논문 2010-1-10

# 오류검출이 가능한 임베디드 시스템용 CAN통신 인터페이스 설계

# Design of CAN Communication Interface possible for Error Detection that use for Embedded System

안종영*, 김성수**, 김영자***, 박상정****, 허강인*****

**Jong-Young Ahn*, Sung-Su Kim**, Young-Ja Kim***, Sang-Jung Park****, Kang-In Hur*****

**요 약** 현재 CAN(controller Area Network)통신은 노이즈에 강인한 시리얼통신으로 전자모듈에 많이 사용하고 있다. 특히, CAN통신은 자동차 전자제어모듈, 엔진제어유닛, 센서모듈 등 자동차 부분에 많이 사용되는 방식 이지만 CAN 통신의 연결노드 우선순위와 오류가 낮아 공장자동화 라인에도 적용되고 있다. CAN통신은 기본적으로 전기 노이즈에 강인하여 적용가능한 분야가 다양하다고 사료된다. 본 연구에서는 High-Speed, Full-CAN에서 2개의 CAN Node를 사용하여 임베디드 시스템용으로 오류 검출이 가능한 CAN통신 인터페이스를 설계 하였다.

**Abstract** Now the CAN(controller Area Network) is using electronic modules as a serial communication which is very robust to noise. Especially the CAN is using for automotive part that very popular in which automotive electronic control module, engine controller unit, sensor modules, etc. but the CAN has the order of priority to linking node and also has fault confinement so using in these features that is applied to in factory automation product line. The CAN communication is basically very robust to electric noise so varied applying to others part. In this paper, we suggest to CAN interface for embedded system that is possible for error detection using two CAN nodes on Hi-speed, full-CAN.

**Key Words :** CAN, Embedded system, Multi-CAN, Error detection

## Ⅰ. 서 론

현재 통신 분야에서도 오픈시스템 네트워크에 연구성과를 보이고 있다. 생산의 비용을 절감 및 시스템의 성능을 최적화하려는 관심에 따라 경쟁성을 유지하기 위해 많은 제조업체들은 오픈 시스템의 네트워크에 투자를 하고 있다. 특히, MAP(Manufacturing Automation Protocol)은 하위 레벨의 센서나 작용기(actuator), 각종 제어기기 등의 비용이나 실시간 특성, 규격으로 인한 호환성의 문제가 대두 되었고, 이러한 문제해결의 필요성으로 새로운 국제 표준 네트워크가 필요했고 그 결과 하위 레벨의 통신 네트워크 필드버스 시스템을 개발하게 되였다.

CAN(Controller Area Network)은 1988년 Bosch와 Intel에서 차량용 네트워크 시스템을 목적으로 개발되었으며 물리 계층과 데이터 링크 계층만으로 구성되어 있다. 현재 ISO 11898 및 11519-2 표준안으로 지정되어 있

*정회원, 한국폴리텍2대학 컴퓨터 정보과 (동아대 박사과정)
**정회원, 한국폴리텍2대학 컴퓨터 정보과
***한국폴리텍2대학 컴퓨터 정보과
***한국폴리텍2대학 컴퓨터 정보과
*****동아대학교 전자공학과 교수
접수일자 2010.1.15, 수정일자 2010.2.10

다. CAN의 응용분야들은 데이터 링크 계층에서 규정된 기본 기능만으로는 수행할 수 없는 많은 서비스를 필요로 하기 때문에 응용계층에 관한 연구가 수행되었다.

CAN통신은 자동차 ECU(Electronics controller Unit) 간의 정보공유의 필요성과 늘어나는 차량 센서의 공용화, 독립적인 성향의 전자 모듈간의 네트워크의 요구로 개발되었으면 특히 Noise환경에 강인한 통신의 필요에서 개발의 의미가 깊다고 할 수 있다.

현재 각종 자동화 산업 분야로 적용범위가 확장된 방식으로 자동차뿐만 아니라 FA나 선박 의료기기, 산업기기 등 다방면에서 도입이 진행되고 있다. 전송 속도는 최대 1Mbit.sec이며 전송방식은 이벤트가 발생할 때 시스템이 작동 되는 Event Trigger형과 미리 정해진 시간이 되면 시스템이 작동하는 Time Trigger 형으로 분류 되어 진다. CAN에서는 1개의 Node가 통신경로를 점거하여 지연이 발생할 수 있다.

