SIEMENS VDO

CAN (Control Area Network) Common Training

Marcel Liviu Sodinca – SV I IP TSR PV INT

SIEMENS VDO

CAN History Timeline

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

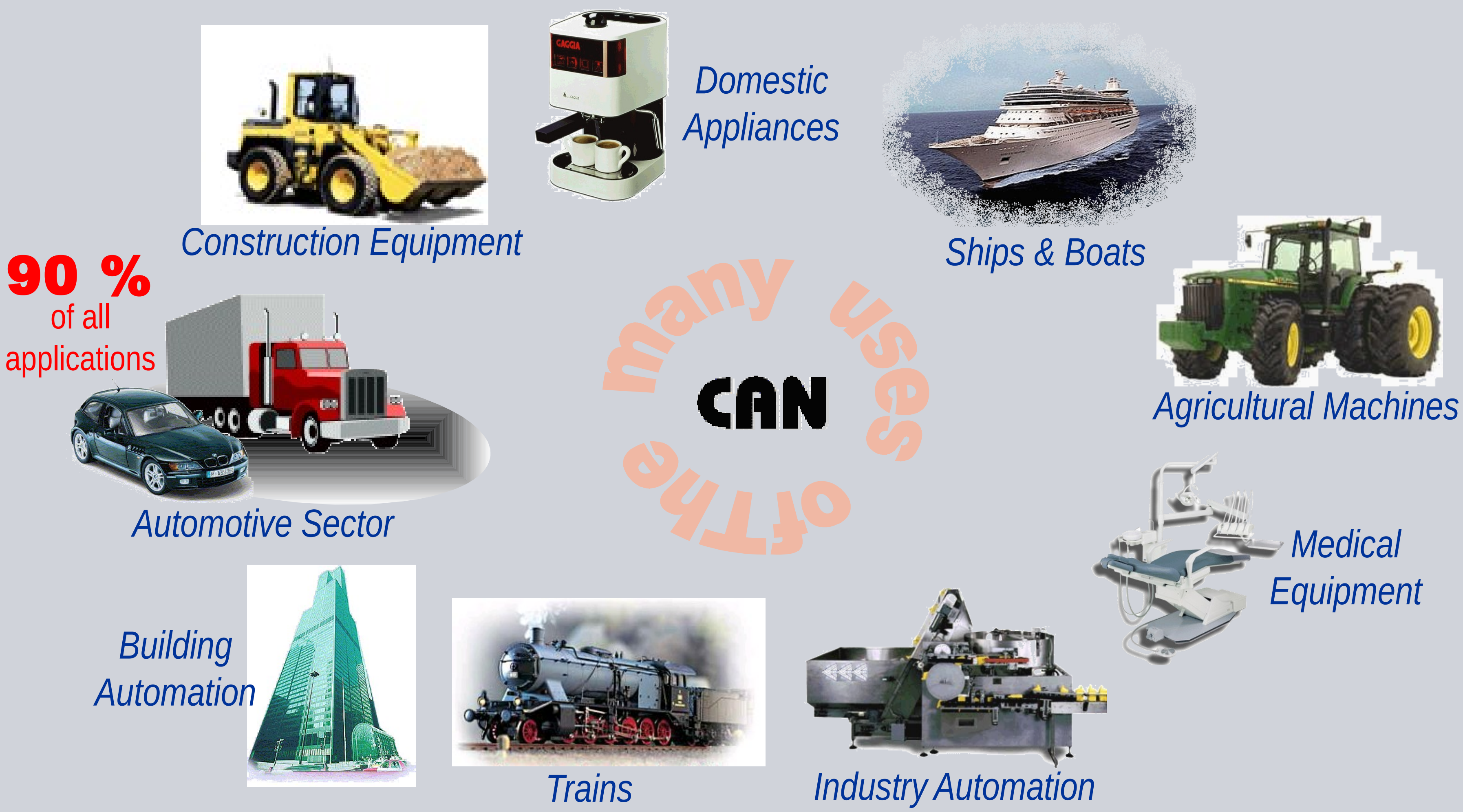
1988 — **First CAN Controller on the market: Intel 82C526**

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

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

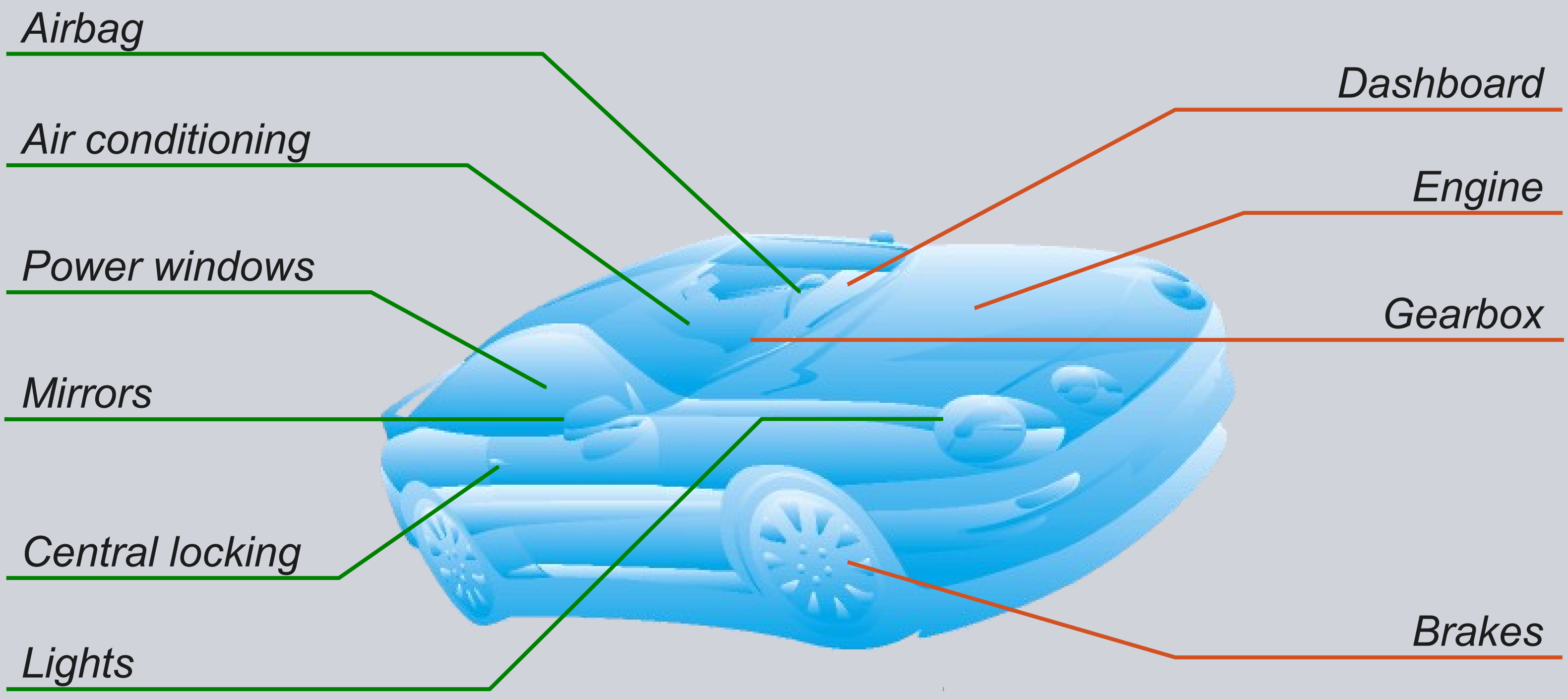
*Source: CAN in Automation (CiA) e.V.

Applications examples for CAN Bus



SIEMENS VDO

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.

- **"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.**

Bus characteristics

- ## 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.**

SIEMENS VDO

Hardware characteristics

- ## Hardware message acceptance filtering

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

- ## Sophisticated hardware error management

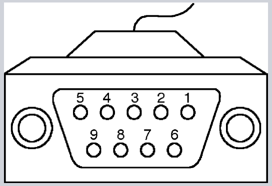
  Combination of different error prevention and detection methods, automatic re-transmission of messages detected as erroneous
  ➔ very high transmission security among field bus systems.

- ## Various transmission media

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

- ## Standardized connector

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



Types of Serial Data Transmission

Serial data transmissions may be classified in:

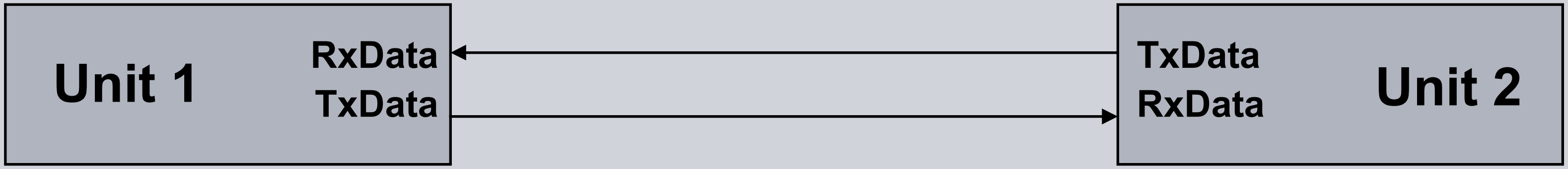
- **Synchronous Serial Data Transmission**

  Data is synchronous transmitted to a clock signal

  (e.g. data shift register, I^2C bus, etc.)

- **Asynchronous Serial Data Transmission**

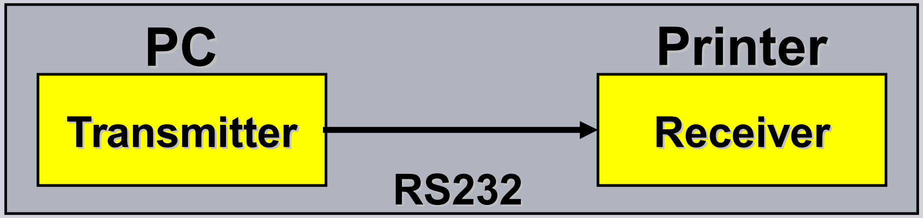
  Transmitter and Receiver are synchronized via a start bit (e.g. RS232, CAN, etc.)



