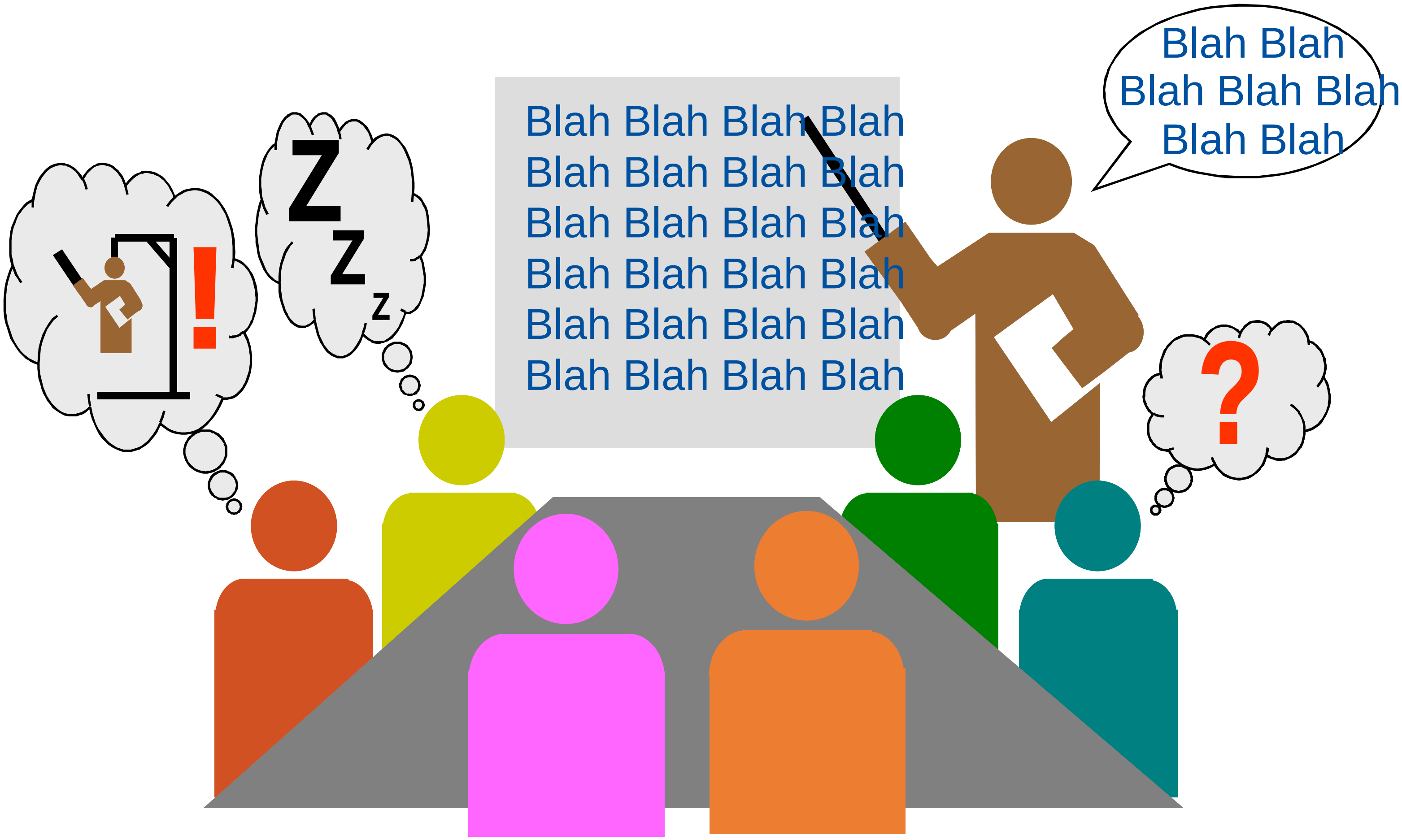
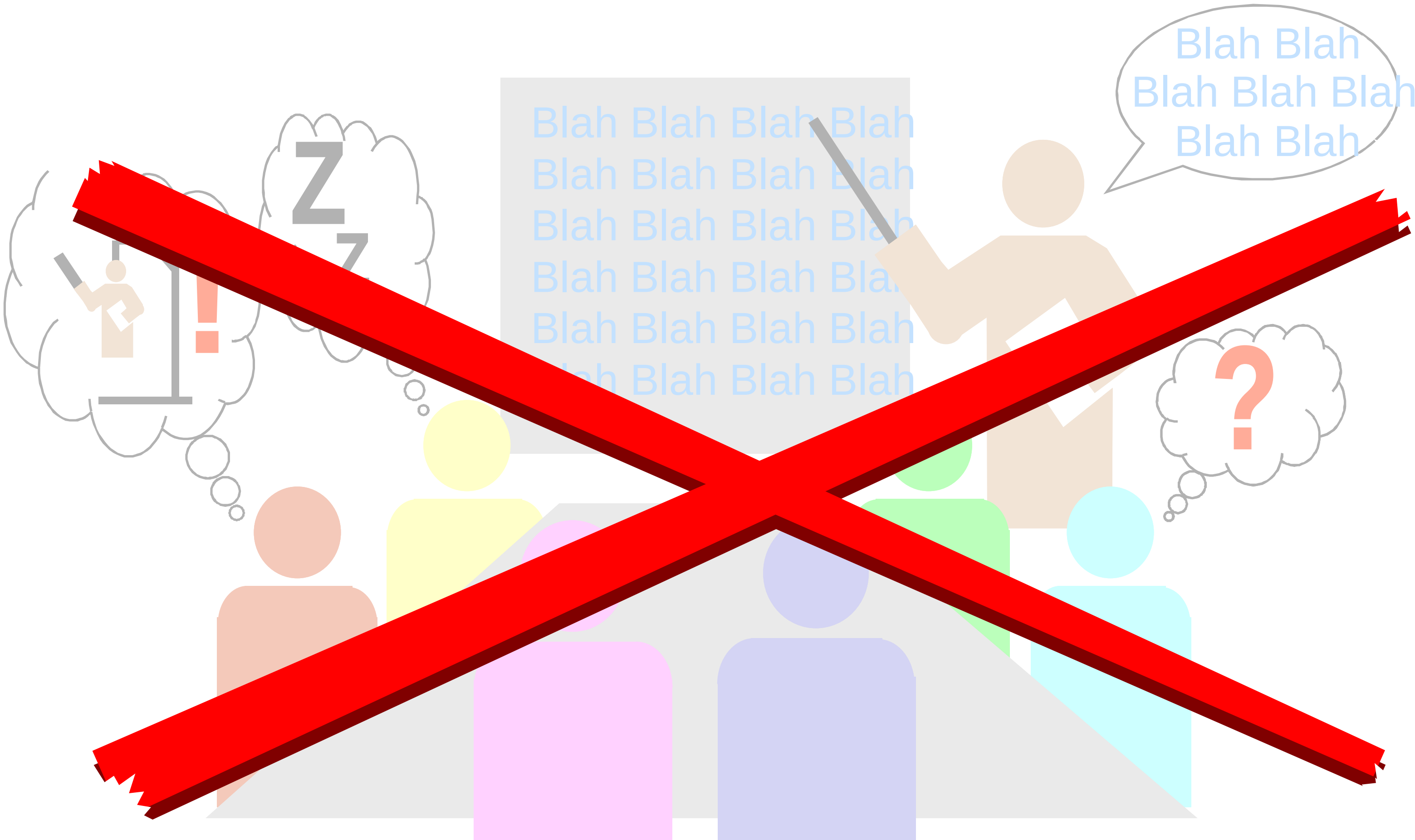
CAN

What should NOT happen during this training



What should NOT happen during this training



Any questions?

Questions?

Please ask!

Overview

CAN Controller Area Network Training

- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

CAN History Timeline

CAN

Time

1999 — **150.000.000 CAN controllers sold***
More than 80 different controllers available from 19 manufacturers

1993 — **ISO 11898 standardization**
World-wide standardization leads to the widespread use of CAN in mass-market applications, e.g. automobiles

1992 — **CAN in Automation (CiA) founded**
International CAN manufacturers and users organization
http://can-cia.de

