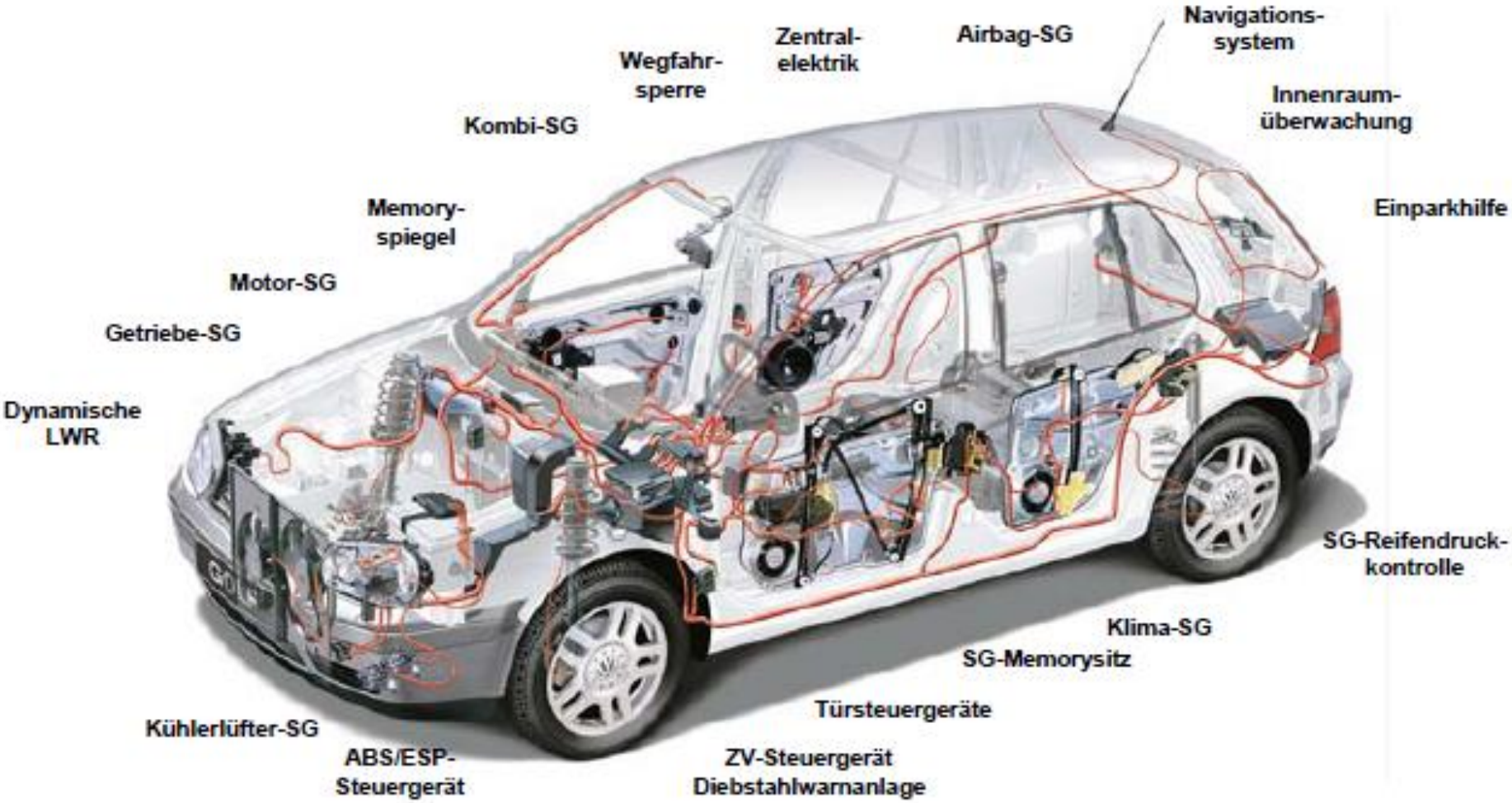
Inhalt

- CAN-Bus Übersicht
- Can-Bus im OSI Schichtenmodell
- OSI Schicht 1 (phys. Schicht)
- OSI Schicht 2 (Sicherungsschicht)
- CAN-Bus im STM32 Mikrocontroller
- CAN-FD
- CAN-Open (OSI Schicht 7)