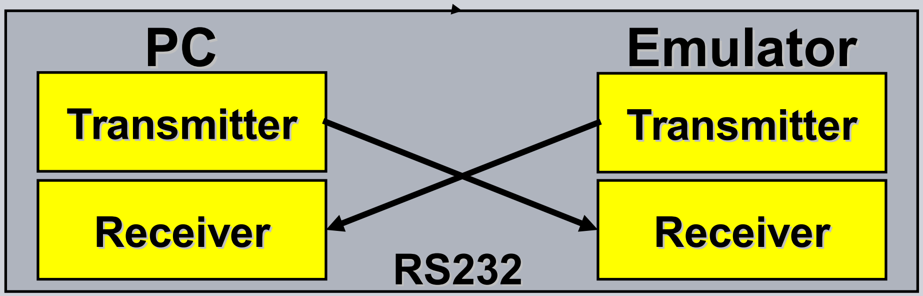
Types of Serial Data Transmission :

Serial data transmissions may also be classified in:

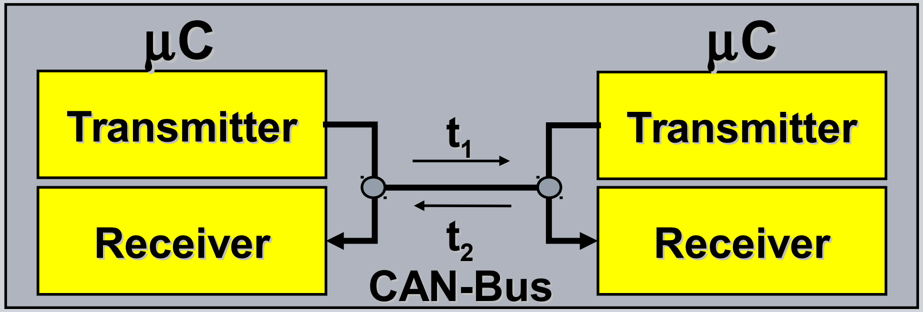
- **unidirectional**



- **bidirectional full duplex**



- **bidirectional half duplex**



Serial Data Transmission with and without Protocol:

- **Data Transmission without Protocol**

  => only data frames:

  - **RS232** — **e.g. UART/USART plus Transceiver**
  - **RS485** — **e.g. UART/USART plus Transceiver**

- **Data Transmission with Protocol**

  => Address + Data + Data-check + Signaling-information:

  - **I²C** — **e.g. I²C-Controller plus Transceiver**
  - **CAN** — **e.g. CAN-Controller plus Transceiver**

ISO/OSI Layer structure of CAN node

SIEMENS VDO

(International Standard Organization's Open System Interconnect)

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

ISO/OSI Layer structure of CAN node

SIEMENS VDO

(International Standard Organization's Open System Interconnect)

7 Application Layer

6 Presentation Layer

5 Session Layer

4 Transport Layer

3 Network Layer

2 Data Link Layer

1 Physical Layer

Electronic Control Unit

not specified by
the CAN Specification

Implemented in Hardware

Acceptance filtering, error management, de-/encryption, (de-)stuffing, arbitration, frame setup, acknowledgement

Encoding / decoding, bit timing, synchronization, wiring, connectors, bus, transceiver characteristics

ISO/OSI Layer structure of CAN node

SIEMENS VDO

(International Standard Organization's Open System Interconnect)

7	**Application Layer:**	Interface between the data communication environment and the application.	
6 : 3	**Presentation Layer:** **Session Layer:** **Transport Layer:** **Network Layer:**	**Not explicitly specified in CAN**	
2	**Data Link Layer:**	**Logical Link Control** **(Object Handling)**	**Acceptance Filtering** **Overload Notification** **Error Recovery Management**
		Medium Access Control **(Transfer Handling)**	**Data Encapsulation / Decapsulation** **Message Framing incl. Stuffing / Destuffing** **Medium Access Control incl. Arbitration** **Error Detection / Signalling / Confinement** **Acknowledgement Handling**
1	**Physical Layer:**	**Physical Signalling**	**Bit Encoding / Decoding** **Bit Timing** **Frame / Bit Synchronization**
		Physical Medium Attachment	**Driver / Receiver Characteristics**
		Transmission Medium	**Cable** **Connectors**

How to remember the seven layers

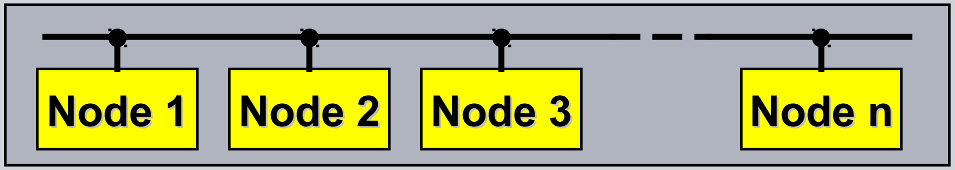
7	Application Layer	All
6	Presentation Layer	People
5	Session Layer	Seem
4	Transport Layer	To
3	Network Layer	Need
2	Data Link Layer	Data
1	Physical Layer	Processing

SIEMENS VDO

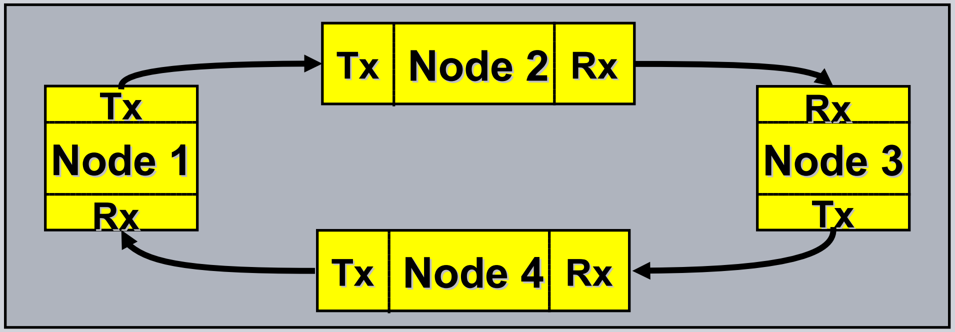
Bus structure

The most used bus structures:

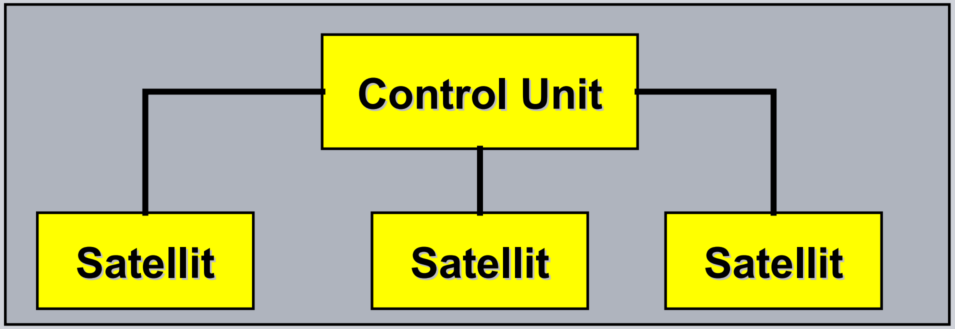
- **linear bus**
  e.g. Ethernet, PROFIBUS



- **ring bus**
  e.g. InterBus-S



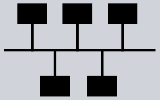
- **bus star**
  e.g. USB

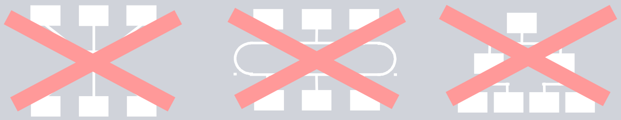


CAN Bus structure

- **Linear bus structure**

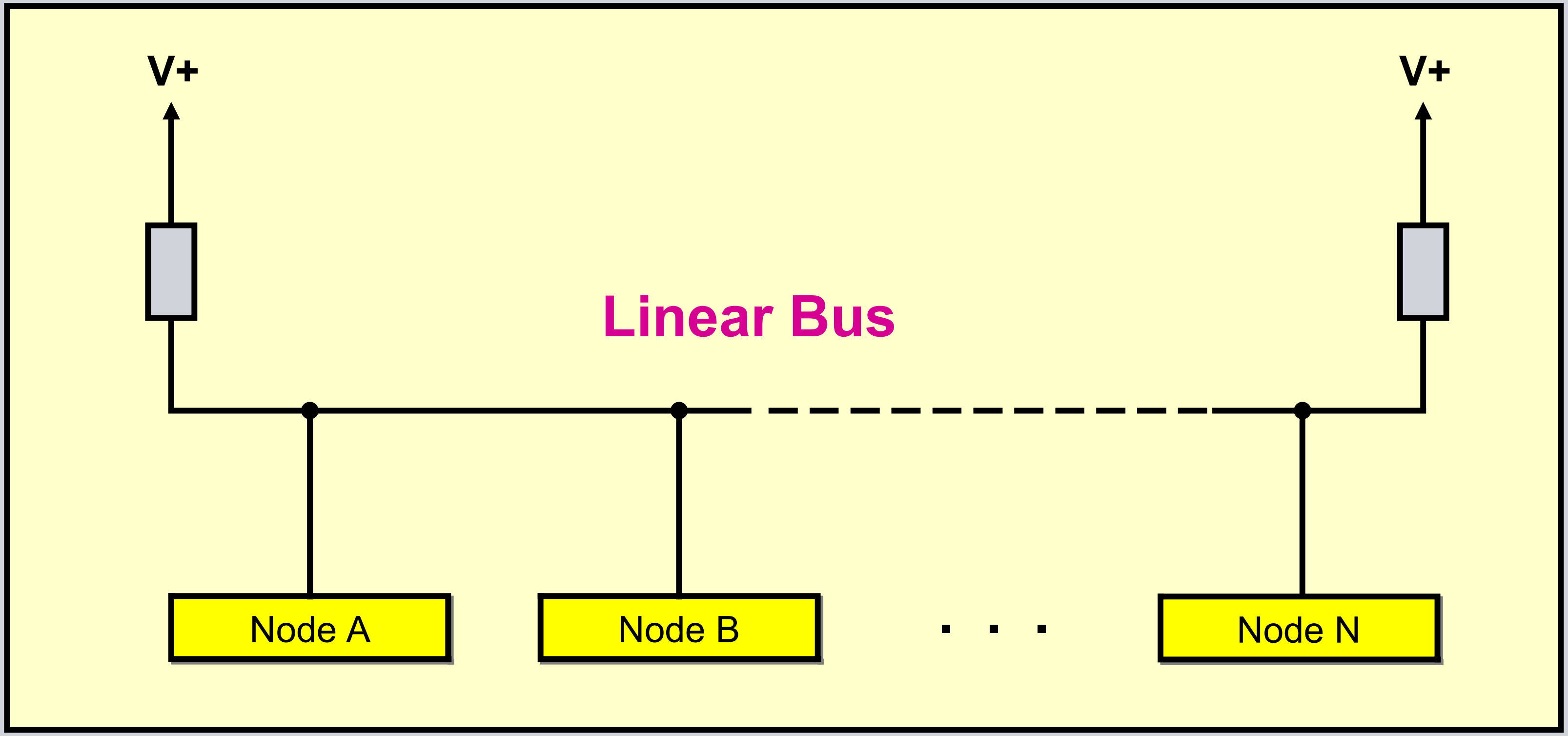
No star structure, no ring structure, no tree structure !





