

CAN

(Controller Area Network)

↳ 32비트 영장기
전차 통신
1대 1 통신
고속 (500k-1M)

2023. 2학기

Kookmin Univ. EMCO Lab.

◆ 제어보드 간의 CAN 통신을 이용한 모터 구동

- CAN 통신 \Rightarrow 선으로 두 개만 있으면 됨.

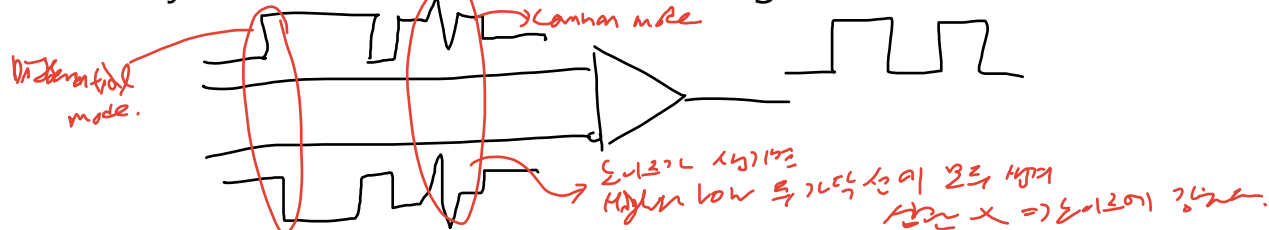
독일의 Robert Bosch사에서 자동차 전장용으로 개발.
자동차뿐만 아니라 여러 가지 산업용 장치에 응용이 확대.
최근 제작되는 많은 마이크로 컨트롤러, DSP 등에 기본적으로 내장되어 출시

- 특징

- 멀티 마스터(Multi-master) 구조
- 메시지 ID로 우선 순위 판별
- 장치 어드레스가 아닌 메시지 어드레싱 방법
- 데이터 수: 최대 8 바이트
- 전송 속도: 최대 1 M BPS
- 국제 표준으로 지정 (ISO11898, ISO11519)
- 0~8 byte로 설정 가능한 32개 Message Buffers(MB)

저장, 인접해온 신호, 같은 ID가 여러 개 있음
 \Rightarrow ID에 있는 수를 정하는 통신 ID의 수를
(이런 것들 통신 속도 ↓)

통신속도	최대 거리
1Mbps	40m
500kbps	100m
250kbps	200m
125kbps	500m

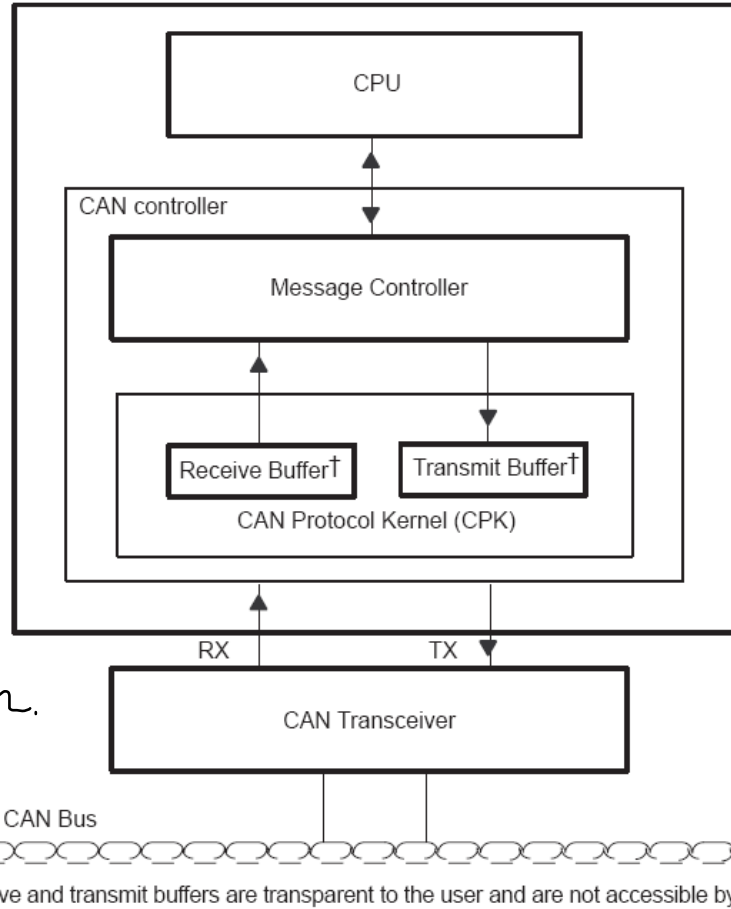
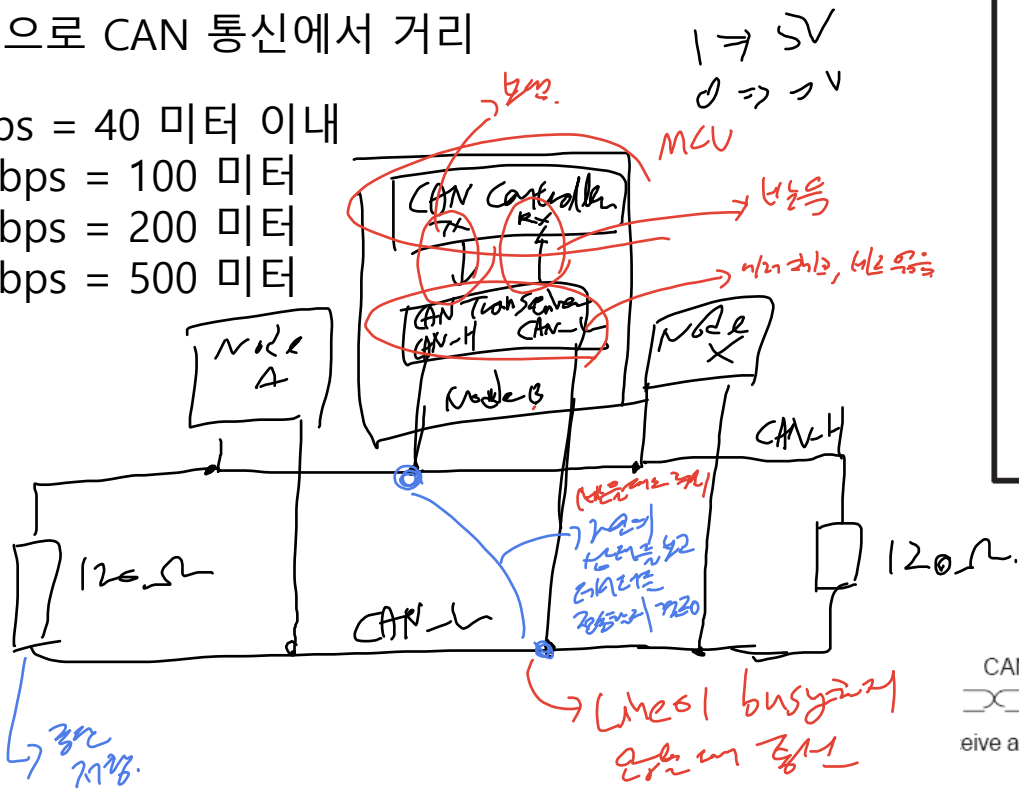