CAN


Navigations-
system
Airbag-SG
Zentral-
elektrik
Wegfahr-
sperre
Innenraum-
überwachung
Kombi-SG
Einparkhilfe
Memory-
spiegel
Motor-SG
Getriebe-SG
Dynamische
LWR
SG-Reifendruck-
kontrolle
Klima-SG
SG-Memorysitz
Türsteuergeräte
Kühlerlüfter-SG
ABS/ESP-
Steuergerät
ZV-Steuergerät
Diebstahlwarnanlage

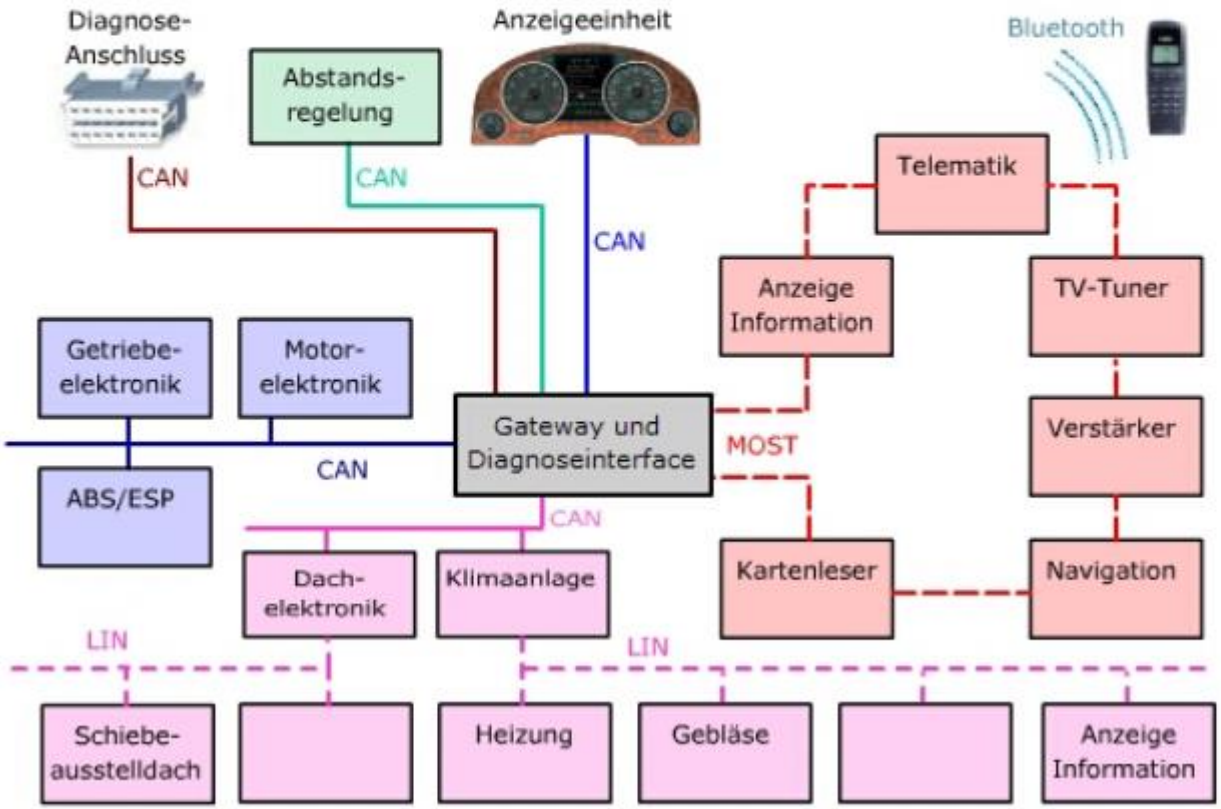
Eigenschaften CAN Bus

- Ereignisgesteuert
- CSMA/CR
- Inhaltsbezogene Nachrichtenbezeichnung
- Prioritätsgesteuerte Nachrichten
- Multimaster
- Synchronisationsmöglichkeit
- Fehlererkennungsmöglichkeit
- Broadcast
- Nachrichtenwiederholung
- Fehlertoleranz

Bussysteme im KFZ

Aspekt	CAN	LIN	FlexRay
Steuerung	Ereignis	Zeit	Zeit
Zeitverhalten	Nicht deterministisch	Deterministisch	Deterministisch
Architektur	Multi-Master	Master-Slave	Multi-Master
Buszugriff	Zufällig (CSMA/CA)	Kontrolliert (Delegated Token)	Kontrolliert (TDMA)
Adressierung	Nachrichten	Nachrichten	Nachrichten
Maximale Datenrate	1 Mbit/s	20 Kbit/s	20 Mbit/s
Maximale Anzahl Datenbyte/Frame	8	8	254
Medium	Twisted Pair	Single Wire	Twisted Pair, Plastic Optical Fiber
Topologie	Linie, Stern	Linie	Linie, Stern