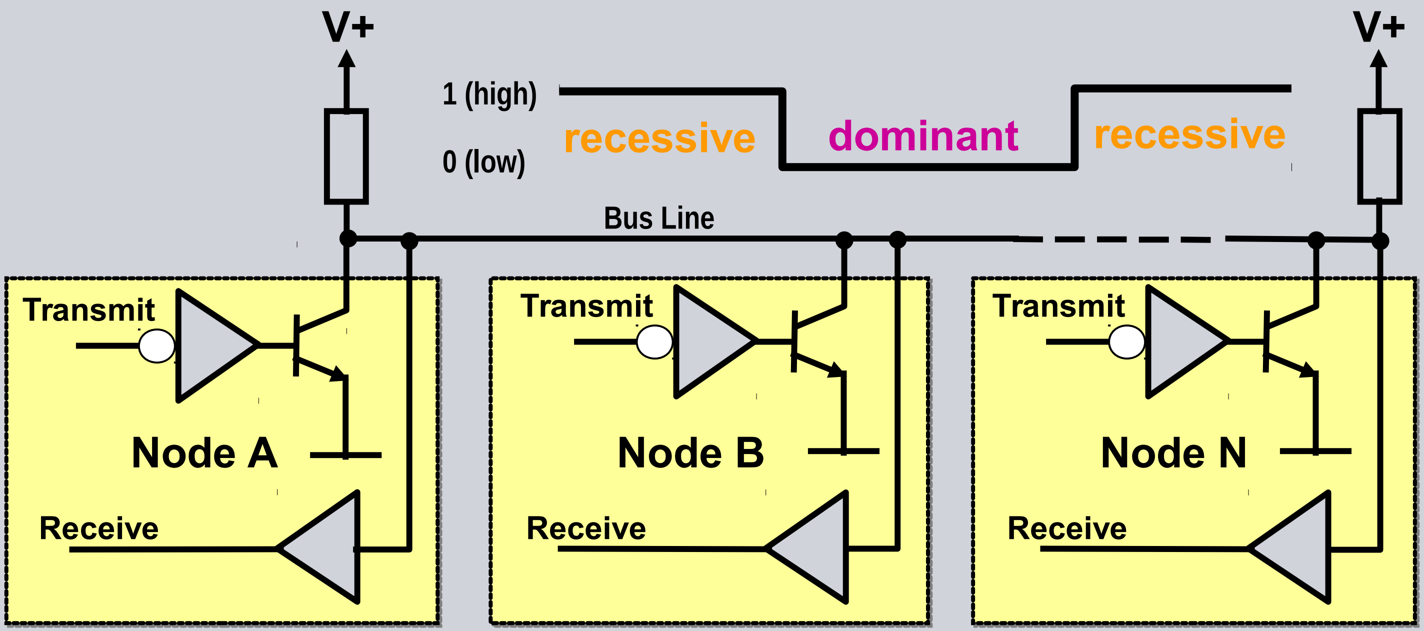
How the CAN networks works

➜ **CAN uses serial half duplex data communication with a linear bus structure**



Physical Transmission

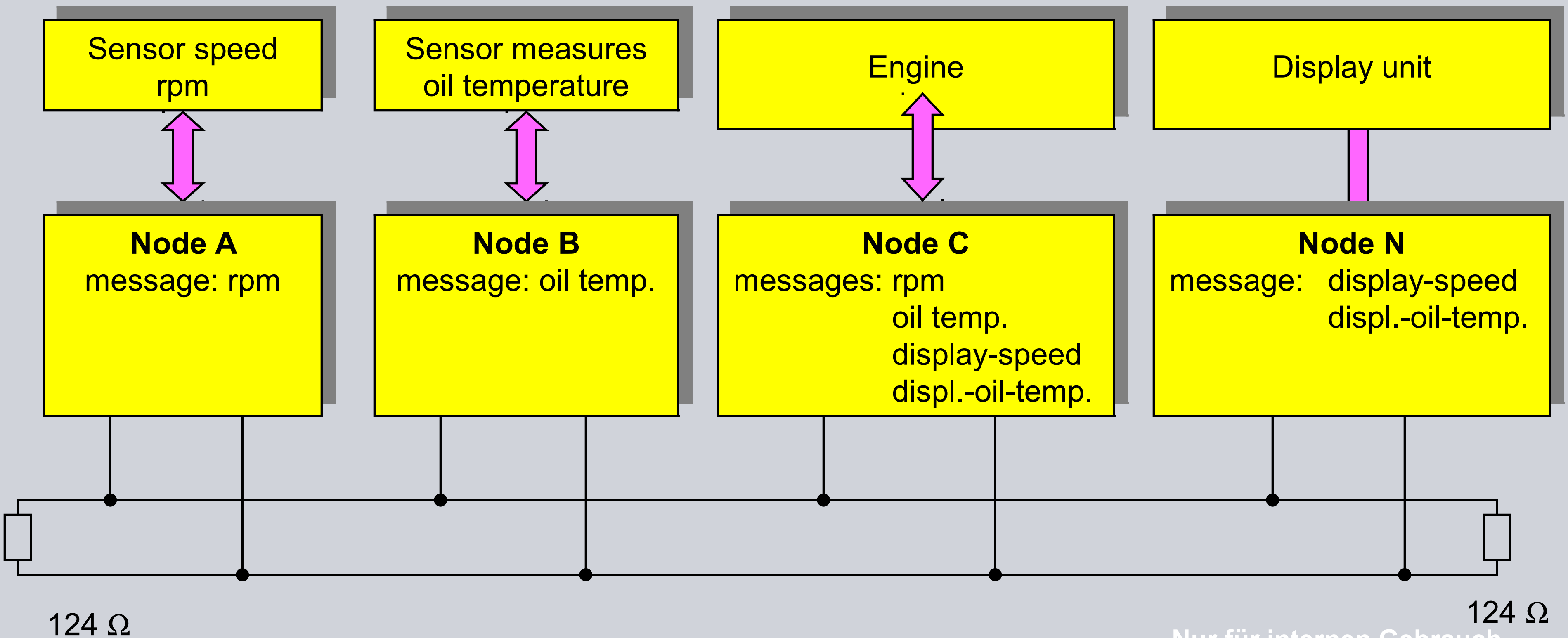
- Transmission with Carrier Sense Multiple Access with Collision Avoidance **CSMA/CA**
- **dominante** Bits (low) win against **recessive** Bits (high) on the bus



How the CAN networks works

➜ **CAN doesn't address stations with physical addresses, but instead of this the message gets an identifier**

An identifier is used as in form of symbol on the programming level.



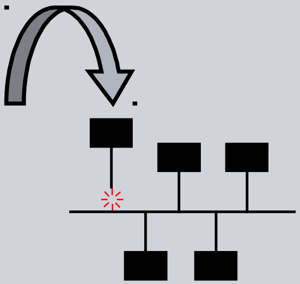
Bus stations (Nodes)

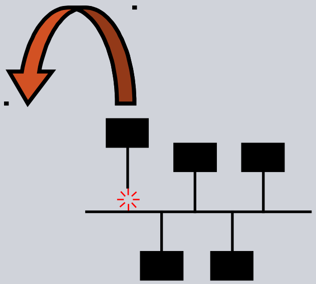
- **Typically 3 to 40 nodes per bus**

  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.

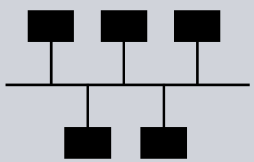




Advantages of the CAN bus

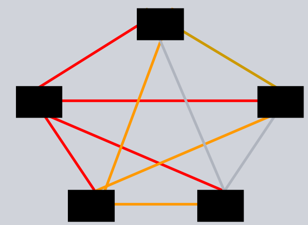
Advantages of

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

CAN Physical layer

The Physical Layer is responsible for:

Physical Signalling	**Bit Encoding / Decoding**
	Bit Timing
	Frame / Bit Synchronization
Physical Medium Attachment	**Driver / Receiver Characteristics**
Transmission Medium	**Cable**
	Connectors

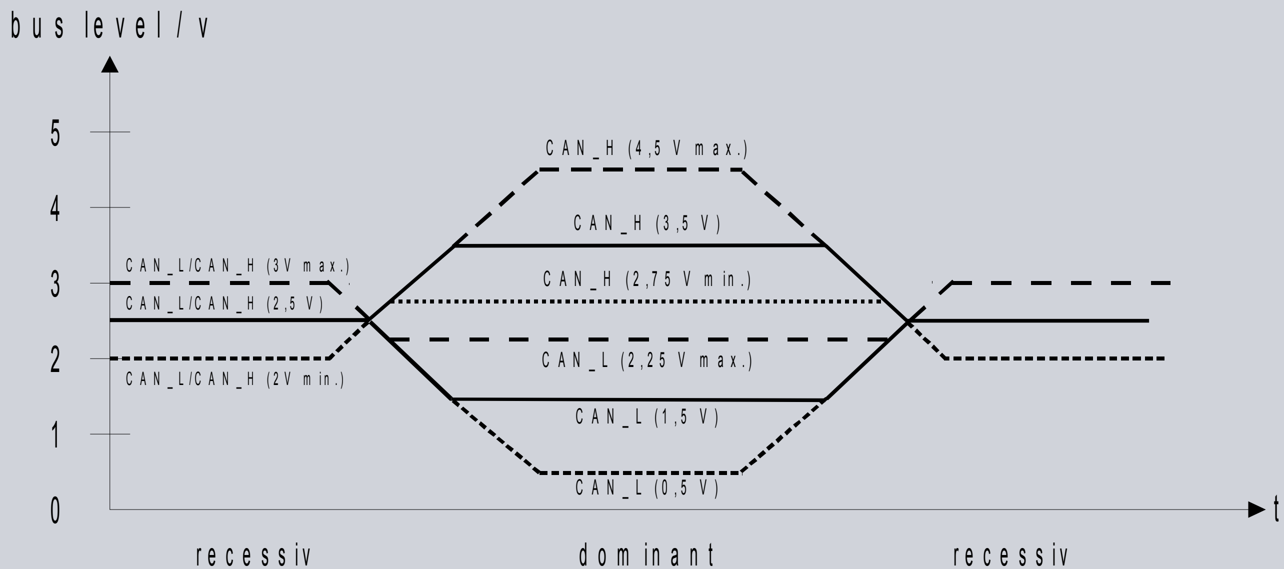
There specifications concerning both the Physical Signalling and the Physical Medium Attachment:

CAN High Speed Transmission	**ISO/DIS 11898**
CAN Low Speed Transmission	**ISO/DIS 11519-2**

These Specifications do **not define** the **Transmission Medium** (Cable and Connectors)

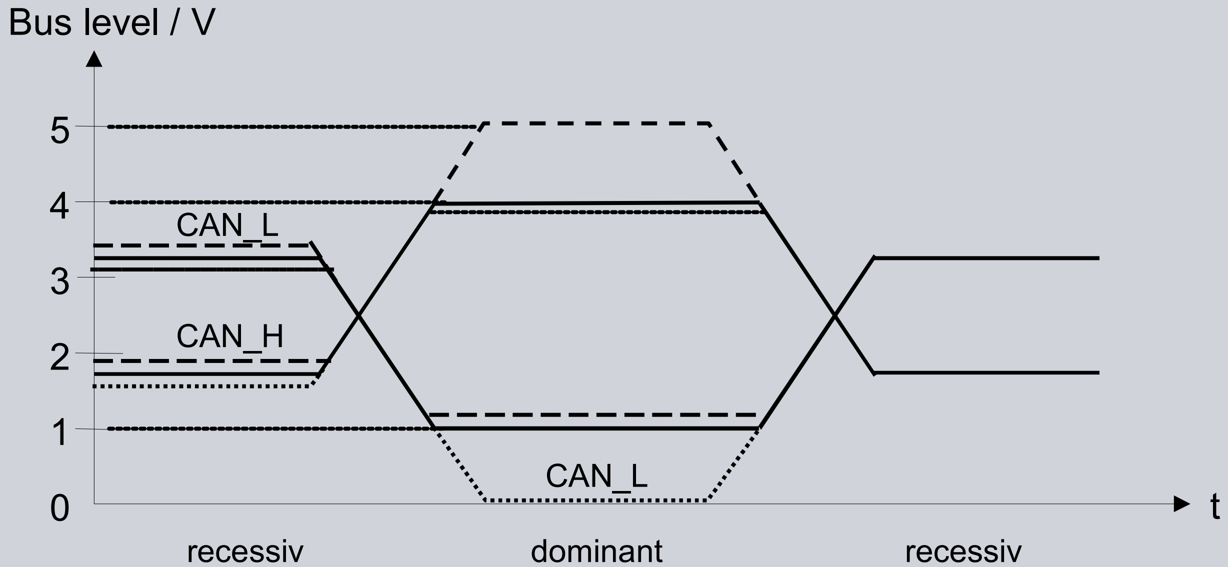
CAN - Physical aspects

- ➜ **Transmission rate up to 1 MBit/s needs bus driver circuits (transceiver) according to ISO/DIS 11898**
- ➜ **CAN High Speed Transmission (ISO/DIS 11898)**
  - ♦ Transmission rate 125 Kbit/s up to 1 MBit/s
  - ♦ max. bus length depend on transmission rate (e.g. 1 MBit/s max. 40 m bus length)
  - ♦ up to 30 nodes



➜ CAN Low Speed Transmission (ISO / DIS 11519-2)

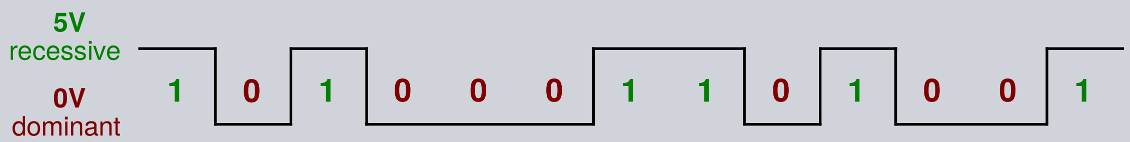
- Transmission rate 10 KBit/s up to 125 KBit/s
- max. bus length depend on the distributed capacity of the line
- up to 20 nodes



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.