1988 — **First CAN Controller on the market: Intel 82C526**

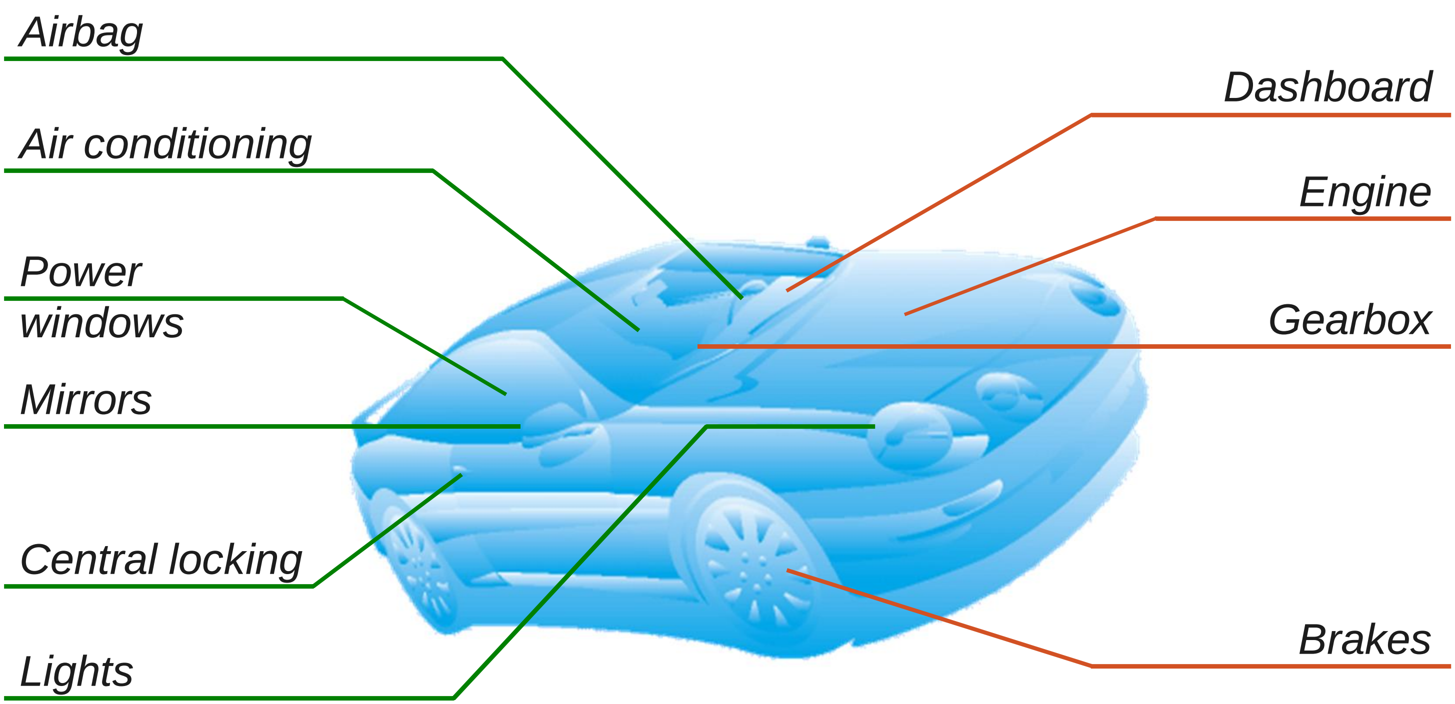
1985 — **First specification of the CAN Protocol**

1983 — **BOSCH starts developing the CAN Protocol**
Intended use: Power train applications

*Source: CAN in Automation (CiA) e.V.

Application examples for the CAN Bus

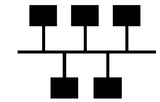
Typical examples for CAN in automobiles



Car Body components
Typically low-speed CAN Bus

Power Train components
Typically high-speed CAN Bus

Bus characteristics

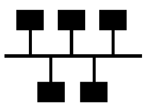


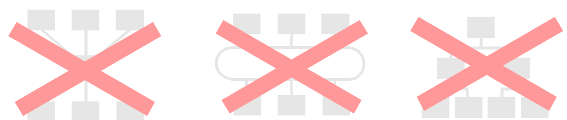
- **Serial data communications bus**

  Inexpensive and simple, but slower than parallel bus.

- **Linear bus structure**

  No star structure, no ring structure, no tree structure !





- **Sensor / actor bus**

  Example: rain sensor = sensor, windscreen wipers = actor,
  both connected via the same bus.

Data rates

- “Good” real-time capabilities

  Small latency (“fast enough”)
  ➔ indispensable for automotive applications.

- Data rate dependent on bus length

Data rate: 1 Mbit/sec	➔	Bus length: 40 meters,
Data rate: 125 kBit/sec	➔	Bus length: 500 meters,
Data rate: 50 kBit/sec	➔	Bus length: 1000 meters.

Typical definitions:

Low-speed:	25 kBit/sec	up to	125 kBit/sec.
High-speed:	500 kBit/sec	up to	1 Mbit/sec.

Transmission principles

- ## Multicast / broadcast philosophy

  CAN messages do not include references to sender or receivers, but to information contents.

- ## Bus access principle: CSMA/CA

  <u>C</u>arrier <u>S</u>ense <u>M</u>ultiple <u>A</u>ccess with <u>C</u>ollision <u>A</u>voidance

  Carrier Sense: Every node monitors the bus level, all the time.

  Multiple Access: Every node can start a transmission any time when the bus is free.

  Collision Avoidance: When several nodes start transmission at the same time, <u>all but one</u> withdraw from sending.

Hardware characteristics

- Hardware message acceptance filtering

  Only leaves messages through which are of interest for the node
  ➔ reduces CPU load.

- Sophisticated hardware error management

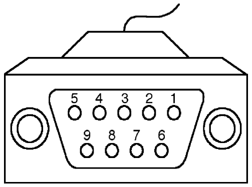
  Combination of different error prevention and detection methods, automatic re-transmission of messages detected as erroneous
  ➔ very high transmission security among field bus systems.

- Various transmission media

  Twisted-pair (dual-wire) cable, single-wire cable or optical fiber.

- Standardized connector

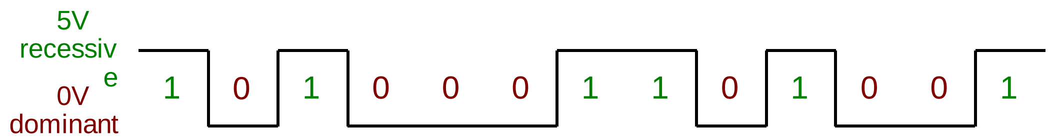
  Recommended: 9-pin D-sub connectors (DIN 41652).



Signal coding

➢ NRZ (non-return-to-zero) coding

Example: Voltage levels: 0V (dominant), 5V (recessive)



Characteristics of NRZ coding:
Voltage level stays the same for consecutive bits of same polarity.

Note:
Different voltage levels are defined for different purposes.

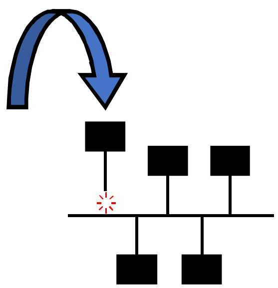
Bus stations (Nodes)

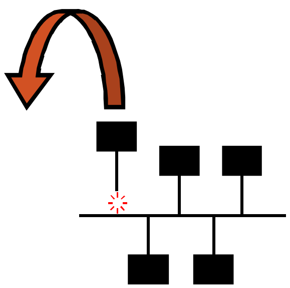
- Typically 3 to 40 nodes per bus

  No limit for node number defined in CAN specification.
  Number of nodes depends on capabilities of CAN transceivers.
  Bus load usually gets higher with more nodes.

- Hot plug-in / plug-out

  Connect / disconnect nodes while the bus is up and running.





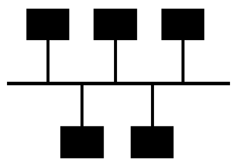
Advantages of the CAN Bus

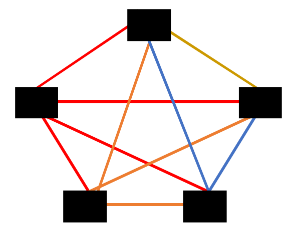
Advantages of

the CAN Bus

over

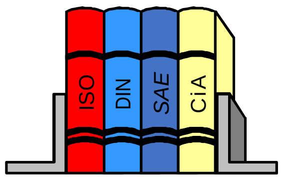
conventional cabling





- Less wires
- Less connections
- Less weight
- Less EMI problems
- Less space requirements
- Easier addition of new nodes
- Lower assembly costs
- Increased transmission reliability

International Standards



"The great thing about standards is that there are so many to choose from."

- International CAN standards

ISO 11898	"Road vehicles: CAN for high-speed communicatio
ISO 11519	"Road vehicles: Low-speed serial data communicat Low-Speed CAN / VAN"
ISO 11992	"Road vehicles: Electrical connections between towing and towed vehicles"
EIA RS-485	"Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems" (formerly used for CAN Physical Layer)

