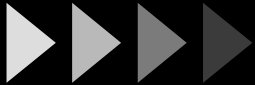


In Vehicle Network



스마트카 협동과정 기석철



충북대학교
CHUNGBUK NATIONAL UNIVERSITY

E-campus/강의자료
sckee@cbnu.ac.kr





Network의 분류

❖ Parallel Communication

- ISA (Industry Standard Architecture)
- P-ATA (Parallel Advanced Technology Attachment)

❖ Serial Communication

- I²C (Inter-Integrated Circuit)
- SPI (Serial Parallel Interface)
- UART (Universal Asynchronous Receiver/Transmitter)

	직렬통신	병렬통신
설명	· 한 번에 한 비트씩 전송	· 동시에 여러 비트를 전송
장점	· 병렬통신 소자보다 가격이 저렴함 · 장거리 전송 가능	· 전송속도가 직렬통신에 비해서 빠름 · 추가 변환로직 필요 없음
단점	· 저속, 추가 변환로직 필요	· 통신거리의 제한 · 구현의 어려움 · 직렬통신 소자에 비해 가격이 비쌈

	비동기 직렬통신	동기 직렬통신
설명	데이터 처음과 끝에 시작, 정지 신호를 삽입해 데이터 송수신	디바이스 사이에 동기를 취하고 타이밍에 따라 데이터 송수신
종류	UART(RS232:20Kbps)	I ² C(100Kbps), SPI(20Mbps)
장점	송수신 시 IDLE 신호 필요 없음	데이터 전송 속도가 빠름
단점	동기 통신에 비해 느림	동기를 유지하기 위해 IDLE 신호 필요



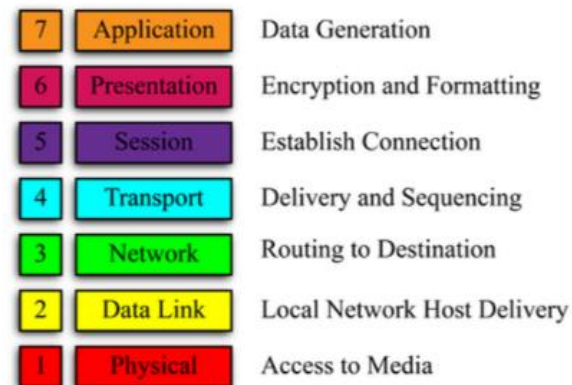
CAN 탄생 배경



- ❖ 1990년대 초까지 1-to-1 communication → nC_2 조합 소요
- ❖ Single Network Protocol 필요
- ❖ 1992년 Bosch에서 multiple access communication protocol 정의 (CAN)

- OSI (Open System Interconnection) Model
- Layer 1 (Physical), 2(Data Link) 만 정의
- Layer 3 ~ 7은 Chip업체에서 설계하여 양산
- Low Speed CAN (ISO 11519 → ISO 11898-3)
- High Speed CAN (ISO 11898)

OSI Model



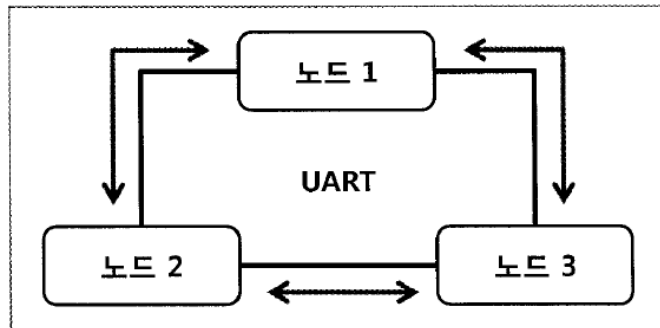
- ❖ CAN (Controller Area Network)은 지금까지도 가장 많이 사용되고 있음
 - 연간 약 4억개의 CAN node 거래되는 것으로 추정
 - 통상적으로 500kbps ~ 1Mbps 속도



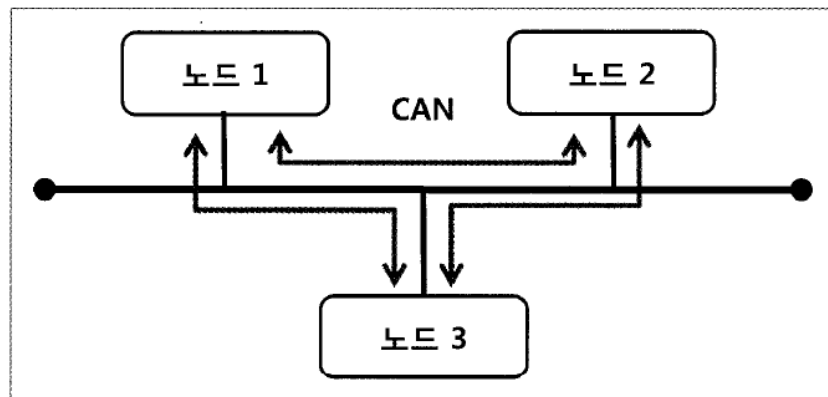


Basic Principle

❖ Normal Serial Communication (1-to-1 method)



❖ CAN (Multi-Master or Broadcasting method)

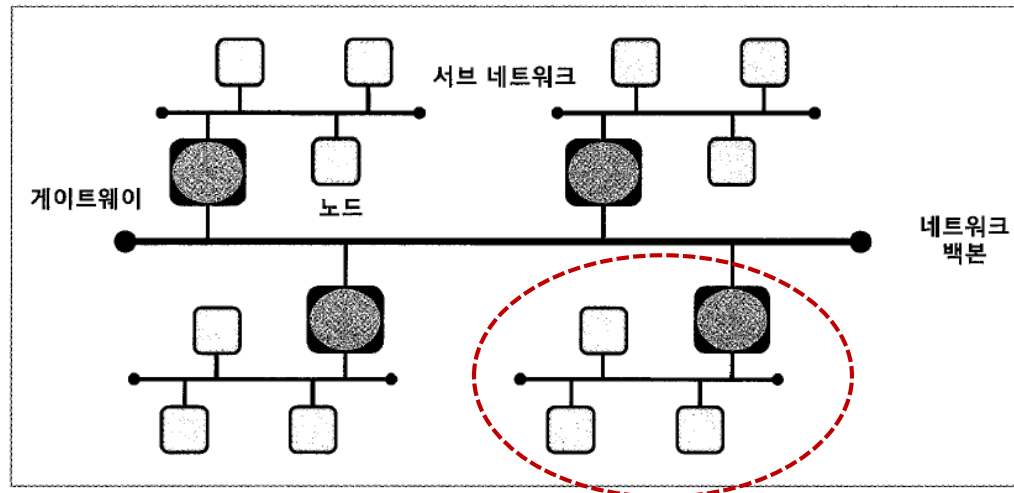




Basic Principle

❖ Message Identifier

- CAN 2.0A (Standard CAN) 에서는 11 bits ($2^{11} - 2^4 = 2032$) * Reserved 4 bits 제외
- CAN 2.0B (Extended CAN) 에서는 29 bits (5 million 이상)
- Standard CAN에서 2032개 이상의 통신이 필요한 경우 별도 Gateway 필요



OSI Model



❖ Data Link Layer

- LLC (Logical Link Control) : Message filtering, Overload notice, Error recovery, ...
- **MAC** (Medium Access Control) : Framing, Arbitration, Ack., Error Detection, ...