CAN Protocol and Frame Types

CAN Protocol and Frame Types: 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**

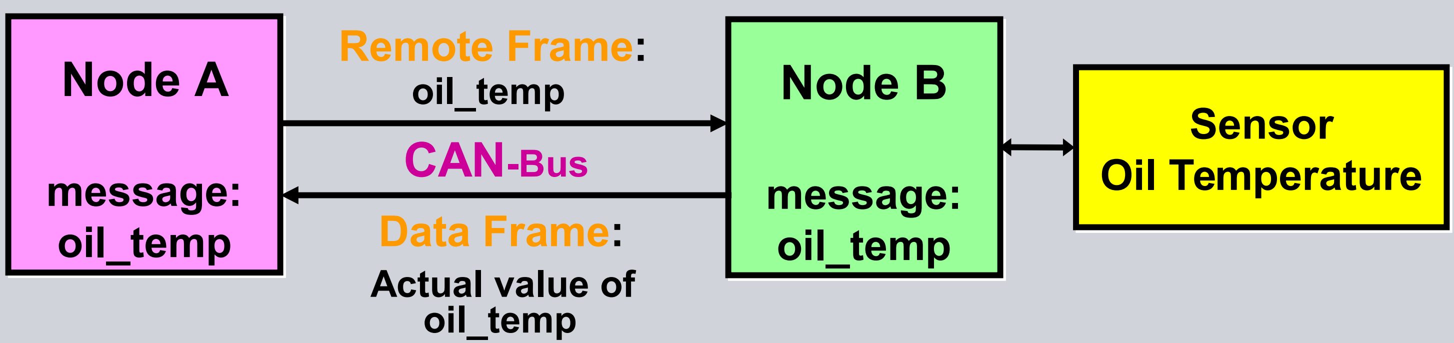
CAN Message Format

- **IFS/IDLE** Idle Level of Data Line (high)
- **SOF** Start of Frame (low)
- **ID** Standard CAN: 11 Bit Identifier
  Extended CAN: 29 Bit Identifier
- **Control** Remote, ID-Extension, reserved Bit(s), Data Length Code
- **Data** 0 bis 8 Data bytes
- **CRC** Cyclic Redundancy Check + CRC Delimiter
- **ACK** Acknowledgement - Receiver Response + Delimiter
- **EOF** End of Frame (7 bits recessive)
- **IFS** Inter Frame Space (3 bits recessive)
- **IDLE/SOF** Bus Idle Level or SOF of the next message

CAN Transmission Types

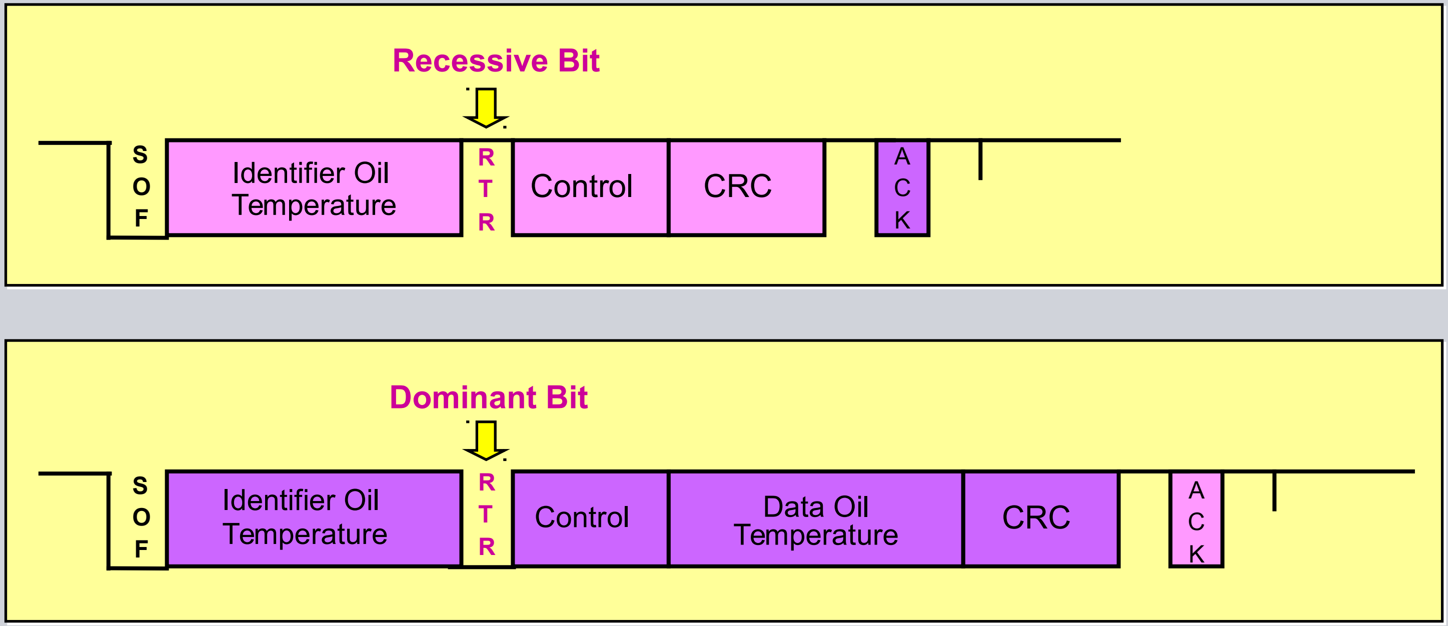
- ➪ **Data Frame** — Transmission of up to 8 byte user data with Standard or Extended Frame (Remote-Bit **RTR = 0**)
- ➪ **Remote Frame** — Destination node is able to require data from another node by sending a Remote Frame (available with Standard and Extended Frames) (Remote-Bit **RTR = 1**)
- ➪ **Error Frame** — Each CAN node detecting an transmission error sends an “Error Frame” (Sequence of 6 bit with the same level)

Example:



CAN Remote and Data Frame

Node A transmits
Remote Frame
(request data)



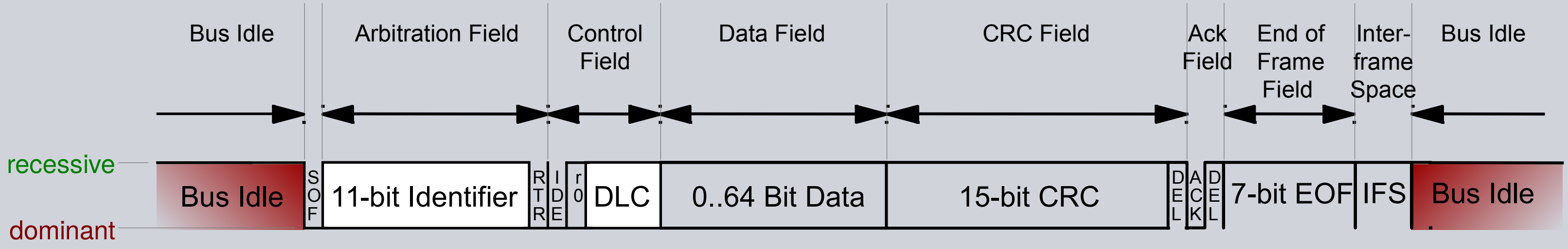
Node B transmits

requested
Data Frame

CAN Protocol and Frame Types

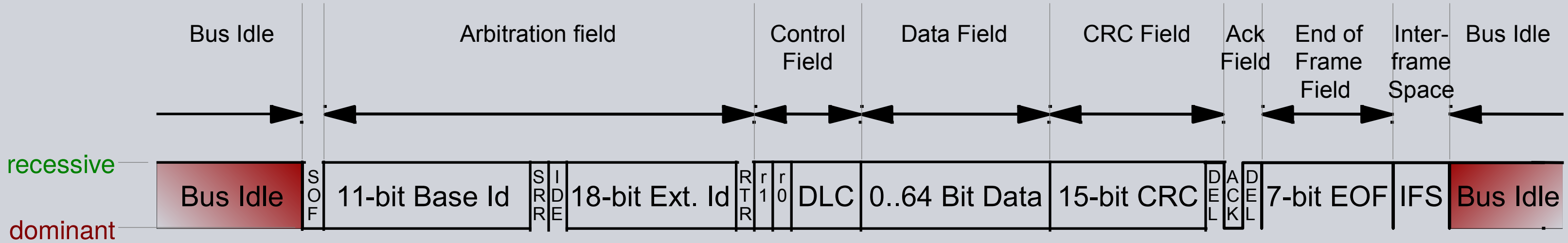
Standard Format (CAN 2.0A): 11-bit Identifier

