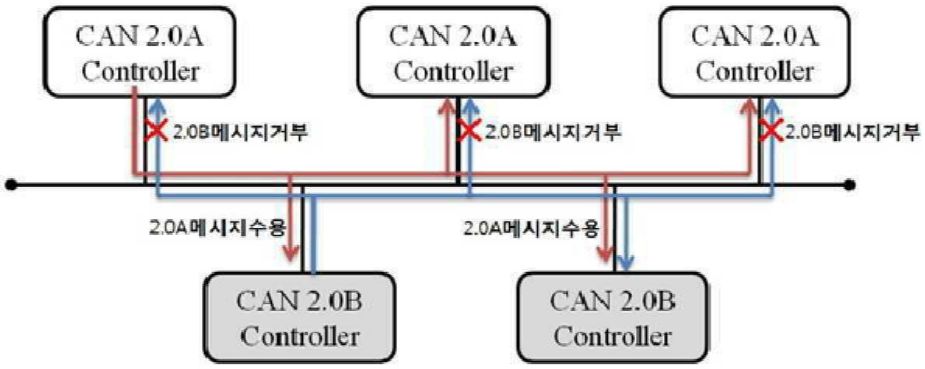
CAN 2.0A(Standard Format)



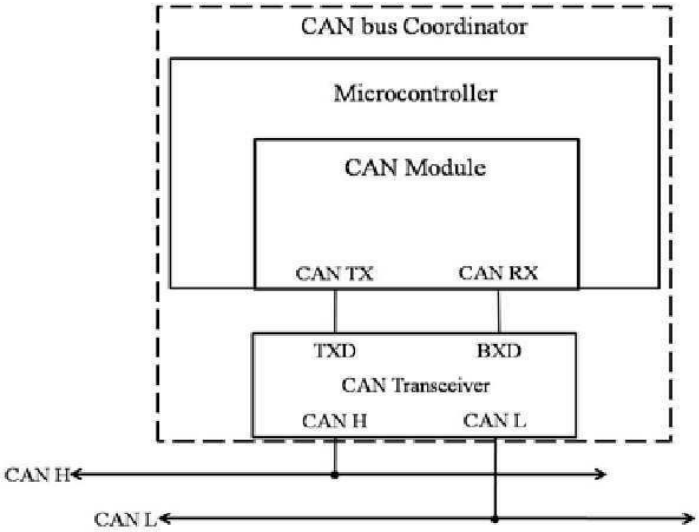
CAN 2.0B(Extended Format)