❖ Physical Layer

- **PLS** (Physical Signaling)
- PMA (Physical Medium Attachment)
- MDI (Medium Dependent Interface)

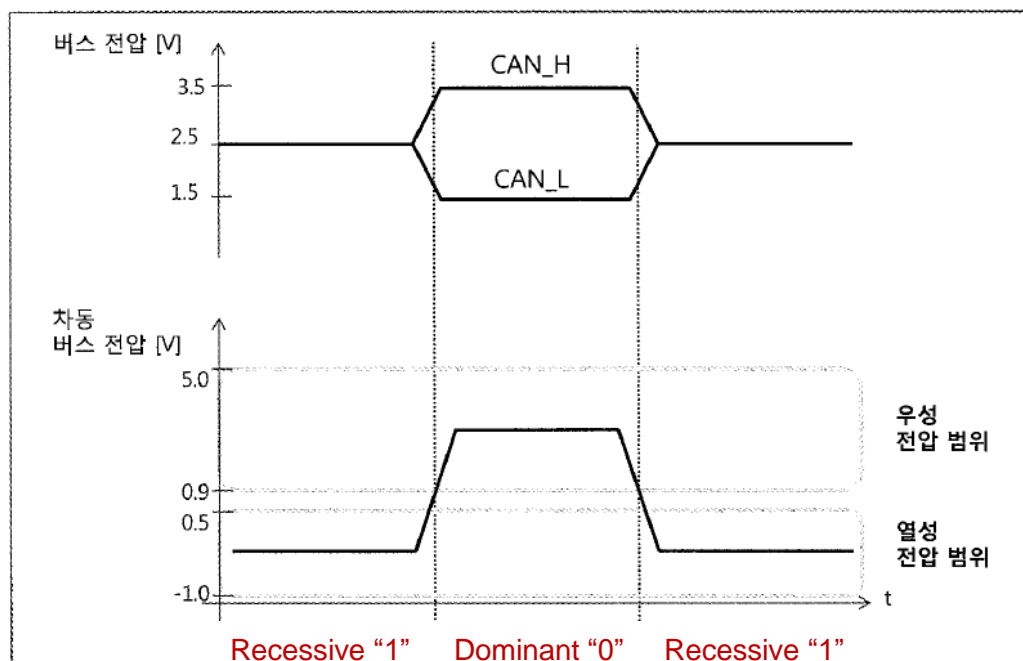
표준	OSI 모델		구현
사용자 정의	애플리케이션 계층		소프트웨어, 보드 레벨
...			
CAN 표준 정의	Data Link	LLC	온칩 하드웨어
		MAC	
		PLS	
	Physical	PMA	오프칩 하드웨어
		MDI	
ISO 표준 범위	TM(Transmission medium)		



PLS



- ❖ Twisted pair cable, Differential Signal (CAN_H, CAN_L)
- ❖ Digital "1" (Recessive Bit)
 - CAN_H = CAN_L = 2.5V, Difference = 0V
- ❖ Digital "0" (Dominant Bit)
 - CAN_H = 3.5V, CAN_L = 1.5V, Difference = 0.9V 이상

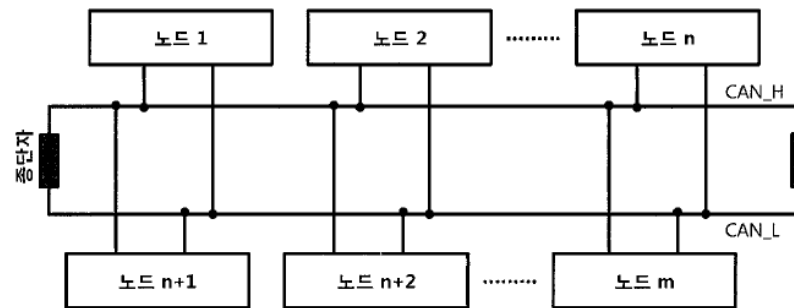


Differential Signal



❖ Terminator

- 송신신호가 수신단에 전달되지 못하고 되돌아오는 신호를 제거하기 위함



❖ Differential Signal의 신호 특성

- 원거리 전송에 따른 신호 감쇄에 강건함

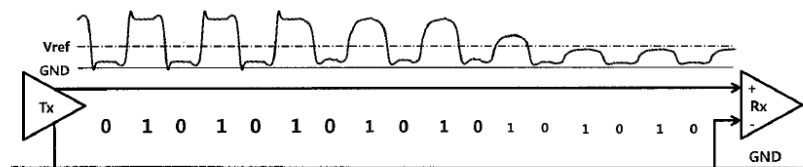


그림 B 단일종단신호

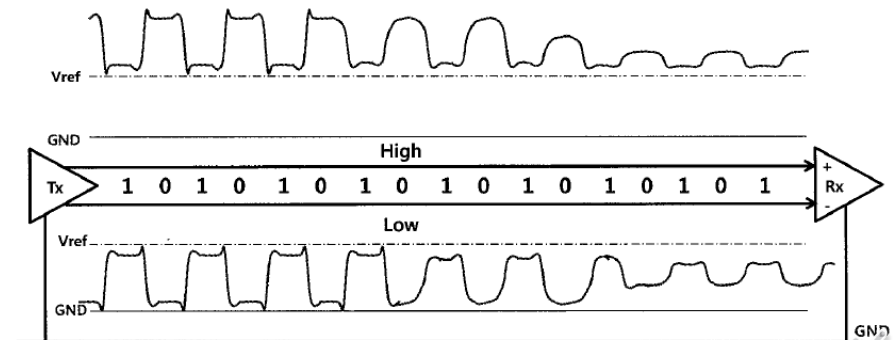


그림 C 차동신호





Multiple Access Protocol

- ❖ 메시지에 우선순위 부여해 충돌 방지, Dominant bit “0” 이 우선 순위가 높아서 ID가 가장 작은 메시지가 우선 순위
- ❖ CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
 - 반송파(Carrier Frequency)를 이용해 충돌을 검출하는 방식(ex. 유선 LAN)
 - 전송량이 증가하면 패킷 충돌이 자주 발생하여 데이터 손실도 증가하는 단점
 - ① 선로가 Idle 상태이면 전송 시작
 - ② 선로가 Busy 상태이면 Idle 상태가 될 때까지 waiting
 - ③ 전송 중 충돌이 감지되면 Jamming Signal 발생 후 전송 중단
 - ④ 임의의 시간 대기 후 처음부터 다시 전송 시도
- ❖ CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)
 - 충돌을 미리 감지해 전송을 중단하는 방식(ex. 무선 LAN, CAN)
 - 충돌은 자주 발생하지 않으나, 네트워크 접근 빈도가 증가하면 예비 신호로 인해 속도가 느려지는 단점
 - ① 선로에 다른 Node의 데이터 송신 여부 확인을 위해 반송파 감지
 - ② 다른 Node가 송신 중이면, 임의의 시간을 waiting
 - ③ 재 반송파를 감지해 다른 반송파가 없는지 확인 (송신 지연시간까지 고려하여 확인)
 - ④ 반송파가 없으면 데이터 전송 시작



Message Frame



CAN message의 종류

프레임		설명
데이터 프레임	Data Frame	일반적으로 전송되는 데이터 메시지
원격 프레임	Remote Frame	특정 메시지 식별자의 데이터 요청 메시지
오류 프레임	Error Frame	오류를 판단한 노드가 전송하는 메시지
오버로드 프레임	Overload Frame	버스 안정화가 필요할 때 전송하는 메시지



