

차량 통신 네트워크 기술

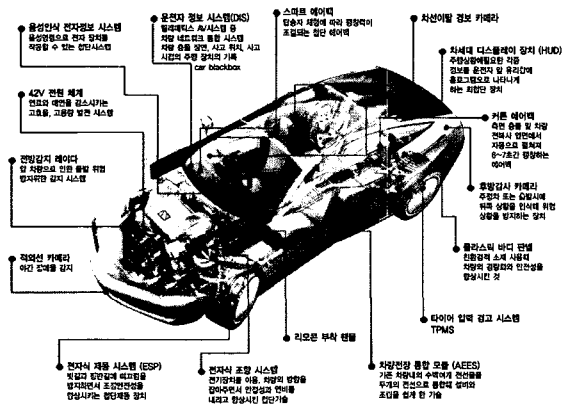
임명섭
전북대학교

요약

사무실과 가정의 컴퓨터, 대중화된 휴대폰 사용 그리고 인터넷으로 특징지어지는 정보통신 기술덕택에 현대인은 어느 정도 시간과 공간의 제약을 극복하고 있다. 그러나, 아직도 현대인은 예전에 비해 가까워진 지구촌을 여러 가지 이동체를 타고 이동을 하여야한다. 특히 현대인에게 있어서 자동차는 가정과 직장을 이어주고 업무 목적지와 휴식을 위한 휴양지를 찾아가기 위한 실질적인 이동 수단으로서 가정과 직장에 이은 또 하나의 정보통신기술이 필요한 중요한 영역이다.

따라서 미래형 자동차는 편의 주행, 쾌적 주행을 제공하고 그리고 안전 주행이 보장되는 지능형 자동차의 수요가 예견되고 있으며 이를 구현하기위해 첨단 정보통신, 전자, 제어 기술이 요구되고 있다. [1][5][6]

이상과 같은 지능형 자동차 관련 응용 분야는 위 그림과 같이 자동차 여러 부위에 장착되는 첨단 전자제품들에 의해서 구현이 된다. 그러나, 기존의 자동차와 달리 미래형 지능형 자동차에서는 많은 전자제품들이 장착됨에따라 소요되는 전원의 용량이 증가하게되어 기존 12V에서 42V 시스템으로 바뀔 예정이다. 또한 각종 센서로부터 정보신호를 받아서 정보처리를 하고 Actuator를 제어하기 위해서 많은 전자제품들간에 연결되는 신호선들의 배선이 복잡해짐에따라 생산부서에서의 공정비용이 증가하게된다. 또한 향후 석유 에너지의 고갈에 따른 전기 자동차로의 전환이 예상되는데 위에서 언급된 많은 전자장치들간 신호를 주고 받기위해 차량내 여러부위로 퍼져있는 배선들이 차지하는 무게가 상당하므로 차체의 무게를 가볍게 해야하는 차세대 전기 자동차의 성능 향상을 위해서 효율적인 In Vehicular Network 기술이 요구된다. 또한 향후 자동차에 장착된 많은 전자장치들의 고장 진단 및 내장된 SW를 효율적으로 갱신하기 위해서는 여러 전자장치들이 하나의 버스로 연결되는 In Vehicle network이 필수적이다.



I. In Vehicular Network 기술 개요

차량 내 네트워크는 직렬 데이터 버스를 통해 분산된 전자 모듈들 간에 데이터를 전달하는 방법이다. 직렬 네트워크가 없을 경우 모듈 간 통신은 전용의 점대점 선을 요구하게 되고, 부피가 크고, 비경제적이고, 복잡하며, 직경 8cm 이상의

굵은 와이어 다발을 설치해야하는 어려움을 초래할 것이다. 결국 이러한 기존의 병렬식 인터페이스로는 경제성이나 기술적인 면을 고려할 때 더이상 경쟁력을 가지지 못한다.

이에 대해 직렬 데이터 버스를 적용하여 한 가닥의 선으로 시분할 멀티플렉싱을 통해 신호를 조합함으로써 필요한 선의 수를 줄일 수 있다. 정보는 ABS, 시트 제어, 온도 조절, 대쉬보드 디스플레이 등 각 기능을 제어하는 각각의 제어 모듈에 보내진다.

이로써 각 기능을 위한 전용 선의 수가 감소되어 이에 대한 배선 작업의 비용을 줄고 시스템 비용, 무게, 신뢰도, 서비스 용이성, 설치 등이 개선된다. 그리고 차량 속도, 엔진 온도 등과 같은 공통의 센서 데이터가 네트워크 상에 유용하므로 데이터 공유가 가능하며, 이에 따라서 여분의 센서에 대한 요구가 사라진다. 또한 직렬 데이터 버스 상에 새로운 하드웨어의 추가나 신뢰도의 감소 없이 기능을 부가할 수 있는 장점이 있다.

특히 오늘날 차량에 전자 장치의 탑재가 증가함에 따라 이와 같은 네트워킹에 대한 요구는 더욱 명백해지고 있다.

In Vehicular network 기술은 그림.1과 같이 차량내 데이터의 속성과 전송속도에 따라 여러 가지 표준이 존재한다. 차량내 간단한 개폐장치용 제어 신호는 20Kbps급의 LIN(local interconnect network)방식, 현재 많이 쓰이는 1Mbps 급은 CAN(controller area network)방식, 그리고 최근에 표준화 작업 및 개발이 이루어지고 있는 10Mbps 급은 FlexRay 방식, 그리고 멀티미디어 정보통신용 신호전송으로서 MOST

(media oriented system transport) 방식이 대표적으로 쓰이고 있다.