Overview

CAN Controller Area Network Training

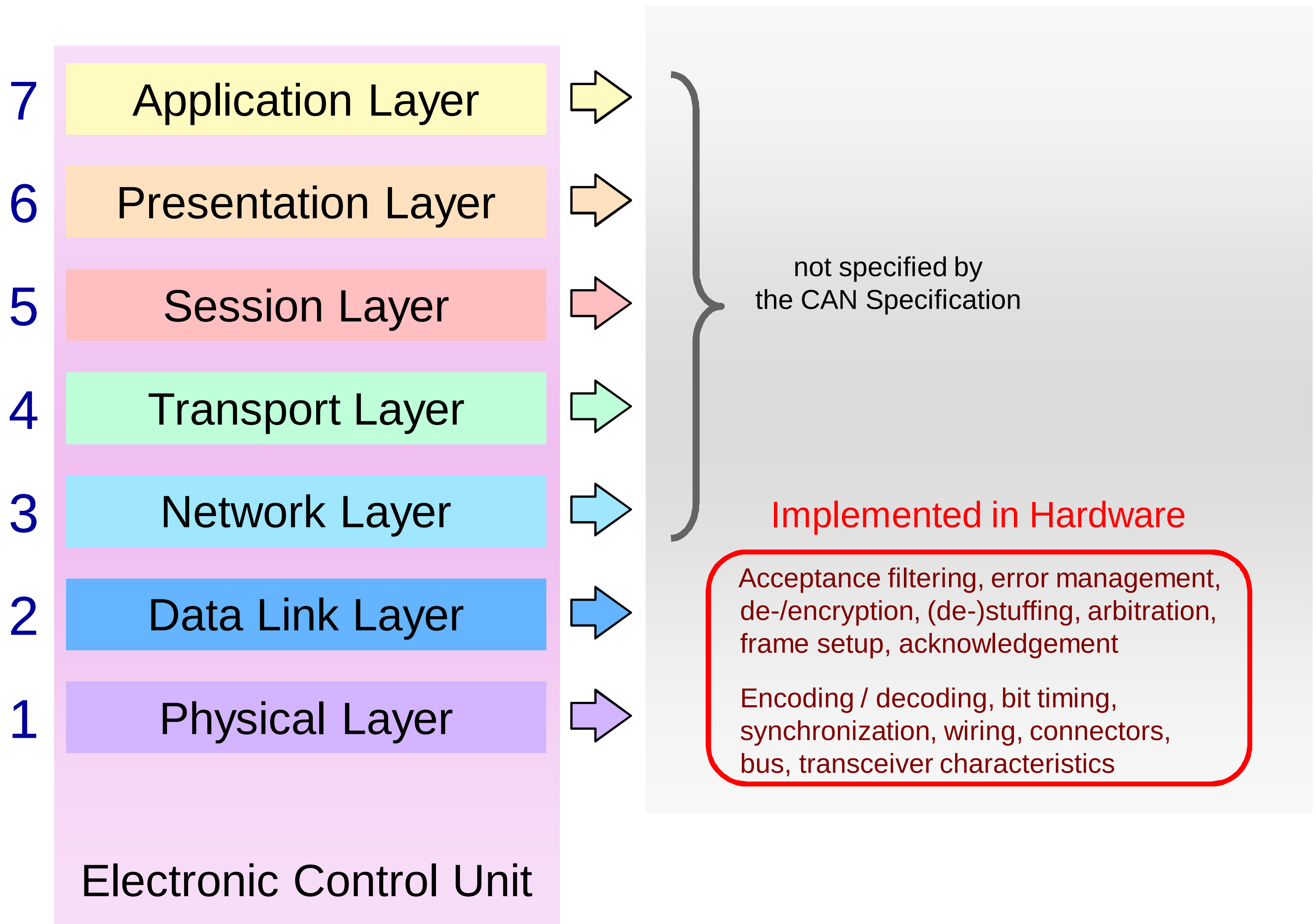
- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

The ISO/OSI seven-layer model

7	Application Layer	Applications, operating system
6	Presentation Layer	Conversion of data formats
5	Session Layer	Task synchronization, buffers, connection setup and monitoring, access rights control
4	Transport Layer	Address conversion, routing, segmentation
3	Network Layer	Setup of logical connection, transport protocol
2	Data Link Layer	Transmission security, frame setup, error management
1	Physical Layer	Electrical / mechanical characteristics: Transmission medium, wiring, connectors, encoding, signals

Electronic Control Unit

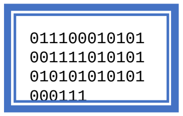
CAN within the ISO/OSI seven-layer model



CAN Controller Area Network Training

- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

Frames: Overview



- **Frame: “Envelope” for transmission data**

  Exact frame format is defined in CAN specification

- **Note: CAN Frame ≠ CAN Message !!!**

  A CAN *message* can be spread out over several CAN *frames*

- **Four different frame types:**

Data Frame:	Transmission of regular data
Remote Frame:	Remote request for data transmission
Overload Frame:	Indication of bus overload situations
Error Frame:	Indication of transmission errors

Overview

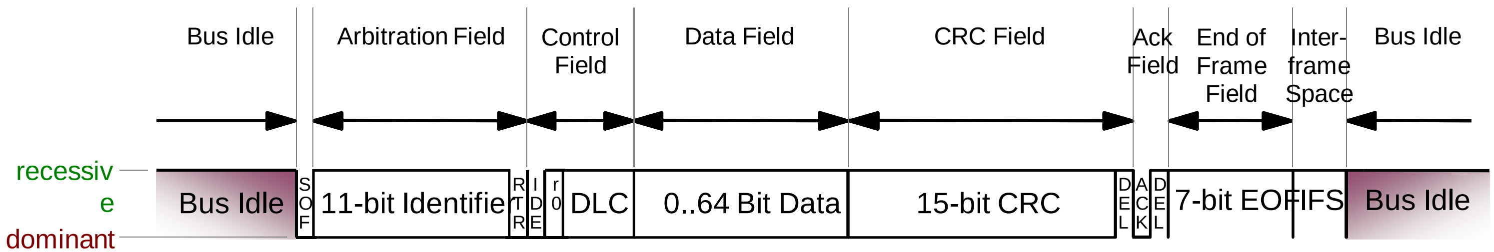
CAN Controller Area Network Training

- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

Data Frame: Formats

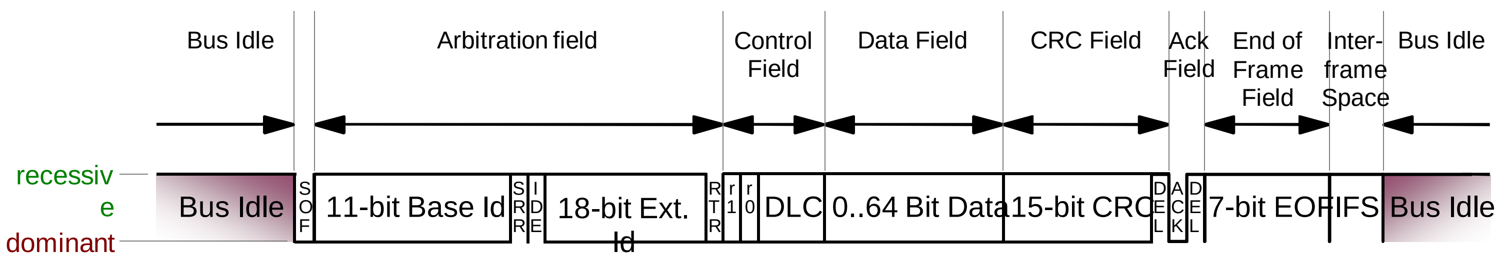
Standard Format (CAN 2.0A): 11-bit Identifier

2^{11} = 2048 identifiers possible

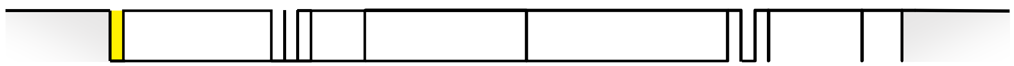


Extended Format (CAN 2.0B): 29-bit Identifier

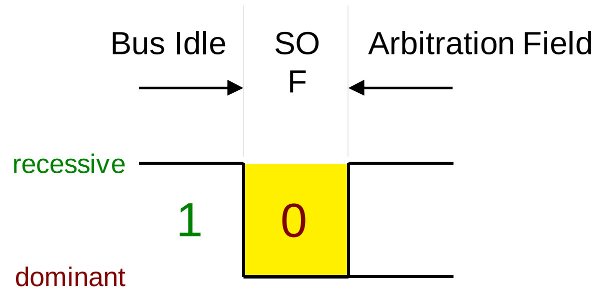
2^{29} = 536.870.912 identifiers possible



Data Frame: Start Of Frame (SOF) bit

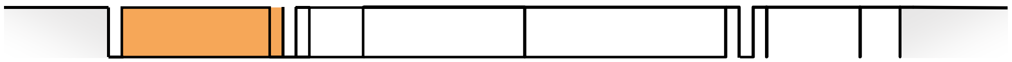


Start Of Frame (SOF) bit

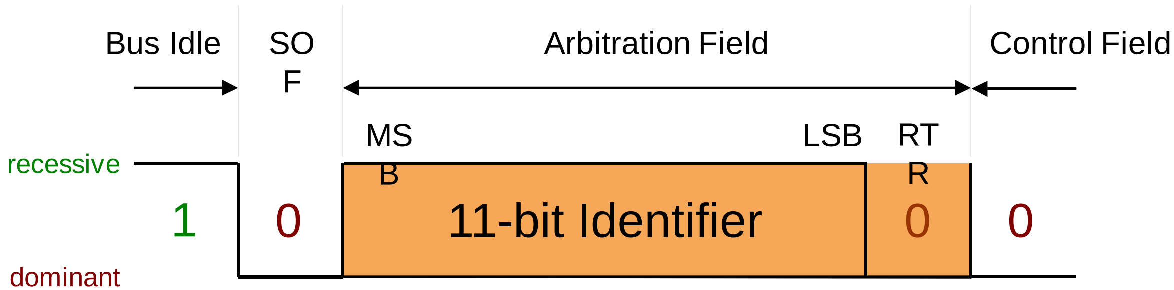


- marks the start of any CAN frame
- is always a dominant bit
- provides a falling edge for hard synchronization of transmitter and receivers

Data Frame: Arbitration Field



Arbitration Field



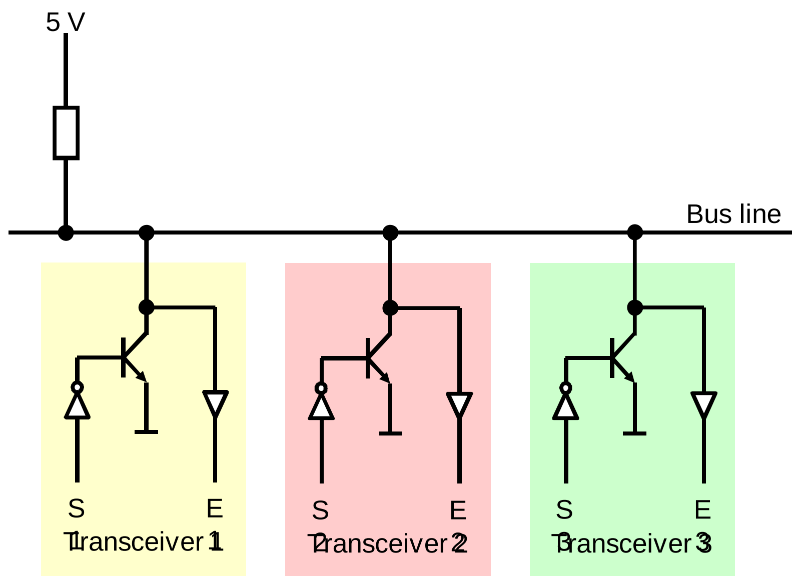
- contains the Identifier (11 bit for CAN 2.0A) which is used for arbitration
- Identifier determines frame priority: low identifier = high priority
- the highest seven bits of the identifier must not be all recessive
- Remote Transmission Request (RTR) bit is always dominant in a Data Frame

