

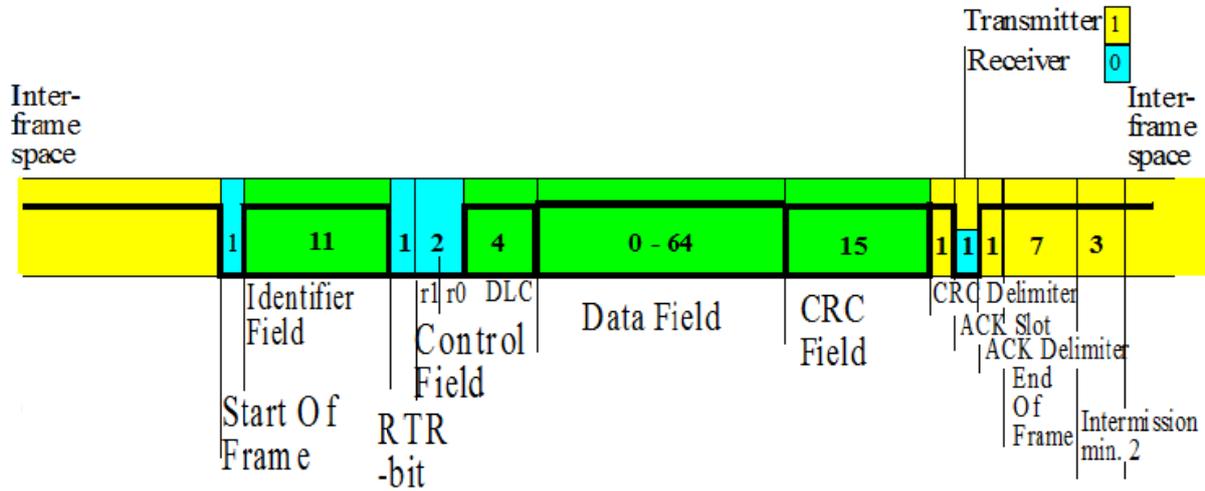
CAN의 일반적인 특징

CAN 특징

- 메시지 우선순위에 의한 버스-액세스
 - CSMA/CR (Carrier Sense Multiple Access / Collision Resolution)
- 버스 액세스 충돌을 중재로 해결
 - 비트 단위
 - 비 파괴
 - 명시된 대기 시간 허용
- 메시지 식별자
 - CAN 은 노드 주소가 없습니다. 모든 노드는 모든 메시지를 수신하며 그것을 사용할 것인지 아닌지를 스스로 결정합니다.
- 포괄적인 ERROR 검사
 - 서로 다른 다섯 개의 검사들
 - 연결된 모든 노드 참여
- 데이터 일관성 보장
 - 한 메시지는 모든 노드들에 의해 수용 또는 어떤 노드에 의해서도 수용되지 않습니다
- 서로 다른 버스 관리 방법들이 CAN 시스템에 적용될 수 있습니다. 예를들면,
 - 비트-단위 중재
 - Master/Slave
 - 직렬 연결 방식
 - TDMA
- Higher Layer Protocol 이 항상 필요 합니다
 - CAN 은 단지 하나의 low level specification 일 뿐입니다
- CAN 의 성능은 선택한 Higher Layer Protocol 에 의해 제한됩니다
 - Market segment
 - Real-time 필요 조건
 - Product Administration 필요 조건
 - 기타

CAN 메시지

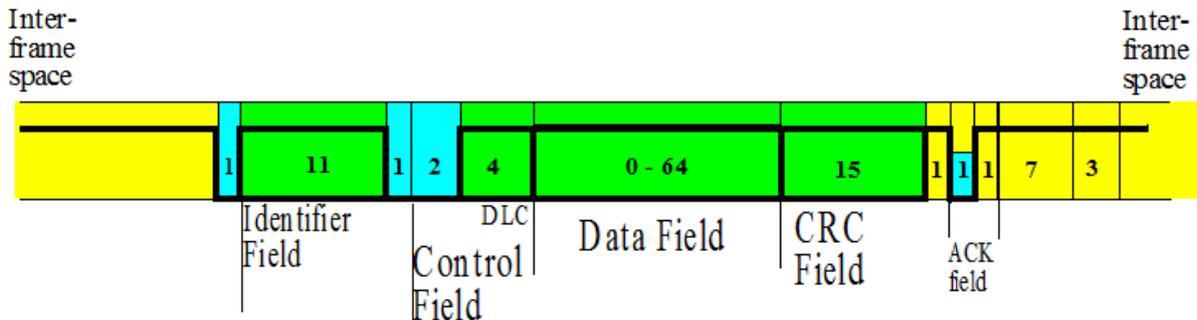
CAN Data Frame Std



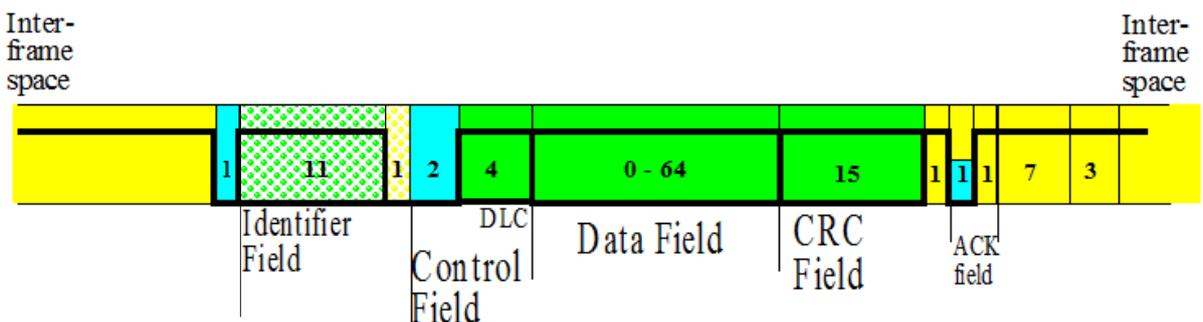
Bit values

- 0
- 0/1
- 1

CAN 프레임



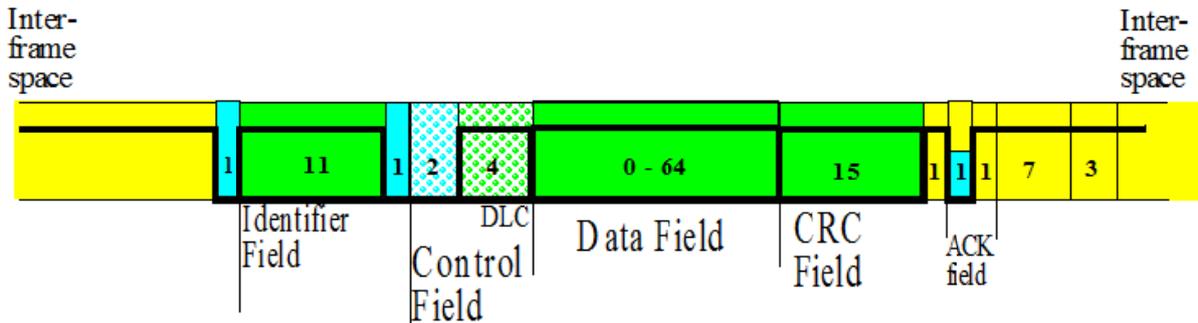
우선 순위와 식별



식별자 영역

- 11 비트 또는 29 비트. 11 비트는 위와 같습니다.
- 중재가 이 부분에서 이루어집니다.
- 충돌시 이 부분이 메시지의 우선순위를 설정합니다.
- Remote Transmit Request bit (RTR) 는 이 영역의 일부입니다.
- CAN-컨트롤러들은 실행되는 여과에 의한 식별로 메시지의 이 부분을 지원합니다.