차량 내의 모든 전자장치들을 연결할 수 있는 단일 네트워크를 구성하기 위해서 1980년대 후반 독일의 Bosch사에서 최초로 CAN을 제안하였고, 차량내 저속 제어 데이터 통신 네트워크를 위하여 LIN이 개발되었다. 1993년 고속 제어 데이터 전송을 위해 ISO 11898(고속 애플리케이션용)과 ISO 11519(저속 애플리케이션용)의 CAN 규격이 제정되었다. 최근에는 차량내 멀티미디어 데이터 통신을 위해서 MOST가 연구되고 있다. 이들은 제조 산업과 항공 우주 산업들에서도 폭넓게 응용되고 있다. CAN관련 반도체 chip 제조업체로는 Atmel, Bosch, CiA, Dallas, Hitachi, Inicore, Microchips, Mitsubishi, Motorola, NEC, NI, STM, TI 등이 있다.

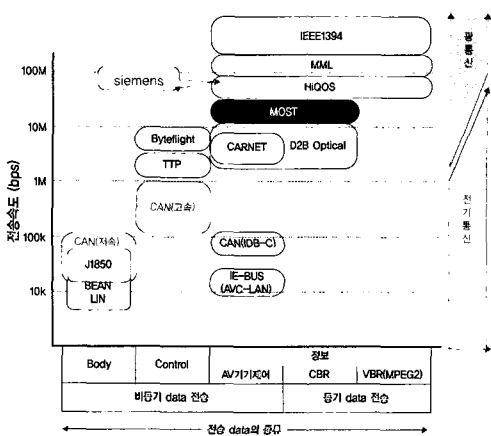
국내에서는 현대 모비스 카트로닉스 연구소, 현대 오토넷 이천 연구소, 현대차 남양연구소, 만도 기흥 중앙 연구소, 쌍용 자동차 연구소, GM 대우 연구소 등에서 IVN 관련 표준화 및 모뎀 기술 개발을 위한 연구가 진행되고 있다. 이러한 노력에도 현재 배선수의 10%를 줄이는 정도에 그치고 있다. 제어신호를 송·수신하는 통신선을 하나로 통합하는 IVN 개발로 여러 다발의 배선 무게를 줄일 수 있어 향후 전기자동차의 성능 향상에 전인차 역할이 기대된다.

다음 장에서는 차량 내에 쓰이는 주요 In Vehicular network의 종류에 대해 좀 더 알아보고 LIN, MOST, CAN과 FlexRay 기술에 대한 개발 환경을 간단히 소개한다.

II. LIN (Local Interconnect Network)

LIN은 차량에서의 분산된 전자 시스템을 위한 저비용의 직렬 통신 시스템으로 현존하는 자동 다중화 네트워크를 보완하기 위해 CAN을 토대로 고안된 것이다. LIN은 네트워크 상에 간단하면서 속도가 중요치 않은 액츄에이터와 스마트 센서와 같은 간단한 기능의 ECU를 컨트롤 하는데 사용되며, 그 주요 기능은 다음과 같다.

- 개선된 ISO 9141을 바탕으로 저비용의 single-wire 구현
- 최대 속도 20Kbit/s (EMI-이유로 제한)

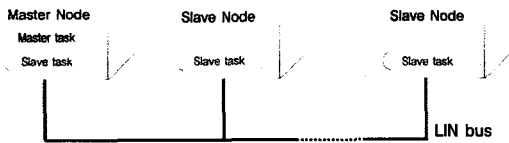


(그림 1.1) 데이터 속도별 IVN 의 분류

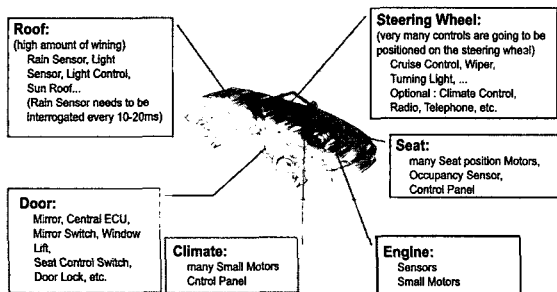
- Single Master / Multiple Slave 개념으로 중재 불필요
- SCI(UART) Data 구조 기반
- 신호 전송을 위한 보증된 대기 시간. 따라서 예측 시스템 가능.
- 다른 slave 노드에서 H/W나 S/W를 변경하지 않고도 LIN 네트워크에 노드를 추가
- slave 모드에서 크리스탈 또는 세라믹 공진회로 (resonator) 없는 self synchronization

1. LIN communication 개념

LIN 네트워크는 1개의 Master 노드와 여러 개의 Slave 노드로 구성되며, Master노드는 Master Task와 Slave Task 두 부분으로 구성되며, Slave 노드는 Slave Task만을 포함하고 있다. Master Task는 LIN 버스 상에 어떤 노드가 데이터를 전송 할지를 결정하고, Slave task는 Master Task에서 요청한 데이터 전송을 수행한다. 즉, CAN과 달리 LIN은 Master 노드에서 모든 네트워크 관리를 처리 한다. 아래 그림은 LIN 네트워크의 구조를 나타낸다.