CAN-Bus Gateway

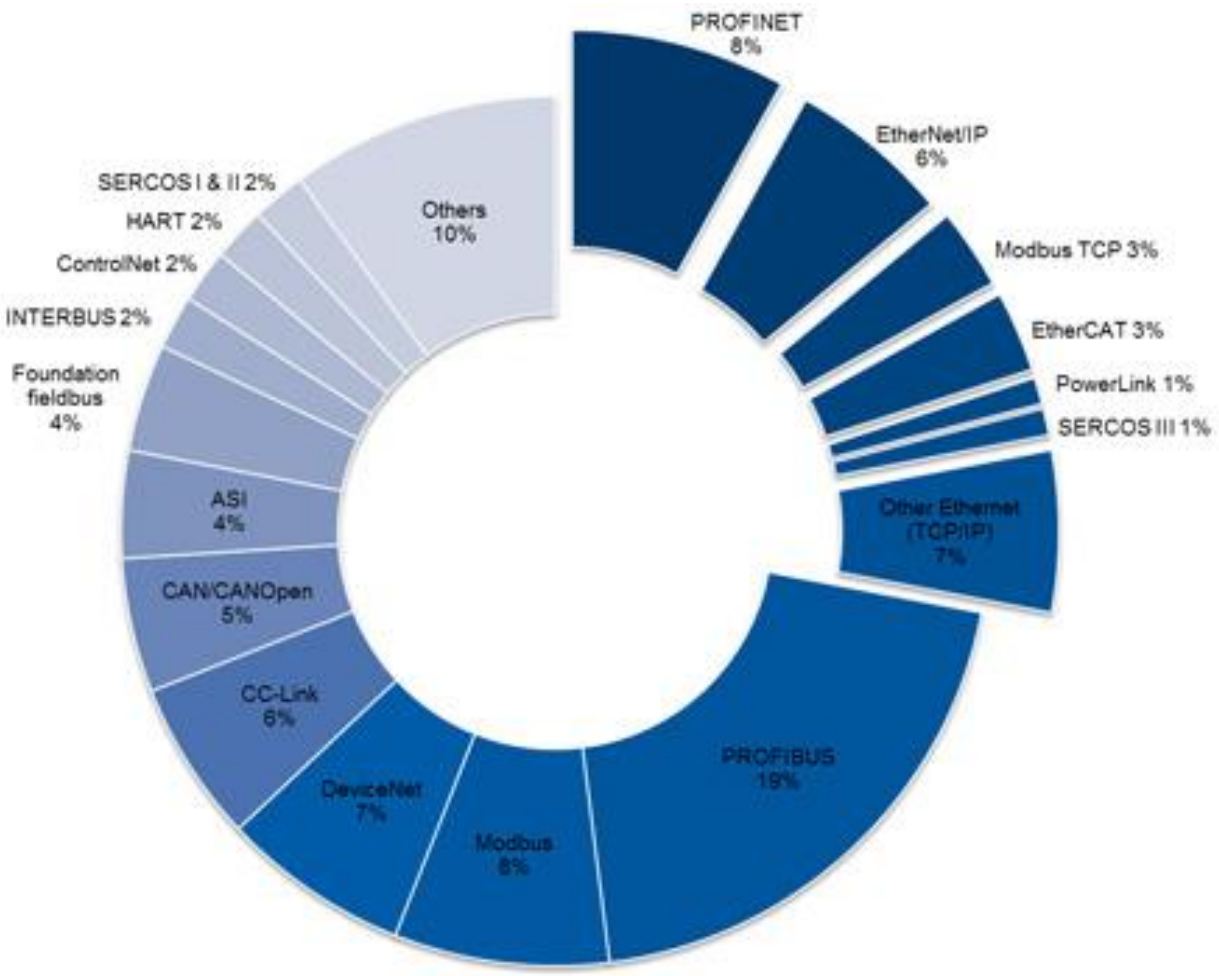


CAN-Bus	Varianten. In Abhängigkeit von der Geschwindigkeit, mit der er Daten übertragen kann, wird er in den Bereichen Antrieb, Komfort und zur Diagnose eingesetzt. Low Speed bis High Speed, 100 bis 500 kBit/s incl. Diagnose CAN
LIN	Der **LIN-Bus** (Local Interconnect Network) ist ein preiswertes dem CAN-Bus untergeordnetes Sub-Bussystem; er wird für die Ansteuerung bestimmter lokaler Komfort-Komponenten oder für Sensoren verwendet. - max. 20 kBit/s
MOST	(Media Oriented Systems Transport) optisches Hochgeschwindigkeits-Multimedia-Bussystem, meist ringförmig, Sehr hohe Übertragungsrate (23 MBit/s)