Data Frame

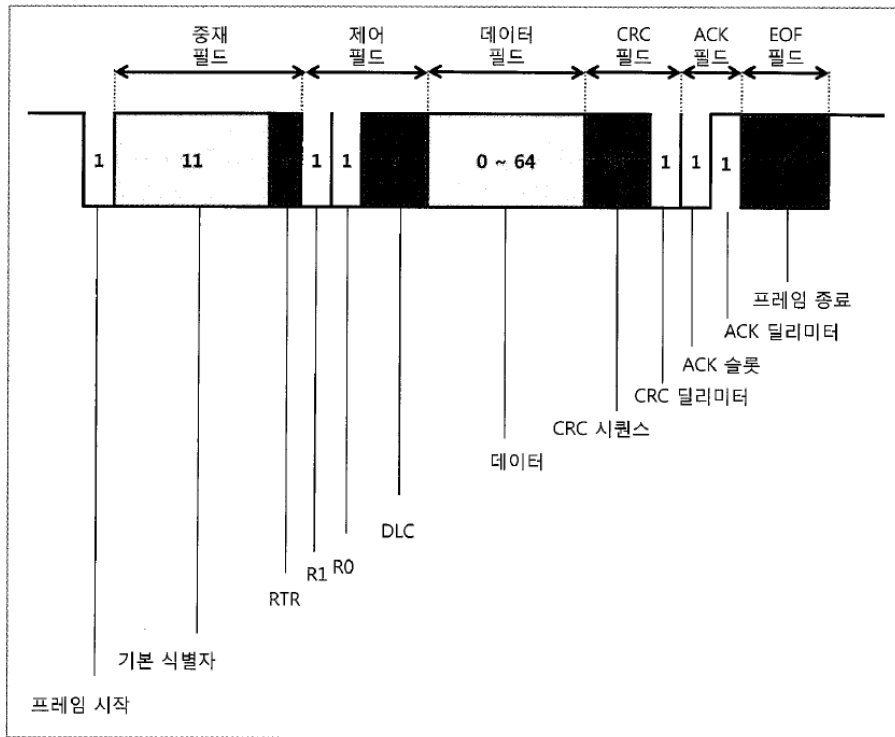


그림 8-7 표준 CAN 메시지 프레임의 구조

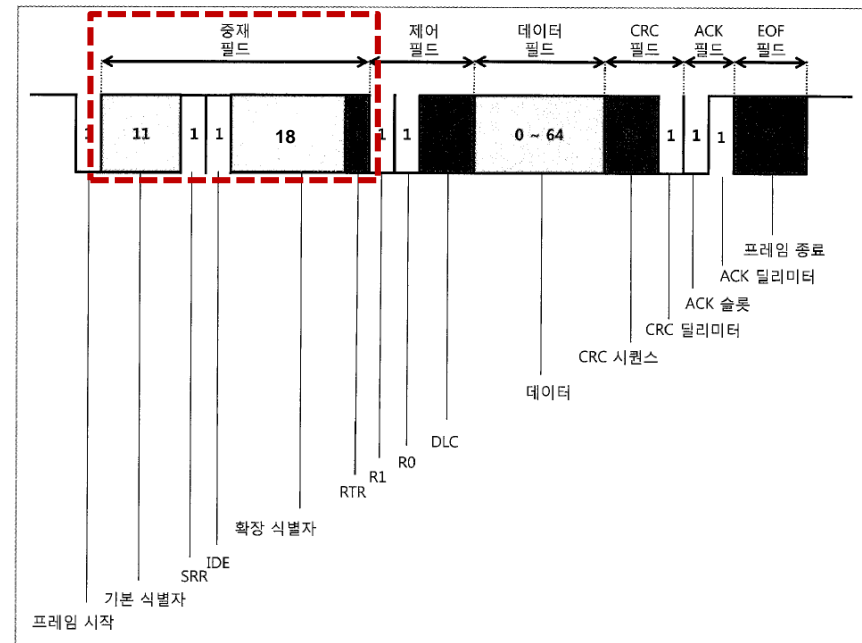


그림 8-8 확장 CAN 메시지 프레임의 구조





Data Frame

❖ Start of Frame (SOF)

- Frame의 시작으로 항상 Digital “0”

❖ Arbitration Field (Standard CAN : 12bits, Extended CAN 32bits)

- Identifier (11 bits)
- Substitute Remote Request (SRR, 1bit) : 확장 CAN에만 있으며, 표준 CAN과 확장 CAN을 중재할 목적. ID가 같으면 표준 CAN이 우선 순위 높음. 항상 Digital “1”
- Identifier Extension (IDE, 1bit) : Extended Identifier 있음을 알림
- Extended Identifier (18bits)
- Remote Transmission Request (RTS, 1bit) :
 - Digital “0” : 메시지가 있는 data frame
 - Digital “1” : Data가 없는 remote frame

❖ Control Field (6bits)

- R1, R0 : Reserved bits, Digital “0”
- Data Length Code (DLC, 4 bits)





Data Frame

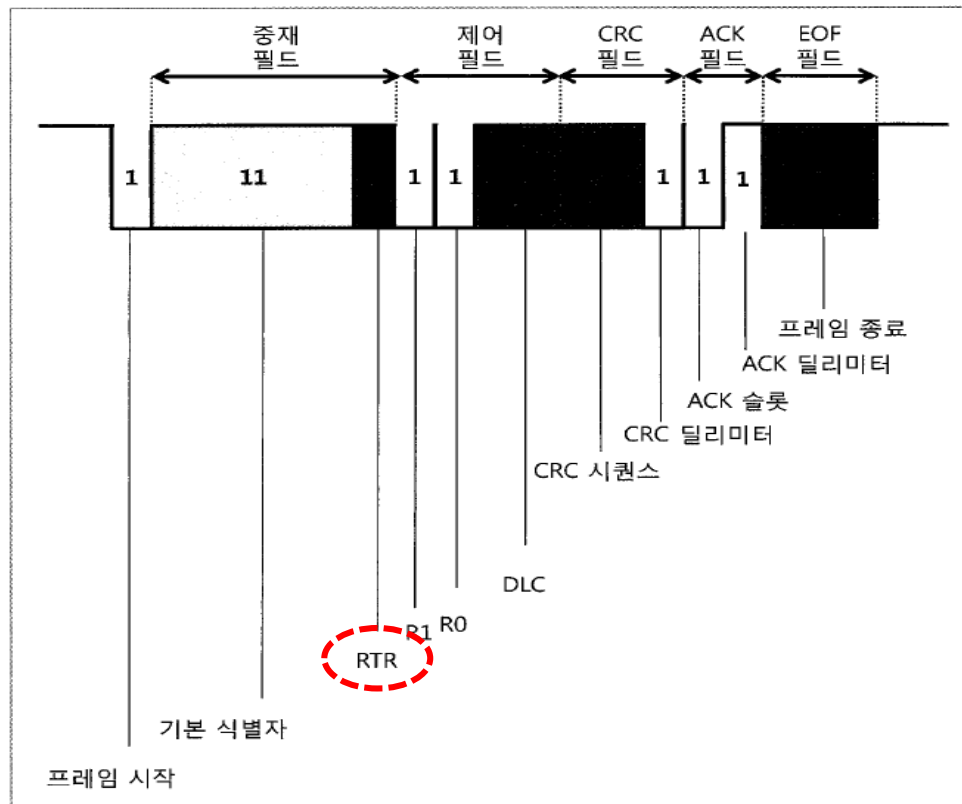
- ❖ Data Field (0~64bits)
 - 0 ~ 8byte 데이터로 구성되며, **MSB**부터 먼저 송신
- ❖ Cyclic Redundancy Code Field (CRC, 16bits)
 - CRC Sequence (15 bits) : 통신 오류를 판단, **CRC** 연산 방식 참조
 - CRC Delimiter (1bit) : CRC field 경계를 알려주며 항상 Digital “1”
- ❖ Acknowledge Field (2bits)
 - ACK Slot (1bit) : CRC field 이전의 값은 전송 오류 없이 수신되었음을 알려주기 위한 time slot, 송신기는 Digital “0”을 송출하고 오류 없이 수신한 수신기는 Digital “1”로 대체해서 송신기에 전송
 - ACK Delimiter (1bit) : ACK field 경계를 알려주며 항상 Digital “1”
- ❖ End of Frame Field (7bits)
 - Data frame의 끝을 알려주며 모두 Digital “1”로 구성
- ❖ Intermission Field (3bits)
 - Data frame과 인접 frame 사이의 메시지가 없는 대기 상태를 알리는 field로 3bits 이상의 Digital “1”로 구성





Remote Frame

- ❖ Remote Frame은 수신기가 송신기에게 Data를 요청할 때 사용
- ❖ Data Frame에서 data field를 제외하고 구조가 동일
- ❖ RTR bit에 의해 Data frame과 구분



Error Frame



- ❖ 현재 송신 **Frame**의 오류 상황을 공지할 목적으로 사용
- ❖ 수신기에서 감지한 오류를 **Frame bit insertion rule**을 손상시켜, 현재 버스에 접근한 노드에 전송
- ❖ 현재 전송중인 장치는 **Frame** 오류를 감지하고 전송 중지
- ❖ **CAN**에서 감지하는 **Error**의 종류
 - **Bit Error** : 송신기에서 감지, 메시지의 **Bit level**과 **Channel level**이 다를 경우
 - **Stuff Error** : 연속해서 5개 이상의 동일 극성 데이터가 전송되는 경우
 - **Ack. Error** : **Ack. Slot**에 수신기가 “1”을 발송하지 않는 경우
 - **CRC Error** : 송신기에서 전송한 **CRC**와 수신기에서 연산한 **CRC**가 다를 경우
 - **Format Error** : 각 **Field**의 경계 **bit**가 “1”이 아닌 경우