2^{11} = 2048 (in reality only 2032) identifiers possible

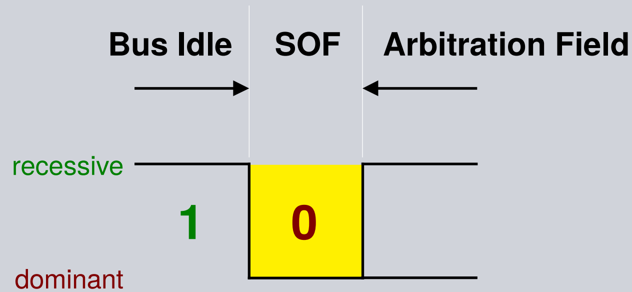


Extended Format (CAN 2.0B): 29-bit Identifier

2^{29} = 536.870.912 identifiers possible

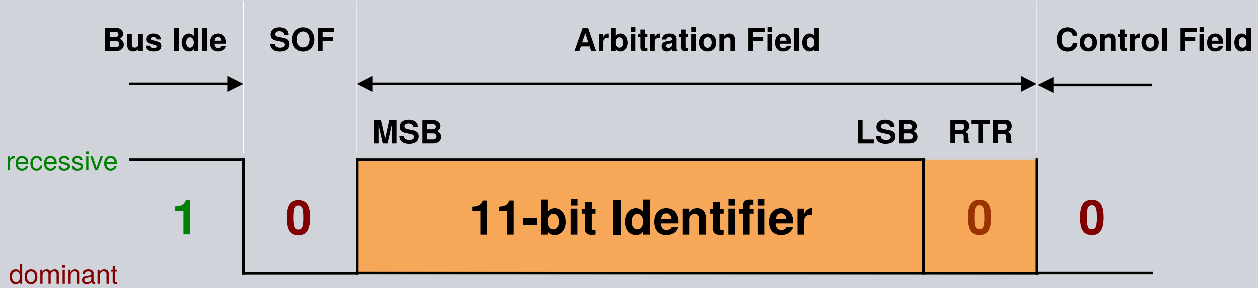


Start Of Frame (SOF) bit



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

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 on CAN

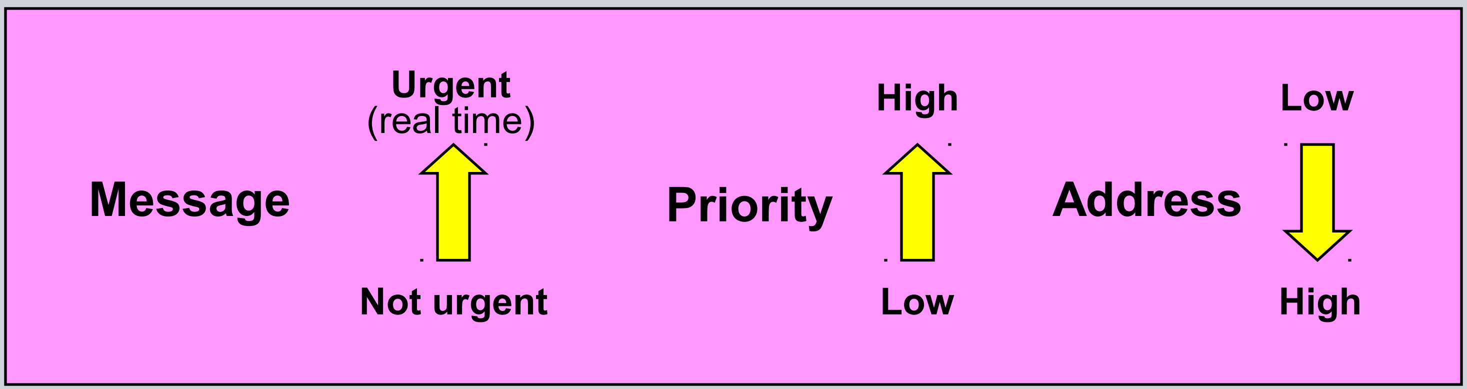
- **Arbitration = the allocation of bus access rights**
- **Arbitration occurs when several nodes start transmission on the bus at the same time**
- **Arbitration procedure:**
  1. All controllers monitor the bus while transmitting simultaneously
  2. A dominant bit (“0”) pulls the bus voltage level 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

SIEMENS VDO

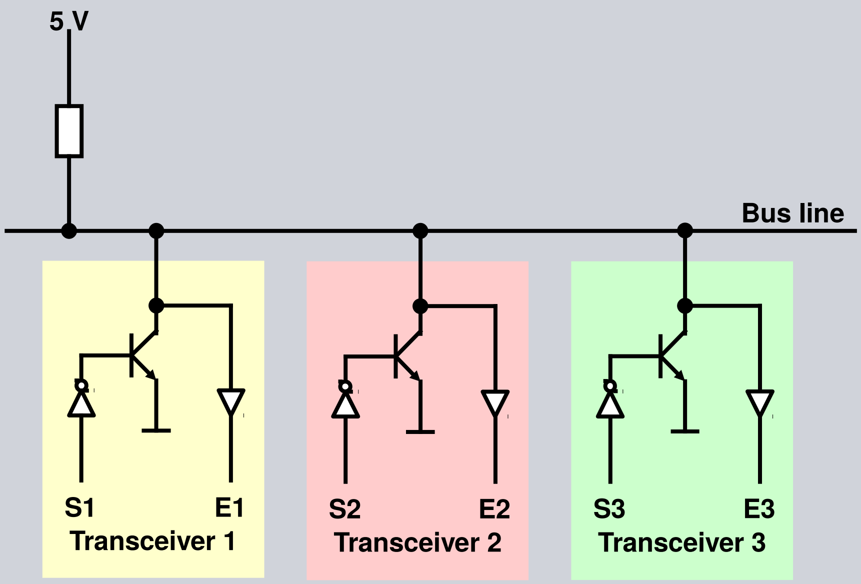
Arbitration on CAN

Message Prioritization

Each message includes an identifier. This identifier is used receive selected messages from the CAN bus data stream and for a simultaneous priority access of different messages.



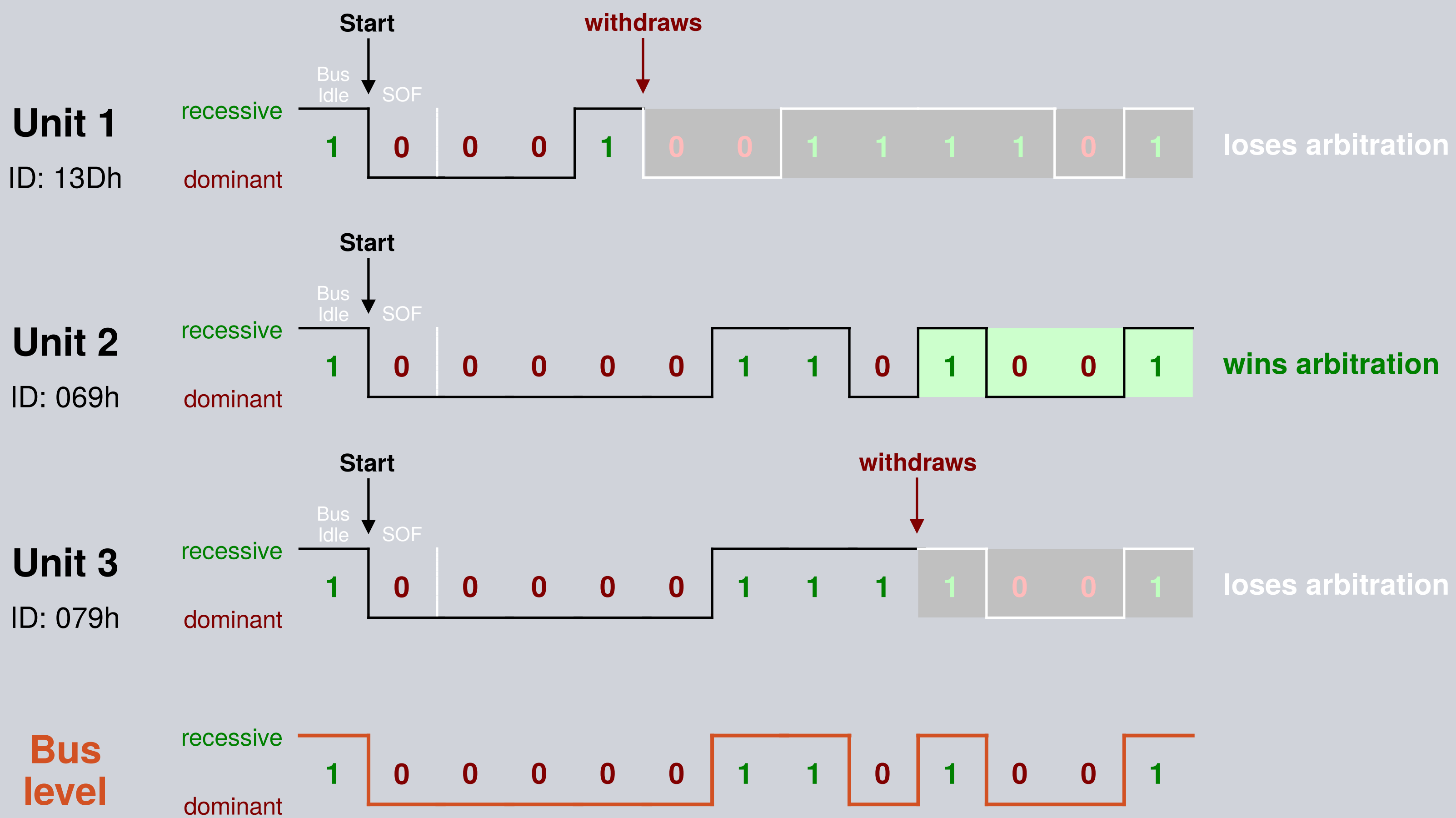
Hardware representation of recessive and dominant bus levels



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

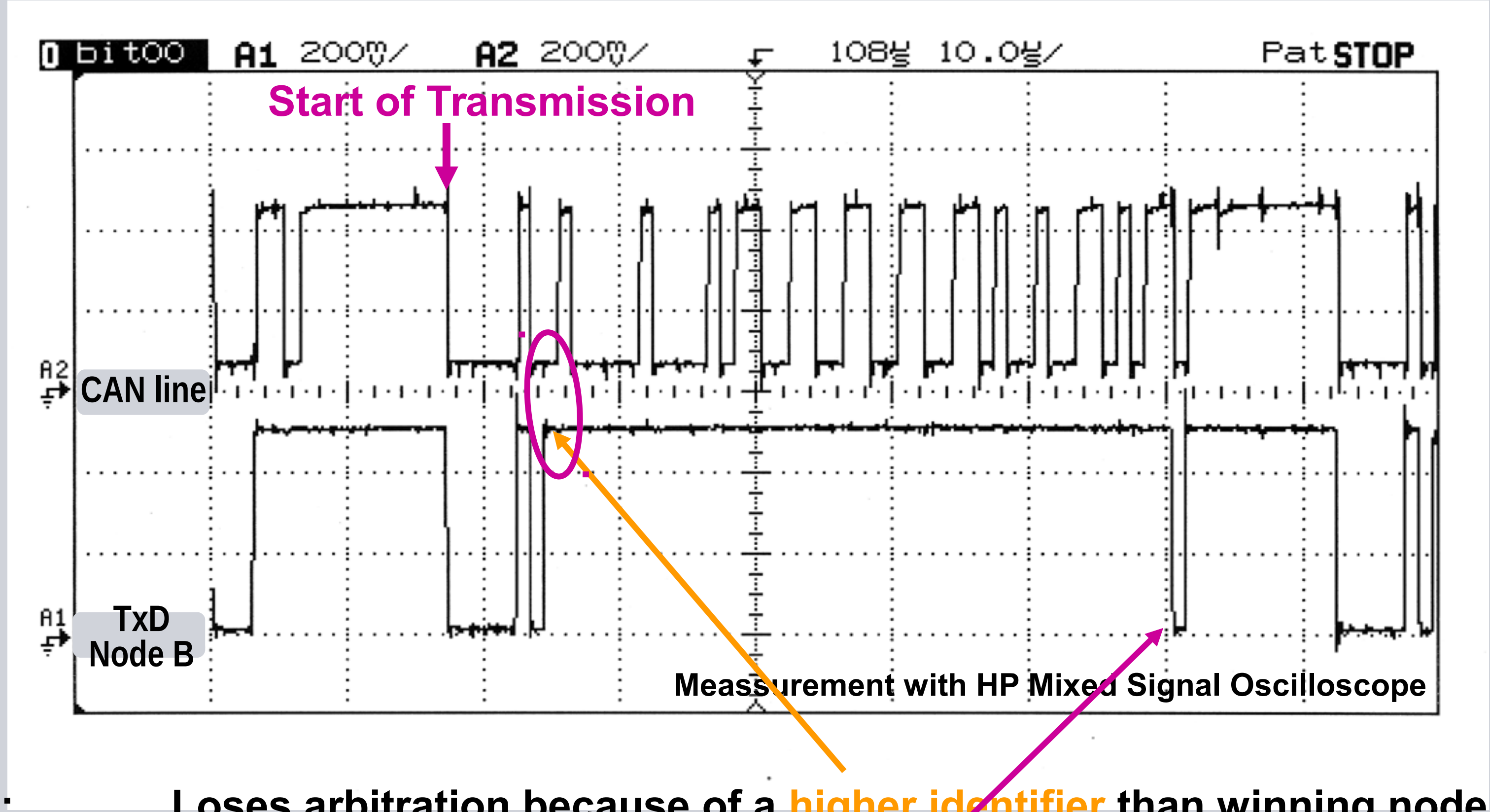
Arbitration example.