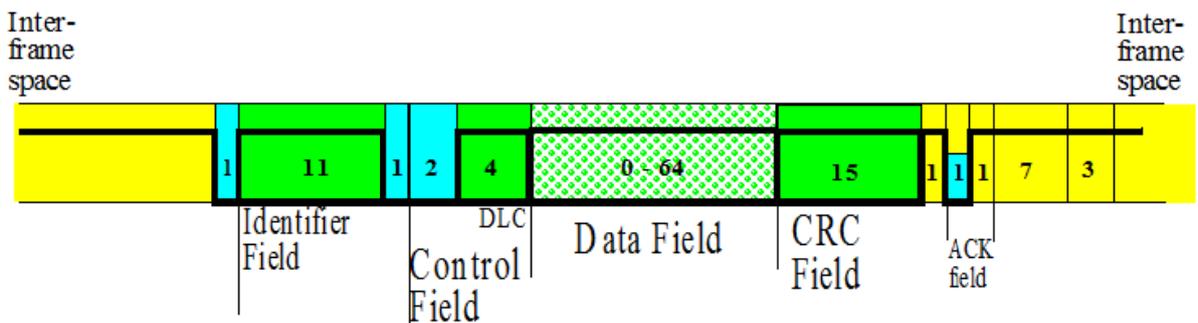
Control 과 Data length 영역



제어 영역

- 주요 함수 Data Length Code DLC
- DLC 는 0.8 의 값을 가질 수 있습니다 (8 이상의 값은 8로 해석 됩니다).
- 두 개의 비트들이 저장되어 Extended frames 을 나타내는데 사용 됩니다.
- Standard frames 에서는 저장된 비트들이 fix dominant bits 입니다.

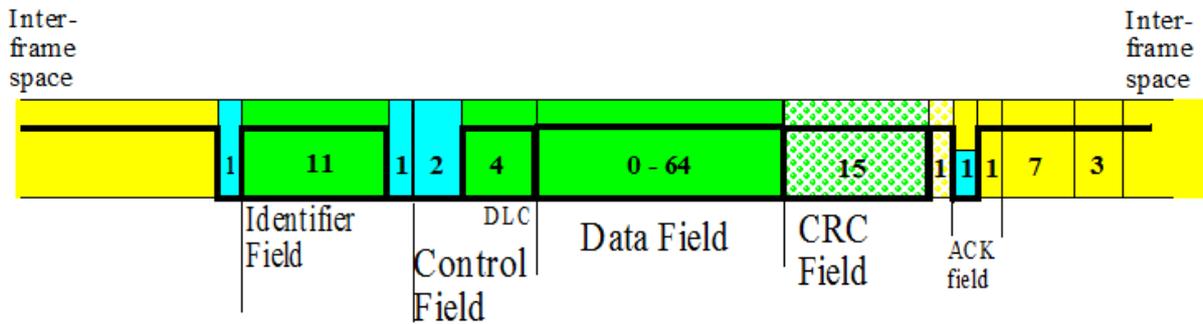
데이터 영역



데이터 영역

- 이 영역은 0 에서 8 byte 까지 될 수 있습니다.
- 이것은 항상 full 8 bit bytes 입니다.
- 이 바이트는 어떤 값도 가질 수 있습니다.
- 어떤 CAN 컨트롤러들은 ID 식별을 데이터 영역 안으로 확장할 수 있습니다.

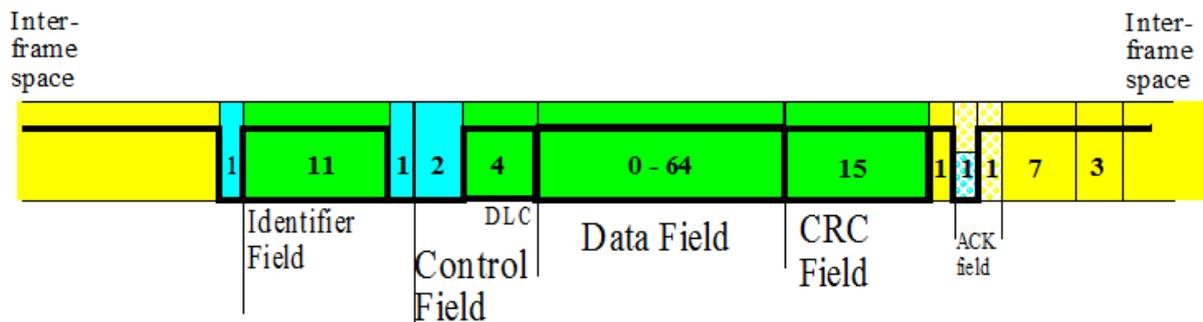
CRC 영역



CRC 영역

- 메시지에서의 비트 검사합니다.
- CRC 는 이러한 짧은 종류의 메시지에 최적입니다.
- CRC check 는 CAN 통신 오류 검사 중 한 가지입니다.