Error Frame



5회 이상 연속된 값을 보낼 때 경계 구분 error 누적 가능

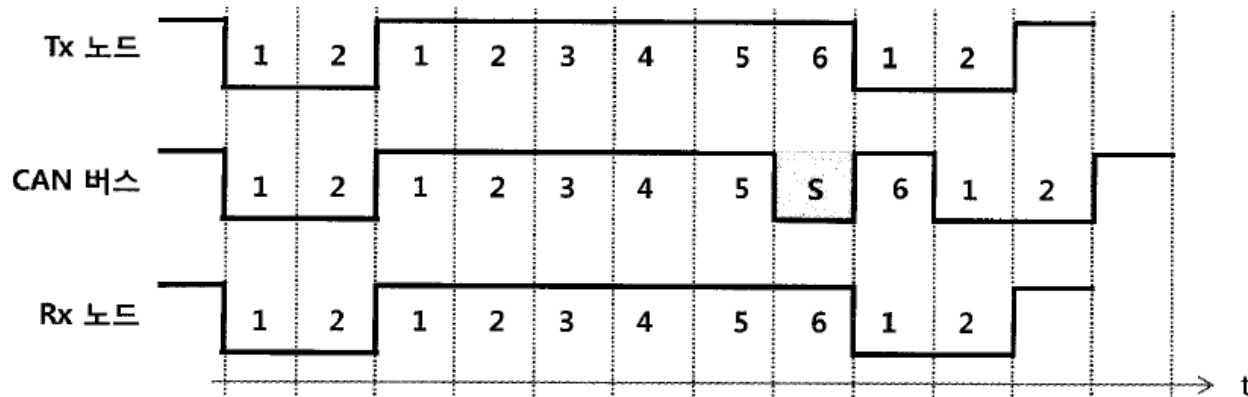
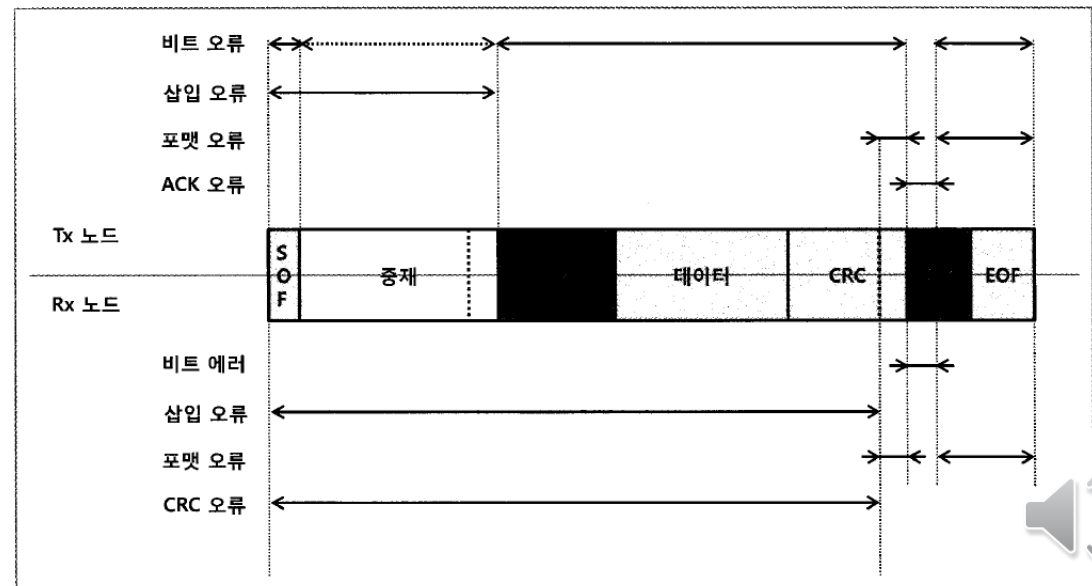


그림 B 삽입 비트 송수신의 예

송수신 노드별로
검출 가능한 전송 오류





Error Frame

- ❖ Error Flag (6~12bit) : 오류를 감지한 노드는 6bit “0” 발송
- ❖ Error Delimiter (8bit) : Error Frame 경계 신호로 항상 “1”

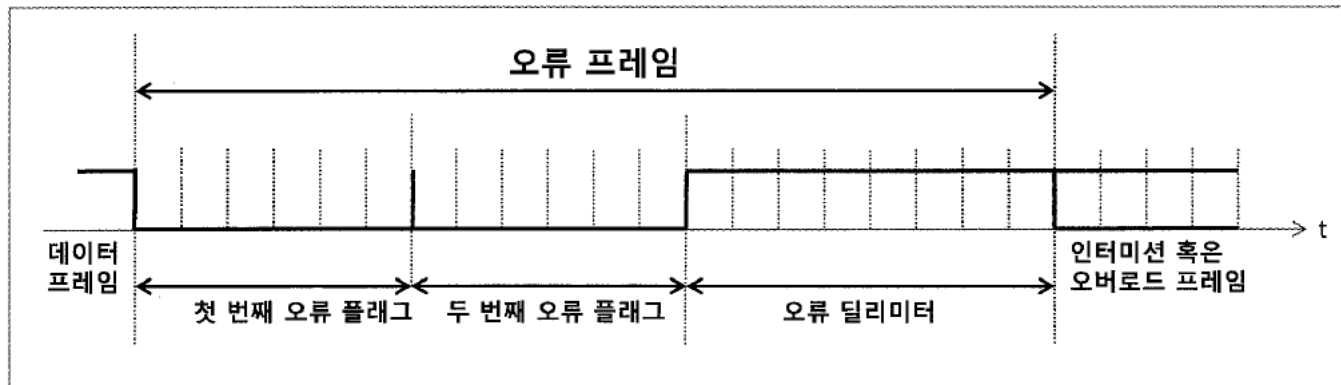
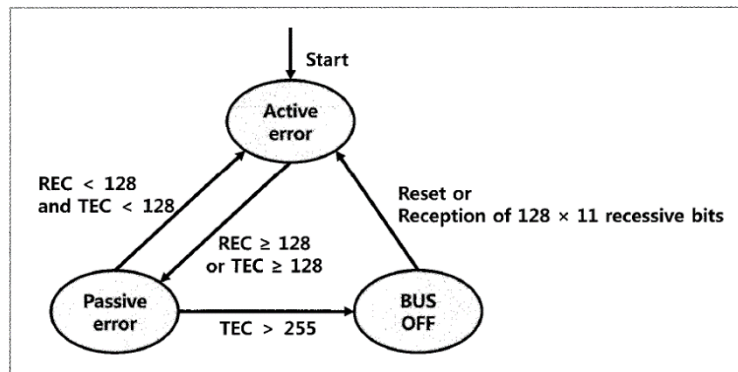


그림 8-13 오류 프레임 구조

- ❖ Error Frame에 의한 통신망 영향을 줄이기 위해 Error Handling 처리



- 전송 실패 시 Counter 1 증가, 성공 시 1 감소
- 128보다 작으면 Active Error State
- 128보다 크면 Passive Error State, 전송 제한
- 256보다 크면 CAN 접속 제거

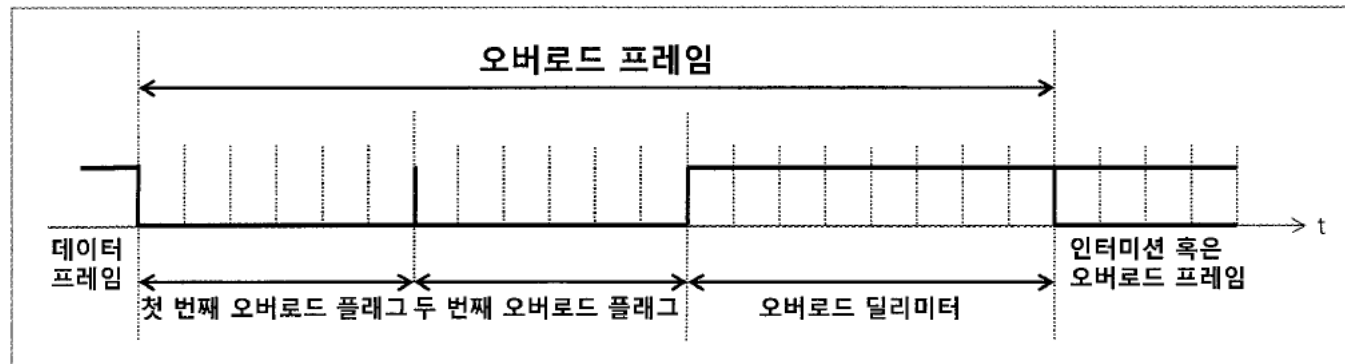
→ Reset 또는 128 x 11 bit 시간 동안 대기





Overload Frame

- ❖ Bus 상태를 안정시킬 목적으로 전송하는 Frame
 - 이전 메시지 처리를 마치지 못해 지연이 필요한 상태
 - Intermission Field의 1,2 bit가 “0” 인 경우
 - Error Delimiter Bit 또는 Overload Delimiter Bit의 마지막 Bit가 “0”인 경우
- ❖ Overload Frame 구조