따라서 본 논문은 기존의 CAN통신의 오류를 판단할 수 있도록 2개의 CAN node를 사용하여 1개의 통신 Node에서 전송되는 데이터를 또 다른 Node로 feed back 받아서 오류 검출을 가능하게 하였다.[1]

## II. CAN 통신

CAN 프로토콜의 특징은 멀티 마스터구조를 가지며 메시지 ID에 의하여 Bus access우선순위가 주어진다. 전송 중단 재개 후 자동 재전송의 특징을 갖는다. 기본적으로 Bitwise arbitration 방법으로 통신한다.

CAN은 OSI 1계층인 물리계층과 2계층에 해당하는 데이터 링크 계층에 관한 표준이 정의되어 있다. 물리계층에 관한 표준이 실제로 신호가 어떻게 전송되는가에 대한 정의가 되어있다. 데이터 링크 계층은 CAN에서 특별하게 전달(transfer) 계층과 객체(object) 계층으로 구분된다. 전달 계층은 CAN 프로토콜에서 커널과 같은 역할을 하며 전송되는 메시지를 받았을 경우 이를 객체 계층에 알리고 객체 계층에서 수신해야 하는 메시지를 받는다.

전달 계층은 비트 타이밍과 동기화, 메시지 프레임화, 조정(arbitration), 확인(acknowledgement), 에러의 감지와 신호를 담당한다. 객체 계층은 메시지 필터링과 핸들링에 관련한 사항을 담당한다. 전송계층에서 이루어지는 CAN의 정보전달(information routing)은 시스템 유연성 즉, CAN 네트워크의 노드나 응용계층에 소프트웨어/하드웨어적 수정 없이 노드를 부가할 수 있다.

그리고, 메시지 전달은 식별자(identifier)에 의한 메시지 필터링이 가능하다.

데이터를 필요로 하는 노드에서는 다른 노드로 데이터를 요구하는 메시지를 보낸다. 데이터 프레임과 원격 프레임은 같은 인식자를 사용하여 조정되며 메시지 필터링으로 한 새 이상의 노드가 수신 가능하며 같은 메시지로 동시에 동작이 가능하다. 또한, CAN네트워크에서 메시지는 동시에 모든 노드에 수신된다. 그러므로 멀티캐스트와 에러 핸들링 개념으로부터 데이터의 일관성을 얻는다. 시스템마다 설정값이 다를 수 있으나 한 시스템 내에서의 속도는 단일하며 고정되어 있다.[2]

두 개 이상의 노드에서 동시에 버스접근을 시도했을 경우 충돌이 발생하며 이 충돌은 인식자를 사용한 비트와이즈(bitwise) 조정에 의해 해결된다. 데이터 전달시 전송의 보장을 위해 에러감지 신호, 자가진단이 각 노드에서 구현되어지며 변형된 메시지(corrupted message)가 들어올 경우 노드에서 에러를 인식하여 표시하면 자동으로 재전송을 하게 된다. 최대 복원시간은 29비트가 소요된다. CAN 노드에서 파라미터 실패가 나타나게 되면 그 오류를 낸 노드는 동작을 멈추도록 하며 모든 받는 쪽에서는 메시지가 들어올 때마다 올바른 것인지 점검하게 되는데, 올바른 메시지이면 인식하고 잘못된 것이면 표시하게 된다.

버스가 유휴상태(idle)일 때 전송할 메시지를 가진 어느 유닛은 전송을 시작할 수 있다. 동시에 두 개 이상의 노드에서 메시지 전송을 시도할 경우 높은 우선순위의 메시지가 우선권을 가지고 전송된다.

CAN은 BasicCAN과 FullCAN의 두 개의 일반적인 접근법이 있다. 데이터가 처리되는 방식에서 서로 다른데 BasicCAN은 CAN메세지 전송과 수신을 처리 하기위한 호스트 CPU를 필요로 하나 FullCAN은 메시지 전송, 메시지 수신 그리고, 최대 16개 메시지들의 저장을 CAN 컨트롤러에서 처리하는 방식이다.