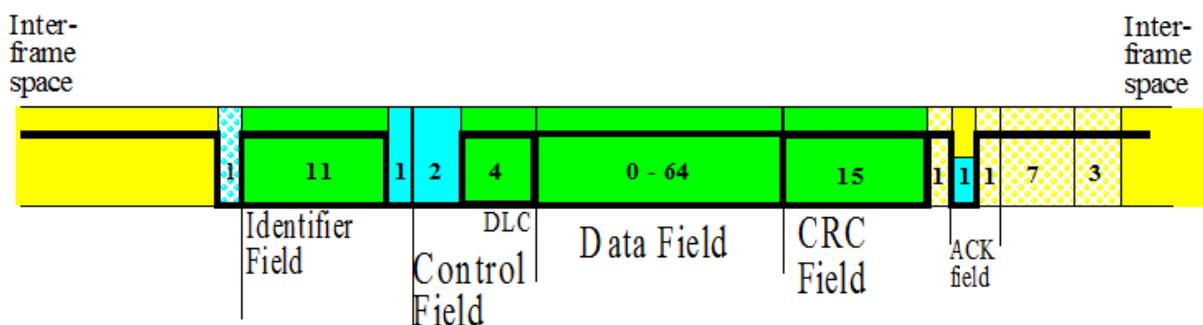
Acknowledgement 영역



확인응답 영역

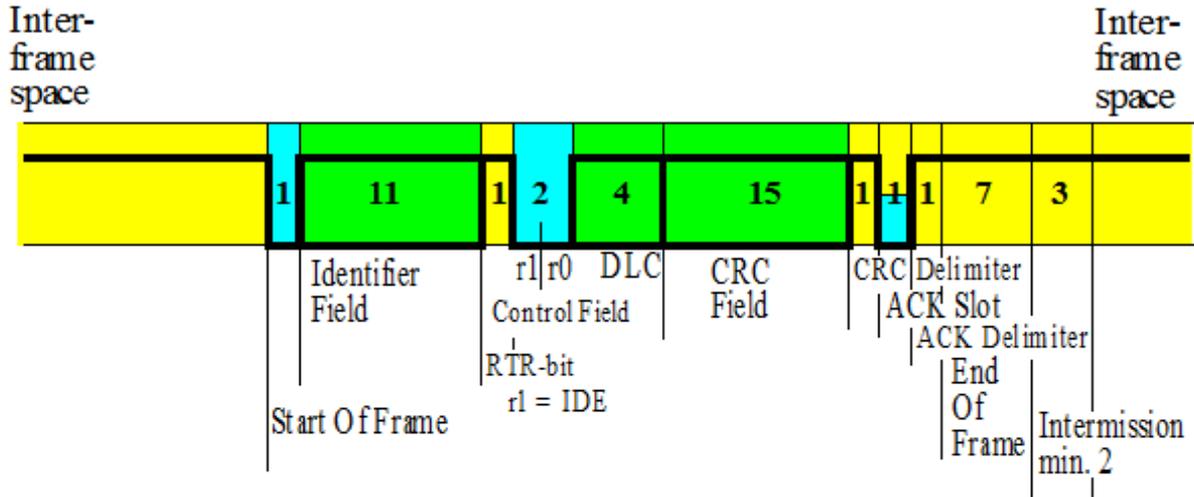
- 이것은 수신에 대한 확인응답으로, 적어도 한 개의 수신기는 메시지를 성공적으로 수신했음을 보증합니다.
- 송신기는 ACK bit 를 1 로 설정합니다.
- 모든 수신기는 메시지가 이상 없이 발견되었을 때 ACK bit 를 0 으로 설정합니다.

고정 값 비트



- 이러한 비트들은 모든 메시지 프레임들에 대해 고정된 값을 갖습니다.휴식(intermission) 비트에 대한 추가적인 규칙들이 있습니다.

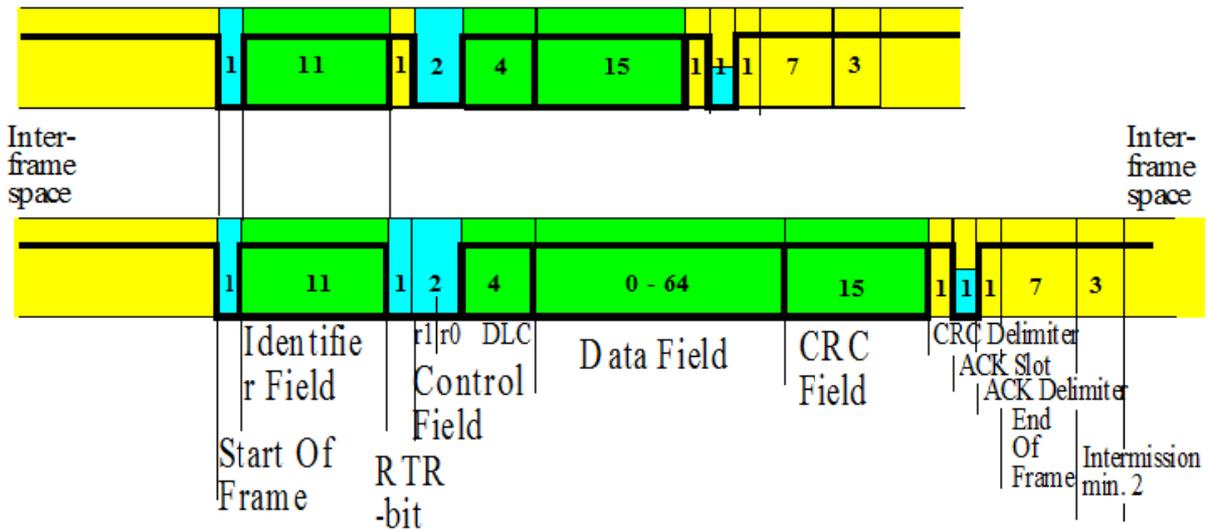
CAN Remote Frame Std



Bit values

- 0
- 0/1
- 1

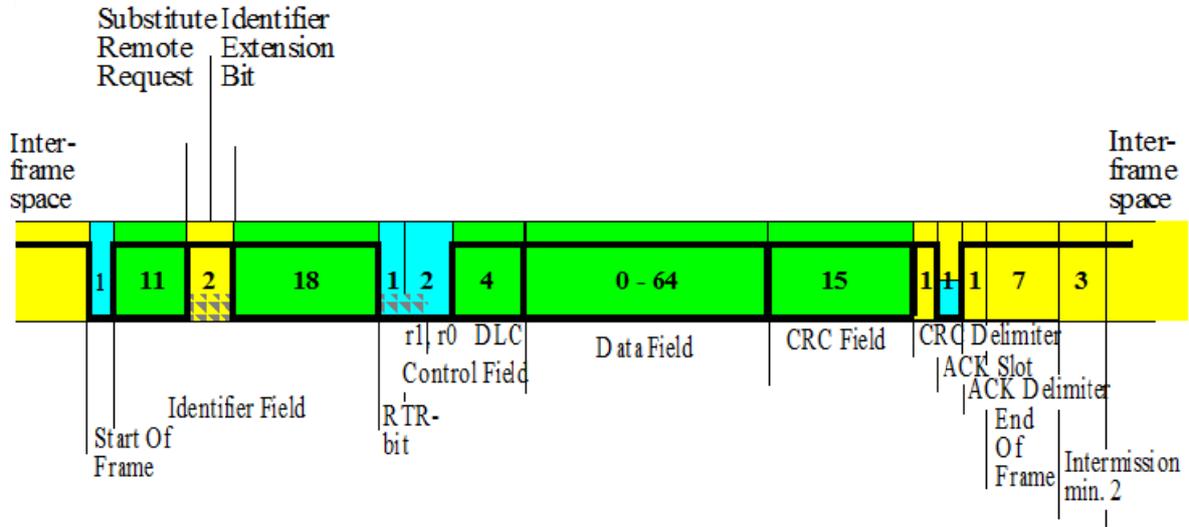
데이터 프레임과 비교한 Remote Std



주의: 모든 원격 요청(remote requests)들에 있는 DLC는 관련 데이터 메시지에 있는 DLC와 동일해야만 합니다!