1. CAN 통신

=> multi master 구조 CAN

- CAN 통신 거리
- 일반적으로 통신 속도가 높으면 거리가 짧아지고, 통신 속도가 낮으면 먼거리까지 통신
- 정확한 통신 거리는 선로의 특성 설치 상태에 따라 달라질 수 있으므로 얼마라고 딱 단정할 수는 없다.
- 일반적으로 CAN 통신에서 거리

1Mbps = 40 미터 이내
500Kbps = 100 미터
250Kbps = 200 미터
125Kbps = 500 미터

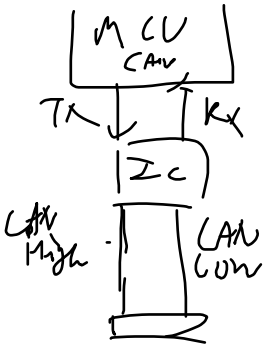


CAN 통신 장점

비동기 통신 방식과 CAN 통신 비교:

비동기(SCI) 통신은 1문자 단위로 통신이 이루어진다. 일반적으로 장치 간에 데이터 통신을 할 때 여러 문자로 구성된 통신 프레임을 사용하여 통신한다. 이러한 경우에 비동기 통신은 매 문자 단위로 소프트웨어가 개입하여 프로토콜을 형성 또는 분석해 주어야 하므로 소프트웨어의 부담이 크다.
예를 들어 아래와 같이 1바이트 단위의 데이터를 조합하여 프로토콜을 형성하여 통신하는 경우 1 문자 단위로 인터럽트가 발생하여 처리를 해주어야 한다. 또한 에러 체크를 위해 SUM 기능도 소프트웨어적으로 형성해 주어야 한다.

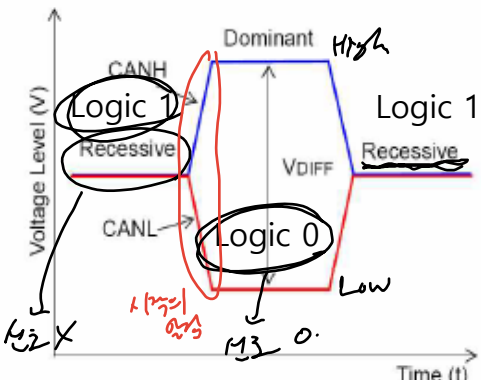
비동기 통신에 비해 CAN 통신에서는 하드웨어적으로 프로토콜 관리를 해준다. CRC 체크, 데이터 송수신, 응답 처리, 충돌 시 재 전송 등을 CAN 모듈에서 하드웨어적으로 처리해 준다. 정상적 통신 동작 완료 또는 에러 발생 시에만 인터럽트를 요구하여 소프트웨어에서 처리를 하므로 프로그램 부담이 적어진다.