SIEMENS VDO

Arbitration example.

Example: Node A1 and A2 are accessing the CAN bus synchronously.



Node B: Loses arbitration because of a higher identifier than winning node.
Node B: Switches to receive mode and acknowledges the transmitted message.
Winning node: Transmits with a lower identifier ➠ wins arbitration on the CAN bus.

SIEMENS VDO

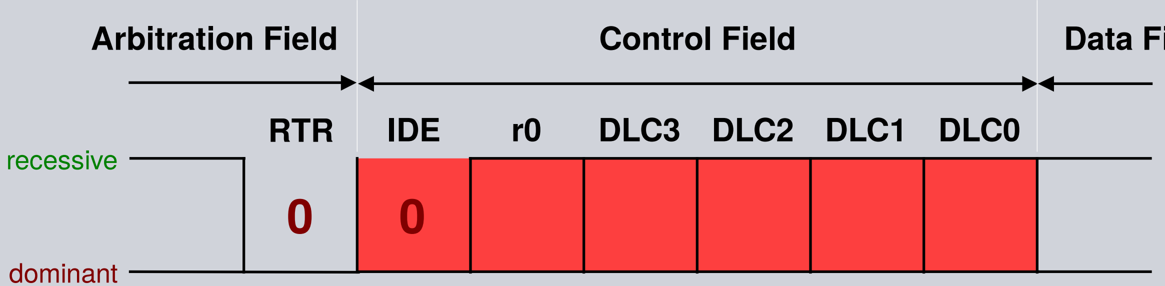
Arbitration consequence

!

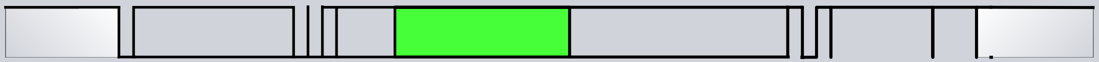
Consequence:

Frames carrying information of high priority should have a low identifier

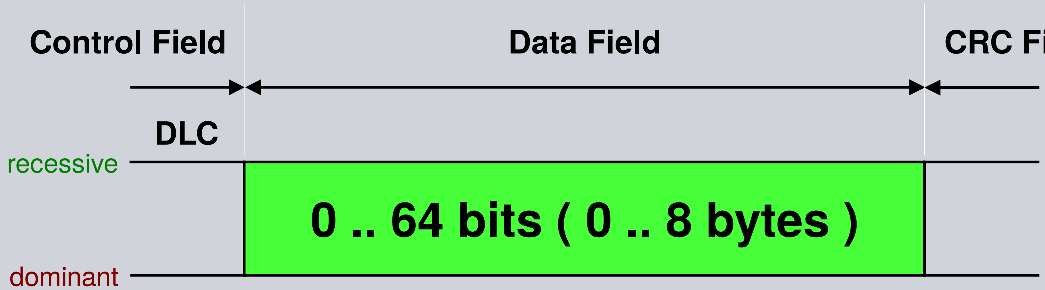
Control Field



- 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

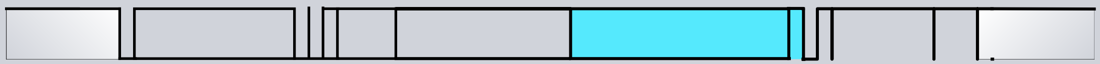


Data Field

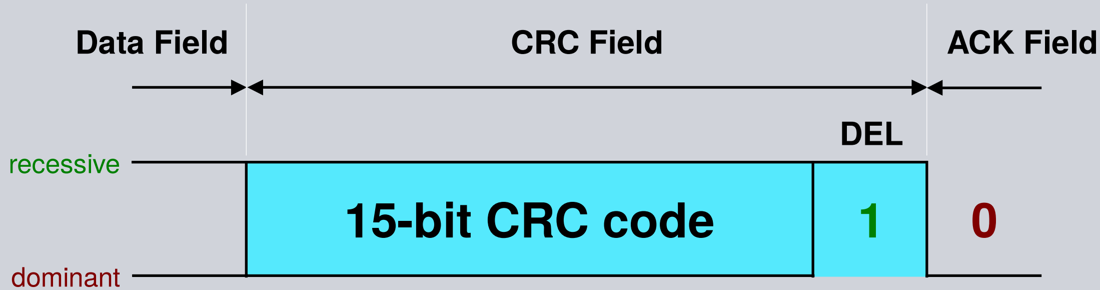


- **contains the actual information which is transmitted**
- **number of data bytes may range from 0 to 8 <u>in units of bytes</u>**
- **number of data bytes is given in the Data Length Code (DLC)**
- **transmission starts with the first data byte (byte 0), MSB first**

Data Frame Format

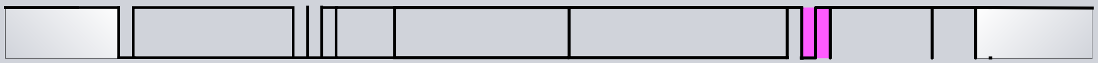


CRC Field

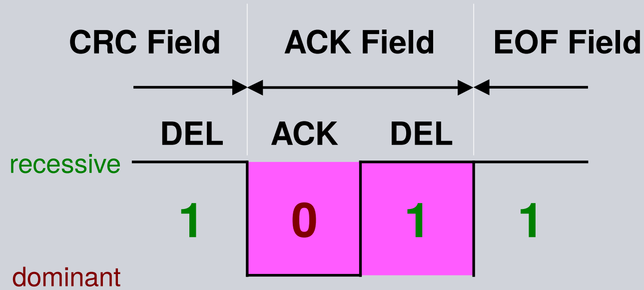


- contains the 15-bit Cyclic Redundancy Check (CRC) code
- CRC is a complex, but fast and effective error detection method
- the CRC Field Delimiter (DEL) marks the end of the CRC field
- the CRC Field Delimiter is always recessive

Data Frame Format



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

Data Frame Format

ACK Field

2 bit:	**ACK Slot**	and	**ACK Delimiter**
Transmitter:	recessive		recessive
Receiver:	dominant		

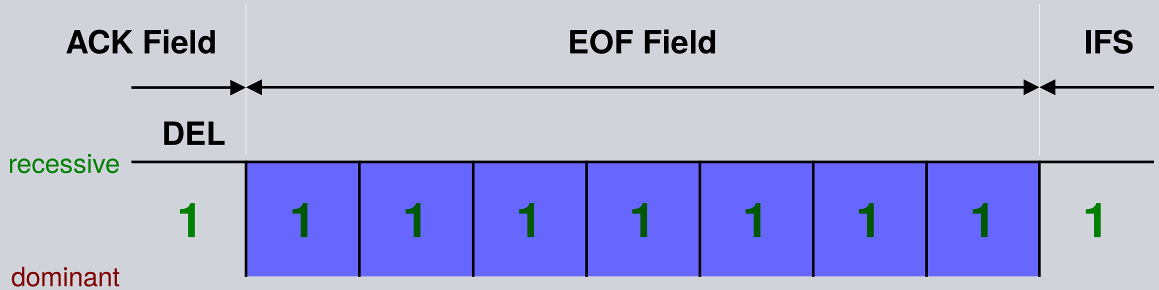
Acknowledgement

Each Receiver which received a valid message sends during the ACK slot a dominant bit.
Each Transmitter get the information that minimum one node received the message correct. If the transmitter gets no acknowledgement a retransmission is started.

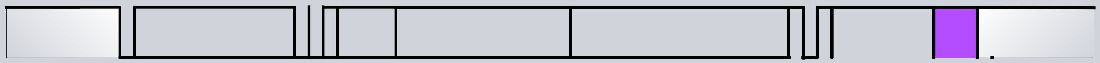
A recessive ACK bit

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

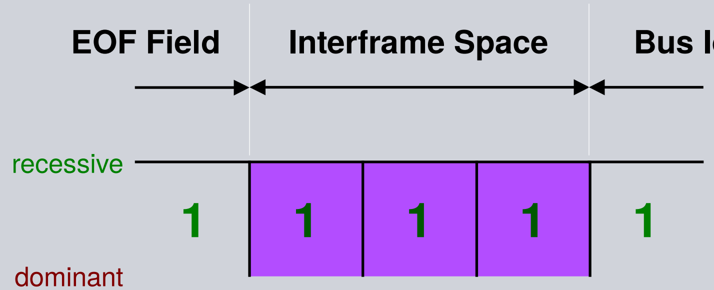
End Of Frame (EOF) Field



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

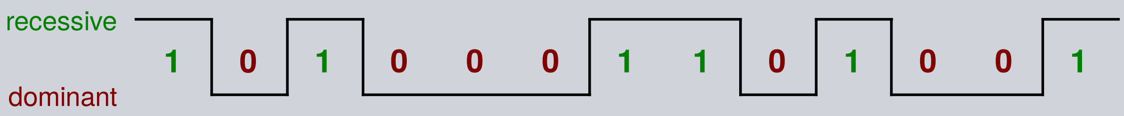


Interframe Space (IFS)



- 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

Problems when using NRZ coding



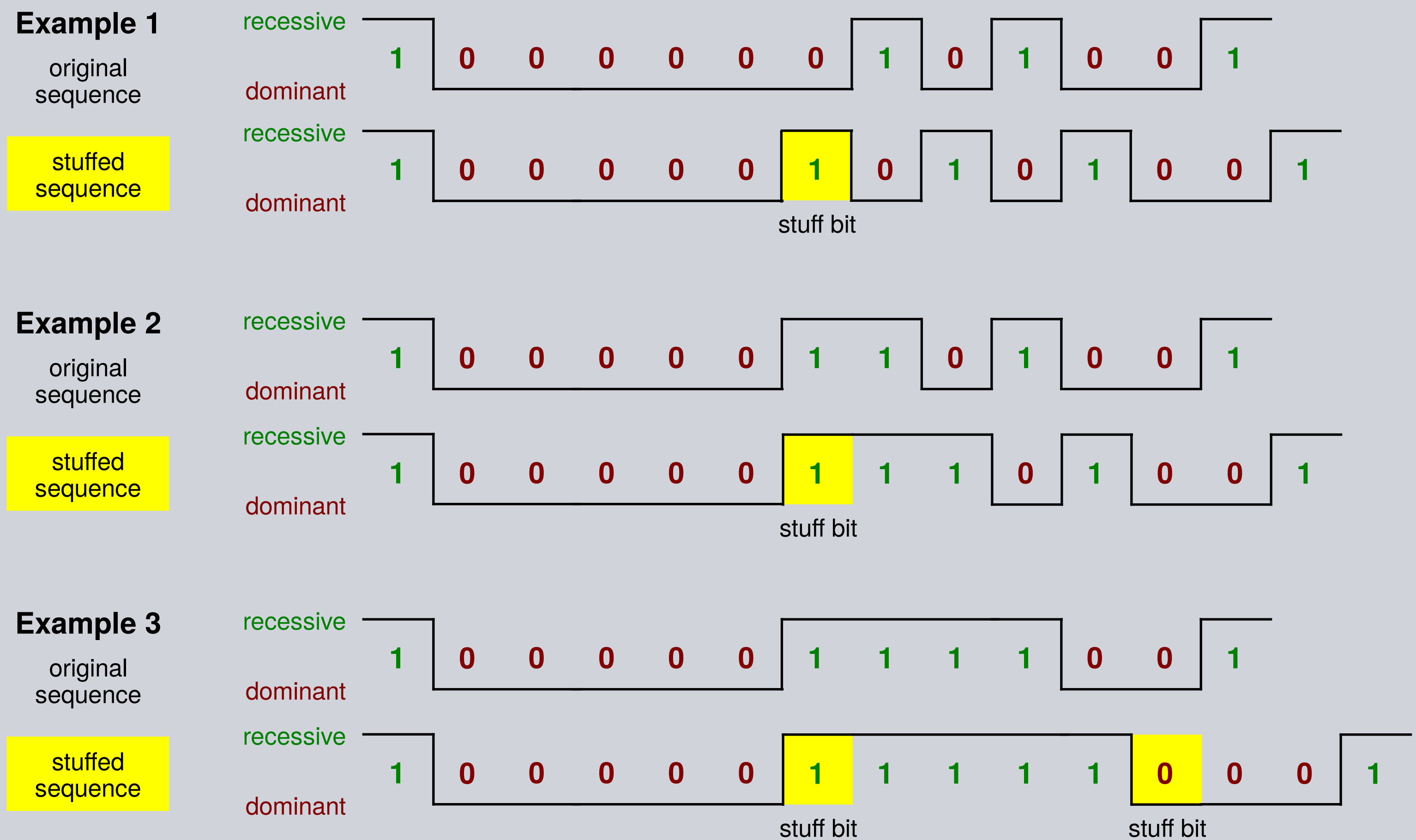
- long sequences of bits of the same polarity
- no changes in voltage level for a longer time
- no falling edges for synchronization
- synchronization between sender and receiver may be lost