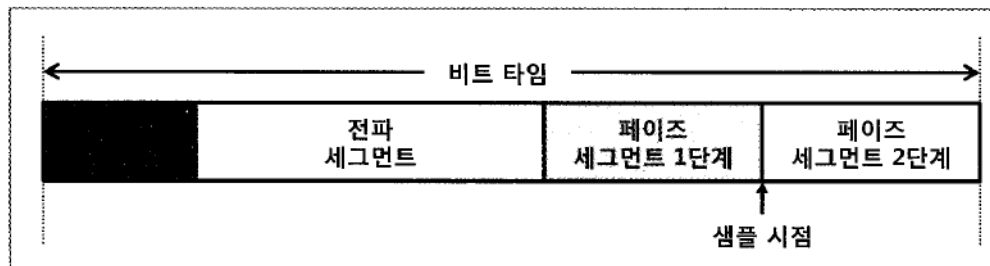
- ❖ Overload Flag (6~12bit) : 오류를 감지한 노드는 6bit “0” 발송
- ❖ Overload Delimiter (8bit) : Overload Frame 경계 신호로 항상 “1”





Bit Time Segment와 Re-synchronization

- ❖ 각 노드의 발진기 f_{osc} 가 상이하거나, 발진 주파수 오차 df 문제를 Re-synchronization으로 해결
- ❖ Bit Timing Segment
 - Synchronization Segment : 항상 1 Time Quanta (T_q)
 - Propagation Segment : 전파 지연 시간 상쇄 목적
 - Phase Buffer Segment 1, 2 : 재동기화 목적



- 동기 실패 시 Phase Buffer Segment 값을 변경하여 재동기화
- RJW (Re-synchronization Jump Width) 를 1 ~ 4 T_q 에서 변경

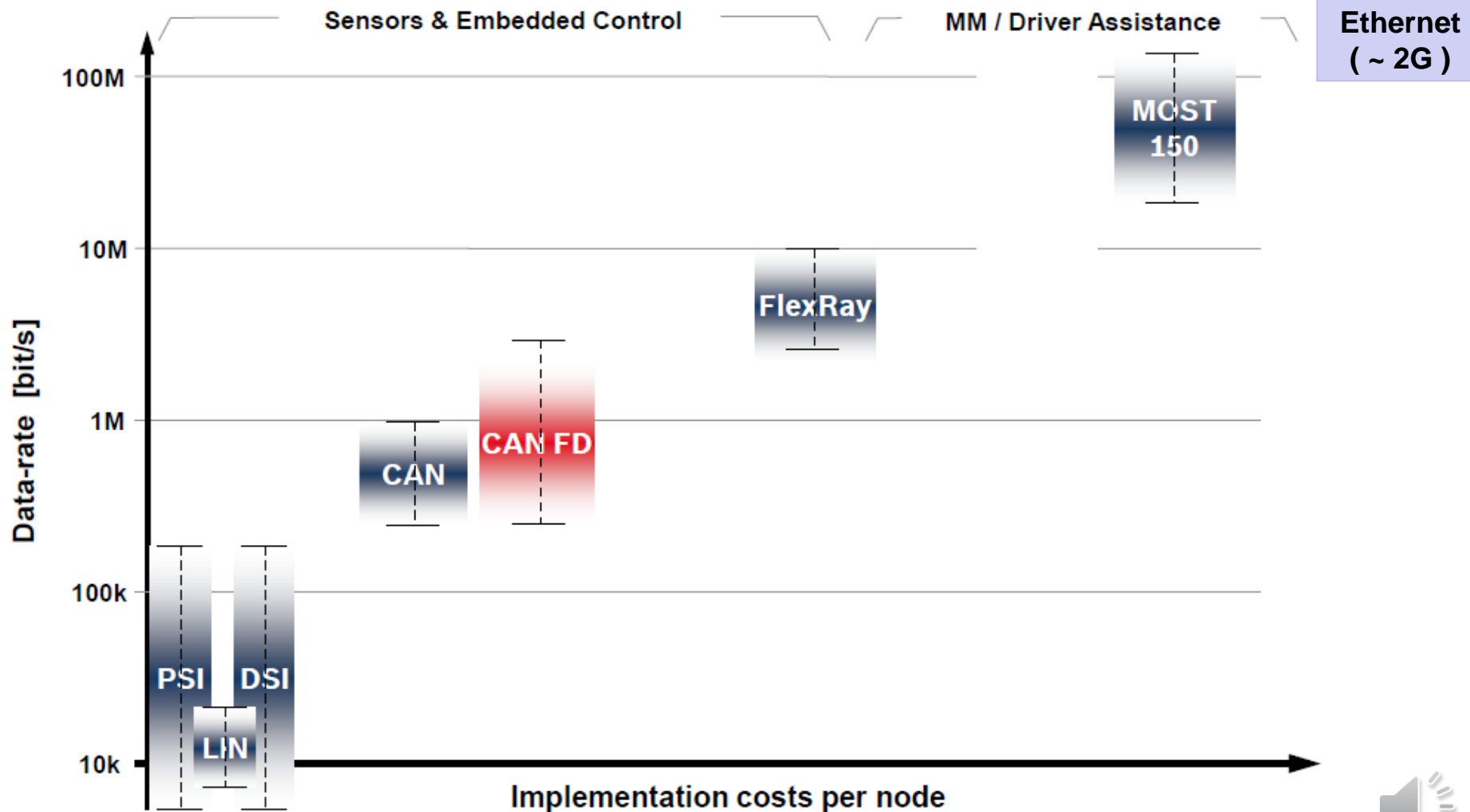
T_q : Time Quanta, BRP : Brad Rate Prescaler, f_{sys} : System frequency

$$T_q = BRP/f_{sys} \rightarrow F_q = f_{sys}/BRP$$





CAN with Flexible Data-Rate



CAN-FD



Speeding up CAN

Unchanged

- CAN arbitration
- CAN acknowledge mechanism

New

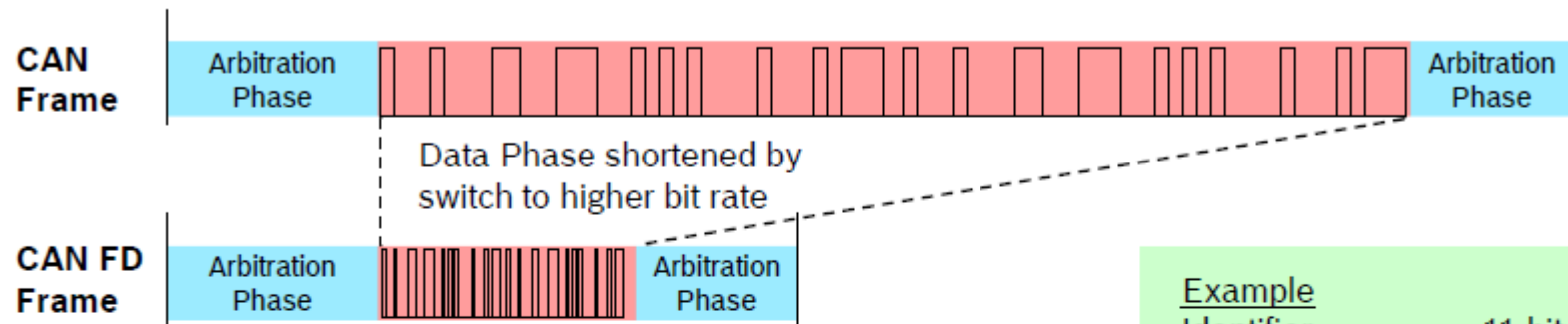
- switch to higher bit rate for transmission of
 - Data Length Code
 - Data Field
 - Frame CRC
- data fields with more than eight bytes possible
 - configured by unused DLC codes “1001” to “1111”
 - 12, 16, 20, 24, 32, 48, 64 bytes
- new CRC polynomials for longer data fields, HD=6
 - 17 bit: up to 16 byte data fields, 21 bit: up to 64 byte data fields