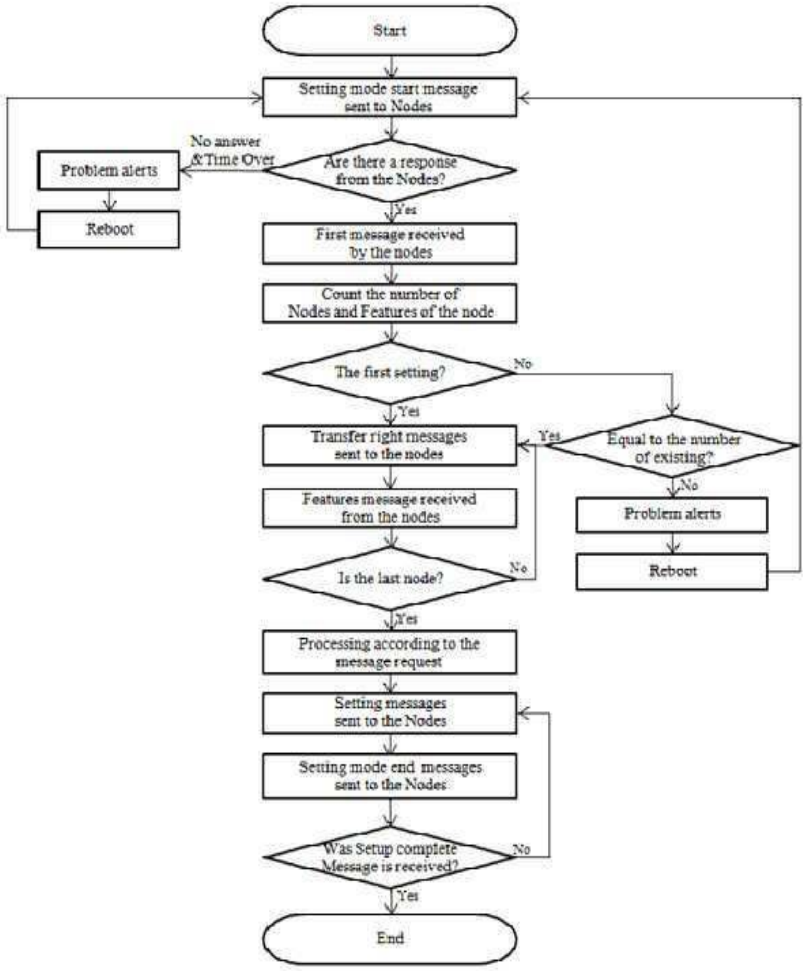
도면11



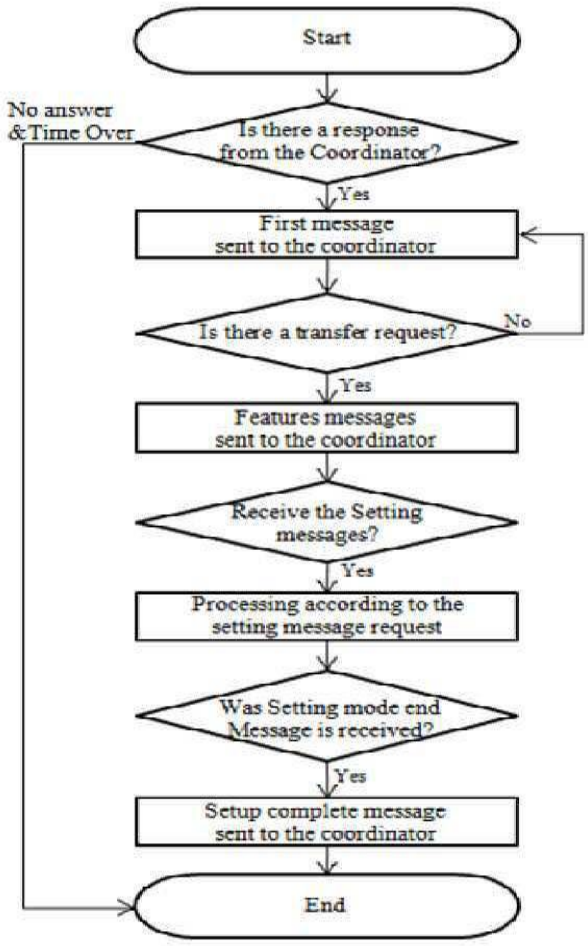
도면12


CAN bus Coordinator
Microcontroller
CAN Module
CAN TX
CAN RX
TXD
BXD
CAN Transceiver
CAN H
CAN L
CAN H
CAN L

도면13


Start
Setting mode start message sent to Nodes
No answer &Time Over
Are there a response from the Nodes?
Problem alerts
Reboot
Yes
First message received by the nodes
Count the number of Nodes and Features of the node
The first setting?
No
Yes
Transfer right messages sent to the nodes
Yes
Equal to the number of existing?
No
Problem alerts
Reboot
Features message received from the nodes
Is the last node?
No
Yes
Processing according to the message request
Setting messages sent to the Nodes
Setting mode end messages sent to the Nodes
Was Setup complete Message is received?
No
Yes
End