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

SIEMENS VDO

Bit Stuffing: example

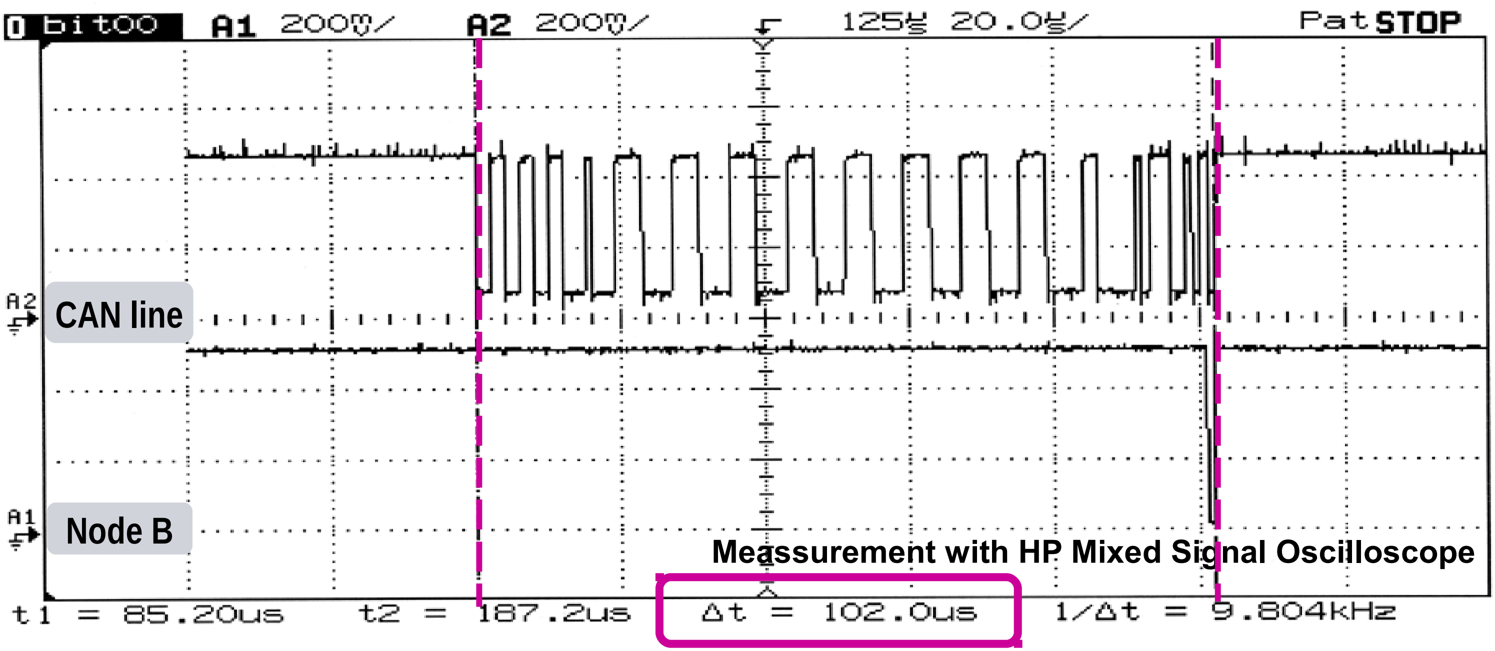


Bit Stuffing: example

The contents of a frame - address, data, control, CRC bits - defines the absolute bit count.

Transmission of a frame with 8 byte data (F0 F0 F0 F0 F0 F0 F0 F0)

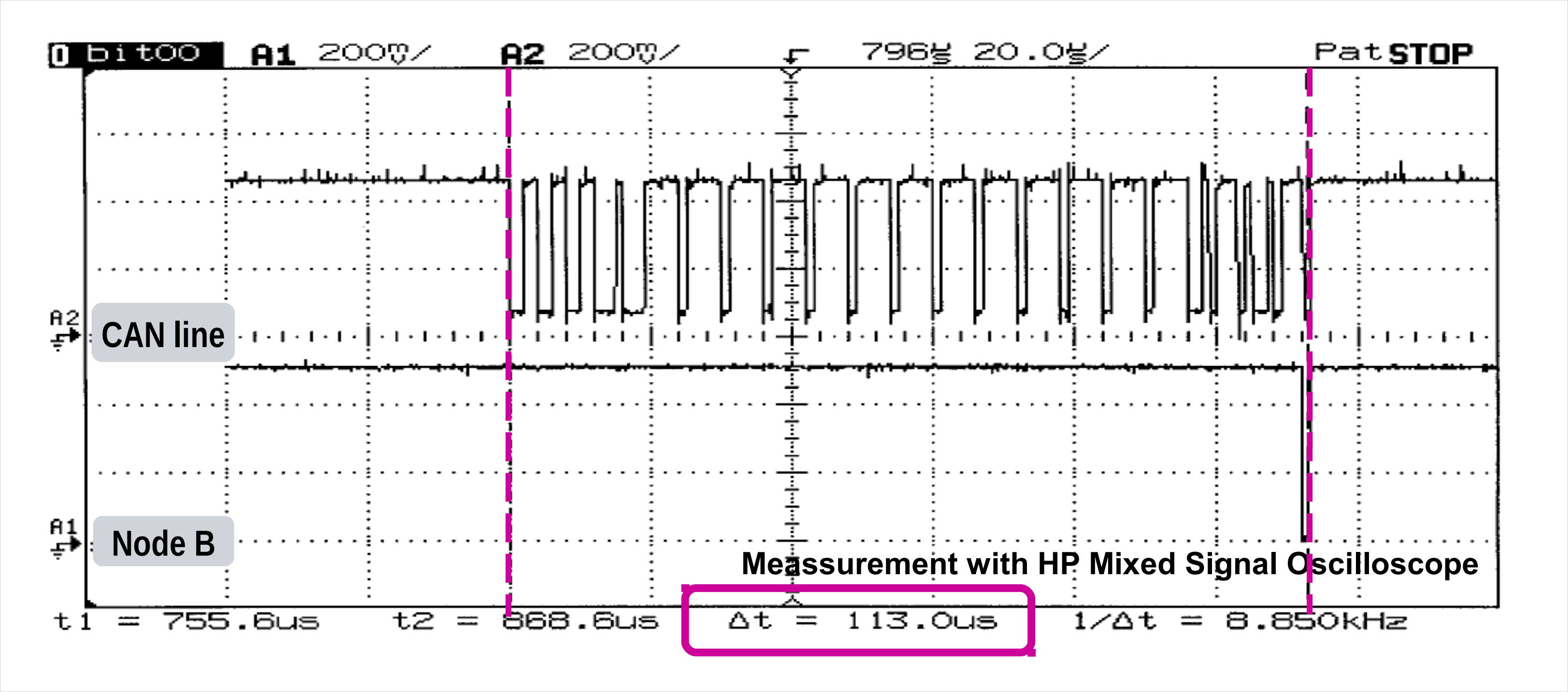
and address 333H. Absolute bit count: **102 Bits**.



Bit Stuffing: example

Transmission of a frame with 8 byte data (FF FF FF FF FF FF FF FF)

and address 333H. Absolute bit count: **113 Bits** (11 additional stuff bits).

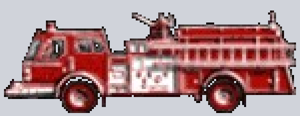


Data Frame Format : maximum size

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

Error Management



Fault Types in a CAN Bus System

There are different types of faults in a CAN bus system:

- **Global Faults**: - **lead to a breakdown of the system wide communication (e.g. short circuits or interrupts of the line, deadlock of the bus due to a permanent transmitting node**
- **Local Faults**: - **are limited to malfunctions of a node ore node components (e.g. interrupts in a bus connector, transmit problems of a transmitter, etc.)**
- **Temporary Faults**: - **disturb the CAN bus system for a limited time**
- **Permanent Faults**: - **disturb the CAN bus system all the time, they are not reversible**

Following error types are detected:

- **Bit error** — **transmitted and received bit are different**
  (except in arbitration, acknowledgement and passive error)
- **Bit stuffing error** — **more than five bits of equal polarity inside of a frame are detected**
- **CRC error** — **receive CRC code doesn't match with the calculated code**
- **ACK error** — **transmitting node receives no dominant acknowledgement bit**
  (no receiver accepts the transmission message)
- **Form error** — **fixed-form bit field contains one ore more illegal bits**
  e. g. violation of end of frame EOF format, CRC- or ACK-delimiter

Error type: Bit error

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

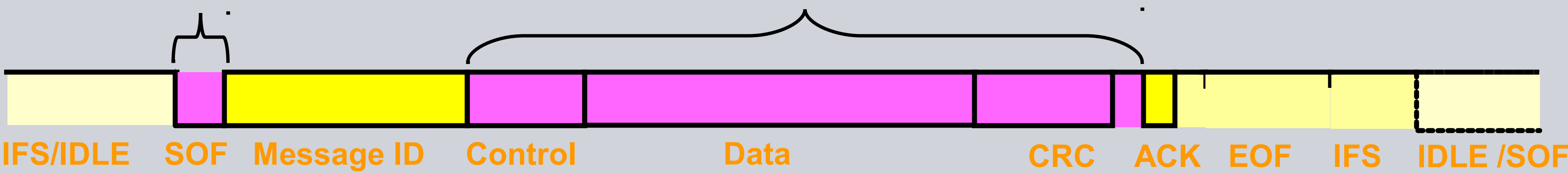
Exceptions

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

SIEMENS VDO

Error type: Bit error

Bit Error Monitoring Area



Error type: **Bit stuffing error**

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 not apply to this section

Error type: Bit stuffing error

SIEMENS VDO

Error type: CRC error

Error description

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

Method of detection

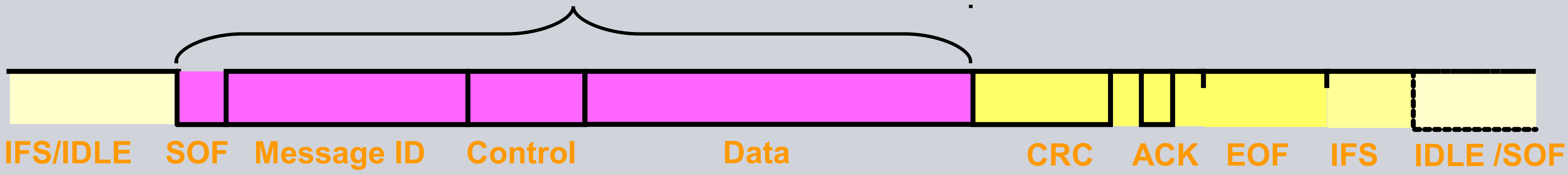
- Receiver calculates CRC code immediately after reception of the Data Field
- Receiver compares calculated CRC code with the one contained in frame