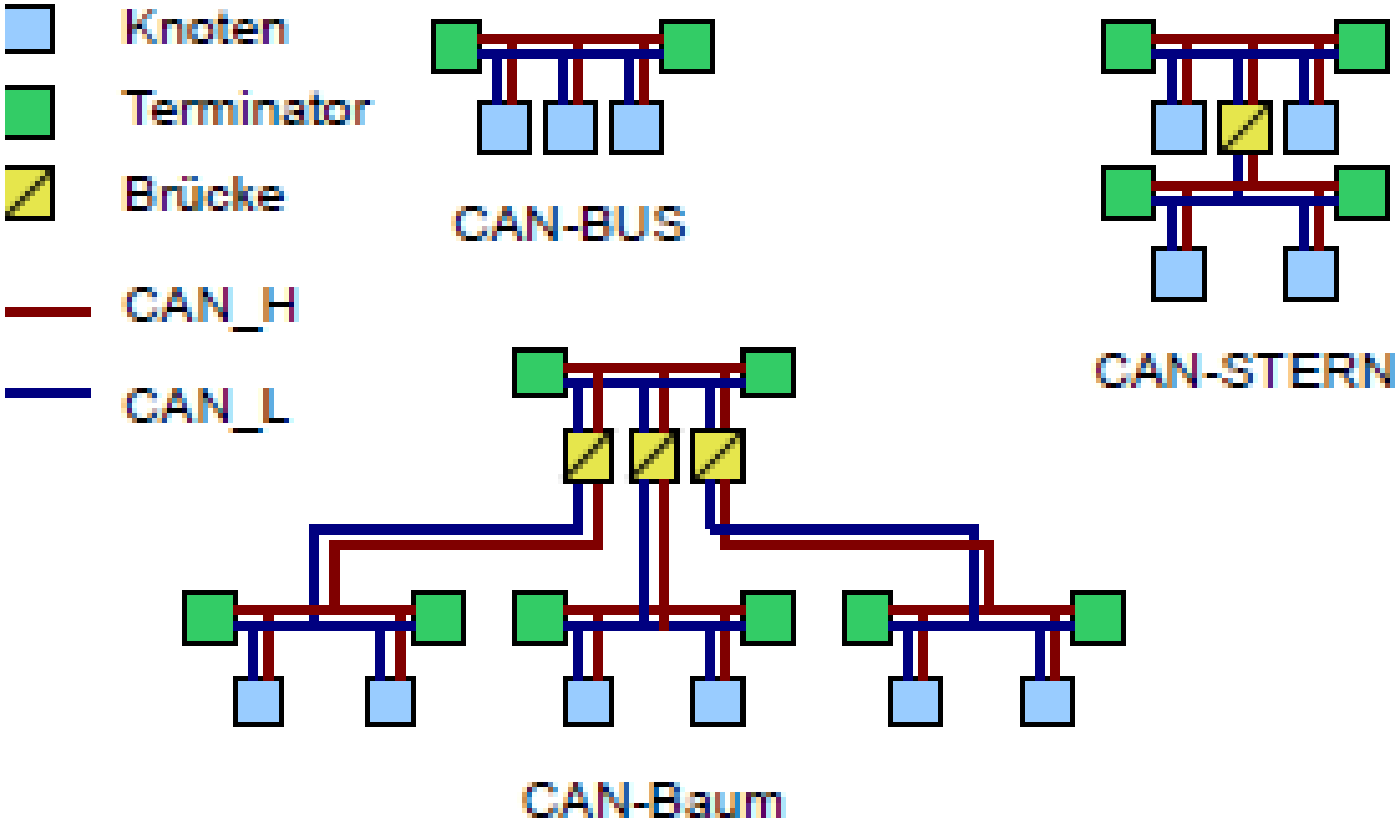
CAN-Bus Verbreitung

- ✓ Automobilindustrie
- ✓ Verkehrstechnik
- ✓ Automatisierungstechnik
- ✓ Gebäudeleittechnik
- ✓ Fernsehen
- ✓ Medizintechnik
- ✓ Landwirtschaft
- ✓ Schifffahrt
- ✓ und Kaffeemaschinen!!