CAN-FD

Speeding up CAN



Example

Identifier	11 bit
Data Field	32 byte
Arbitration Phase	1 Mbit/s
Data Phase	4 Mbit/s

→ average bit rate 3.1 Mbit/s

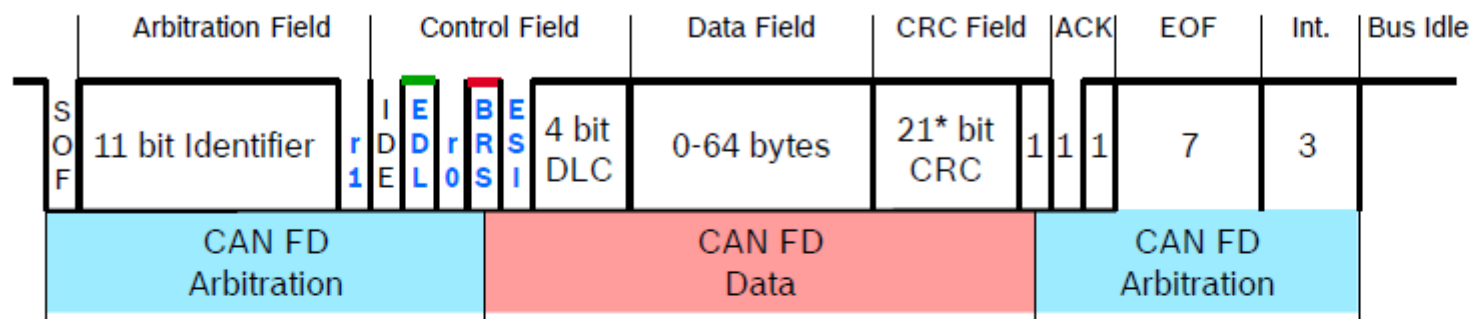
- Based on existing CAN
 - well known technology, minimized risk
 - **changes limited to HW:** protocol controller
 - for bit rates up 1 Mbit/s standard CAN transceivers usable
 - **no changes to SW:** with 8 bytes data field (legacy SW fully compatible)
 - even higher data rate possible by data fields >8 bytes and SW change
- Costs similar to CAN





CAN-FD

CAN FD Standard Frame



* 17 bit CRC for data fields with up to 16 bytes

- CAN FD Arbitration Phase
 - length: 30 bit times*
 - data rate: max. 1 MBit/s
- CAN FD Data Phase
 - length: 86 bit times* (8 data bytes)
 - data rate: > 1 MBit/s
- Remote Frames always in CAN Format
 - RTR bit replaced by reserved bit r1
 - r1 takes part in CAN arbitration
 - reserved for protocol expansion

EDL – Extended Data Length

Substitutes first reserved bit in standard frames

EDL = recessive indicates CAN FD frame format
(new DLC-coding and CRC)

EDL = dominant indicates standard CAN frame format

r1, r0 – reserved bits

Transmitted dominant, reserved for future protocol variants

BRS – Bit Rate Switch

BRS = recessive: switch to alternate bit rate

BRS = dominant: do not switch bit rate

ESI – Error State Indicator

ESI = recessive: transmitting node is error passive

ESI = dominant: transmitting node is error active

* bit stuffing not considered



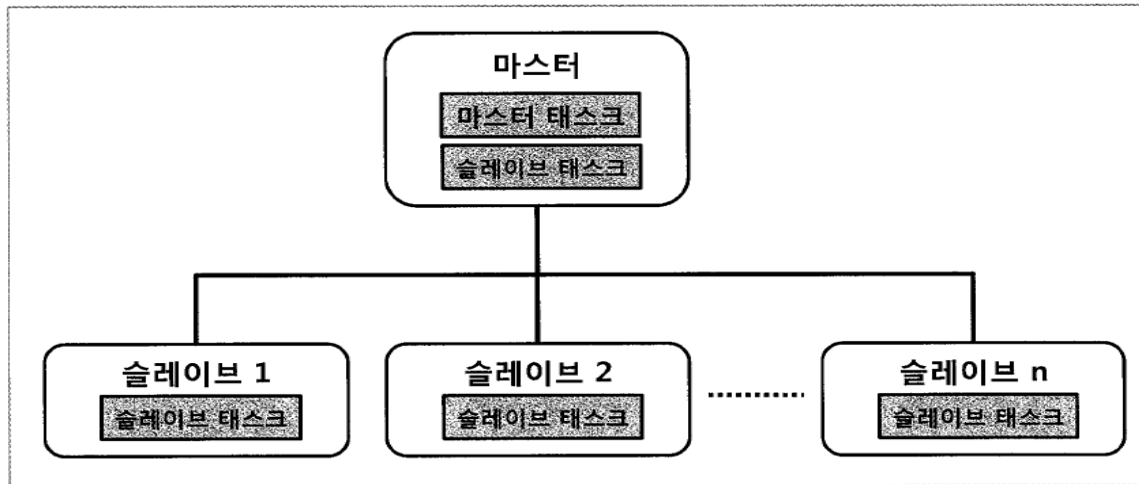
LIN



- ❖ LIN = Local Interconnection Network
- ❖ 기존 직렬통신보다 우수한 다중 통신 시스템 구축 가능
- ❖ 오직 1개의 Master와 여러 개의 Slave Node로 구성되기 때문에 Multiple Master를 지원하는 CAN 보다는 기능이 제한적
- ❖ Master가 제어하므로 데이터 충돌이나 Bus 중재가 필요 없음
- ❖ 저가의 8bit MCU와 Standard Serial Asynchronous Receiver/Transmitter 를 사용해 CAN 보다 낮은 비용으로 차량 네트워크를 구성 가능
- ❖ 단일 선로를 사용하기 때문에 Noise에 약해 전송 거리가 짧고, 전송 속도도 20kbps로 제한적임



OSI Model



계층	OSI 모델	설명	LIN 사양서 (Rev1.3)
7	애플리케이션	작동중인 사용자 프로그램	애플리케이션 프로그램의 인터페이스 명세
6	프리젠테이션	데이터 포맷 규정	없음
5	세션	메시지 데이터 순차 처리 기술	없음
4	전송	메시지와 패킷 간 변환 작업 (재전송, 오류복구)	전송 계층 명세(Rev2.0)
3	네트워크	데이터 전송 방법 (라우팅, 어드레싱)	없음
2	데이터 링크	데이터 비트 정렬과 패킷 조직 (체크섬, 프레임িং)	진단 명세 프로토콜 명세
1	물리	신호 해석 물리적 특징 기술	물리 계층 명세



Application Area



Body Domain

- 미러: 전동 도어, 전동 미러, 눈부심 방지 룸 미러
- 윈도우: 끼임 방지 기능을 포함하는 전동 윈도우
- 도어: 운전자, 조수석, 승용석의 도어 열림/잠김
- 선루프: 전동 개폐
- 자동 와이퍼: 강우 센서를 사용해 자동으로 와이퍼 작동
- 파워 시트: 운전자를 인식한 자동으로 시트 조절
- 에어컨: 각종 센서를 사용한 에어컨

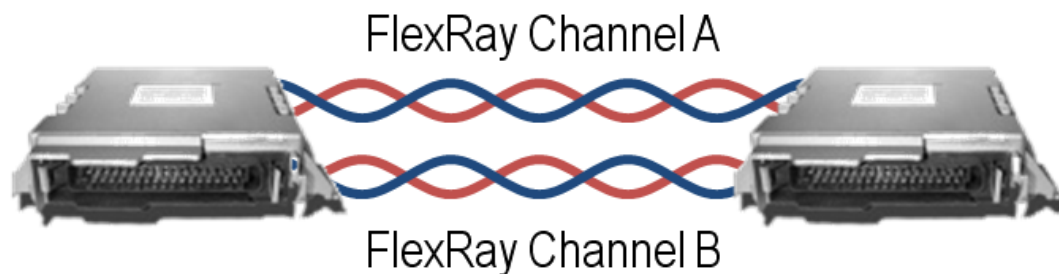




FlexRay 개요

❖ FlexRay Consortium

- 2005년 Protocol 2.1 발표 → **X by Wire** 지향
- 2006년 BMW X5 suspension에 최초 적용



- Freescale Semiconductor
- Robert Bosch GmbH
- NXP Semiconductors
- BMW AG
- Volkswagen AG
- Daimler AG
- General Motors