도면14



도면15



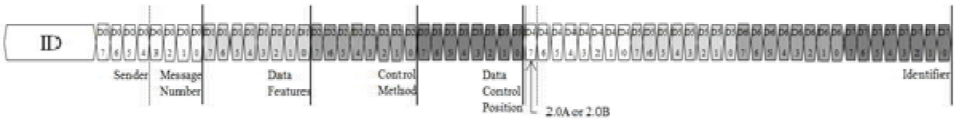
도면16



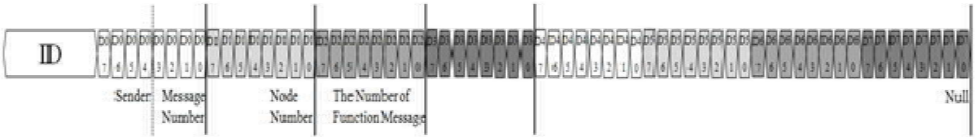
도면17



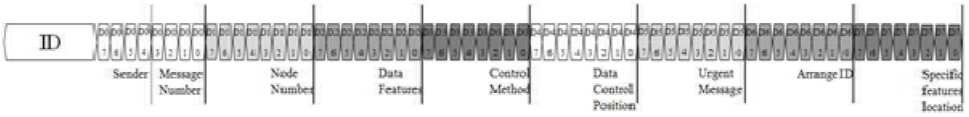
도면18



도면19



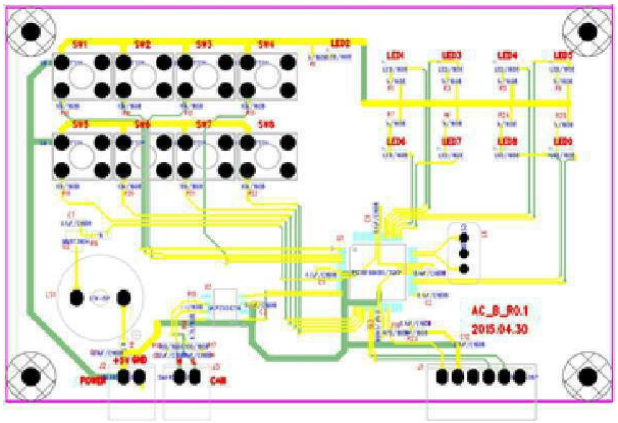
도면20



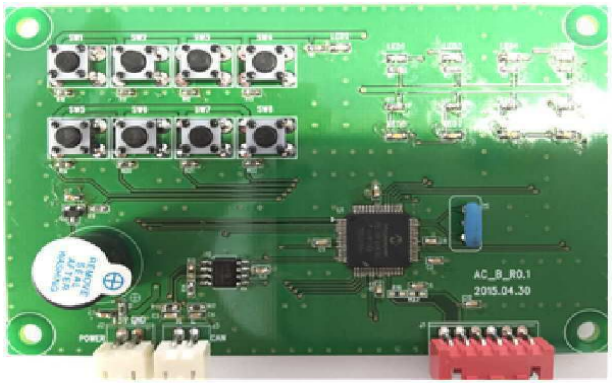
도면21



도면22