Feldbusse-Marktanalyse



CAN Topologie

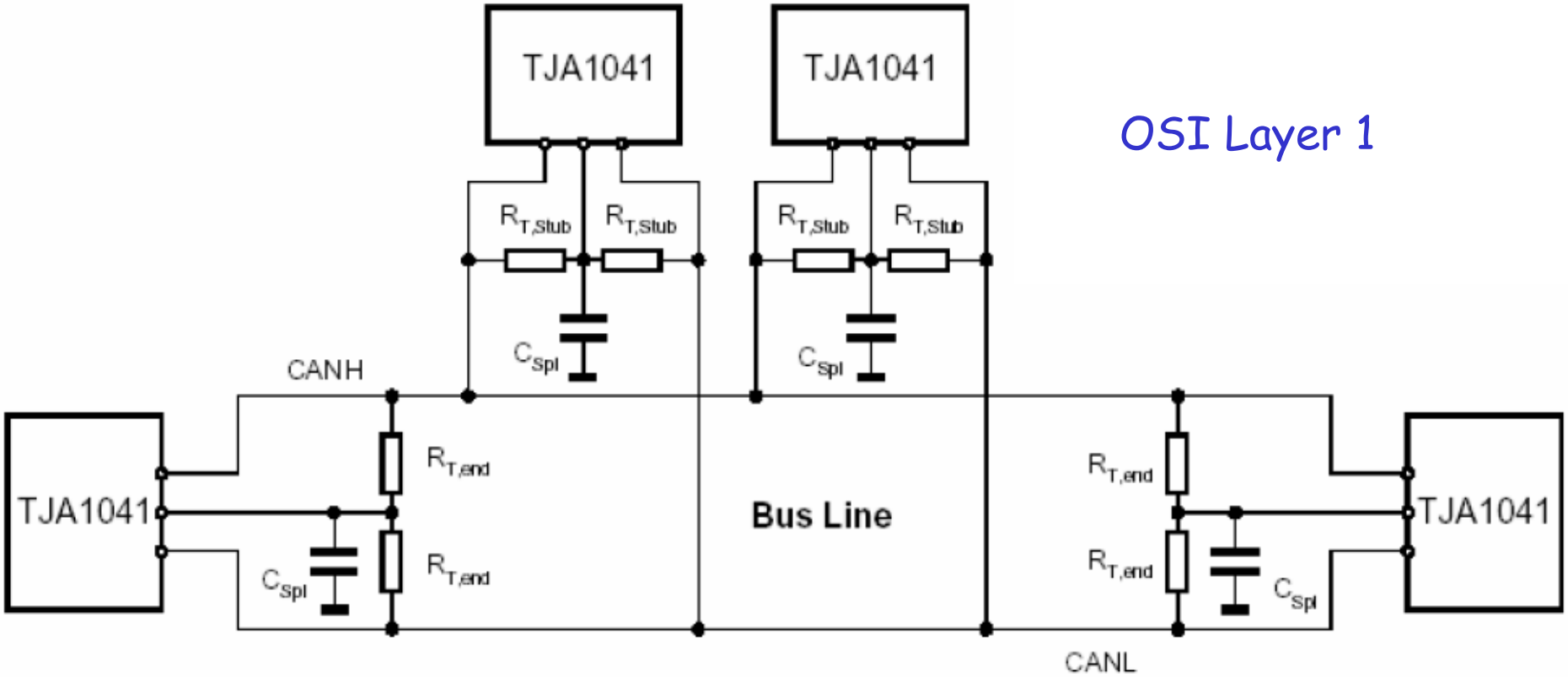


OSI Layer 1

CAN

- Für CAN werden lineare Busstrukturen empfohlen. Das beste Signalverhalten ergibt sich bei Abschluss der Busleitungen mit 120Ω an jedem Ende. Die Länge soll 40m bei 1Mbit/s nicht überschreiten.
- Kurze Zweigleitungen (Stubs) verzerren grundsätzlich die Signalform. Ihr Einfluss kann durch hochohmige Abschlusswiderstände (1,5kΩ) verbessert werden. Sie sind auf 30cm Länge zu begrenzen.
- Zur Verbesserung des AM-Verhaltens wird der Abschlusswiderstand geteilt und der Mittelpunkt über einen Kondensator an Masse gelegt.