- Fault Tolerant 위해 동일 메시지 중복 전송
- 데이터 량이 중요한 경우, 다른 메시지 전송도 가능
- TDMA(Time Division Multiple Access)
- Cluster : 1 또는 2개의 Channel로 연결된 node들의 집합
- Cluster 내의 모든 node들은 Global clock을 공유
- Node는 host와 통신 controller로 구성
- Host는 application program, OS, MCU 포함





CCP 개요

❖ CCP (CAN Calibration Protocol)

- Calibration, Measurement, Diagnostics device와 ECU간의 표준 Protocol
- Audi, BMW, Mercedes-Benz, Porsche, VW 등이 ASAM(Association for Standardization of Automation and Measuring Systems) 전문위원회 설립 하여 1999년 Version2.1 배포

❖ 응용 분야

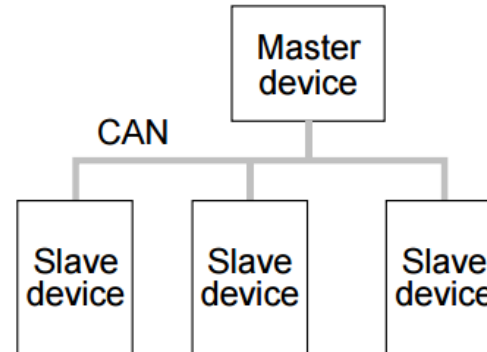
- 특징
 - OSI Model layer 7 interface 지원
 - Physical layer는 통상적으로 CAN 표준으로 구현
 - 1 to N communication 이므로 여러 개 ECU와 통신 가능
 - 기본적으로 Monitor program interface
- 적용 부문
 - ECU에서 실시간 메모리 관리(Read/Write)
 - ECU에서 실시간 데이터 취득
 - ECU 제어 알고리즘 실시간 바로잡기
 - 보드 레벨과 실제 차 환경에서 ECU 평가
 - 개발자의 설계 변경 평가
 - ECU 플래시 리프로그래밍
 - 실험실 환경에서 시뮬레이션 평가





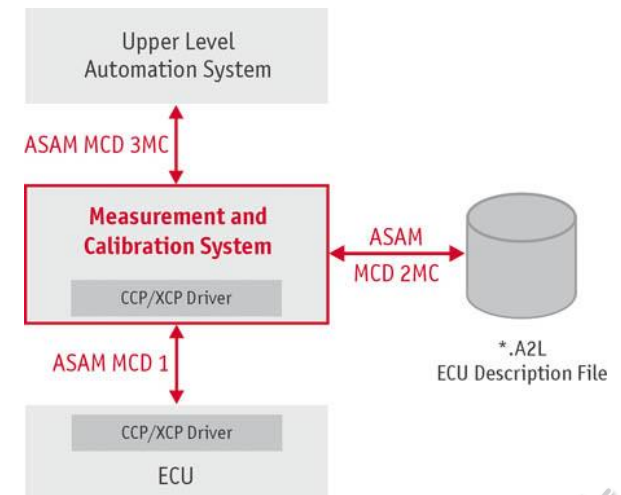
Protocol

❖ Master-Slave structure



❖ ASAM 3 Step Interface

- ASAM-MCD-1a : 제어 장치에 대한 직접적인 HW I/F (CAN, K-Line, ...)
- ASAM-MCD-1b : Monitor program과 PC와의 SW I/F (Device driver I/F)
- ASAM-MCD-2 : Slave의 Calibration parameter와 Measurement signal을 기술하는데 사용되는 DB file
- ASAM-MCD-3 : Calibration 시스템의 원격 제어를 위한 표준 protocol (Ethernet or RS-232)



MCD(Measurement Calibration Diagnostics)





KWP2000 개요

❖ KWP (KeyWord Protocol)

- OBD (On-Board Diagnostics) 차량의 상태를 감지하고, 통보하는 기능
- MIL (Malfunction Indication Lamp)
- Repair Shop의 장비와 차량 제어기 사이의 통신 Protocol

❖ OBD 역사

연도	주요 사건
1960년대 말	CARB(캘리포니아 대기 자원국)는 LA 지역의 심각한 대기 문제의 대응책으로 1966년 차량 모델부터 배기가스 제어 시스템 장착을 의무화했다.
1969년	폴크스바겐에서 스캔 기능의 온보드 컴퓨터 시스템을 최초로 소개했다.
1975년	연료 주입 시스템에 실시간 제어 시스템이 필요했던 닛산은 표준이 정해지지 않는 상태에서 무엇을 감시하고 어떻게 보고할지에 대한 단순한 OBD를 상용차(DATSUN 280Z)에 구현했다.



KWP2000 개요



1980년	GM은 차량 조립 공정에서 ECM(Engine Control Module) 검증용의 인터페이스와 프로토콜인 ALDL(Assembly Line Diagnostic Link)을 구현했다. 이는 GM 내부적인 표준으로 OBD-II라 불렸지만, 기술표준은 정하지 못했다. 매우 적은 차량 검증용으로 오직 GM 공장 내에서 사용된 ALDL은 160 버드 레이트 속도로 PWM 신호를 사용했다. 이와는 별도로 캐딜락 모델에는 고장 코드, 액추에이터 테스트, 센서 데이터를 제공하는 OBD를 장착했다.
1986년	GM은 반이중 통신방식(Half-Duplex)의 UART 신호를 사용하고 속도를 8192 버드 레이트로 한 단계 진화시킨 ALDL 프로토콜을 발표했다.
1987년	CARB에서 1988년부터 생산되고 캘리포니아에서 판매되는 모든 차량에 OBD-II라 불리는 OBD 기능 포함을 의무화했다. 하지만 커넥터 형태나 데이터 프로토콜은 표준으로 정하지 못했다.
1988년	SAE는 표준화한 진단통신 신호방식과 커넥터 표준안을 제시했다.
1991년	CARB는 캘리포니아에서 팔리는 모든 차량에 기본적인 OBD 기능을 장착하는 법률을 제정했다. OBD-II로 불리는 이때의 OBD는 프로토콜뿐만 아니라 커넥터, 커넥터 위치가 표준으로 규정되지 않았다.
1994년	미국 전역에서 추진된 배출 테스트 프로그램이 동기가 돼 CARB는 1996년 이후 캘리포니아에서 판매되는 모든 차량에 장착해야 하는 OBD-II 표준을 발표했다. 이 표준은 SAE에서 제안한 DTC(Diagnostic Trouble Code) 코드와 커넥터 표준안을 포함하고 있다.
1996년	OBD-II는 미국에서 팔리는 모든 차량에 의무적으로 적용하도록 법제화했다.
2001년	유럽연합(EU)은 유럽연합 내에서 팔리는 모든 M1(승객 8명이 하, 차량 무게 2,500Kg 이하) 가솔린 차량에 OBD-II를 개선한 EOBD(European OBD) 장착을 의무화했다(디젤 차량은 2004년 이후 적용됐다).
2008년	미국에서 팔리는 모든 차량에 ISO15765-4(Diagnostic on CAN part4) 표준 구현을 의무화했다. EPAO(Environmental Protection Administration Office)는 2008년 7월 1일까지 중국에서 생산된 경차에 OBD 기능 구현을 의무화했다.
2010년	미국에서 팔리는 지정된 상업용 엔진에 대해 HDOBD(Heavy Duty OBD) 구현을 의무화했다.