2. LIN의 대상이 되는 애플리케이션



(그림 2.1) LIN network 구조와 적용 예

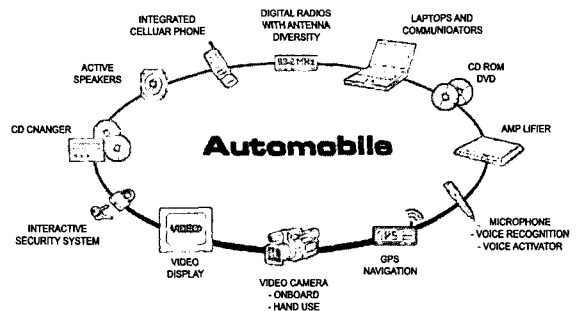
III. MOST (Media Oriented System Transport)

MOST는 대용량의 멀티미디어 정보를 전달하기 위해 광케이블(Plastic Optical Fiber: POF)을 이용하여 오디오, 비디오 및 제어 정보를 전송하는 시리얼 통신 시스템이다.[5]

동기화된 데이터 통신에 기반하고 있는 MOST는 다양한 기능과 향상된 성능의 멀티미디어 네트워크 기술로써 자동차 내에서 적용될 수 있는 다양한 분야의 멀티미디어 시스템을 위한 통신 매체로 적용되고 있는 추세이다. 그러나 로열티 문제와 복잡한 인터페이스 및 고비용이 단점이다.

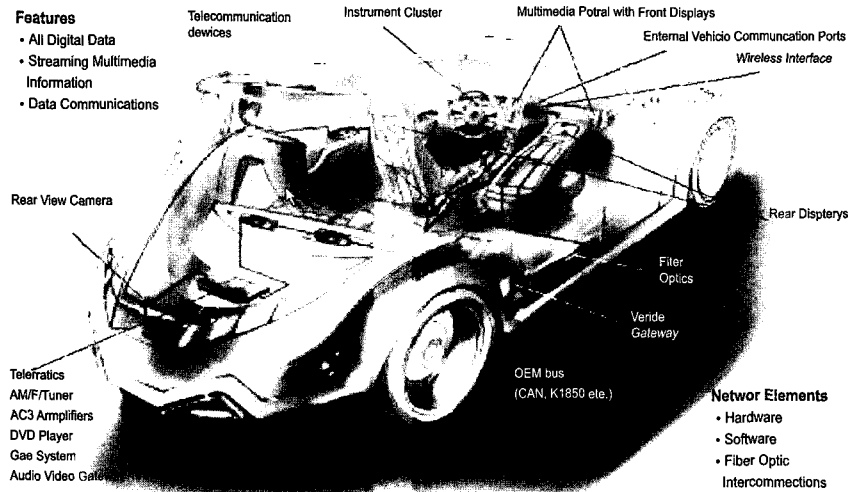
1. MOST의 특징

MOST 네트워크는 일반적으로 Ring Topology 형태를 갖추며, 최대 64개(1 Timing Master + 63 Slaves)의 MOST 장치로 구성될 수 있다. 경우에 따라서는 Star Topology도 가능하며, 안전상 중요한 어플리케이션 구현을 위하여 Double Ring Topology 구성도 사용 가능하다. 플러그 앤 플레이가 가능하여 MOST 네트워크 상에 장치의 추가 및 제거가 용이하다.



(그림 3.1) MOST 네트워크 구조

MOST는 스트리밍 데이터 (Synchronous Data Transmission) 전송과 패킷 데이터 (Asynchronous Data Transmission) 전송을 위한 대역폭으로 최대 약 23Mbaud가 사용되며, 사용자 설정이 가능한 60개의 채널로 구성된다.



(그림 3.2) MOST 네트워크 응용 예

2. MOST의 애플리케이션

MOST와 관련된 자동차내 멀티미디어 서비스 규격으로는 AMI-C(automotive multimedia interface collaboration) 단체가 형성되어 MOST외에도 IDB1394가 적용되고 있다.

CAN의 주요 응용 분야는 승용차, 트럭과 버스, 버포장도로 차량, 기차, 선박 전자장치, 비행기 및 우주 전자장치, 공장 자동화, 산업 기계 제어, 기중기와 에스컬레이터, 건물 자동화, 의료 장비 및 장치, 비산업 제어, 비산업 장비 등이다.

IV. CAN(Controller Area Network)

CAN은 초기에 자동차 산업(Automotive Industry) 분야에 적용하기 위해 고안된 serial network 통신방식으로 근래에는 자동차 분야뿐만 아니라 산업 전 분야에 폭 넓게 적용되고 있다.[1~4] 2가닥의 꼬임선(Twist Pair Wire)으로 구현된 CAN 버스로 임베디드 시스템 (또는 마이크로 컨트롤러) 사이에서 통신망을 형성하며 외부의 요인 (노이즈 등) 등에 강인성을 가져 통신 에러율을 최소화 하여 높은 신뢰성을 가지고 있고 최대 110개의 노드를 하나의 network상에 연결하여 통신할 수 있다. 통신 속도는 실시간 제어가 가능한 1Mbps (ISO 11898 규격)의 고속 통신을 제공하며 더불어 자동차 환경 (자동차 엔진 룸의 경우 다양하고 심각한 전기적 인 노이즈 상존)과 같은 심각한 노이즈 환경에 적합하도록 에러 검출 및 에러 보정의 기능이 있다.