OSI Layer 1



Quelle: Philips semiconductors

CAN

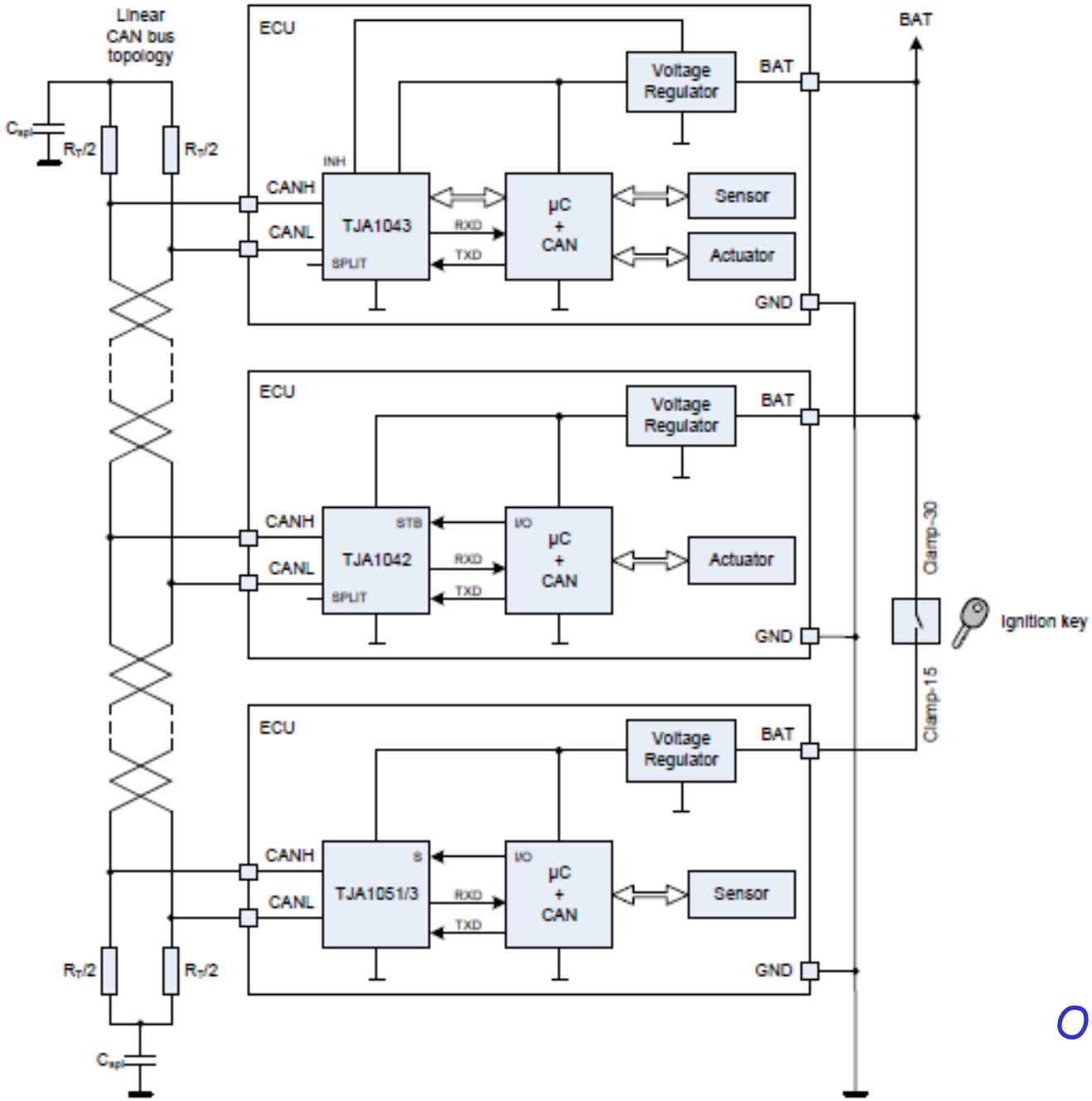


Fig 9. High speed CAN application example

Typische Topologie im KFZ

OSI Layer 1

CAN

Steckerbelegung

PIN	Signal	Beschreibung
1	-	(reserviert)
2	CAN_L	Signal CAN_L
3	CAN_GND	Signal GND
4	-	(reserviert)
5	-	(reserviert)
6	CAN_SHLD	Signalabschirmung
7	CAN_H	Signal CAN_H
8	-	(reserviert)
9	CAN_V+	Optionales Power Supply V+