Arbitration

- Arbitration = the allocation of bus access rights
- Arbitration occurs when several nodes start transmission on the bus at the same time
- Arbitration procedure:
  1. All controllers monitor the bus while transmitting simultaneously
  2. A dominant bit (“0”) pulls the bus voltage level to zero
  3. When a controller transmits “1”, but observes “0” on the bus, it has lost arbitration
  4. Controllers who lost arbitration retreat immediately and retry later
  5. Arbitration is won by frame with lowest identifier = highest priority

Recessive and dominant bus levels

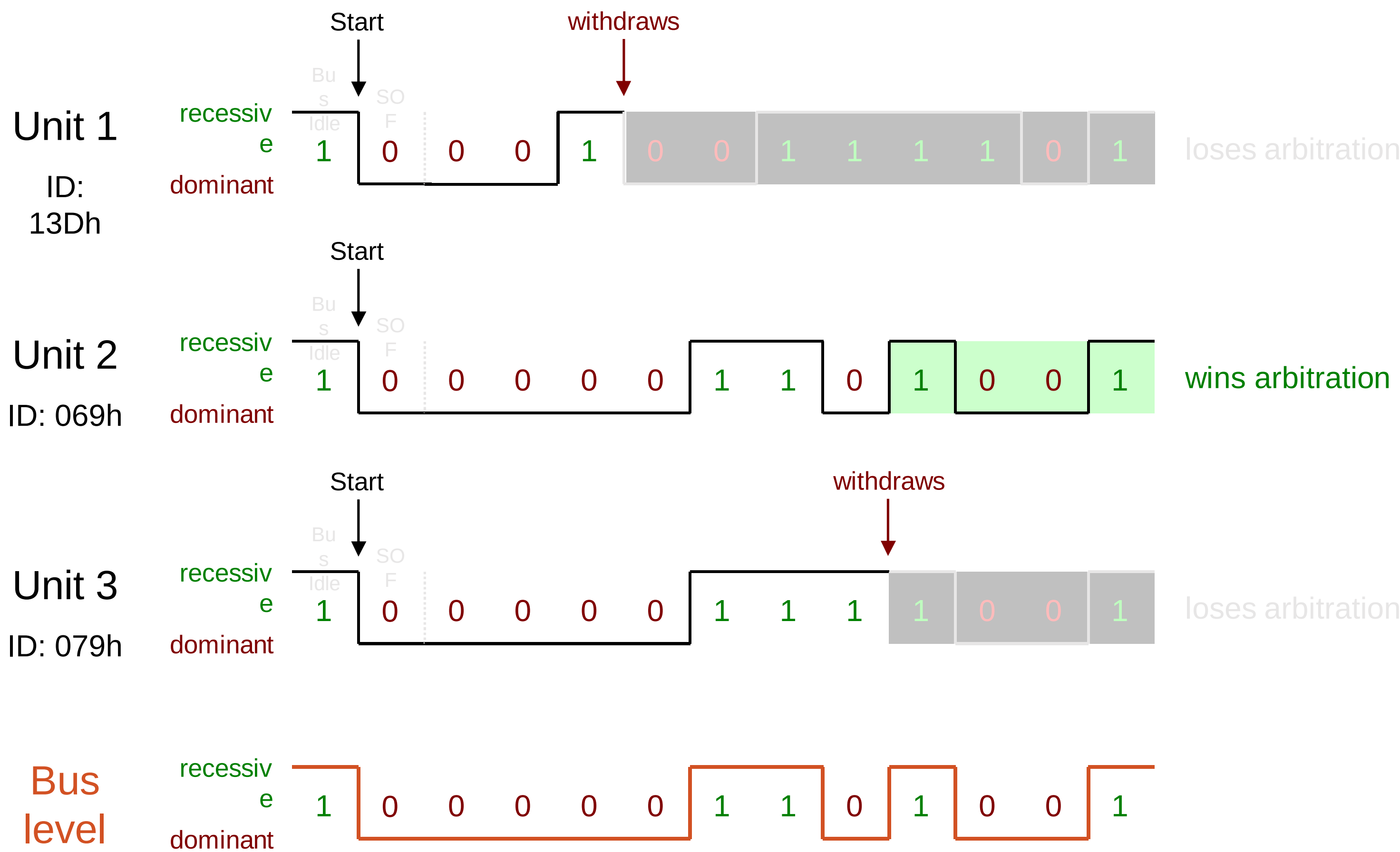
Hardware representation of recessive and dominant bus levels



Wired-AND / Open-Collector circuit

S1	0	1	0	1	0	1	0	1
S2	0	0	1	1	0	0	1	1
S3	0	0	0	0	1	1	1	1
Bus	0	0	0	0	0	0	0	1

Arbitration: Example



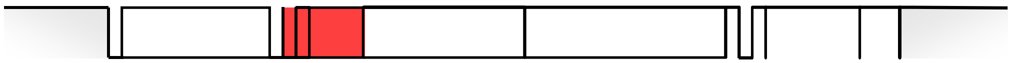
Arbitration: Identifier ↔ Priority

!

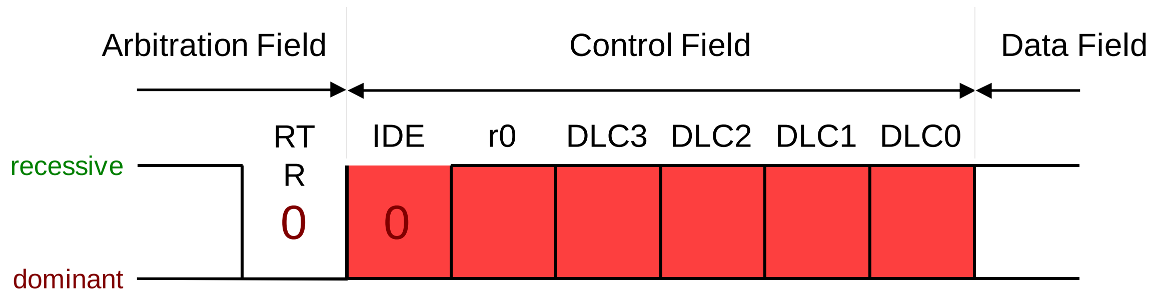
Consequence:

Frames
carrying information
of high priority
should have
a low identifier

Data Frame: Control Field

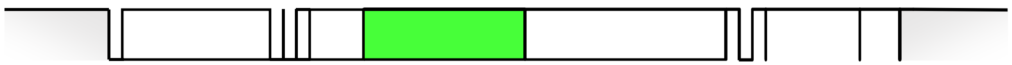


Control Field

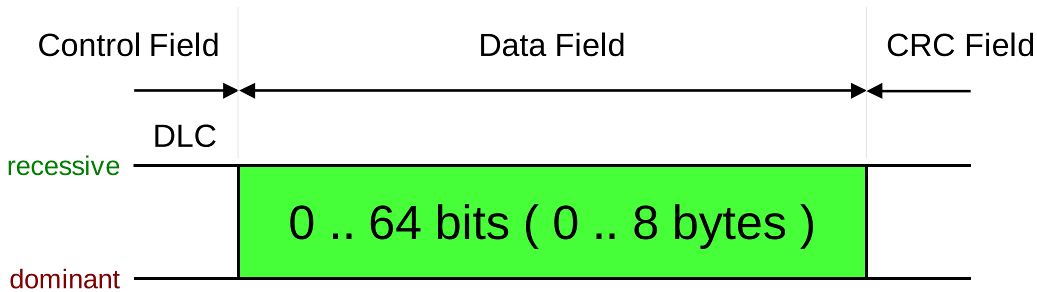


- Identifier Extension (IDE) bit is dominant for Standard Frames and recessive for Extended Frames
- r0 bit is not used (“reserved for future extensions”)
- Data Length Code (DLC, 4 bits) indicates number of data bytes in Data Field; may take values ranging from 0 to 8, other values are not allowed

Data Frame: Data Field

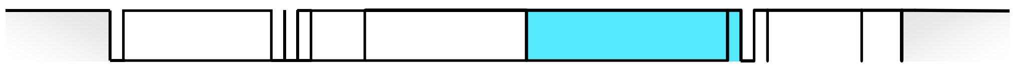


Data Field

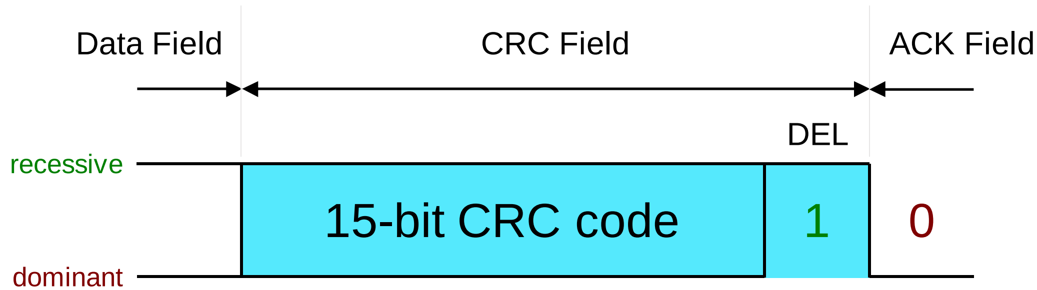


- contains the actual information which is transmitted
- number of data bytes may range from 0 to 8 <u>in units of bytes</u>
- number of data bytes is given in the Data Length Code (DLC)
- transmission starts with the first data byte (byte 0), MSB first