1. CAN 통신

CAN

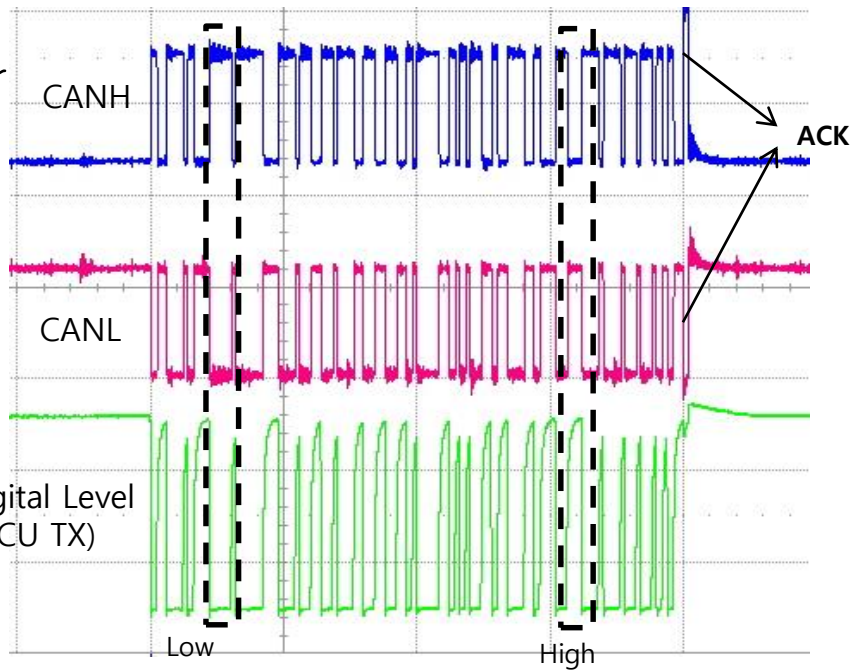
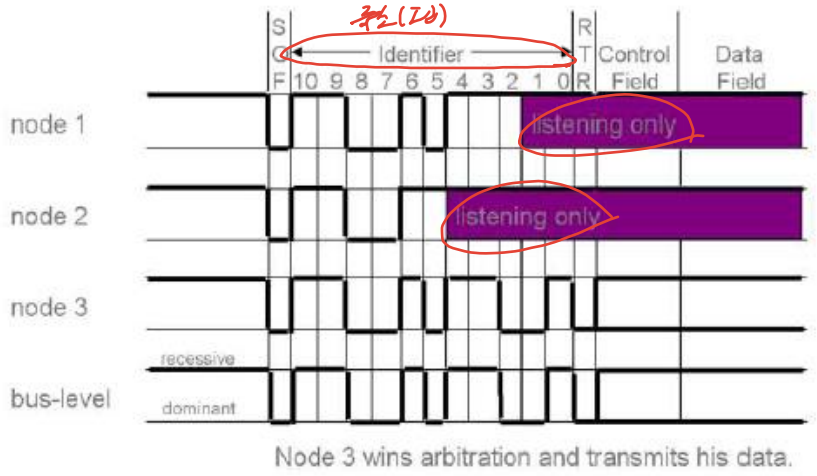
① Physical layer(물리적 특성)