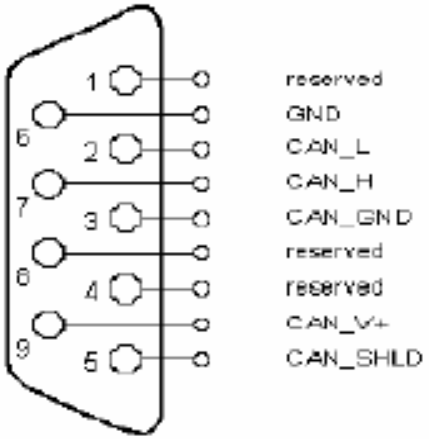


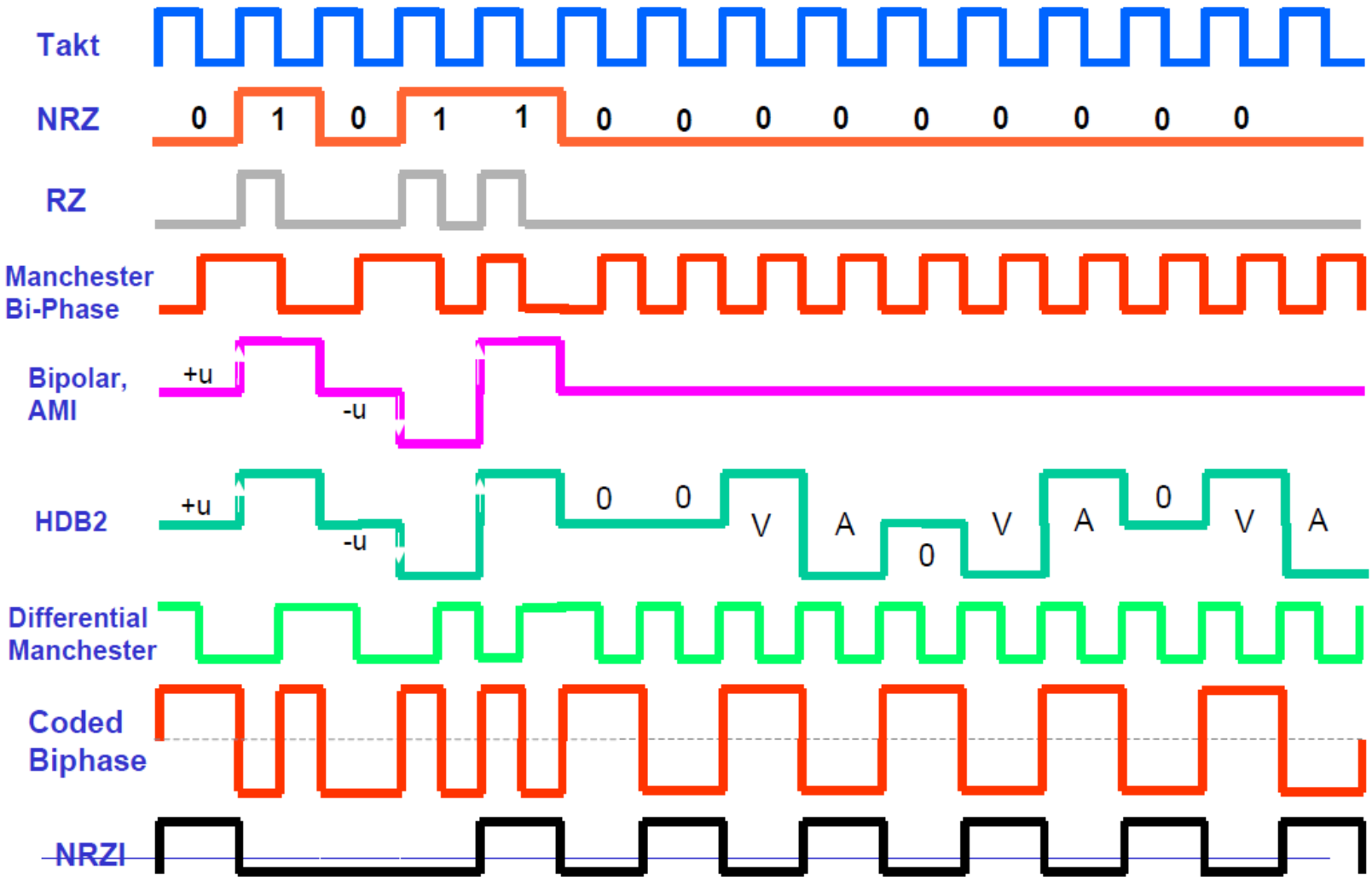
Abbildung 7: Pinbelegung des CAN-Steckers