또한, 표준 CAN에서는 식별자들이 11비트 길이를 가지며 확장 CAN에서는 식별자들이 29비트 길이를 갖게 된다.

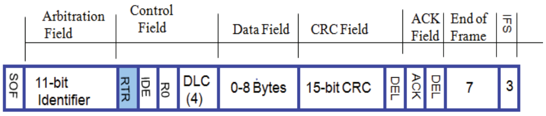


그림 1. CAN 2.0A 표준 데이터 프레임
Fig 1. CAN 2.0A Standard Frame

CAN 프로토콜 버전 2.0 에서 명시된 것에 의하면, V2.0A 로 컴파일하는 CAN 컨트롤러는 반드시 11-비트 식별자를 가진다. 반면 V2.0B 에서는, 11비트 또는 29비트 아무 것이나 될 수 있다.. V2.0B active를 이용하면, CAN 컨트롤러는 표준 포맷과 확장 포맷 모두를 전송하고 수신할 수 있다. CAN 2.0A 는 표준 CAN을 위한 규격이고, CAN 2.0B 는 확장 CAN을 위한 규격이다.[3]

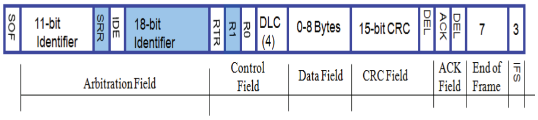


그림 2. CAN 2.0B 확장 데이터 프레임
Fig 2. CAN 2.0B Extended Standard Frame

클래스 A 네트워크들은 전자 부트 작동기(electronic boot release) 등과 같은 편의 기능 또는 고급 기능들에 일반적으로 사용되며 10 kbit/s 미만으로 작동 ,클래스 B 네트워크 10 ~ 125 kbit/s 클래스 C 네트워크들은 파워트레인, 안정성 제어(ABS, 견인 제어, 액티브 서스펜션), 엔진 관리, 변속 같은 실시간 제어 애플리케이션 정보의 빠른 응답 시간 또는 전송이 필요한 모든 애플리케이션 등에서 사용되며 이것들은 125 kbit/s부터 최대 1Mbit/s로 작동한다. 클래스 D 네트워크들은 인터넷, 디지털 TV, x-by-wire 같은 애플리케이션들에서 사용되며, 일반적으로 1Mbit/s 이상의 빠른 속도에서 실행된다. 통상 500Kbit/s 이상을 Hi-CAN, 이하를 Low-CAN 이라고 한다.[4]

## Ⅲ. Feed Back Multi-CAN

CAN은 부여된 ID에 따라서 우선순위가 주어지고 메시지 충돌 시 우선 순위가 높은 ID를 가지는 메시지를 우선 전송시키고 낮은 ID를 가지는 메시지는 중단 하게 한다. 통상적으로 CAN에서 트래픽이 증가하면 평균전송 지연이 발생 한다. 그림 3은 기존의 CAN Network 구조를 나타내고 있다. 각 임베디드 모듈은 하나의 노드를 가지며 각 ID에 따라서 우선순위가 정해지고 메시지를 전송하게 되는데 만일 트래픽의 증가로 평균전송 지연이 발생할 경우 네트워크상에서 메시지간의 충돌이 일어날 가능성이 있을 수 있다.

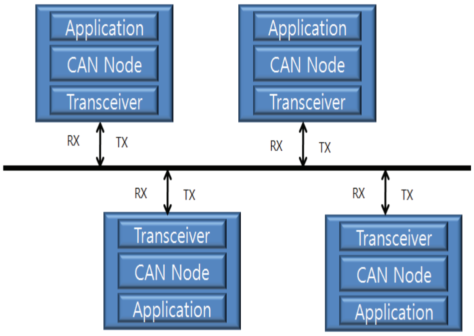


그림 3. CAN 네트워크
Fig 3. CAN Network

최근 CAN controllor로 많이 사용하는 MCU들은 기본적으로 2개 이상의 CAN node를 가지고 있다. 증가하는 network 특성을 고려한 결과로 각기 특징을 가지는 그룹 네트워크를 하나의 MCU로 제어하기 위함이다.