Data Frame: CRC Field

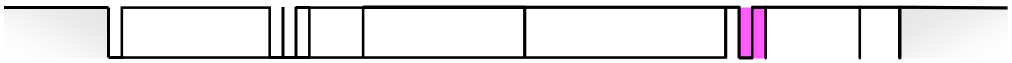


CRC Field

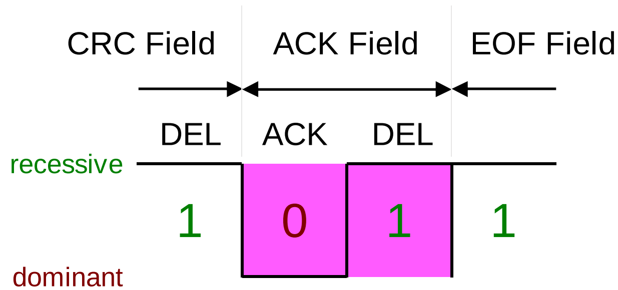


- contains the 15-bit Cyclic Redundancy Check (CRC) code
- CRC is a complex, but fast and effective error detection method
- the CRC Field Delimiter (DEL) marks the end of the CRC field
- the CRC Field Delimiter is always recessive

Data Frame: Acknowledge (ACK) Field



Acknowledge (ACK) Field



- contains the Acknowledgement (ACK) bit
- the Acknowledgement bit can be dominant or recessive
- the ACK Field Delimiter (DEL) marks the end of the ACK field
- the ACK Field Delimiter is always recessive

ACK Bit: Values and interpretation

Acknowledgement procedure:

1. Sender transmits a recessive bit in the ACK bit slot
2. Every receiver which received the frame without an error transmits simultaneously a dominant bit in the ACK bit slot

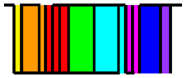
A dominant ACK bit

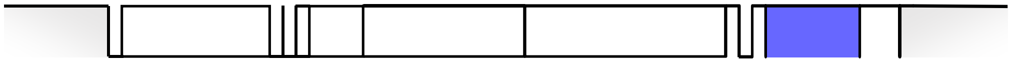
1. means that at least one node received the frame without an error, but
2. does not necessarily mean that the frame was error-free

A recessive ACK bit

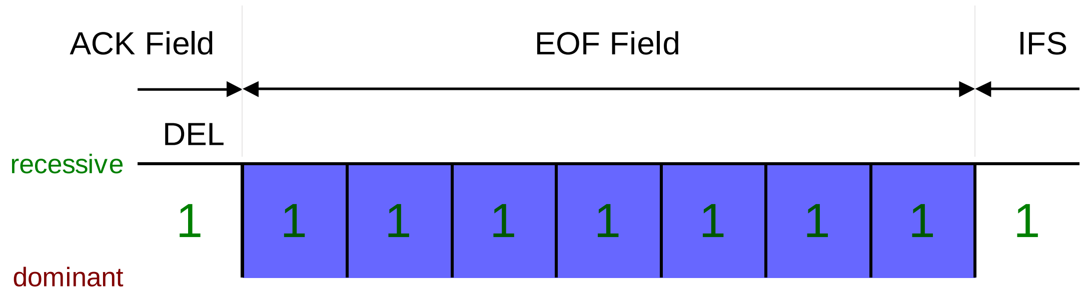
1. could mean that no node received the frame without an error or
2. that no other node is connected to the bus

Data Frame: End Of Frame (EOF) Field



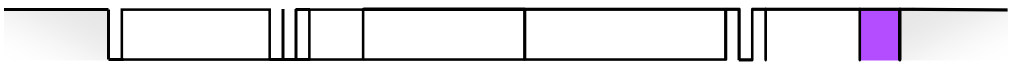


End Of Frame (EOF) Field

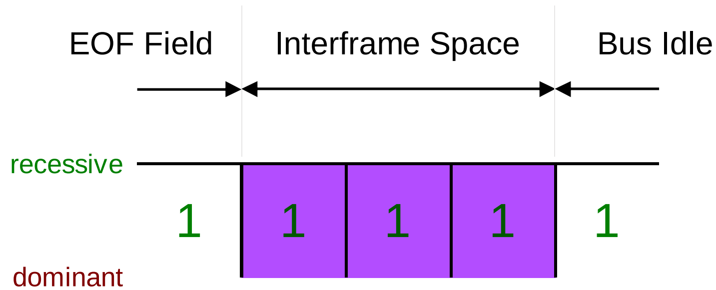


- consists of seven consecutive recessive bits
- marks the end of the Data Frame

Data Frame: Interframe Space (IFS)



Interframe Space (IFS)



- consists of <u>at least</u> three consecutive recessive bits
- no transmission is allowed during the Interframe Space (IFS)
- is needed by controllers to copy received frames from their Rx buffers
- ACK Field Delimiter + EOF + IFS = 11 consecutive recessive bits

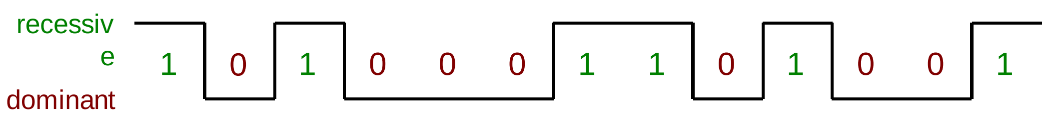
Overview

CAN Controller Area Network Training

- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

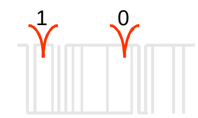
Bit stuffing: Motivation

Problems when using NRZ coding



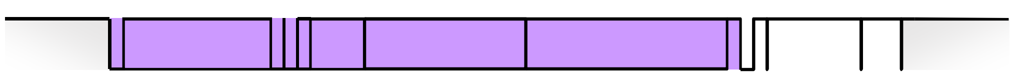
- long sequences of bits of the same polarity
- no changes in voltage level for a longer time
- no falling edges for synchronization
- synchronization between sender and receiver may be lost

Bit stuffing: Approach

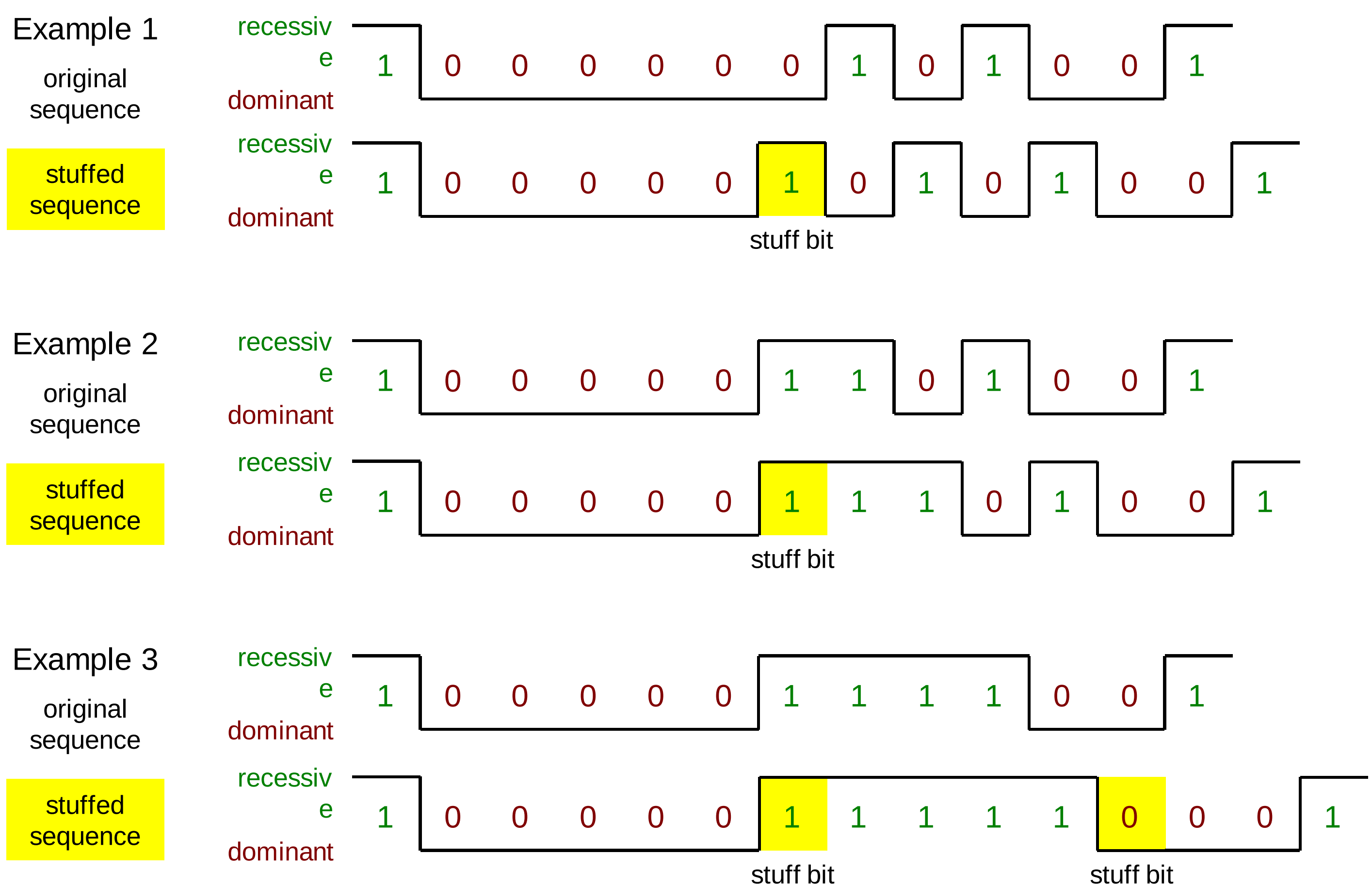


Solution: Bit stuffing

- after five consecutive bits of same polarity, insert one bit of reverse polarity
- CRC code is calculated before bit stuffing is done
- bit stuffing is done by the sender directly before transmission
- de-stuffing is done by the receiver directly after reception
- CRC code check is done after de-stuffing the frame
- bit stuffing is applied to part of the frame from SOF to CRC field