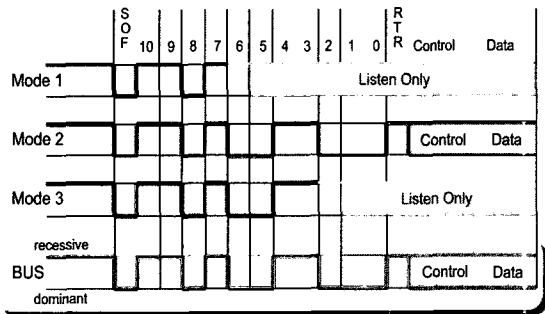
1. CAN의 주요 핵심 기술

- CSMA/CA

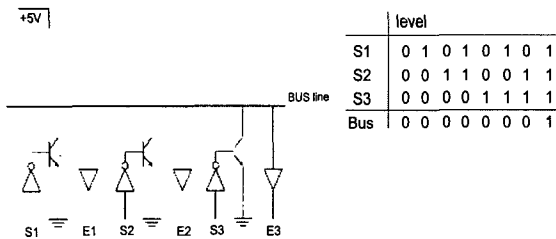
CAN 시스템의 주요 핵심기술은 단일 직렬버스에 연결된 여러 전자장치들간에 데이터를 전송하기 위해 버스 사용권을 획득하기 위한 다중 충돌을 조정하는 CSMA/CR (carrier sensing multiple access/collision resolution)이다. 다중 마스터 구조를 통해 우선순위의 메시지가 직렬 버스에 보내 지는데, (그림 4.1)에서와 같이 2개 이상의 노드가 동시에 메시지를 보내려할 경우 SOF(start of frame) 다음의 11bit로 이루어진 identifier를 이용하는 프로토콜은 메시지 충돌을 중재한다.

물리적으로는 (그림 4.2)에서와 같이 wired AND 로직에 의해 0 bit에 의해 결과 값이 결정된다. 즉 버스 레벨이 0V일 때 우선권이 주어지고 5V일때 버스 사용권을 양보하게 된다. 이와 같이 여러 개의 node에서 발생된 identifier의 값이 0 bit를 많이 가지고 있는 노드가 생존하게 되고 결국 버스 사용권을 획득하게 되어 정보를 전송하게 된다. CAN은 지

연 없이 최소 우선순위를 갖는 메시지에 버스 접근을 승인하는 비파괴적인 중재 메카니즘을 기반으로 하며 중앙집중적인 버스 마스터가 존재하지 않는다. 또한 몇가지 예외점출 메카니즘을 제공하며, 잘못된 버스 노드의 자동적 연결 단절 기능을 두어 나머지 노드들 간의 통신을 유지하도록 한다. 전송 메시지들은 대부분의 버스 시스템들과 마찬가지로 송신기나 수신기의 노드 주소가 아니라 메시지의 내용에 의해 확인된다.



(그림 4.1) CSMA/CA방식의 버스 접속



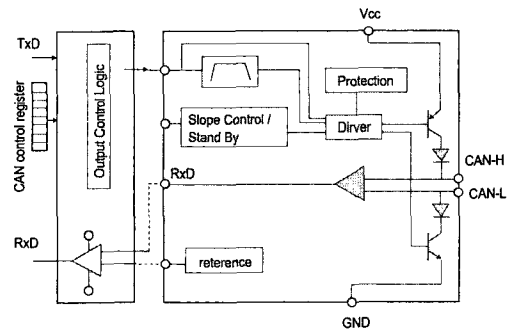
(그림 4.2) CAN 버스 접속부의 Wired AND 구조

두 번째 중요한 요소 기술은 각 노드간에 정보를 정확히 주고 받기위한 동기 기술이다. 각 노드간 상대적 버스 길이가 다르고 노드내의 clock을 발생하는 발진기의 주파수가 정확히 일치하지 않기 때문에 이를 보정할 수 있는 회로가 필요하다.

왼편 그림에서와 같이 주어진 차량내 버스 길이에서 최대 허용 데이터 전송속도를 고려하여 전송지연 시간이 주어진다. 그리고 실제 각 노드들의 송·수신기간에 clock 발생용 발진기의 빠른 위상 과 늦은 위상 오차를 고려해서 데이터가 포착되는 표본 시간을 중심으로 느린 clock timing 조정용 위상 버퍼와 빠른 clock timing 조정용 위상 버퍼를 두어 위상 오차를 보정한다. 이렇게 하여 각 노드들간 주고 받는 데이터의 동기를 유지한다.

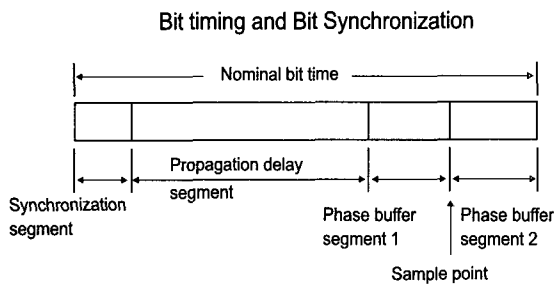
- CAN의 구조

CAN은 다음의 CAN transceiver 와 CAN controller로 구성 되어진다.