아래 그림은 2개의 Multi-CAN을 갖는 MCU kernel구조를 나타낸다.

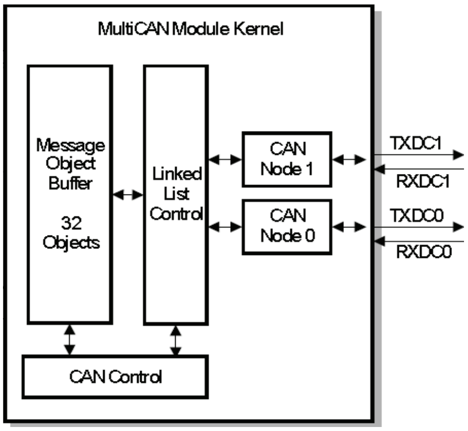


그림 4. Multi-CAN 모듈커널
Fig 4. Multi-CAN Module Kernel

본 논문에서는 11bit의 Arbiter field를 이용하는 CAN2.0A를 사용하여 CAN Node0를 기본통신으로 하여 동 버스 상에 CAN Node1에서 Feed back하는 방식으로

버스에서 일어나는 송수신을 검출하는 방식으로 실시간 모니터링이 가능한 Multi-CAN Feed Back Network을 설계하였다.

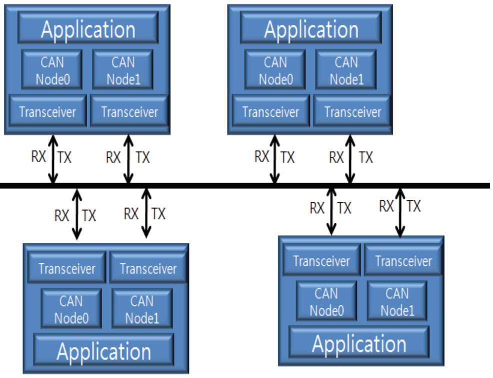


**그림 5. Multi-CAN 모듈 네트워크**
**Fig 5. Multi-CAN Module Network**

상기 그림과 같이 각 모듈별로 Multi-CAN node를 동시에 활용하여 CAN node0에서 Tx전송 시 CAN node1에서 Rx 단에서 바로 송신하는 방법으로 현재의 송수신 상태를 실시간으로 점검할 수 있어 효율적이나 CAN transceiver를 추가하여 사용해야하는 하드웨어적 부담이 있다. 그러나, Multi-CAN을 사용하면 각 32개의 Message-OBJ를 사용하여 64개의 OBJ를 활용 가능케 한다. OBJ의 사용 개수는 MCU가 가지는 SPEC.에 따라 정해지지만 2배를 활용할 수는 있다는데 기인한다.

XC886은 Full-CAN Node를 가지고 32개의 Message object를 가지고 있다. Standard 11-bit ID를 사용 하였고 가능한 8Byte의 데이터를 사용하였다.

CAN 송신은 아래와 같이 수행된다. Message object는 상위 4Byte와 하위 4byte로 분리하여 송신한다. 또한 Message Object는 송수신을 별도로 DATA0(송신)와 DATA1(수신)로 구분한다.[5]

```
void CAN_tx_Operation(void)
{
      unsigned char CAN_Transmit_Data[8]; //8byte
송신데이터선언
      CAN_Transmit_Data[0] = 0x30;
      CAN_Transmit_Data[1] = 0x31;
      CAN_Transmit_Data[2] = 0x32;
      CAN_Transmit_Data[3] = 0x33;
      CAN_Transmit_Data[4] = 0x34;
      CAN_Transmit_Data[5] = 0x35;
      CAN_Transmit_Data[6] = 0x36;
      CAN_Transmit_Data[7] = 0x37;
      CAN_vWriteCANAddress(CAN_MODATAL0);
        CAN_DATA0 = CAN_Transmit_Data[0];
        CAN_DATA1 = CAN_Transmit_Data[1];
       CAN_DATA2 = CAN_Transmit_Data[2];
        CAN_DATA3 = CAN_Transmit_Data[3];

        CAN_vWriteEN(ALL_DATA_VALID);


CAN_vWriteCANAddress(CAN_MODATAH0);
        CAN_DATA0 = CAN_Transmit_Data[4];
        CAN_DATA1 = CAN_Transmit_Data[5];
       CAN_DATA2 = CAN_Transmit_Data[6];
        CAN_DATA3 = CAN_Transmit_Data[7];

        CAN_vWriteEN(ALL_DATA_VALID);
    // 송신데이터를 Message Object에 저장
      CAN_vTransmit(0);  // 송신
 }
```