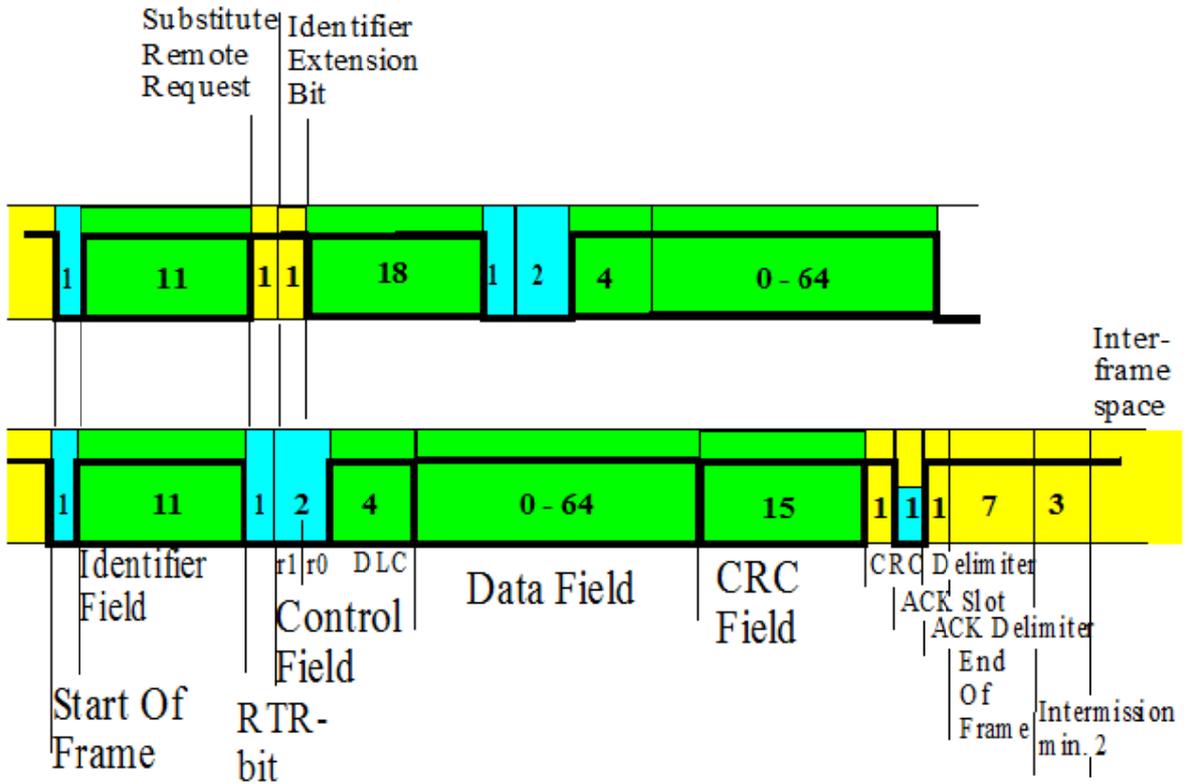
CAN Data Frame Ext.



Bit values

- 0
- 0/1
- 1

CAN Data Frame Ext. and Std



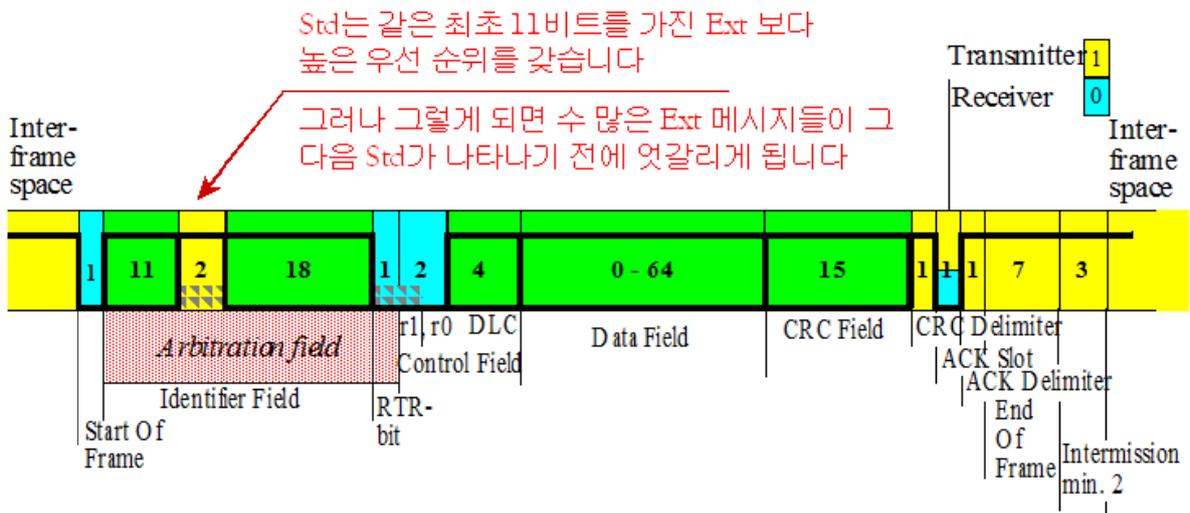
이제 녹색 노드는 버스상에 우성 비트가 있다는 것을 알 수 있으며 중재에서 집니다. 따라서 송신기는 bus-line 레벨을 판단하기 전에 파장이 가장 먼 노드에 도착하고 돌아올 때까지 (내부 지연 포함) 기다려야만 합니다.



최대 비트율은 파장 전파 지연에 따라 다릅니다

- 버스 길이
- Opto couplers
- 내부 지연
- 오실레이터 정확성 (때때로 명기되지 않음)

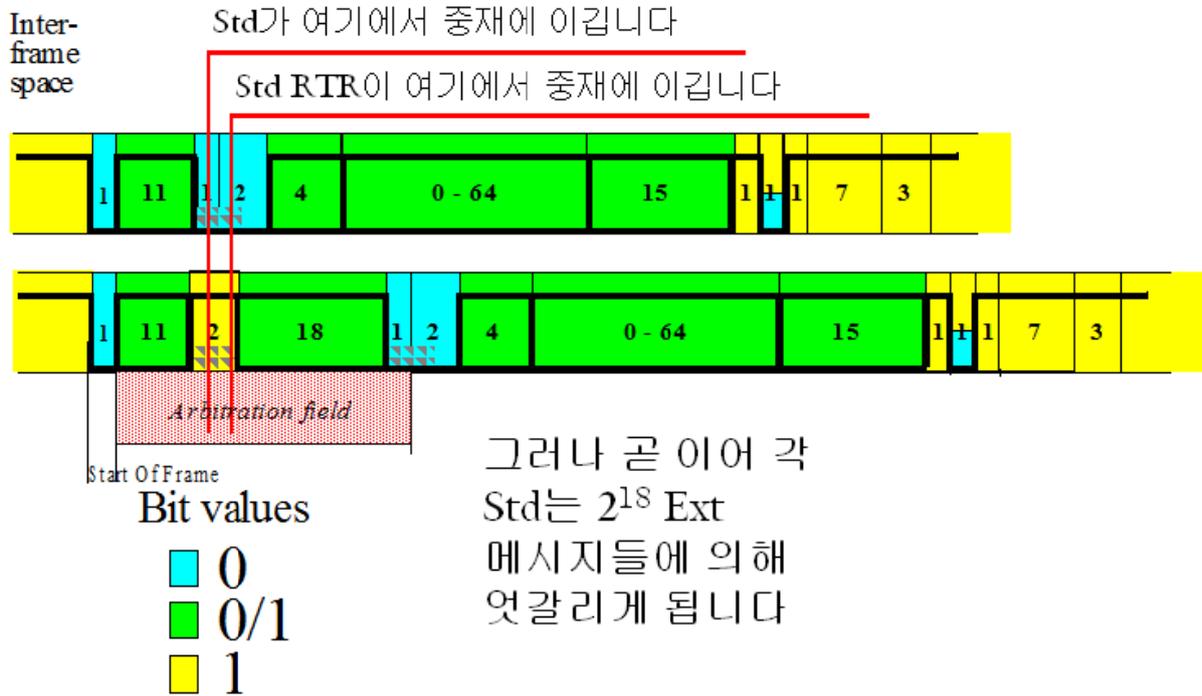
CAN Data Frame Ext.