Efficiency

- Recognizes up to 5 single bit errors per frame $\rightarrow$ Hamming distance: 6
- Recognizes burst errors with lengths of up to 14 bits
- Recognizes all odd numbers of bit errors
- The more bits the CRC code has, the more efficient it is

Error type: CRC error

CRC Error Monitoring Area



Error type: ACK error

Error description

- Acknowledge (ACK) bit is recessive

Method of detection

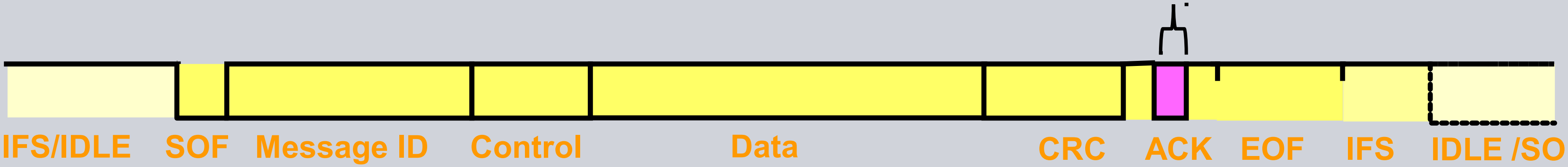
- Sender monitors the bus while transmitting recessive ACK bit
- Sender expects to observe dominant ACK bit on bus
- Acknowledgement error when ACK bit on bus remains recessive

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 type: ACK error

ACK Error Monitoring Area



Error type: Message format error

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

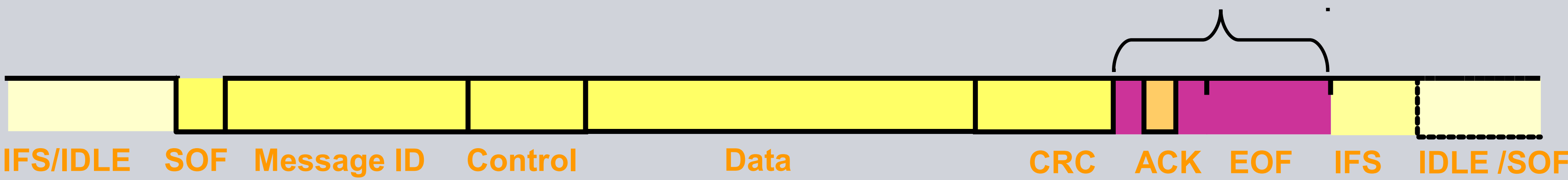
Possible cause

- 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)

SIEMENS VDO

Error type: Message format error

Form Error Monitoring Area

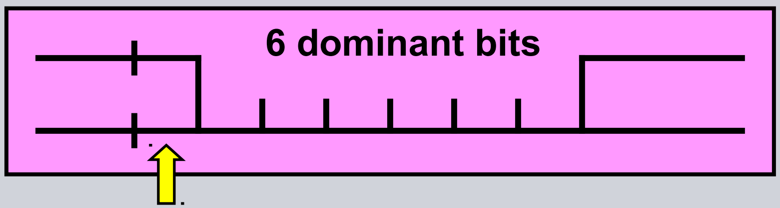


Error Handling

Steps of CAN Error Handling

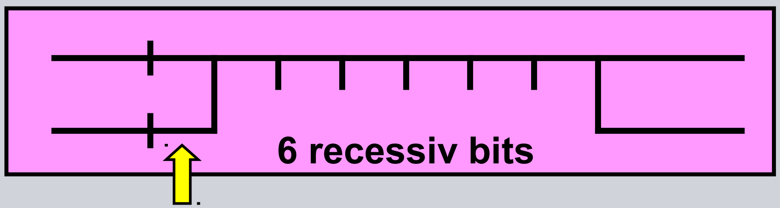
- **Each CAN node which detects an error - local or global error - sends an error flag to inform all other stations about this error (globalitization)**
- **An error flag is a queue of six consecutive bits**

Error active node :



Error condition

Error passive node :



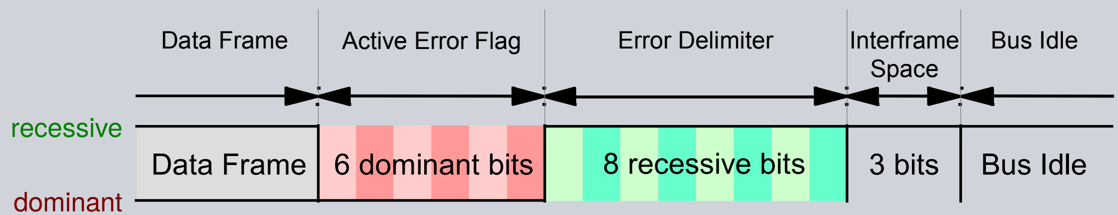
Error condition

- **In the case of one node detects a local error it will send an error flag. Other nodes may detect an error because of this error flag. They send also an error flag.**
- **After receiving/sending an error flag all nodes cancel the message.**
- **Each node which detects an error increments his error counter(s).**
- **The transmitter of a cancelled message retransmits automatically this message.**

Procedure observed after detection of an error (1)

- **Immediate transmission of an Error Frame**

Format of an Active Error Frame:

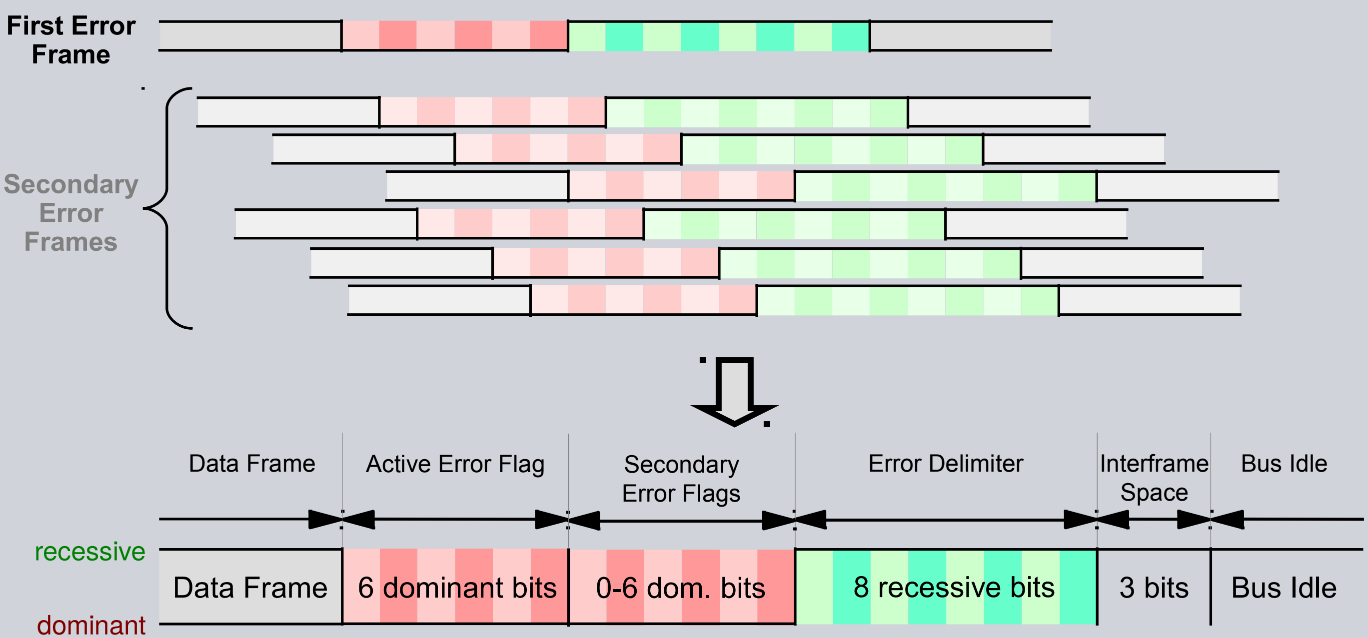


<u>No</u> bit stuffing is applied to Error Frames!

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

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:**



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 state

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.

Node state: 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 state: error passive

A unit is in error passive state when

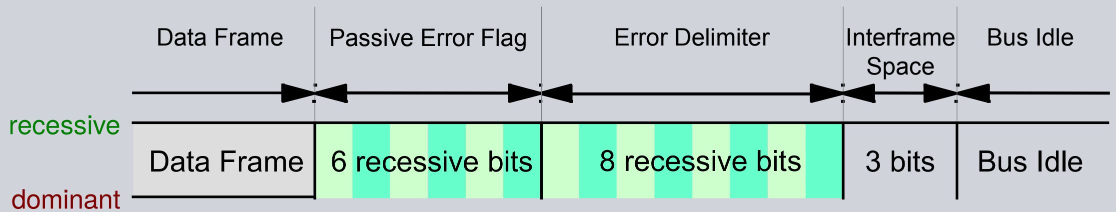
- its Transmit Error Counter (TEC) is greater than 127: TEC > 127 AND / OR
- its Receive Error Counter (REC) is greater than 127: REC > 127

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

Node state: error passive

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 state: bus off

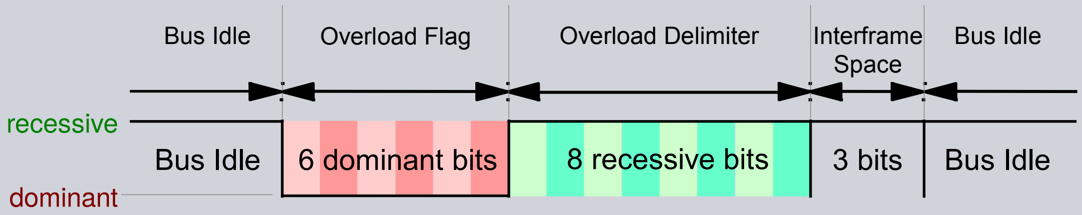
A unit is in bus off state when

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

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**
  ⇒ 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 not influence the error counters (TEC and REC)**

Efficiency of the error management

The probability for **not** discovering an error is

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

Example 1*

*Source: CiA

A CAN bus is used	365 days / year
	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**

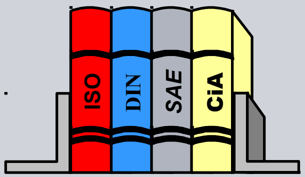
**Source: Kaiser, Schröder: "Maßnahmen zur Sicherung der Daten beim CAN-Bus"

A CAN bus in a car is run at	500 kBit / sec
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

SIEMENS VDO

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 communication”
ISO 11519	“Road vehicles: Low-speed serial data communication - 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)

Thank you!