KWP2000 개요



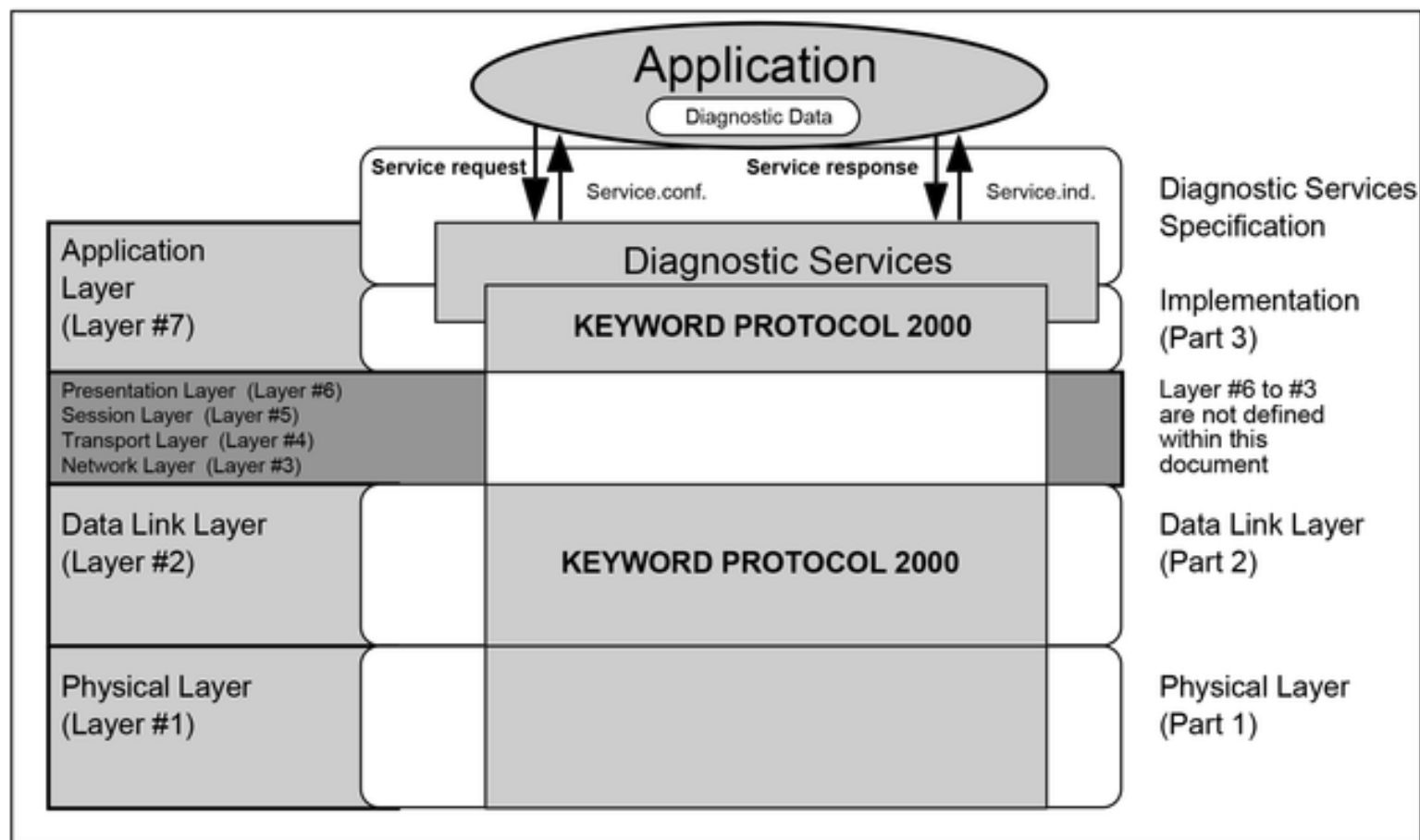
버전	설명															
OBD-I	OBD-I을 법률화한 의도는 자동차 생산업체가 신뢰성 있는 배기 제어 시스템(Emission Control System) 개발을 촉진하게 하기 위해서였다. 하지만 OBD-I은 표준화되지 않은 진단 정보의 전달, 구현의 기술적 어려움 때문에 궁극적으로 실패했다.															
OBD-1.5	1994년과 1995년 사이 GM의 몇몇 자동차에 적용한 OBD-1.5는 OBD-II가 표준화되기 전에 OBD-II 일부분을 구현 적용했다. 커넥터는 총 16핀으로 구성됐으며, 핀 모양은 본문의 그림 12-1과 같다. OBD-1.5(ALDL)의 9번 핀은 데이터 라인, 4번과 5번 핀은 그라운드(GND), 16번 핀은 배터리 전원으로 사용한다. OBD-1.5 표준에 기반을 뒤 생성된 코드를 취득하려면 추가 검증장비가 필요했다. PCM(Power-Train Control Module), CCM(Chassis Control Module), 에어백 시스템, 타이어 압력 경고등 진단 데이터 전송용이 있다.															
OBD-II	<p>OBD-II는 진단 능력과 표준화 면에서 OBD-I을 개선했다. OBD-II 표준은 진단 커넥터, 핀 모양, 프로토콜 신호, 메시지 포맷을 명확하게 기술했다. 기존에 별도로 검증장비에 전원을 인가했던 방식과는 다르게 커넥터의 특정 핀을 통해 자동차 배터리의 전원을 공급받는 방식으로 표준화했다. OBD-II 표준은 DTC 코드 리스트를 추가로 표준화했다. 이 표준화 결과로 검증장비에서 자동차의 종류에 관계 없이 ECU로 진단정보를 요청할 수 있게 됐다. 참고로 DTC 코드는 아래와 같이 5개의 캐리터로 구성됐다.</p> <p>DTC 코드의 5개 캐리터</p> <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>시스템 명칭</td><td>DTC 종류</td><td>서브 시스템</td><td colspan="2">결함 명</td></tr><tr><td>P</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table> <div><div><p>1) 시스템 명칭</p><p>P: 파워트레인</p><p>B: 보디</p><p>C: 새시</p><p>U: 미지정</p><p>2) DTC 종류</p><p>0: 표준 명세</p><p>1: 제조자 정의 명세</p></div><div><p>3) 서브 시스템</p><p>1: 연료 혹은 공기/연료</p><p>2: 연료제어 - 인젝터</p><p>3: 시동 혹은 실화</p><p>4: 배기 제어</p><p>5: 엔진/차량 속도</p><p>6: 마이크로프로세서</p><p>7: 변속기</p><p>8: 변속기</p><p>9: SAE 지정</p><p>0: SAE 지정</p><p>4) 결함 명</p></div></div>	1	2	3	4	5	시스템 명칭	DTC 종류	서브 시스템	결함 명		P	1	1	0	1
1	2	3	4	5												
시스템 명칭	DTC 종류	서브 시스템	결함 명													
P	1	1	0	1												

DTC
(Diagnostic Trouble Code)





KWP 2000





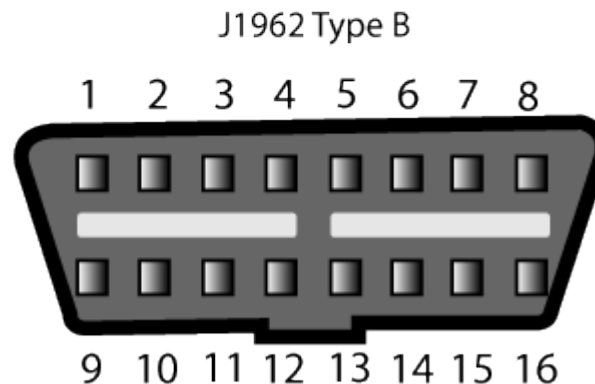
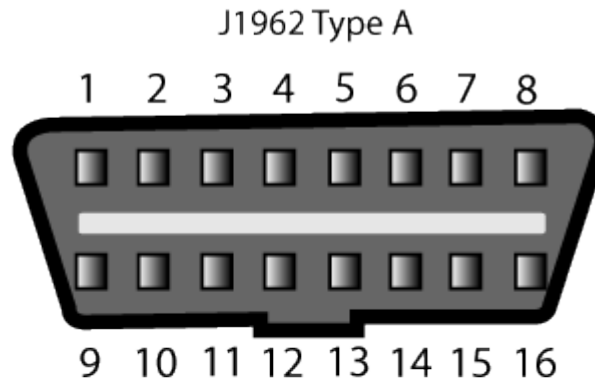
DLC

❖ DLC (Data Link Connector)

- SAE-J1962 표준 (Society of Automotive Engineer)



- VPW-PWM (SAE-J1850)
- CAN (ISO 15765)
- K-Line, L-Line
(ISO 9141-2, ISO 14230-4)

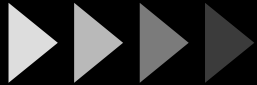


DLC Pinout

1. Manufacturers
2. Bus Positive Line
3. Ford DCL Chrysler CCD
4. Chassis Ground
5. Signal Ground
6. CAN High
7. K Line
8. Manufacturers
9. Manufacturers
10. Bus Negative
11. Ford DCL Chrysler CCD
12. Manufacturers
13. Manufacturers
14. CAN Low
15. L Line
16. Battery Positive



Thank You !



www.chungbuk.ac.kr



충북대학교
CHUNGBUK NATIONAL UNIVERSITY