(voice) (노이즈)
신호-노이즈
High/Low level
기준

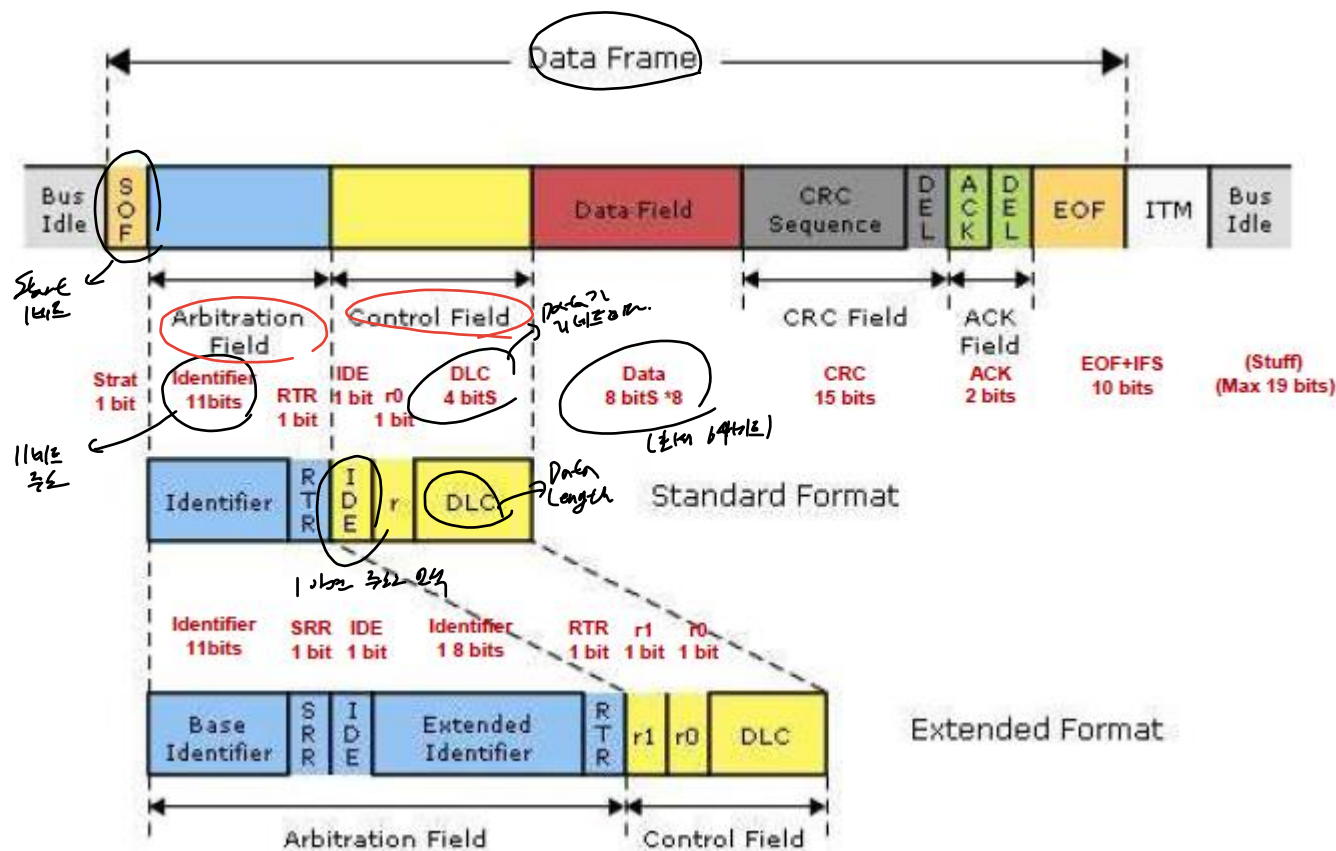
OTA: CAN은 급제 616
↳ 2중중심이 표준

clock (bps) 느리다
↳ 실시간 관계에 따라 1은 3분



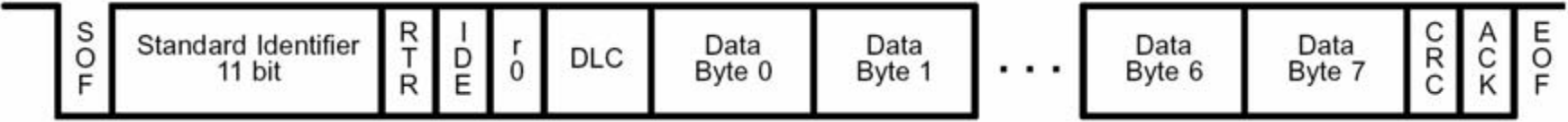
- Dominant bit : Bus에 우선하여 나타나는 bit, Digital Logic "0"
- Recessive Bit : Dominant bit에 대하여 무시될 수 있는 bit, Digital Logic "1"
- Dominant bit가 Recessive bit에 우선하며, 따라서 ID가 낮은 메시지가 우선

② Protocol



1. CAN 통신

기본 Format (버전 2.0A) : 11비트 ID 영역 => 11비트 ID



확장 Format (버전 2.0B) : 29비트 ID 영역.

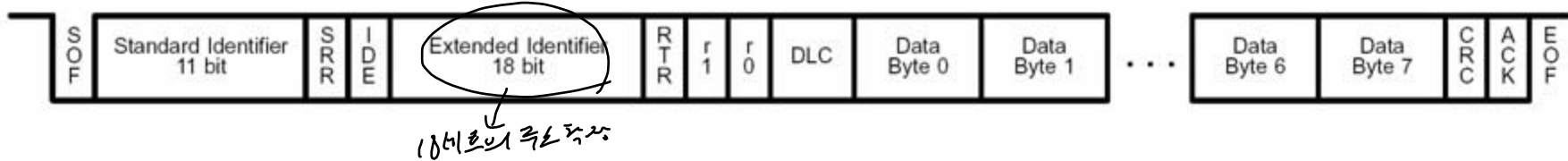
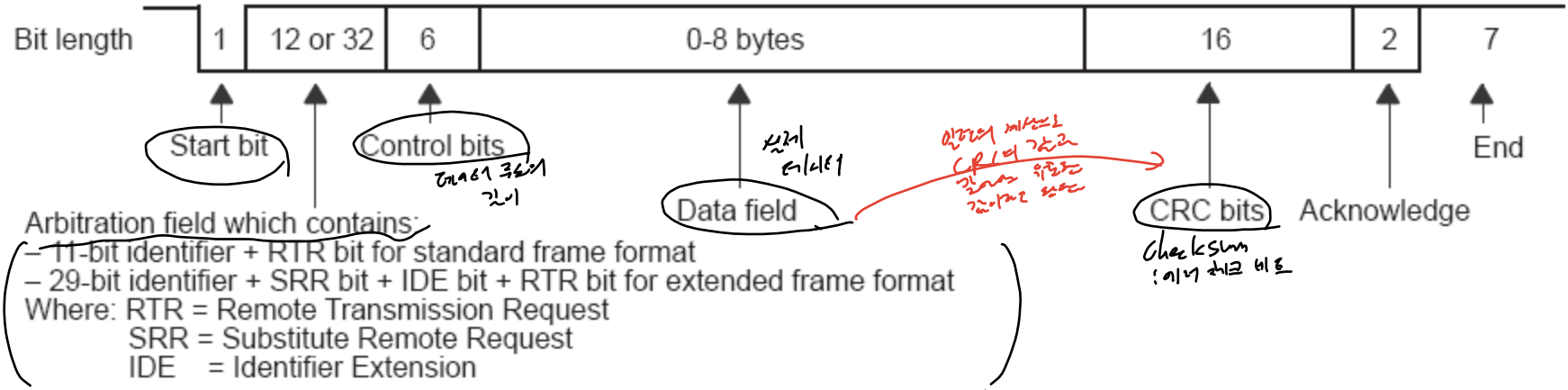