Bit stuffing: Examples



Data Frame: Maximum size

Maximum frame size (for 8 Data Bytes)

	Standard Frame CAN 2.0A (11-bit identifier)	Extended Frame CAN 2.0B (29-bit identifier)
without stuff bits	111 bits	131 bits
with stuff bits	130 bits	154 bits

Overview

CAN Controller Area Network Training

- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

Error types

Types of possible errors

- Bit errors
- Bit stuffing errors
- CRC errors
- Message format errors
- Acknowledgment errors

Error types: Bit errors

Error description

- The bit actually appearing on the bus is different from the one transmitted

Method of detection

- Sending unit constantly monitors the bus while transmitting

Possible cause

- Sending unit hardware is defective

Exceptions

- Arbitration phase (unit loses arbitration)
- Acknowledgement bit (unit gets positive ACK from at least one receiver)
- Sending of a Passive Error Frame

Error types: Bit stuffing errors

Error description

- Data Frame contains six or more consecutive bits of the same polarity

Method of detection

- Receiver detects error when de-stuffing a received frame

Possible causes

- Error of sending unit during bit stuffing and/or transmission
- Bit changed value during transmission, possibly due to EMI/RFI
- An Active Error Frame was sent

Exceptions

- Transmission of ACK delimiter, EOF field and Interframe Space (IFS): 11 consecutive recessive bits, bit stuffing does <u>not</u> apply to this section

Error types: CRC errors

Error description

- CRC code calculated by receiver does not match received CRC code

Method of detection

- Receiver calculates CRC code immediately after reception of the Data Field
- Receiver compares calculated CRC code with the one contained in frame

Efficiency

Use polynomial generator: x15 + x14 + x10 + x8 + x7 + x4 + x3 + 1

- The more bits the CRC code has, the more efficient it is

Error types: Message format errors

Error description

- Frame integrity is not preserved

Method of detection

- Receiver checks received frames for bits or bit fields having a fixed value (e.g. SOF bit, CRC delimiter, ACK delimiter, EOF field, Interframe Space)
- Violation of frame integrity when wrong value in one of these fields

Possible cause

- Transmission error, error in sender and/or receiver
- Transmission of an Active Error Frame during EOF field
- Transmission of an Overload Frame during Interframe Space (IFS)

Error types: Acknowledgement errors

Error description

- Acknowledge (ACK) bit is recessive

Method of detection

- Sender monitors the bus while transmitting recessive ACK bit
- Sender expects to observe dominant ACK bit on bus
- Acknowledgement error when ACK bit on bus remains recessive

Possible cause

- No other nodes are connected to the bus
- Not one single receiver acknowledges that the received frame was error-free

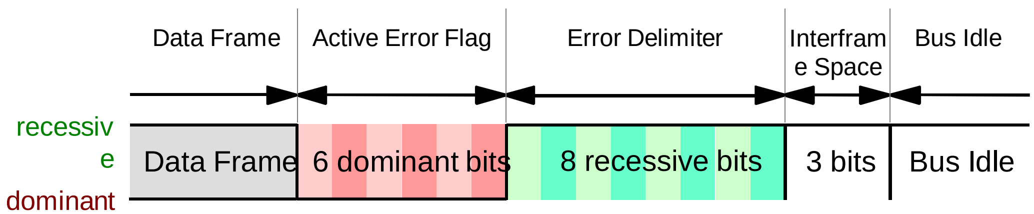
  → Cause of error is very likely to be found in sender

Error management (1)

Procedure observed after detection of an error (1)

- Immediate transmission of an Error Frame

Format of an Active Error Frame:



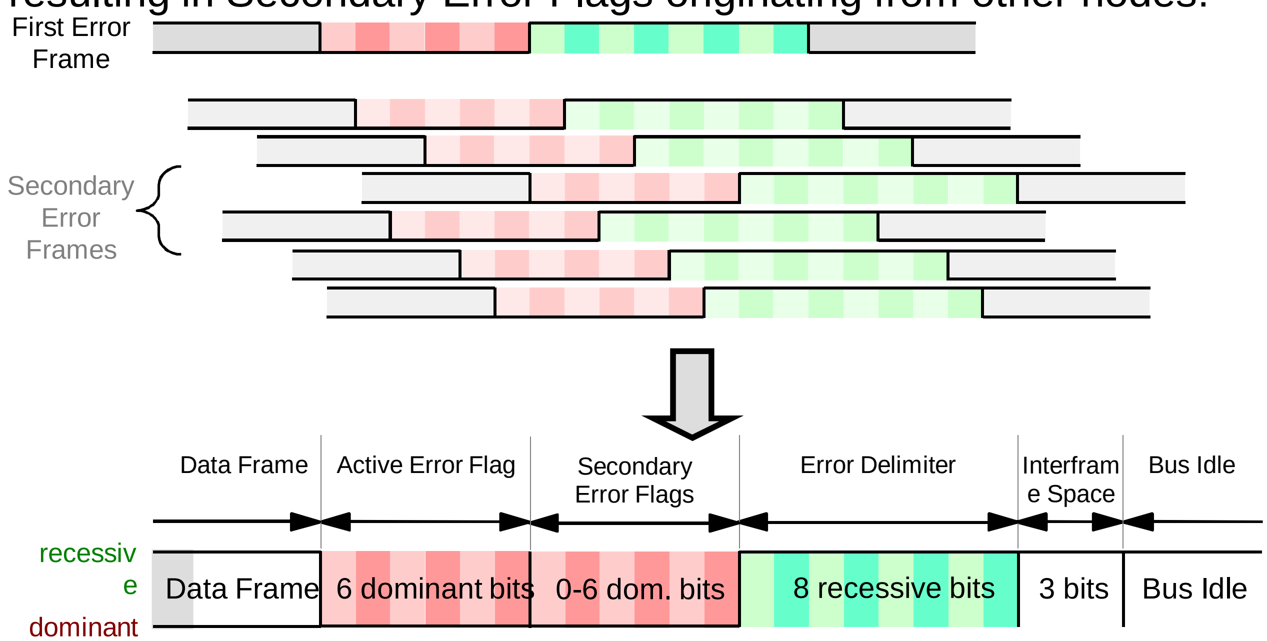
No bit stuffing is applied to Error Frames!

- Error Frame violates the bit stuffing rules $\rightarrow$ Other receivers are instantly informed that an error has occurred (unless they already found out)

Error management (2)

Procedure observed after detection of an error (2)

- As a result, other units also send Error Frames → Error frames may overlap,
  resulting in Secondary Error Flags originating from other nodes:



Error management (3)

Procedure observed after detection of an error (3)

- Sender and receivers reject erroneous frame completely and do not process it any further
- Sender retries transmission later
- All units on the bus increase their error counters
- Recovering from errors can take up to 23 bit cycles (max. 12 bits Error Flag + 8 bits Error Delimiter + 3 bits Interframe Space)

Error counters and node states

Two error counters for each unit

- Transmit Error Counter (TEC)
- Receive Error Counter (REC)

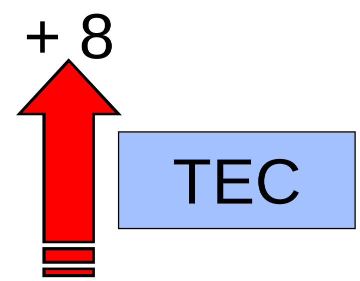
Characteristics of error counters

- TEC and REC start counting at 0 (zero)
- Distinction between sporadic (temporary) and permanent errors possible
- Error-prone units are deactivated automatically after a certain time
- Depending on the values of their error counters, units can assume one of three possible node states: Error active, Error passive, Bus off.

Transmit Error Counter (TEC)

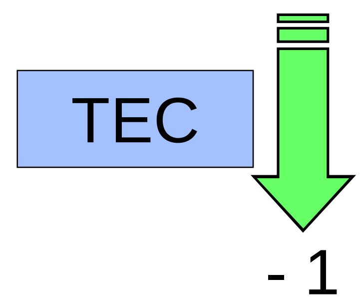
The TEC is increased by 8 when

- the unit detected an error in the frame it just sent.
  It is very likely the unit itself is defective.
- there are several other conditions of lesser importance and exceptions to them.
  Refer to the CAN specification for details.