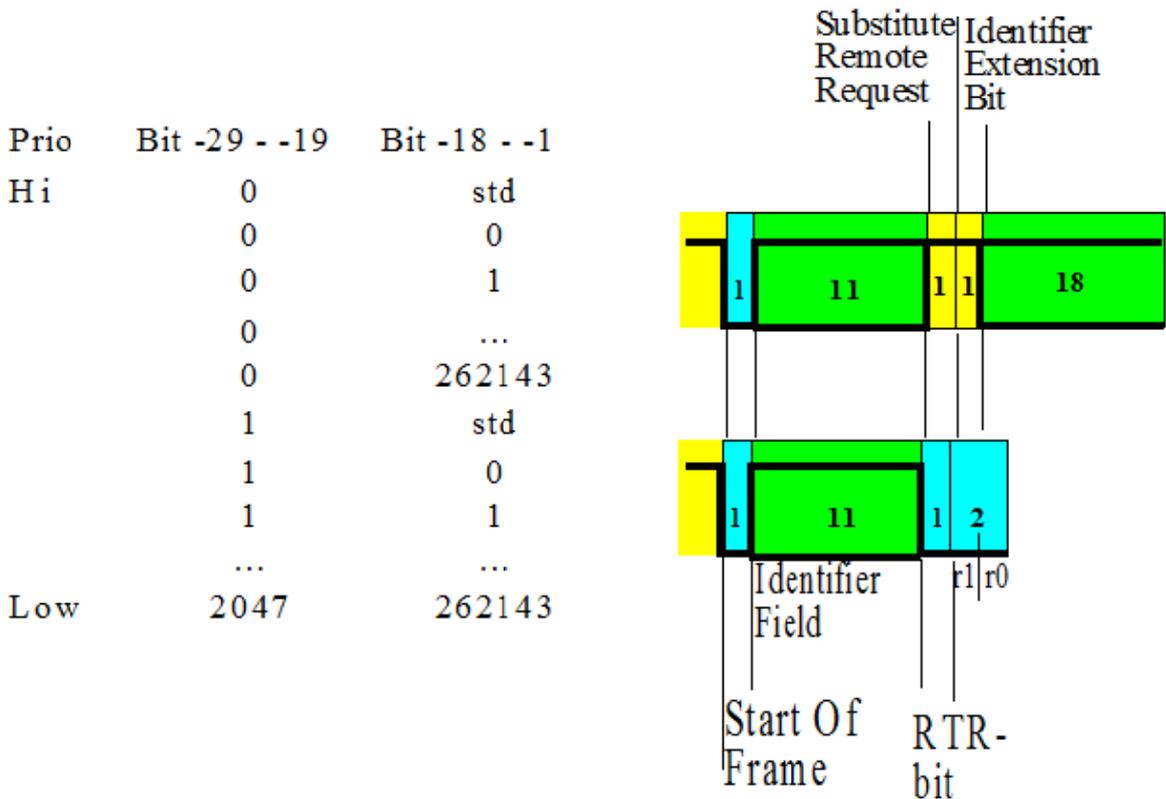
Bit values

- 0
- 0/1
- 1

처음 11비트가 동일할 때 Std 식별자가 Ext. 식별자를 이깁니다



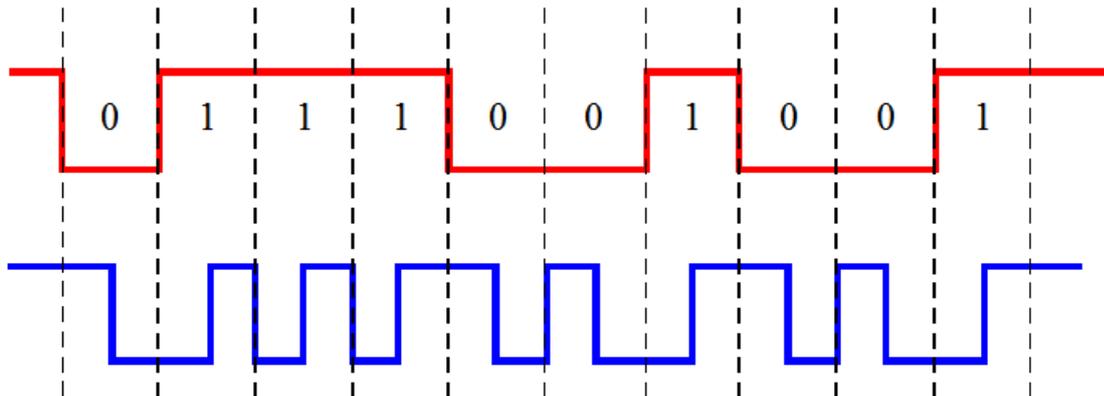
Std vs. Ext. priority



CAN 동기화

NRZ

Non Return to Zero



(Manchester Coding)

CAN은 맨체스터 코드(MC)와 비교해서 전자파적합성(EMC) 장점을 가진 NRZ입니다

BIT TIMING

한 개의 비트 타임은 4 부분들로 이루어집니다:

Synch_Seg, Prop_Seg, Phase_Seg1 and Phase_Seg2

이러한 부분들은 time quantas 의 수로 이루어집니다

Bit Time = Synch_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2

또는 흔히 다음과 같이 표현됩니다

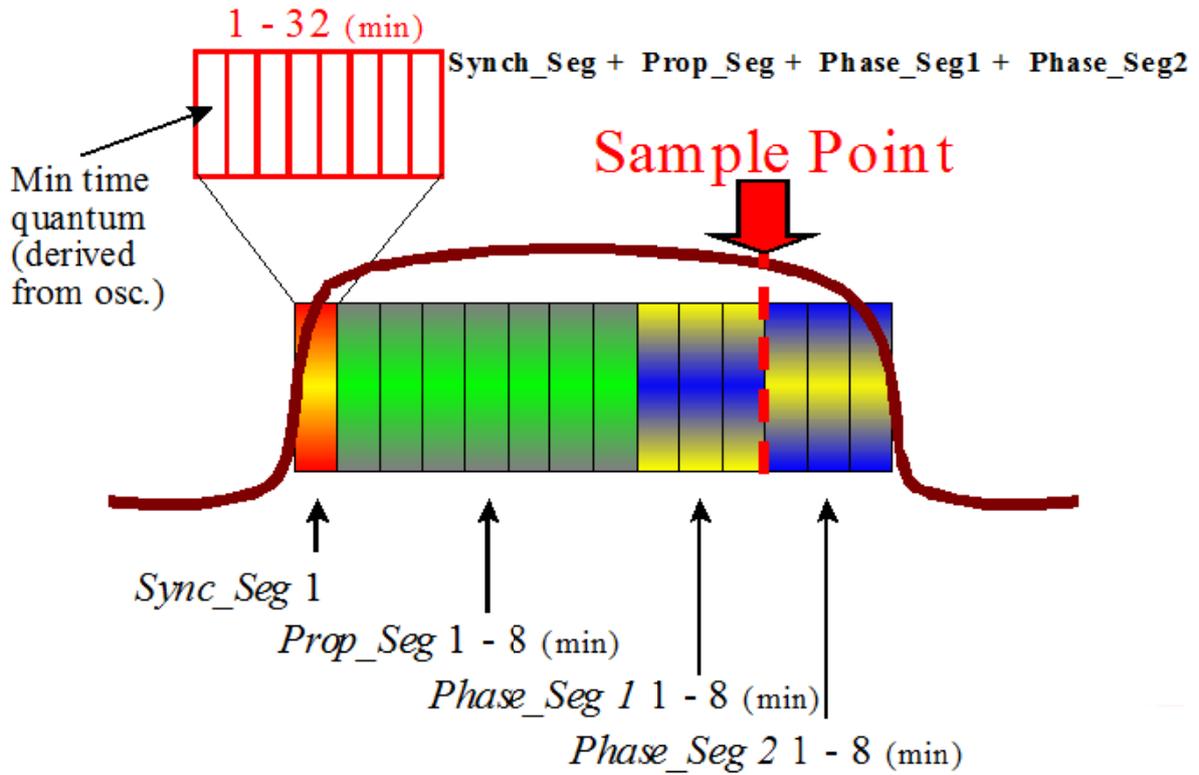
TBIT = TSYNC + TSEG1 + TSEG2

where

- TSYNC = 1
- TSEG1 = [2..16]
- TSEG2 = [1..8]
- TBIT = [4..25]



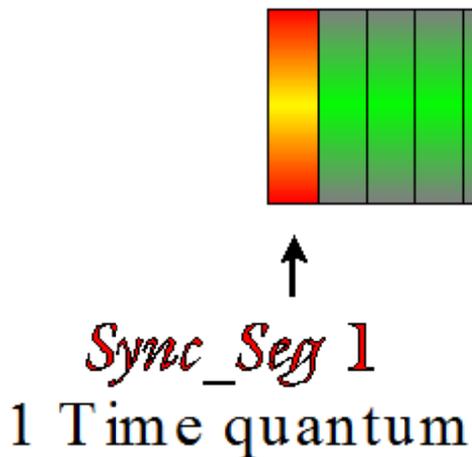
Bit Time =



Total 4 - 25 Time quanta (min)

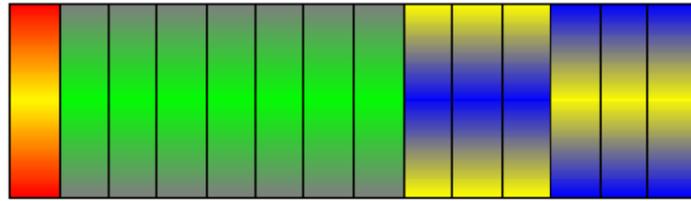
Sync_Seg

버스 상에서 다양한 CAN 노드들을 동기화합니다. 이 세그먼트 내에서 한 개의 edge가 예상됩니다.



Prop_Seg

버스 상에서의 물리적 지연 시간과 노드 인터페이스를 보정합니다.

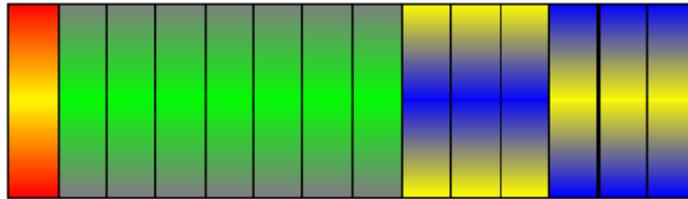


Prop_Seg

1 - 8 Time quanta (min)

Phase_Seg1 & 2

Phase 오류들을 보정합니다.



Phase_Seg 1

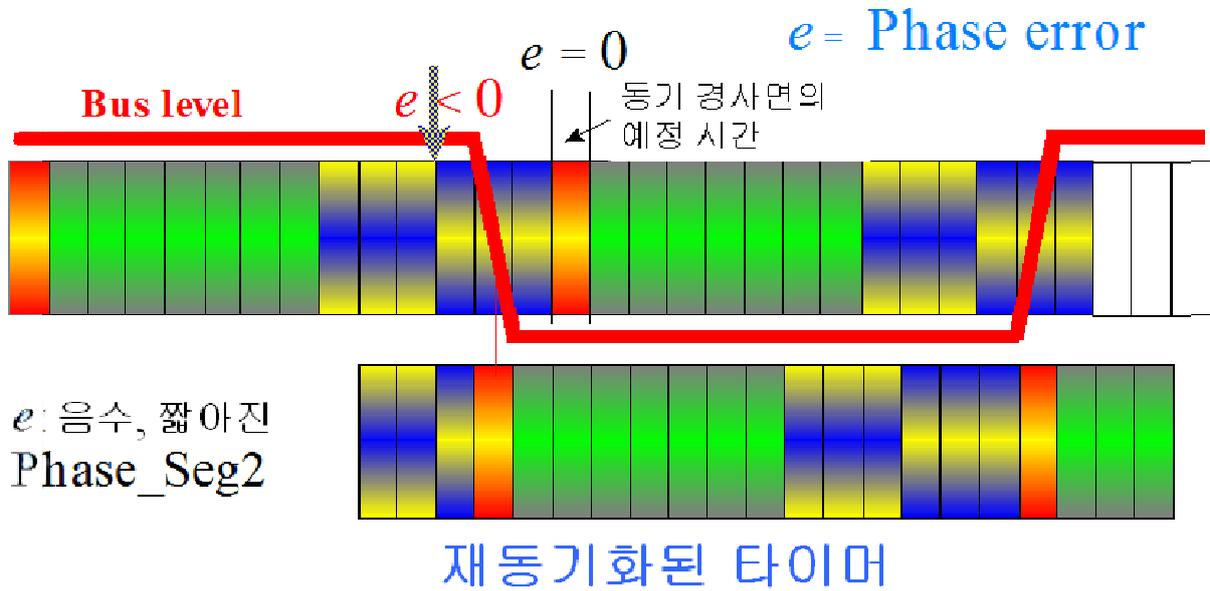
Phase_Seg 2

Phase_Seg1 1 - 8 Time quanta (min)