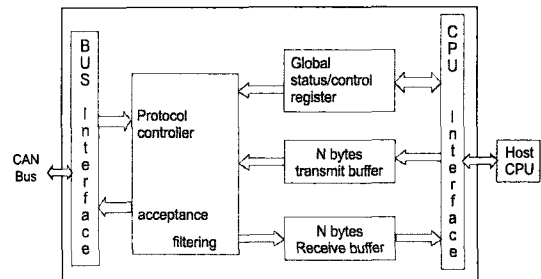


(그림 4.4) CAN transceiver

- bit timing 동기



(그림 4.3) CAN bit timing 동기



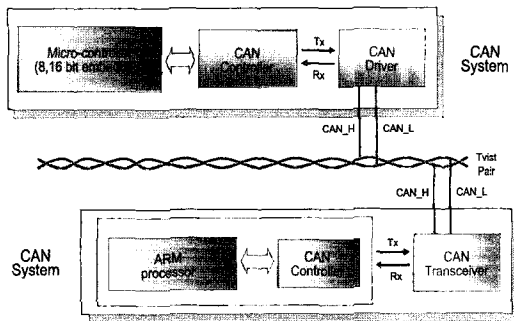
(그림 4.5) CAN controller

CAN transceiver는 bus에 직접 연결되는 물리 계층 소자이고 수신된 신호를 CAN controller에 전달하고, 역으로 CAN controller에서 전달받은 신호를 물리계층의 조건에 맞게 변환한 후 버스에 실어 다른 노드로 전송한다.

CAN controller는 CAN transceiver에서 받은 신호를 이용하여 동기를 맞춘 후 데이터 format에 따른 프로토콜을 분석을 절차에 의해 버스 사용권 조정획득 여부를 결정하고 host processor와 연동하여 데이터를 전송한다. 실제 차량내 버스에 새로운 전자장치를 연결할 때 노드별 우선 순위를 정하기 위한 identifier의 입력등 초기화 과정이 이루어진다.

CAN controller의 기능을 요약하면 다음과 같다.

- bus arbitration
- serialization and de-serialization of the frames to be sent or received
- calculation and checking of the cyclical redundancy sequence
- error detection and error signaling
- building of the CAN message formats
- inserting and deleting of the stuff bits
- generation and checking of the acknowledge bit
- synchronization of the received bit stream
- message filtering

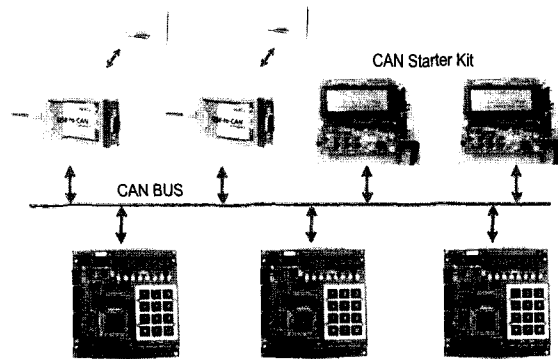


(그림 4.6) ARM 프러세서 기반 CAN 개발 환경

2. CAN의 기술 개발 환경

전북대학교 이동체정보전자 연구센터(VIITRC)에서는 (그림 4.6)과 같은 구성을 갖는 기본적인 CAN 시스템을 H/W로 구현하기위해 CAN controller를 ARM processor가 내장된

ALTERA의 Excalibur chip을 사용하여 CAN의 동작을 VHDL로 설계하였다. Quartus II 4.0 tool을 기반으로 H/W와 S/W를 동시 설계하였으며, 검증을 위한 CAN test board를 자체 제작하여 상용화된 CAN 제품들과 CAN bus에 연결하여 실제 CAN통신이 이루어짐을 확인하였다.

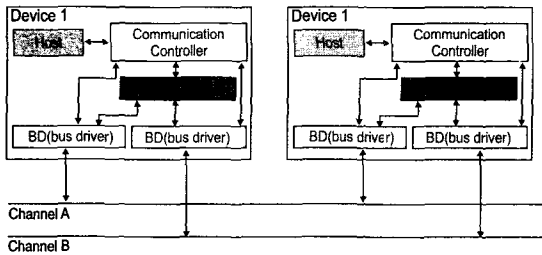


(그림 4.7) 개발된 CAN 시스템과 상용 CAN 제품의 연동 시험

차량의 전자제어 장치가 증가함에 따라 CAN(Controller Area Network)이 한계에 부딪히자 BMW, Motorola, Philips, DaimlerChrysler 등에 의해 2000년도에 설립된 컨소시엄에 의해 개발된 FlexRay는 현재 Bosch, Freescale, Philips, Motorola, Honda, Toyota, Hyundai-Kia Motors 등 전 세계에서 150 여개의 자동차 회사와 반도체회사가 FlexRay Consortium Membership을 구성하였다.[6] 2003년에 첫 번째 프로토콜 사양의 제품이 발표되었고 최근 2006년 봄에는 세 번째 프로토콜 사양의 버전(Version 2.1 Revision)이 발표되었다. 기존에 널리 알려진 CAN은 데이터 전송용량이 1Mbps의 속도로 데이터를 전송하지만 FlexRay는 두개의 채널로 구성되었으며 매개 채널에서 최대로 10Mbps의 속도로 데이터를 전송할 수 있으므로 두개의 채널을 모두 사용할 경우에는 최대로 20Mbps 대용량의 속도로 데이터를 전송하게 된다.[7]

FlexRay 하드웨어는 (그림 5.1)과 같이 Host 프로세스,

Communication controller(CC), Bus guardian(BG)과 Bus driver(BD)로 구성된다. Host 프로세스는 데이터를 공급하여 처리하고 CC는 데이터를 구성하고 액세스를 제어하고 동기를 맞추는 등 FlexRay 통신 프로토콜의 핵심부분이다. CC에서 출력되는 메시지는 BG를 거쳐 BD로 전송되어 각 채널에 전송되어 데이터를 송수신하고 이때 BG는 데이터를 송수신 할 때 버스 액세스(access)들을 감시하고 controller가 할당되어 있는 슬롯에서 액세스를 전송한다.