Figure 1-2. CAN Data Frame | Data Frame (최대 16 bytes)



Note: Unless otherwise noted, numbers are amount of bits in field.

1. CAN 통신

SOF (Start of Frame): 프레임의 시작 의미

Identifier: 표준 프레임(11 비트), 확장 프레임(11+18 = 29 비트)

이 ID로 메시지 우선 순위 결정 및 메시지 수신 여부 결정.

RTR (Remote Transmission Request): 데이터 프레임과 리모트 프레임 구분
데이터 프레임은 항상 0 (dominant) 상태 임.

SRR (Substitute Remote Request): 표준 프레임의 RTR 위치 점유

IDE (Identifier Extension): 표준 프레임과 확장 프레임 구별

r0, r1: 예약 기동 비트 임.

DLC (Data Length Code): 데이터 프레임의 데이터 바이트 수, 범위: 0 ~ 8

Data 8 바이트 크기, MSB(Most Significant Bit)부터 송신 됨

CRC (Cyclic Redundancy Check): 16 비트 체크섬(checksum)

ACK (Acknowledgement): 응답

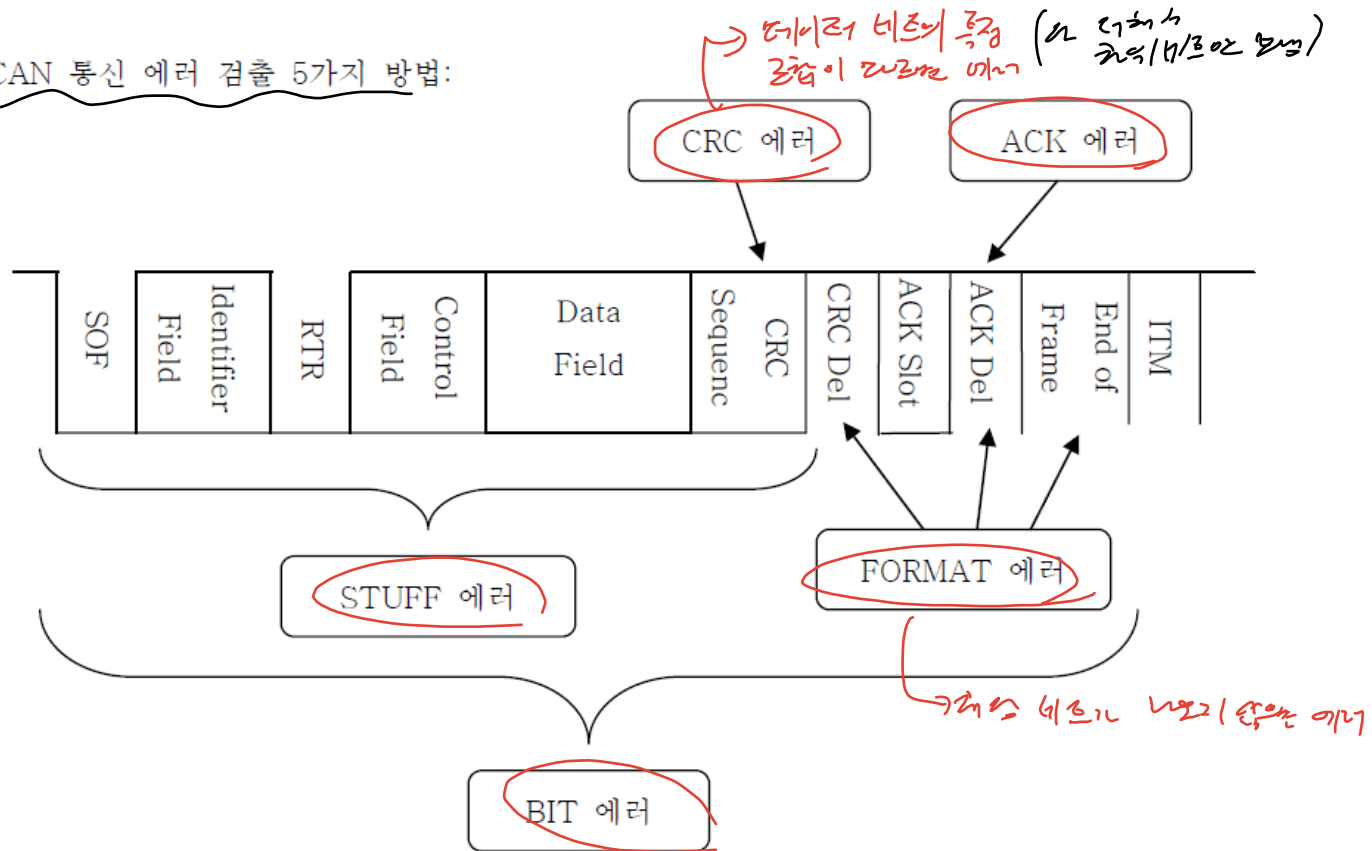
EOF (End of Frame): 프레임의 끝 의미

↳ 송수신은 리시버가 결정해서 "언제" 보내
(리시버의 계산값과 송신하는 송신기의 리시버)
↳ 일치하지 않으면 버림

* RS 232 의 XOR로 모든 데이터 연산하여
받은 데이터와 비교하여 일치하면 성공.

리시버는
받은 데이터 저장

CAN 통신 에러 검출 5가지 방법:



송신 장치 발생 에러:

BIT 에러: 보낸 비트와 받은 비트가 서로 다르다면 비트 에러 발생

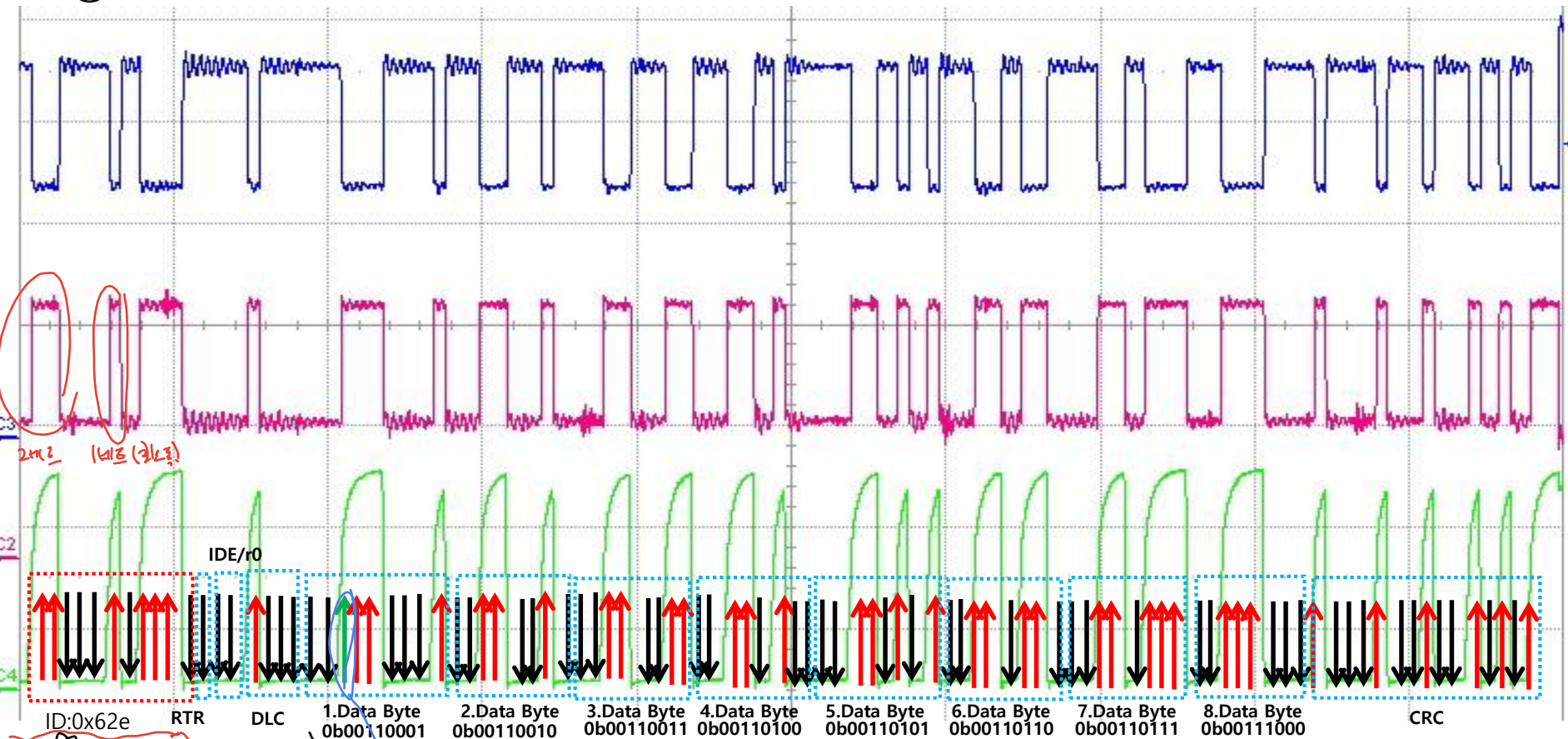
ACK 에러: ACK 슬롯 동안 수신 장치가 0을 보내 주지 않을 때 이 에러는 모든 수신 장치가 접속되어 있지 않을 때 (통신 선로가 끊겼을 때 등) 발생.