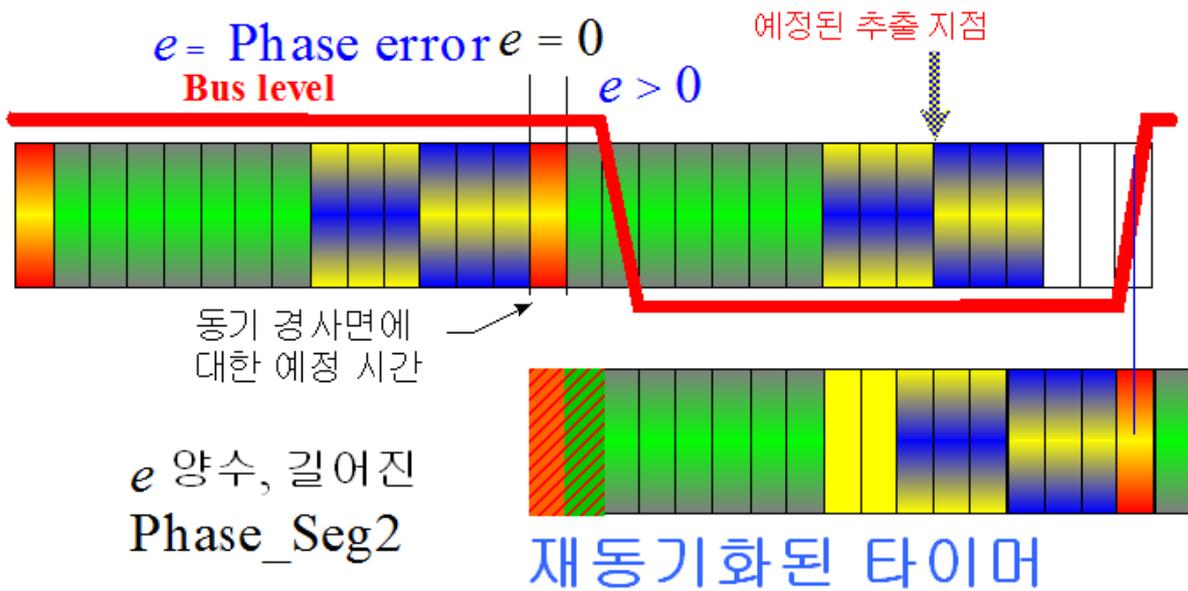
Phase_Seg2 = Phase_Seg1 (or information processing time)