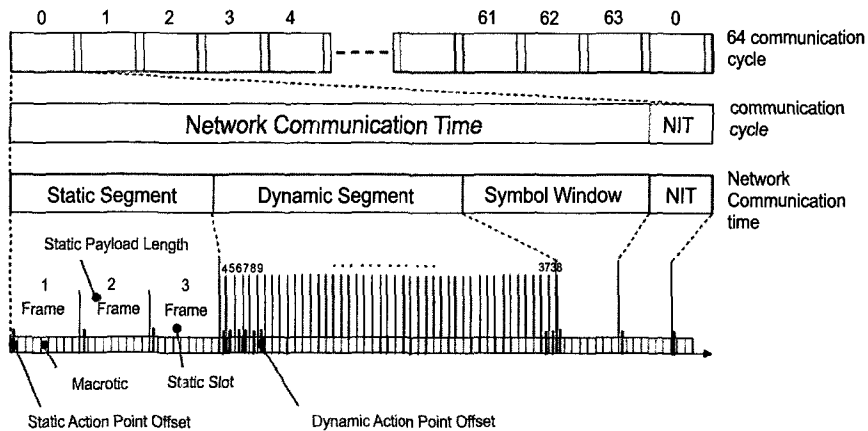


(그림 5.1) FlexRay 노드 아키텍처

FlexRay는 (그림 5.2)와 같이 규칙적인 통신주기를 기준 단위로 정보를 전송하는 TDMA(Time Division Multiple Access) 원칙에 따라 동작하는 다중 접속 방식이다. 따라서 매개의 메시지들에 대한 버스 대역폭을 최대한으로 활용하기 위하여 사이클을 정적(static segment)인 시분할 다중접속(TDMA) 방식과 동적(dynamic segment)인 미니슬롯

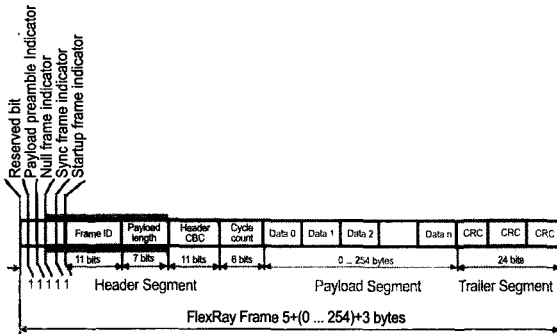
(minislot) 기반 방식으로 분할한다. 정적인 파트에는 고정된 시간 슬롯을 포함하고 있고 또한 매개 프레임은 똑같은 길이를 가지고 있다. 동적인 파트에는 시간 슬롯이 할당되는데 프레임의 길이는 임의로 변화할 수 있다. 여기에서 버스 액세스는 'minislot'으로 불리는 짧은 시간동안에만 enable 된다. 시간 슬롯은 minislot 내에서 오직 버스 액세스가 발생할 때 요구되는 시간에 의해서만 확장되므로 대역폭은 실제로 필요할 경우에만 사용된다. 이렇게 두개의 부분으로 메시지를 분리함으로써 Time-critical 한 메시지가 전송된 후 event에 의해 생성되는 메시지가 전송되는데 message ID(0 혹은 1)를 추가함으로써 '0'을 우선순위를 가지게 함으로써 전송메시지가 중첩될 때 message ID에 의하여 데이터가 서로 경쟁하면서 통신할 수 있다. Symbol window는 미리 정의된 심벌 세트 중 하나의 심벌이 전송되어 테스트 목적으로 사용된다. NIT(network idle time)는 비어있는 통신주기로서 이러한 시간사이에 clock rate를 측정하여 교정할 값을 계산하고 다음 사이클의 동기를 맞추는데 필요한 클럭의 오차 값을 수정한다.

FlexRay에서 데이터 전송은 프레임의 단위로 데이터를 전송한다. 하나의 프레임 포맷은 (그림 5.3)과 같이 header segment, payload segment, trailer segment 등 3개의 부분으로 나뉜다. Header segment 부분은 5 byte로 구성되었고 전송할 메시지의 길이, 우선순위, 네트워크와의 연결 상태 등 전송할 메시지의 기본 정보를 미리 알려주어 데이터 전송속

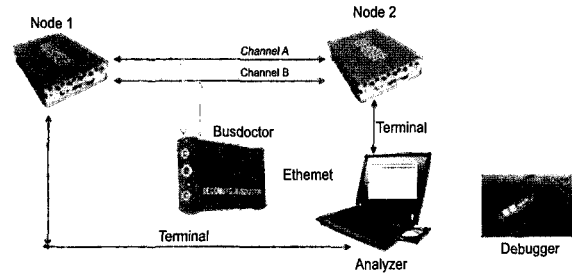


(그림 5.2) FlexRay의 시간제도

도를 크게 높인다. Payload segment 부분은 실제로 전송하려는 메시지를 포함한 부분으로서 최대로 254 byte의 메시지를 추가할 수 있다. Trailer segment 부분은 24bit의 CRC(Cyclic Redundancy Check)로서 전송하려는 메시지의 에러를 검출하는 강력한 도구로 사용되고 있다.