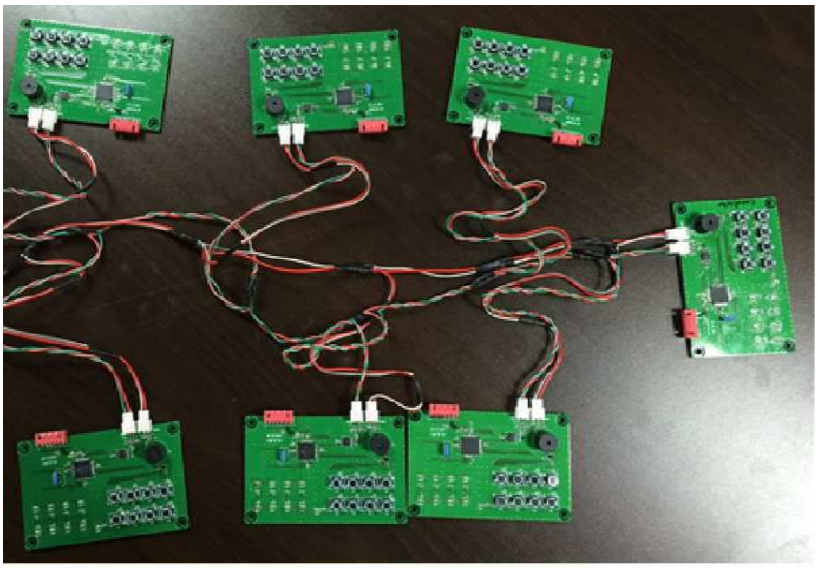
도면23



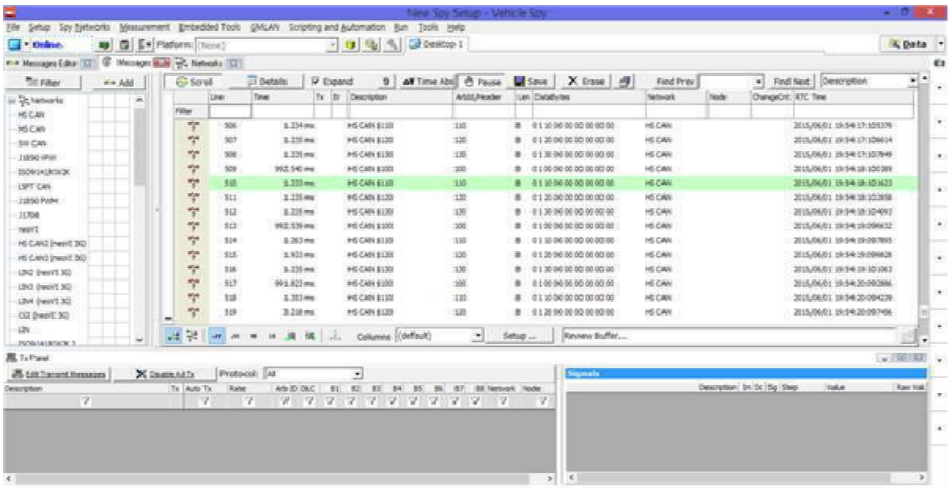
도면24



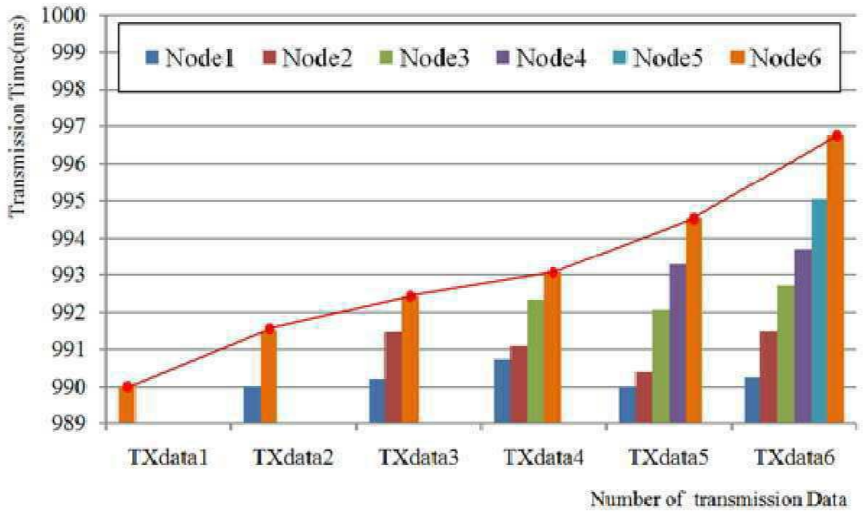
도면25



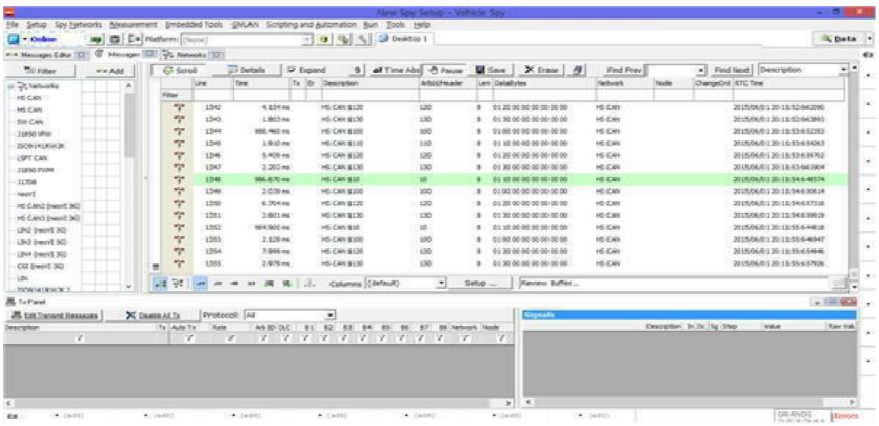
도면26



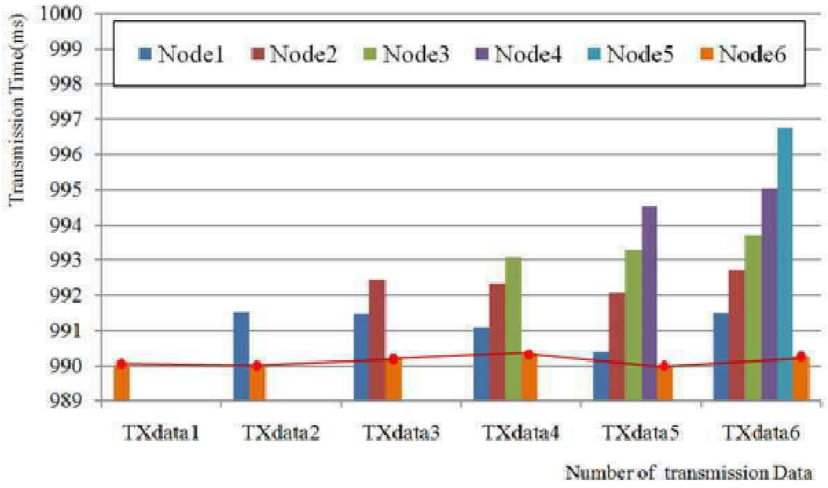
도면27



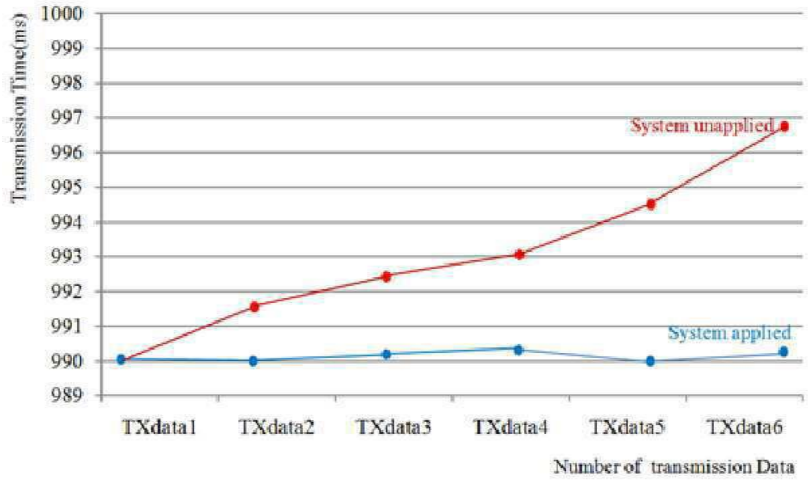
도면28