1. CAN 통신

CAN

③ Data Frame

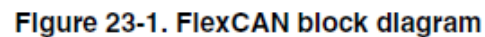
CANH CANL DIGITAL



- Bit stuffing(비트 채움)
1. CAN 컨트롤러에 내장된 오류 방지 메커니즘
 2. 같은 극성의 비트가 연속적으로 6개 이상 발생 하는 것을 방지
 3. 같은 극성의 비트가 5개 연속으로 발생하게 될 경우 다른 극성의 비트 1개를 삽입
- 아래가 5인 경우로 방지 (반복적인 데이터 삽입)

- CAN protocol Ver 2.0B
- 최대 1Mbit/s까지 bit rate 설정 가능
- Standard/Extended data, remote frames
- 32개 Message Buffers(MB) 32개의 버퍼에 저장 가능
- 각 MB는 0~8 byte로 설정 가능
- 각 MB는 Rx(수신), Tx(송신)로 설정 가능
- 수신 또는 송신된 메시지에 대한 16bit time stamp를 사용 할 수 있음

CAN



4. Register

4.1 Module Configuration Register(MCR)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MDIS	FRZ	FEN	HALT	NOT_RDY	WAK_MSK	SOFT_RST	FRZ_ACK	SUPV	0	WRN_EN	LPM_ACK	0	0	SRX_DIS	BCC	0	0	LPRI_O_EN	AEN	0	0	IDAM	0	0	MAXMB						
W																																
Reset	1	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

- MDIS

FRZ

FEN

HALT

NOT_RDY

WAK_MSK

SOFT_RST

FRZ_ACK

SUPV

WRN_EN

LPM_ACK

SRX_DIS

BCC

LPRIO_EN

AEN

IDAM

MAXMB
- : FlexCAN Module Enable

: Freeze Mode Enable

: FIFO Enable

: Halt FlexCAN

: FlexCAN이 Disalbe인 경우 알려주는 bit(Read only)

: Wake up Interrupt Mask

: Reset register(MCR, TIMER, TCR, ECR, ESR, IMASK1, IFLAG1)

: Freeze mode에서 prescaler가 멈췄는지 알려주는 bit(Read only)

: Supervisor Mode

: Warning Interrupt Enable

: FlexCAN이 Disalbe Mode 또는 Stop Mode로 동작

: Self reception Disabled

: Individual Rx masking and queue feature Enabled

: Local Priority Enable

: Abort Enabled

: ID Acceptance Mode

: Maximum Number of Message Buffers

BPS : bit per sec.

4.2 Control Register(CR)_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	PRESDIV								RJW		PSEG1		PSEG2			BOFF_MSK	ERR_MSK	CLK_SRC	LPB	TWRN_MSK	RWRN_MSK	0	0	SMP	BOFF_REC	TSYN	LBUF	LOM	PROPSEG			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- PRESDIV

RJW

PSEG1

PSEG2

BOFF_MSK

ERR_MSK

CLK_SRC

LPB

TWRN_MSK

RWRN_MSK

SMP

BOFF_REC

TSYN

LBUF

LOM

PROPSEG

: Prescaler Division Factor ($Sclock\ freq = CPI\ clock\ freq / (PRESDIV+1)$)

: Resync Jump Width (= RJW + 1)

: Phase Segment1 (Phase Buffer Segment 1 = (PSEG1+1) x one Sclock period)

: Phase Segment2 (Phase Buffer Segment 2 = (PSEG2+1) x one Sclock period)

: Bus Off Interrupt Enable

: Error Interrupt Enable

: CAN Clock Source 선택 bit

: Loop Back Enabled

: Tx Warning Interrupt Enabled

: Rx Warning Interrupt Enabled

: Rx입력에 대한 CAN Sampling Mode 설정

: Automatic recovering from Buss Off state Disabled

: Timer Sync feature Enabled

: Lowest Buffer Transmitted First 설정

: FlexCAN의 Listen-Only Mode 설정

: Propagation Segment (Propagation Segment Time = (PROPSEG+1) x one Sclock period)
- (CLOCK 분배)
100% 2/3 시리얼
매우 중요

4. Register

CAN

4.3 Message Buffer(MB) => 이 이론을 숙지하여 공부

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x0					CODE					SRR	IDE	RTR	LENGTH				TIME STAMP															
0x4	PRIO			ID (Standard/Extended)										ID (Extended) [17:16]		ID (Extended) [15:0]																
0x8	Data Byte 0								Data Byte 1								Data Byte 2								Data Byte 3							
0xC	Data Byte 4								Data Byte 5								Data Byte 6								Data Byte 7							