The TEC is decreased by 1 when

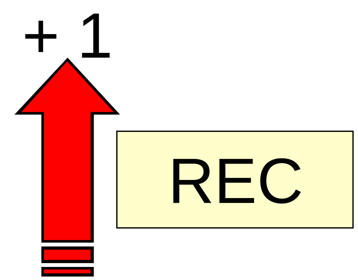
- the unit successfully transmitted a frame,
  i.e. a dominant ACK bit was observed on the bus and Error Frames were neither sent nor received.
- Note: The TEC is <u>not</u> decreased when TEC = 0.



Receive Error Counter (REC): Increase

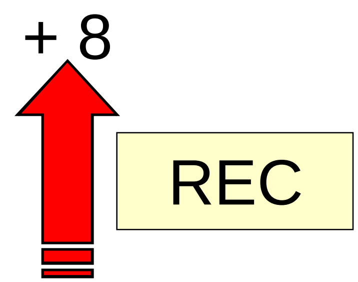
The REC is increased by 1 when

- the unit detected an error in a received frame.



Additionally, the REC is increased by 8 when

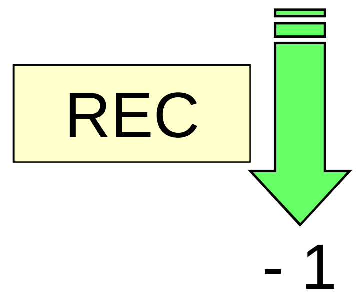
- the unit detected this error autonomously, i.e. not through another unit's Error Frame. This is the case when the unit observes a dominant bit following its own Error Flag on the bus.
- Usually, the REC cannot be greater than ca. 135.



Receive Error Counter (REC): Decrease

The REC is decreased by 1 when

- the unit successfully received a frame,
  i.e. Error Frames were neither sent nor received.
- Notes: The REC is not decreased when REC = 0.
  Usually, the REC is decreased by 8 when REC > 127.



Node states: Error active

A unit is in error active state when

- its Transmit Error Counter (TEC) is less than 128: TEC < 128 AND
- its Receive Error Counter (REC) is less than 128: REC < 128

In error active state a unit

- is fully operational
- sends an Active Error Frame when it has detected an error

Node states: Error passive

A unit is in error passive state when

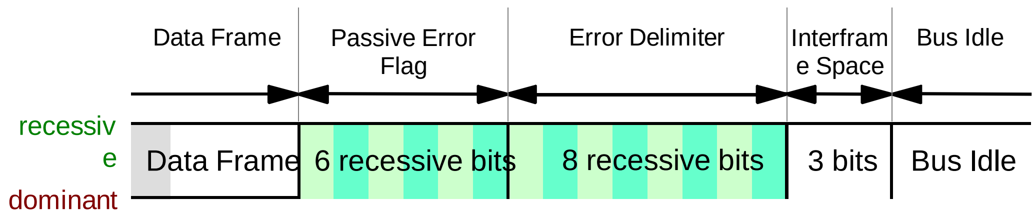
- its Transmit Error Counter (TEC) is greater than 127: TEC > 127 AND / OR
- its Receive Error Counter (REC) is greater than 127: REC > 127

In error passive state a unit

- sends a Passive Error Frame when it has detected an error
- can still receive frames like a unit in error active state can
- has to wait after transmission of a Data Frame for 8 additional consecutive recessive bit cycles on the bus (Suspend Transmission Field) until it is permitted to transmit another Data Frame
- can go back to error active state for TEC <= 127 AND REC <= 127

Passive Error Frame

Passive Error Frame



- a node in error passive state sends a Passive Error Frame in case of an error
- a Passive Error Frame cannot destroy an ongoing transmission on the bus
- the Passive Error Frame might overlap with Active Error Frames

Node states: Bus off

A unit is in bus off state when

- its Transmit Error Counter (TEC) is greater than 255: TEC > 255
- Note: The value of the Receive Error Counter (REC) is of <u>no</u> importance

In bus off state a unit

- is practically disconnected from the bus
- cannot receive and transmit anything any more
- can only leave bus off mode via a hardware reset OR
  a software reset and subsequent initialization carried through by the host
  (CAN specification: TEC and REC are set to zero and the unit must
  receive 128 times a field of 11 consecutive recessive bits)

Node states: Warning level

The Warning Level for a unit is set when

- its Transmit Error Counter (TEC) is greater than 96: TEC > 96 AND / OR
- its Receive Error Counter (REC) is greater than 96: REC > 96

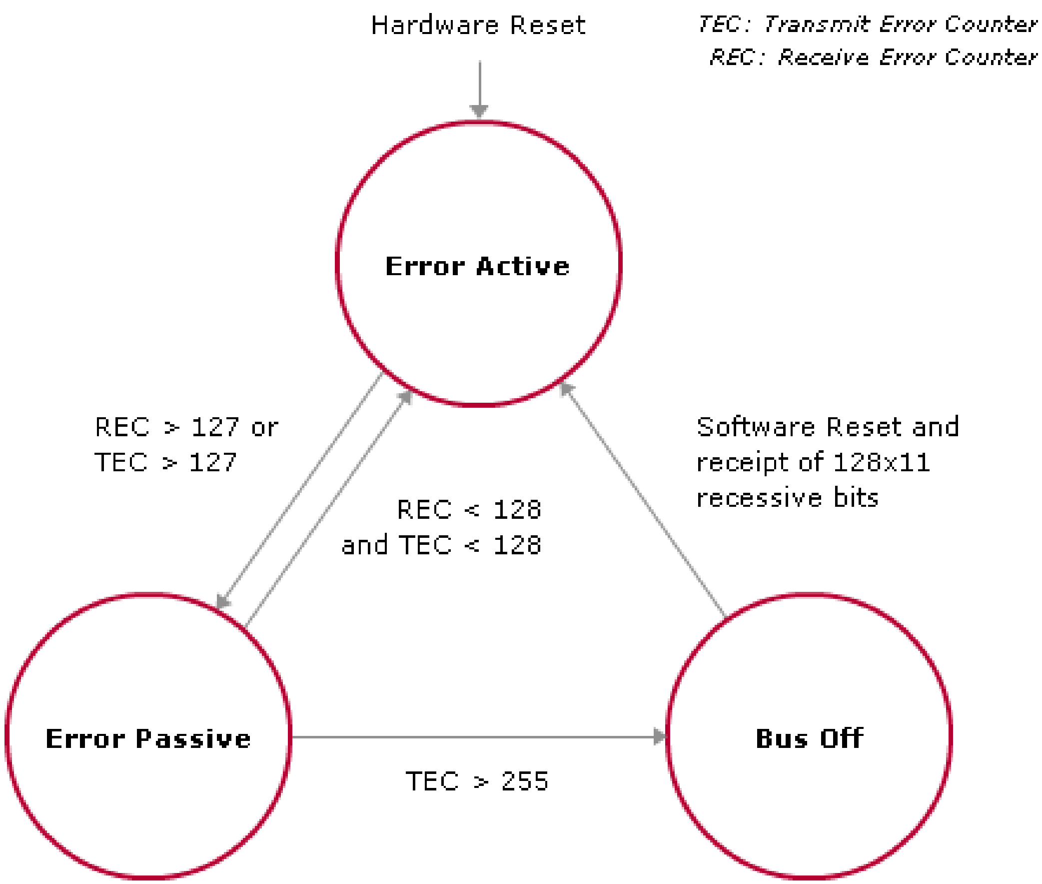
When a unit reaches the Warning Level

- the CAN controller sets its “Error Warning Flag”
- the “Error Warning Flag” is cleared for TEC <= 96 AND REC <= 96