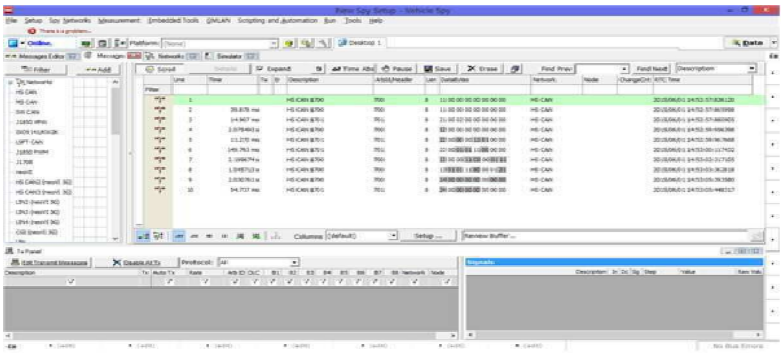
도면29



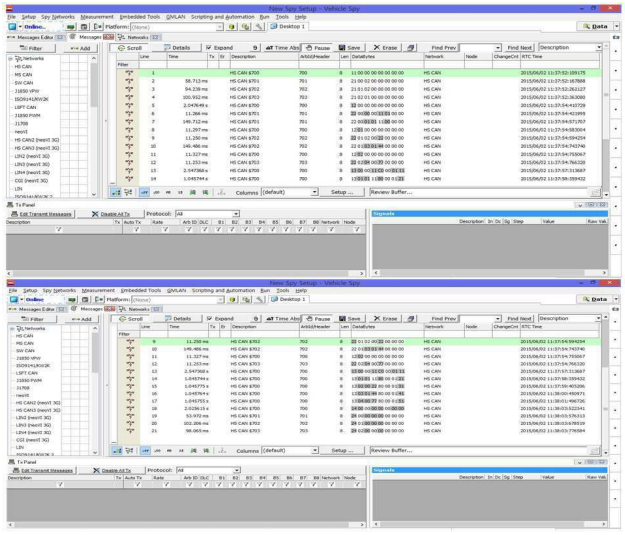
도면30



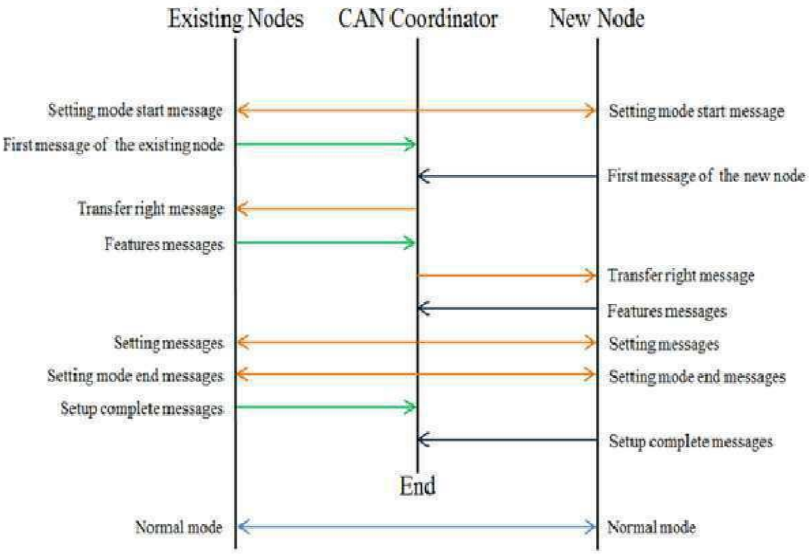
도면31



도면32



도면33


Existing Nodes
CAN Coordinator
New Node
Setting mode start message
Setting mode start message
First message of the existing node
First message of the new node
Transfer right message
Features messages
Transfer right message
Features messages
Setting messages
Setting messages
Setting mode end messages
Setting mode end messages
Setup complete messages
Setup complete messages
End
Normal mode
Normal mode