재동기화

지점 추출 후 (너무 이른 시기에) 발견된 동기 경사면



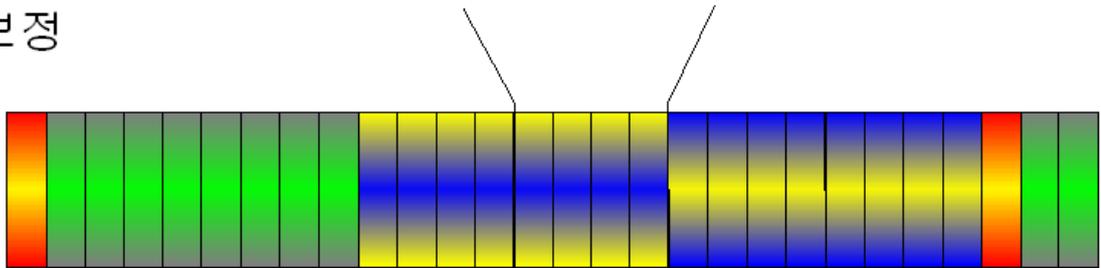
지점 추출 후 (예정보다 더 늦게) 발견된 동기 경사면



동기 점프 폭

한 단계에서
허용된 최대 동기
보정

Max 4(min),
Phase seg 2



SJW = 최대 위상 교정

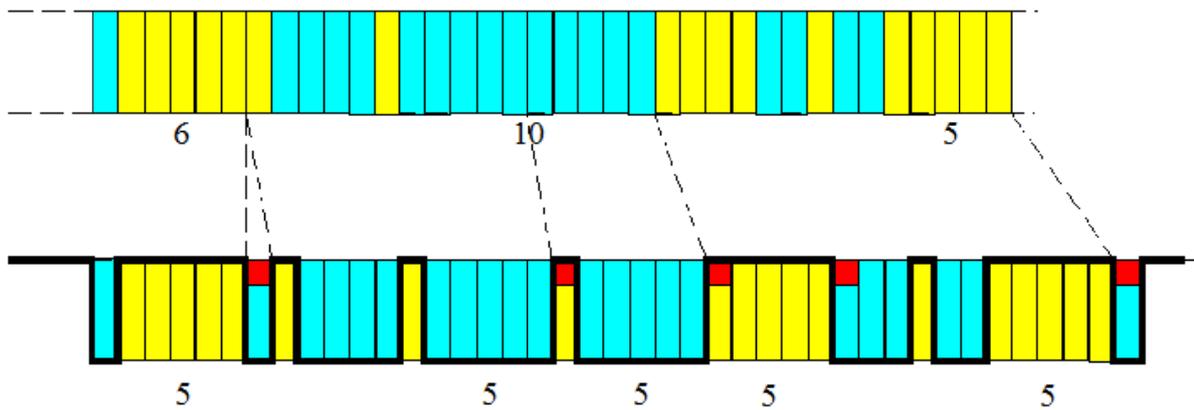


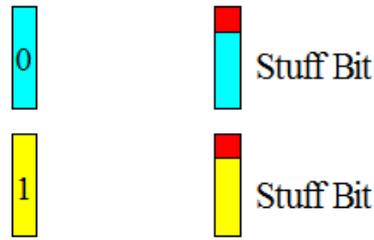
Bit Stuffing



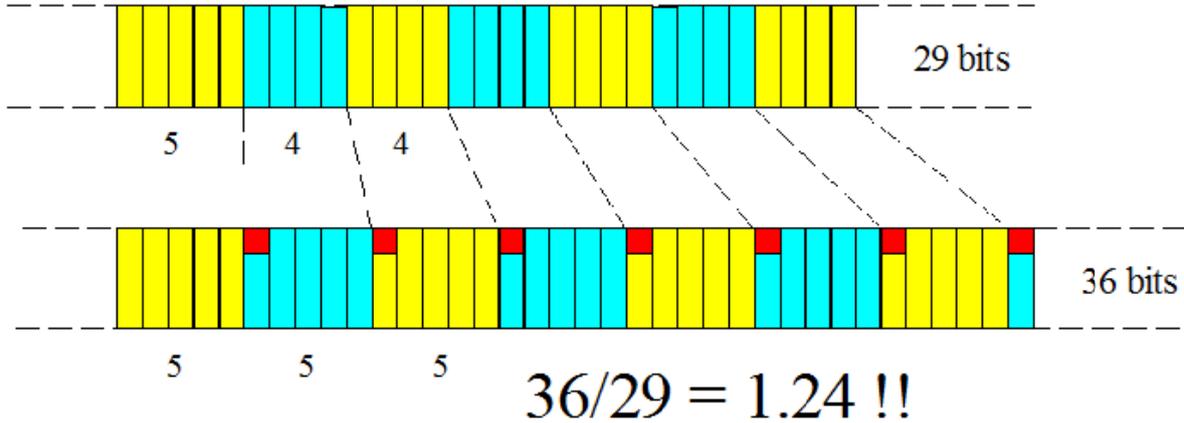
Stuff Bit
Stuff Bit

Sam 극성의 연속적인 다섯
개의 비트들은 한 개의 stuff
bit를 가져옵니다

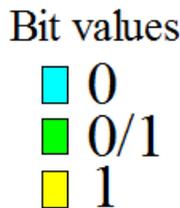
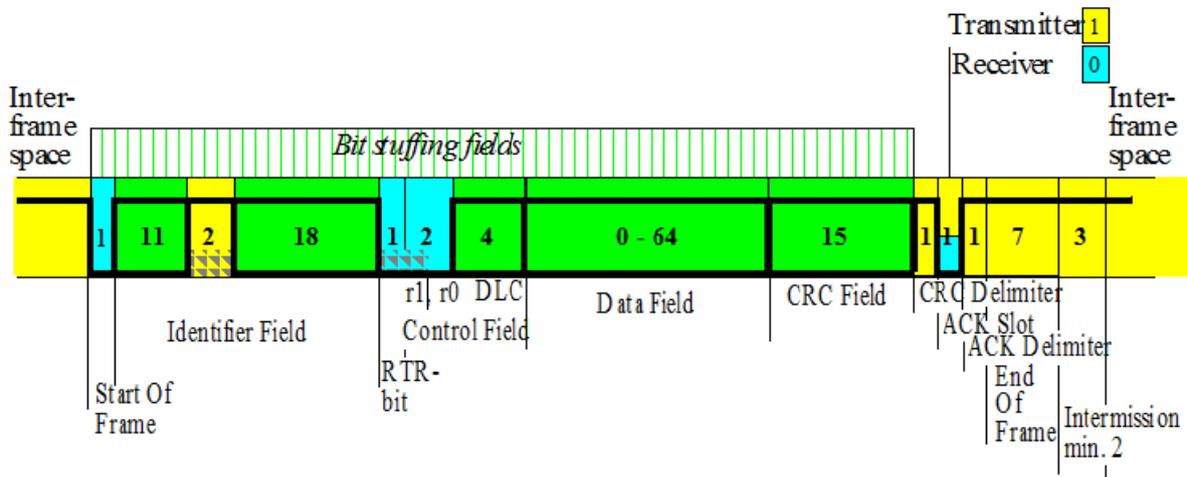




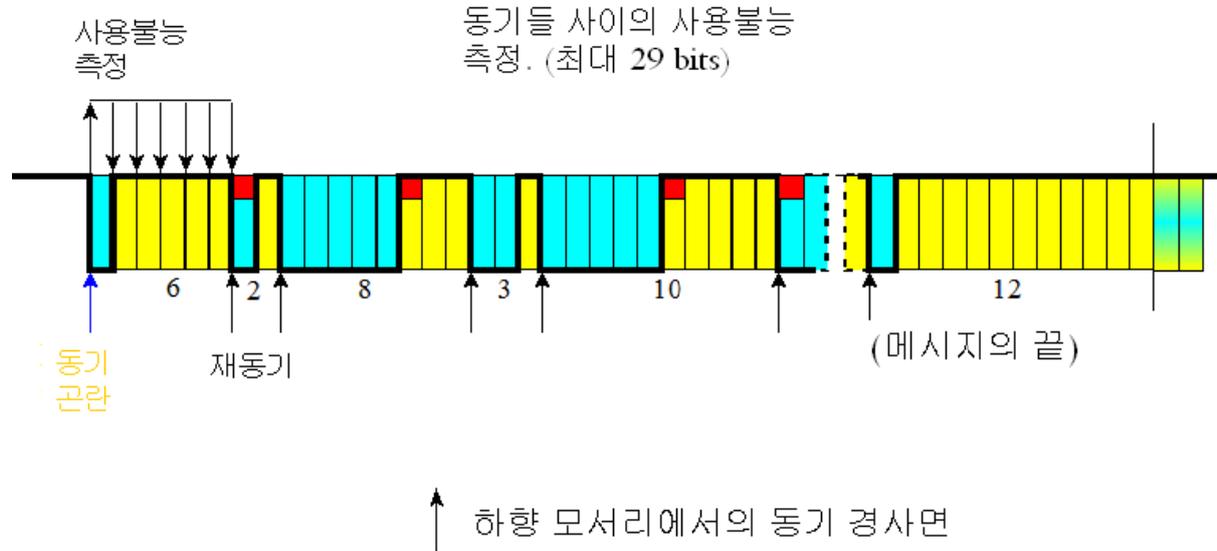
동일한 극성의 연속적인 다섯 개의 비트들은 한 개의 stuff bit를 제공합니다



CAN Data Frame Ext.



Synchronisation



오실레이터 허용 한계 범위

$$(1 - df) \cdot f_{nom} < f_{osc} < (1 + df) \cdot f_{nom}$$

Conditions:

- $df = \min(\text{Phase_Seg1}, \text{Phase_Seg2}) / (2(13 \cdot \text{TBIT} - \text{Phase_Seg2}))$
- $df = \text{SJW} / (20 \cdot \text{TBIT})$
- Max diff. between two osc. is $2df \cdot f_{nom}$

연산

- $T_{scl} = T_{clk} \cdot \text{BRP} \cdot 2 = T_{clk} \cdot (\text{BRP} + 1) \cdot 2$
 - BRP the value in CAN-controller
 - (clk = 16 MHz and BRP = 0: Tclk = 62.5ns and Tsc1 = 125 ns)
- $T_{seg1} = T_{scl} \cdot (\text{TSEG1}) = T_{scl} \cdot (\text{TSEG1} + 1)$
 - TSEG1 the value in CAN-controller
- $T_{seg2} = T_{scl} \cdot (\text{TSEG2}) = T_{scl} \cdot (\text{TSEG2} + 1)$
 - TSEG2 the value in CAN-controller
- $T_{sjw} = T_{scl} \cdot (\text{SJW}) = T_{scl} \cdot (\text{SJW} + 1)$
 - SJW the value in CAN-controller
- $\text{SJW} = [1..4]$

규칙

- $T_{\text{Prop_Seg}} > (\text{All delays}) \cdot 2$
- $T_{\text{Seg2}} \geq 1 T_{scl}$, CAN 컨트롤러가 최소 $2T_{scl}$ 을 요청할 수 있습니다.
- $T_{\text{Seg2}} \geq T_{sjw}$
- $T_{\text{Seg1}} \geq T_{sjw} + T_{\text{Prop}}$

제안 : Phase segment T_{sjw+1} 을 유지