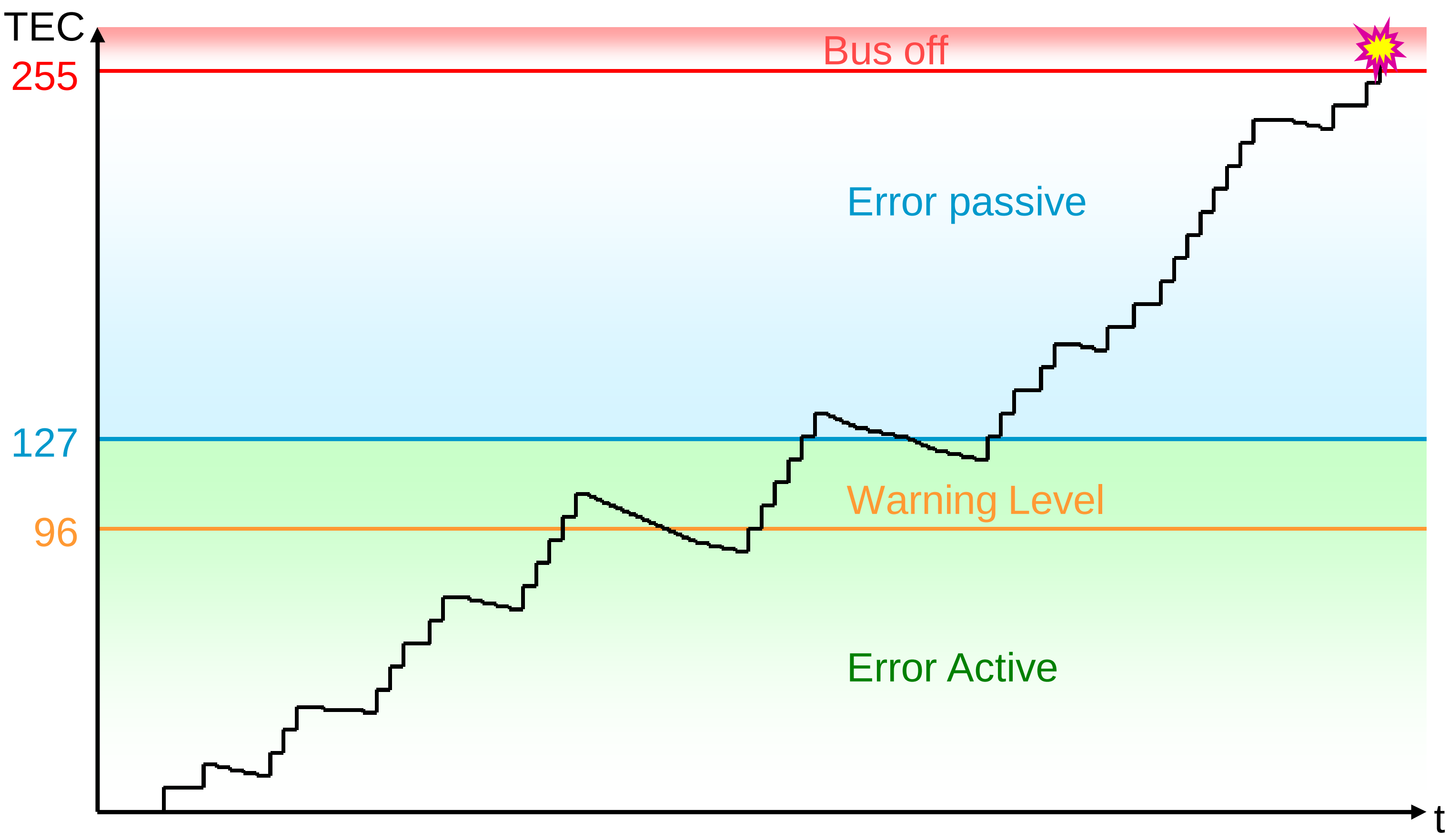
Practical use of the Warning Level

- by constantly checking the “Error Warning Flag”, it can be determined whether a unit gets near the threshold to the error passive state
- unfortunately, by checking this flag, one cannot determine whether a unit is already in error passive state

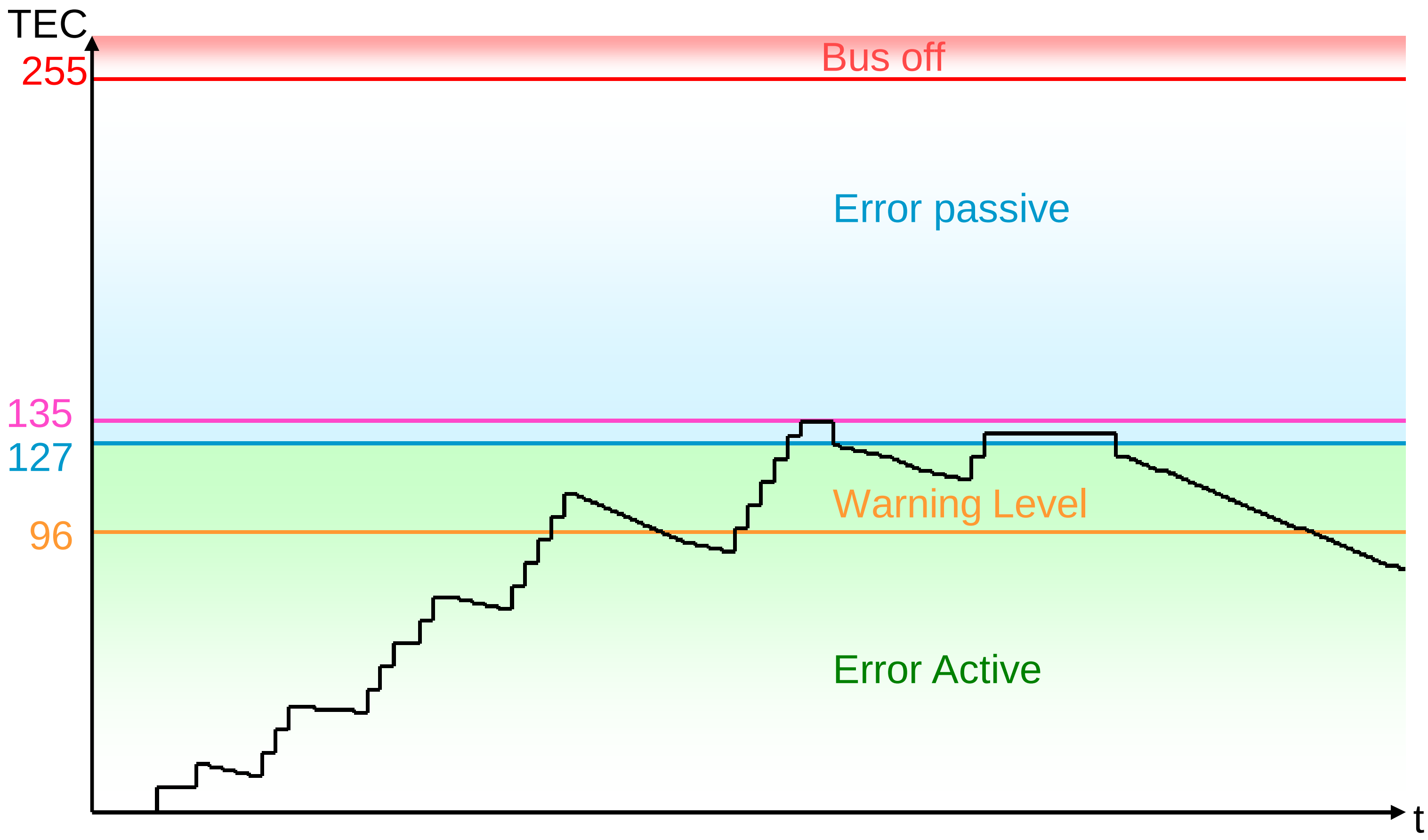
Node: state machine



Transmit Error Counter (TEC): Example



Receive Error Counter (REC): Example



Efficiency of the CAN error management

The probability for not discovering an error is

$$\boxed{4.7 * 10^{-11}}$$

Example 1*

A CAN bus is used	365 days / year	*Source: CiA
	8 hours / day	
with a transmission speed of	500 kBit / sec	
and errors arise every	0.7 seconds	

⇒ in 1.000 years, only one error remains undiscovered

Example 2**

A CAN bus in a car is run at	500 kBit / sec	**Source: Kaiser, Schröder: “Maßnahmen zur Sicherung der Daten beim CAN-Bus”
with an average bus load of	15 %	
an average data frame size of	110 bits	
for an average operating time of	4000 hours	

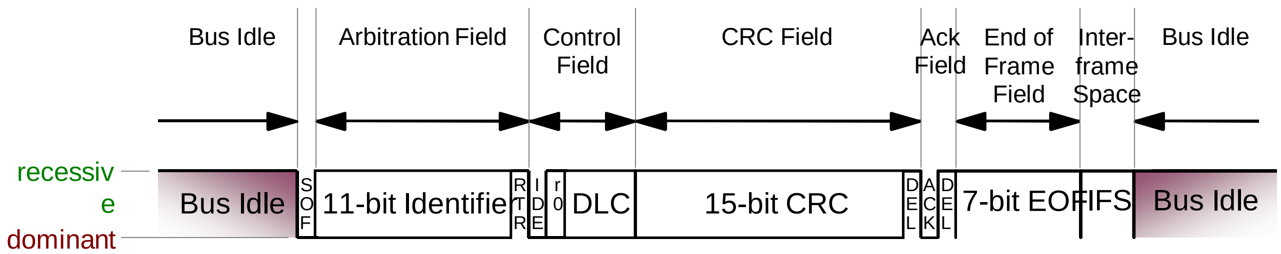
⇒ only one error in 100.000 automobiles remains undetected

CAN Controller Area Network Training

- Introduction / Technical features
- The ISO/OSI seven-layer model and CAN
- Frames
- Data Frames
- Bit stuffing
- Error management
- Remote Frames / Overload Frames

Remote Frame

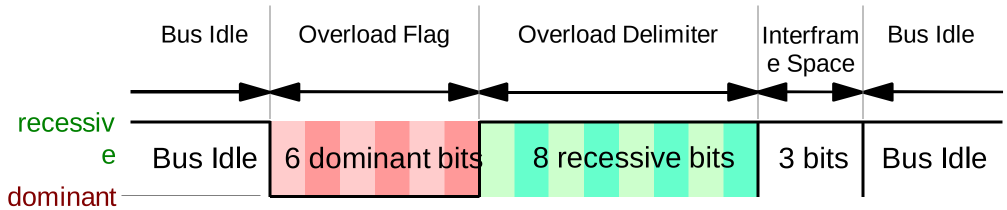
Remote Frame



- Used to request transmission of a specific Data Frame
- Similar to a Data Frame, but without Data Field
- Remote Transmission Request (RTR) bit is recessive
- Same identifier as the Data Frame which is requested
- Note: When Remote Frame is transmitted at the same time as corresponding Data Frame, Data Frame wins arbitration because of dominant RTR bit

Overload Frame

Overload Frame



- Unit sends Overload Frame when at present it cannot receive frames any more due to high workload
- Transmission of an Overload Frame is started during the first two bits of the Interframe Space (IFS) of the preceding frame
- Other units react immediately by also transmitting Overload Frames
  $\Rightarrow$ Overload Flags overlap, resulting in up to 12 consecutive dominant bits
- Implemented in very few (mostly older) controllers, though controllers must still be able to interpret correctly Overload Frames they receive
- Overload Frames do <u>not</u> influence the error counters (TEC and REC)