수신 프로그램은 아래와 같으며 수신 인터럽트 발생 시 미리정의 된 Message object1 로 수신된다. 수신역시 상위 4byte 하위 4byte로 총 8byte가 수신되게 프로그램 하였다.

```
 void SHINT_viXINTR6Isr(void) interrupt
XINTR6INT
 {
 .........
 //  인터럽트 플래그 확인
 if (((IRCON1 & 0x20) != 0))
 {
 //메세지 ID 확인 후

CAN_vWriteCANAddress(CAN_MODATAL1);
 CAN_vReadEN();
      CAN_Recieved_Data316[0] = CAN_DATA0;
      CAN_Recieved_Data316[1] = CAN_DATA1;
      CAN_Recieved_Data316[2] = CAN_DATA2;
      CAN_Recieved_Data316[3] = CAN_DATA3; //
```

```
하위 4byte 수신

   CAN_vWriteCANAddress(CAN_MODATAH1);
   CAN_vReadEN();
        CAN_Recieved_Data316[4] = CAN_DATA0;
        CAN_Recieved_Data316[5] = CAN_DATA1;
        CAN_Recieved_Data316[6] = CAN_DATA2;
        CAN_Recieved_Data316[7] = CANN_DATA3;
//상위 4byte수신
   .......
   }
```

그림6 에서는 Hi-CAN Transceiver 내부를 보여주고 있는데 MCU 단에서 송수신 되어지는 Txd, Rxd 단이 있고 그 입출력은 CANH, CANL 단에서 이루어진다.[6]

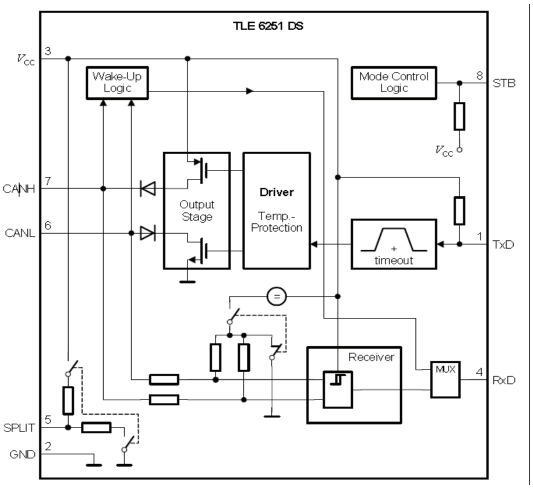


그림 6. Transceiver 내부 블록도
Fig 6. Transceiver Function diagram

표 1. 전송데이터 결과
Table 1. Result of data

| 기존데이터(8-byte) | Feed-back 데이터 |
|---|---|
| CAN_DATA0 = 0x31 | CAN_DATA0 = 0x31 |
| CAN_DATA1 = 0x32 | CAN_DATA1 = 0x32 |
| CAN_DATA2 = 0x33 | CAN_DATA2 = 0x33 |
| CAN_DATA3 = 0x34 | CAN_DATA3 = 0x34 |
| CAN_DATA4 = 0x35 | CAN_DATA4 = 0x35 |
| CAN_DATA5 = 0x36 | CAN_DATA5 = 0x36 |
| CAN_DATA6 = 0x37 | CAN_DATA6 = 0x37 |
| CAN_DATA7 = 0x38 | CAN_DATA7 = 0x38 |

본 논문에서는 임베디드 시스템에 사용 가능한 Multi-CAN Node를 내장한 Full CAN MCU를 사용하였고 Hi-CAN Transceiver를 사용하였다.

전송속도는 500k bps, Stand-by mode, Event trigger 방식을 사용하였다. CAN Node1의 Feed back 수신 상태를 기존 데이터와 비교하였을 때 이상 없이 송수신됨을 알 수 있었고 특히 , 임의 노이즈 시 전송 후 불량 송신으로 Feed Back이 되지 않음을 판단하여 재전송 및 시스템 리셋이 가능하였다.