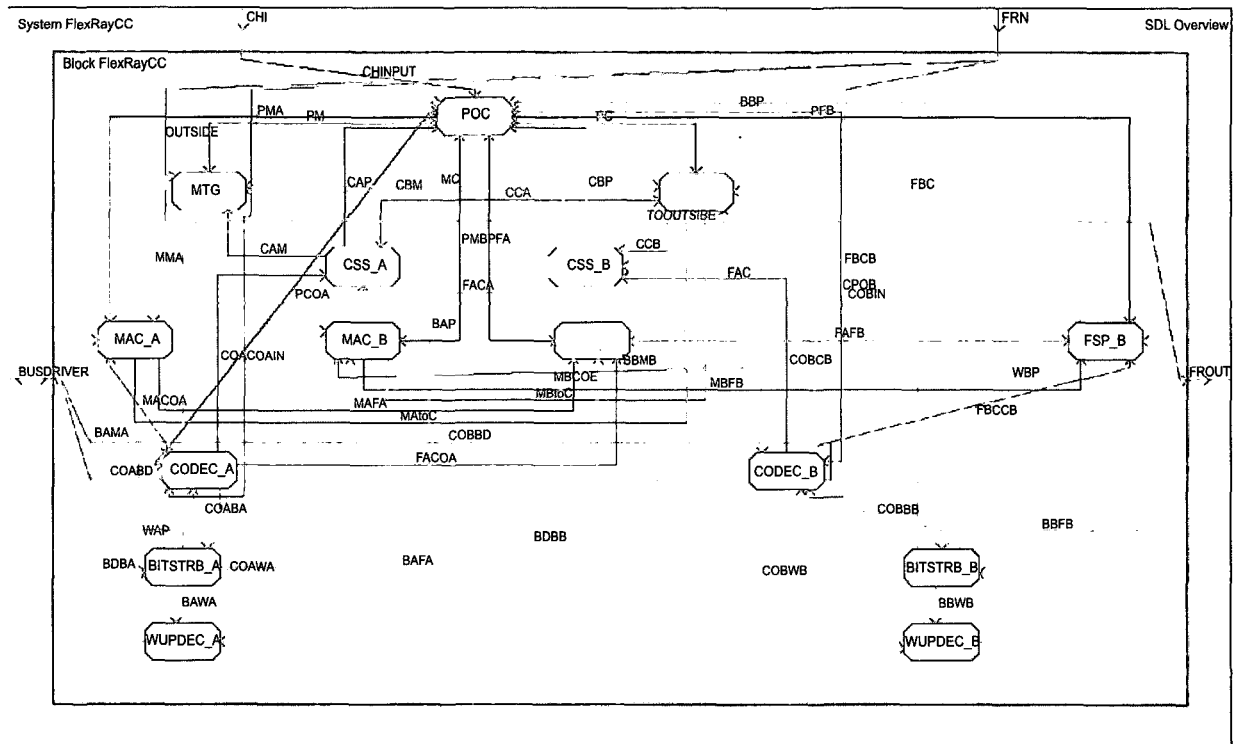


(그림 5.3) FlexRay 프레임 포맷



(그림 5.4) FlexRay 네트워크의 구성

이러한 FlexRay 프로토콜 명세는 SDL(Specification and Description Language) 프로그램 언어로 구현되었다.[8][9] 또한 SDL 프로그램은 그래픽 형식언어(formal language)로 구현되어 있어 C나 C++ 및 기타 언어로 작성된 문서보다 구조 및 원리를 알아보기 쉽고 SDL로 정의된 시스템을 분석하고 각 단계별로 시뮬레이션이 가능하여 오류나 인터페이스의



(그림 5.5) FlexRay CC 메인 블록

불일치를 쉽게 발견할 수 있으므로 시스템의 개발 시간을 줄이고 쉽게 소스를 수정 및 변화 시킬 수 있는 장점을 가지고 있다. SDL로부터 C나 C++, JAVA 등 언어로 자동변환 틀이 포함되어 있어 사용자의 의도 및 목적에 따라 변환 가능할 수 있고 실시간 시스템을 구현할 수 있어 주로 통신 분야에서 많이 사용되고 있다.[10][11]

아래의 (그림 5.4)는 오스트리아의 DECOMSYS GmbH 사의 FlexRay Starter Kit를 사용하여 FlexRay 네트워크를 구성한 예이다. 여기에서 노드와 노드사이에서 두개의 채널을 통하여 데이터를 서로 송수신하게 되는데 이러한 데이터 통신 과정을 monitoring hardware를 통하여 analyzer 프로그램으로 통신 데이터를 실시간으로 분석하고 제어할 수 있다.[12]

전북대학교 이동체정보전자 연구센터(VITRC)에서는 SDL 틀을 이용하여 FlexRay 통신 프로토콜을 구현하였다. 이는 S/W로 먼저 통신 프로토콜을 구현하여 빠른 시간내에 프로토콜을 구현하여 작동여부를 확인할 수 있고 동작원리 및 구조적 특징을 이해할 수 있다. 또한 VHDL을 이용하여 FlexRay 통신 프로토콜을 H/W로 구현하는데 기초가 된다. 아래의 (그림 5.5)는 SDL을 이용하여 FlexRay CC의 메인 블록을 표현한 그림이다.