- CODE

SRR

IDE

RTR

LENGTH
- : Message Buffer의 기능 설정

: Substitute Remote Request

: ID Standard or Extend 설정

: Remote Transmission Request → 보내는 방법 숙지 필요

: 데이터의 크기 설정
- TIME STAMP

PRIO

ID

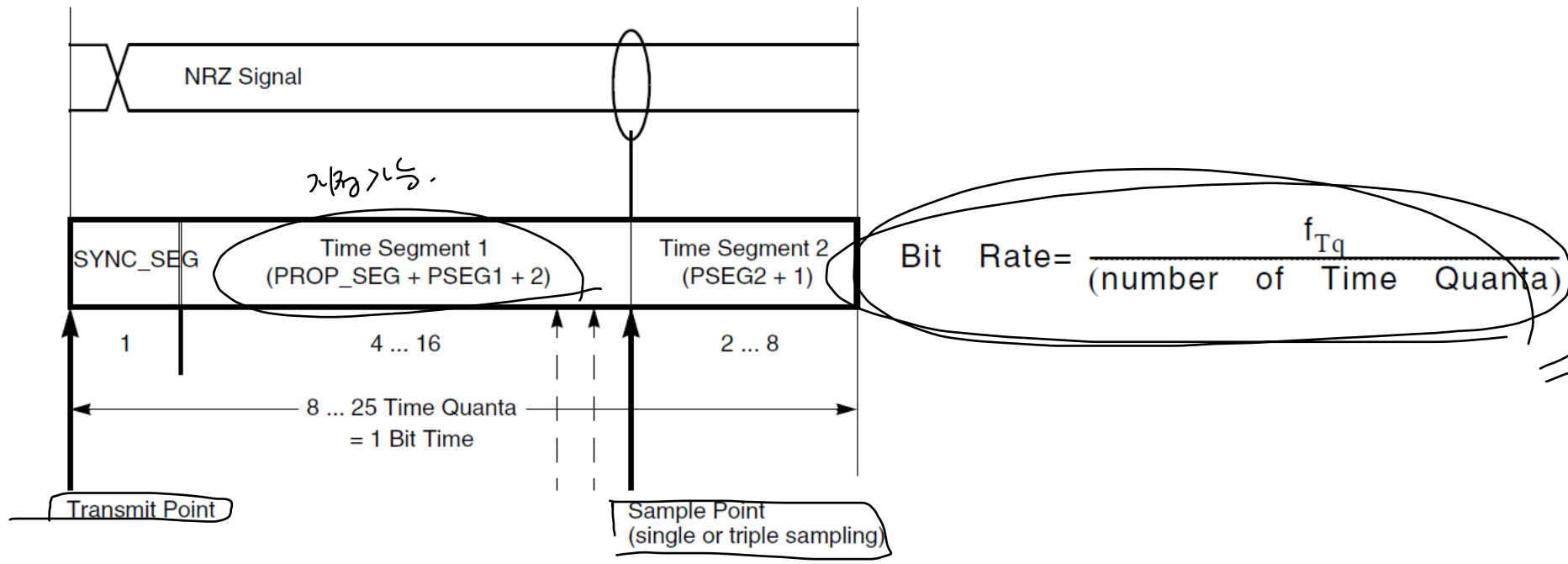
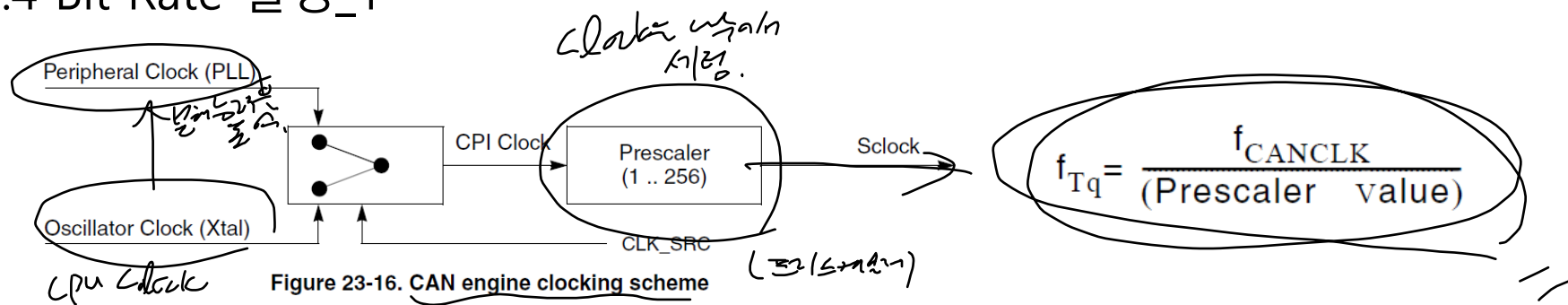
DATA
- : Free-Running Counter Time Stamp

: Local priority

: Frame ID

: Data Field

4.4 Bit Rate 설정_1



4.4 Bit Rate 설정_2

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

Time Segment 1	Time Segment 2	Resync Jump Width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4