## Ⅳ. 결론

Multi-CAN을 이용하여 오류검출이 가능한 네트워크를 설계하여 보았다. 각종 임베디드 시스템에 대해서 CAN통신을 적용할 수 있으며 발생할 수 있는 노이즈 및 물리적인 충돌을 조절할 수 있다. 다중 모듈의 점검 및 오류에 대해 소프트웨어적으로 조절할 수 있어 효율적이다.

결과적으로 2개의 Node로 데이터 Feed-Back을 확인하여 점검하는 방식으로 2개의 Transceiver를 하드웨어적으로 사용해야한다는 단점이 있다. 향후 이러한 단점을 보완한 방식을 연구할 필요성이 있다고 사료되어지며 이러한 문제에 대해서 더 많은 연구를 할 예정이다.

## 참 고 문 헌

[1] CAN Specification, Version2.0. Robert Bosch GmbH. Stutted, 1991

[2] 이기선,이태연,박재홍 자동차 응용분야를 위한 물리계층의 특성을 고려한 CAN프로토콜 진단 장비의 구현에 관한 연구, 한국자동차공학회 2006년도 전기, 전자시스템부분 심포지움 pp.122~127

[3] Road vechicles – Interchange of digital information: CAN for High-speed communication, ISO 11898,1993

[4] Mark Zachos, Controller Area Networks(CAN) for Vehicle Application ,September 17-19, 2007

[5] 안종영, 김성수, 김영자, 박상정 임베디드 시스템에 적용 가능한 CAN통신 인터페이스 설계, 2009년 한국인터넷방송통신TV학회추계대회논문집,

pp26-28 한성대학교 2009
[6] TLE6251DS Data sheet, Infineon -Rev. 3.0, 2005-09-20

※ 본 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음.

## 저자 소개

### 안 종 영(정회원)

- 1993년 : 동아대학교 전자공학과 공학사
- 1996년 : 동아대학교 전자공학과 공학석사
- 1996-2000 ;현대오토넷 전임연구원
- 2001-2003:한국폴리텍 아산캠퍼스 영상매체과 교수
- 2004-2006 : (주)대성전기 선임연구원
- 현 : 동아대학교 대학원 전자공학과 박사과정 한국폴리텍II 인천대학 컴퓨터정보과 초빙교수

<주관심분야 : 음성신호처리, 임베디드 시스템, DSP, 전장 ECU>

### 김 성 수(정회원)

- 1989: 건국대학교 전자공학과 공학사
- 1992: 건국대학교 전자공학과 공학석사
- 1998: 건국대학교 전자공학과 공학박사
- 1992-1996 : 대우전자(주) 중앙연구소 선임연구원
- 1996-현 : 한국폴리텍II 컴퓨터정보과 교수

<주관심분야 : 임베디드 시스템, 연상신호처리, 무선통신>

### 김 영 자

- 1993년 : 군산대학교 전자계산학과 학사졸업
- 2000년 : 순천대학교 컴퓨터교육과 석사졸업
- 2007년 : 군산대학교 컴퓨터과학과 박사졸업
- 2001-현 : 한국폴리텍II 인천대학 컴퓨터정보과 교수

<주관심분야 : 데이터베이스, 임베디드 시스템>

### 박 상 정

- 1991년 : 동아대학교 물리학과 학사졸업
- 1993년 : 동아대학교 전자공학과 공학석사
- 1999년 : 동아대학교 전자공학과 공학박사
- 1999년-현 : 한국폴리텍II 인천대학 컴퓨터정보과 교수

<주관심분야 : 영상처리, 임베디드 시스템>

### 허 강 인

- 1980년 : 동아대학교 전자공학과 공학사
- 1982년 동아대학교 전자공학과 공학석사
- 1990년 경희대학교 전자공학과 공학박사
- 1998년 9월~1989년 8월 일본 쓰쿠바대학 객원연구원
- 1992년 9월~1993년 8월 일본 도요하시대학 객원연구원
- 1984년-현: 동아대학교 전자공학과 교수

<주관심분야 : DSP, 음성인식, 음성합성, 신경회로망>