VI. 차량내 network의 확장 및 DC-PLC를 이용한 경량화

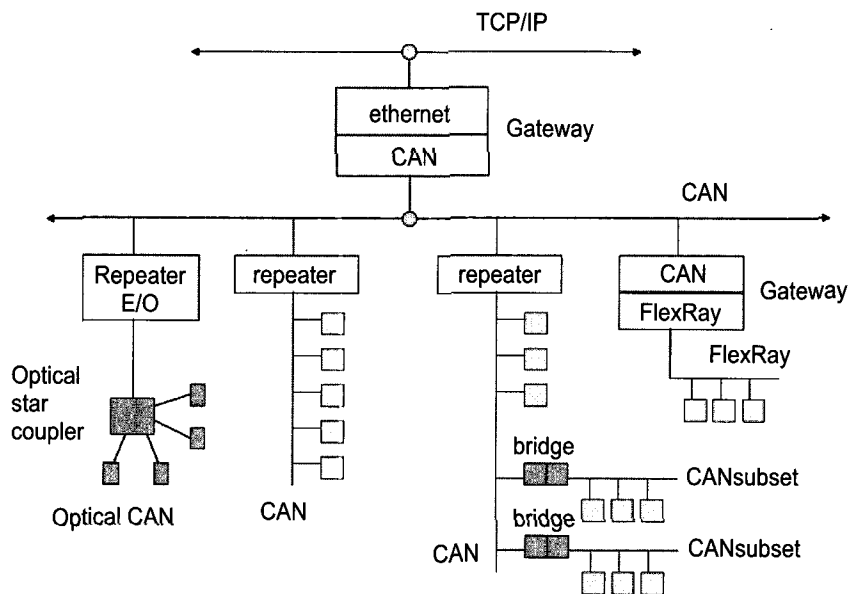
- gateway

차량내에는 목적에 따라 LIN, CAN, FlexRay, MOST 등의 이기종 network이 공존하고, 외부 telematics 단말기와의 연동에 따른 서로 다른 protocol을 가진 network간 정보를 주고 받을 수 있는 상황이 예견되므로 아래와 같은 구조를 갖는 gateway 개발을 작업이 필요하다.

- DC-PLC CAN

미래 지능형 자동차는 차량내 여러 부위에 많은 전자장치가 장착됨에 장치들간에 신호 연결 및 전력선공급을 위한 배선이 많이 소요된다.

향후 차량의 동력원으로 전기를 사용함에따라 차량의 무게를 가볍게 하는 것이 바람직하다. 따라서 DC 전원선에 신호를 공급할 수 있는 DC-PLC CAN이 개발되어 이러한 문제점을 해결 할 수 있다.



(그림 6.1) Gateway를 통한 CAN, FlexRay, MOST 및 외부 망과의 연동

참 고 문 헌

- [1] CAN Specification 2.0 part A and B, Robert Bosch Gmbg, Sept. 1991.
- [2] VHDL Reference CAN User's Manual Revision Revision 2,2, Rebert Bosch Gmbg, 1999.
- [3] Wolfhard Lawrenz, CAN System Engineering - From Theory to Practical Applications. Springer - Verlag, 1997.
- [4] K. Etschberger, Controller Area Network-Basics, Protocols, Chips and Applications. Weingarten, 2001.
- [5] MOST Homepage, <http://www.oasis.de>
- [6] FlexRay Consortium, <http://www.flexray.com>
- [7] FlexRay Communications System Protocol Specification v2.1 Revision A, December 22 2005.
- [8] SDL-Forum, <http://www.sdl-forum.org>
- [9] Telelogic AB, <http://www.telelogic.com>
- [10] Ferenc Belina, Dieter Hogrefe and Amardeo Sarma. SDL with Applications from Protocol Specification. Prentice Hall, 1991.
- [11] G. J. Holzmann, Design and Validation of Computer Protocols, Bell Kaboratories, Murray Hill, New Jersey, 1991.
- [12] DECOMSYS GmbH, <http://www.decomsys.com>



임 명 섭

1976년 ~ 1980년 연세대 전자공학과 학사
 1980년 ~ 1982년 연세대학원 전자공학과 석사
 1984년 ~ 1990년 연세대학원 전자공학과 박사
 1984년 ~ 1985년 대우통신 종합연구소 연구원
 1985년 ~ 1996년 한국전자통신 연구소 책임연구원,
 이동통신기술연구단 신호처리연구실장
 1992년 ~ 1993년 CDMA 개발관련 Qualcomm사와 공동개발
 1992년 ~ 1994년 제2이동 통신사업자 선정 심사위원
 1996년 ~ 현재 전북대학교 전자정보공학부 교수

관심분야 : Wireless LAN (IEEE802.11a) : OFDM, Advanced CDMA : MIMO , smart antenna, UWB, 지능형 자동차용 network 통신 및 전자제어장치