Tabelle 1: Pinbelegung gemäß CiA Draft Standard 102 V2.0

OSI Layer 1

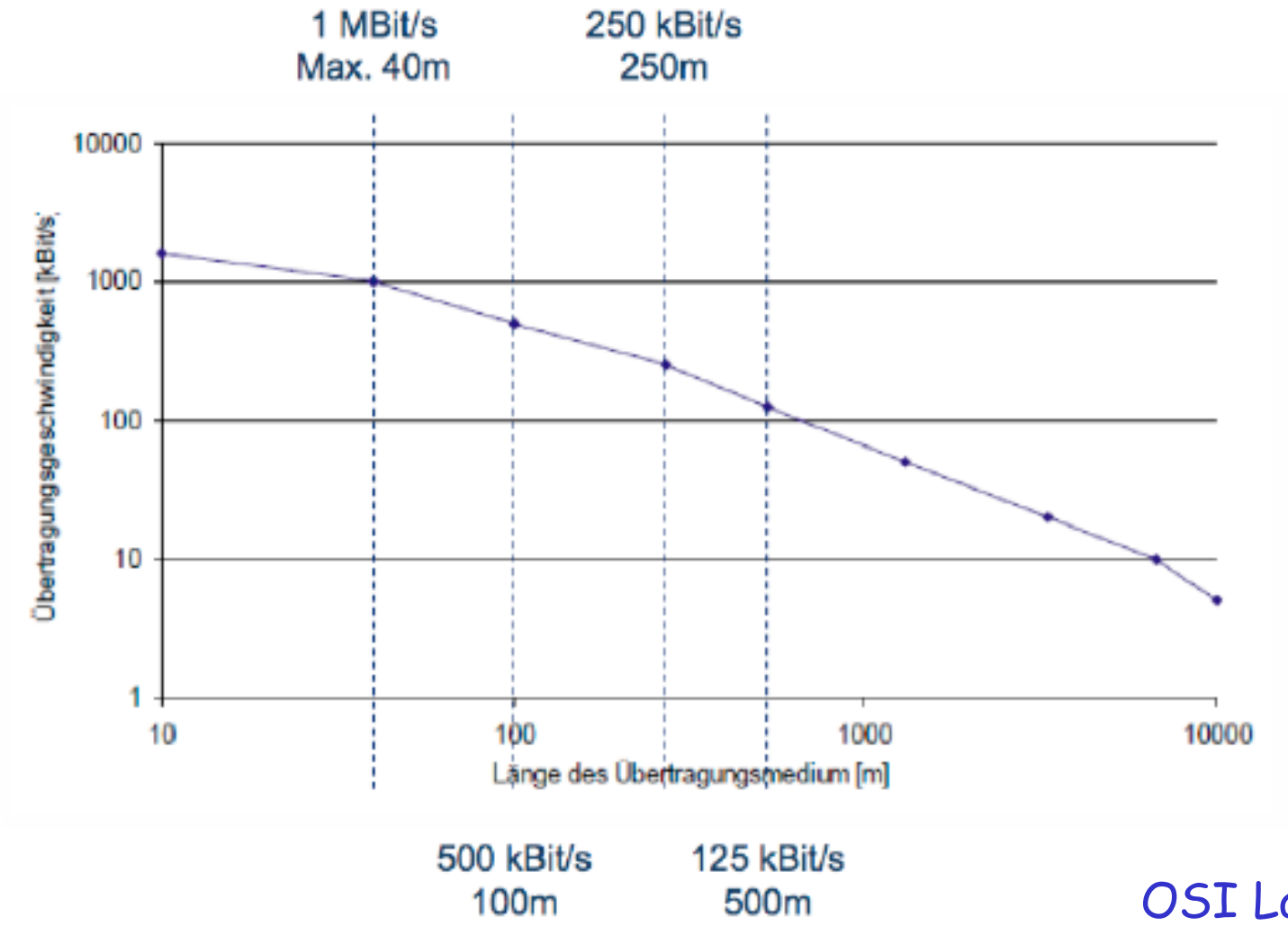
CAN Bitübertragung

Fakultät für Feinwerk- und Mikrotechnik, Physikalische Technik
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN



OSI Layer 1

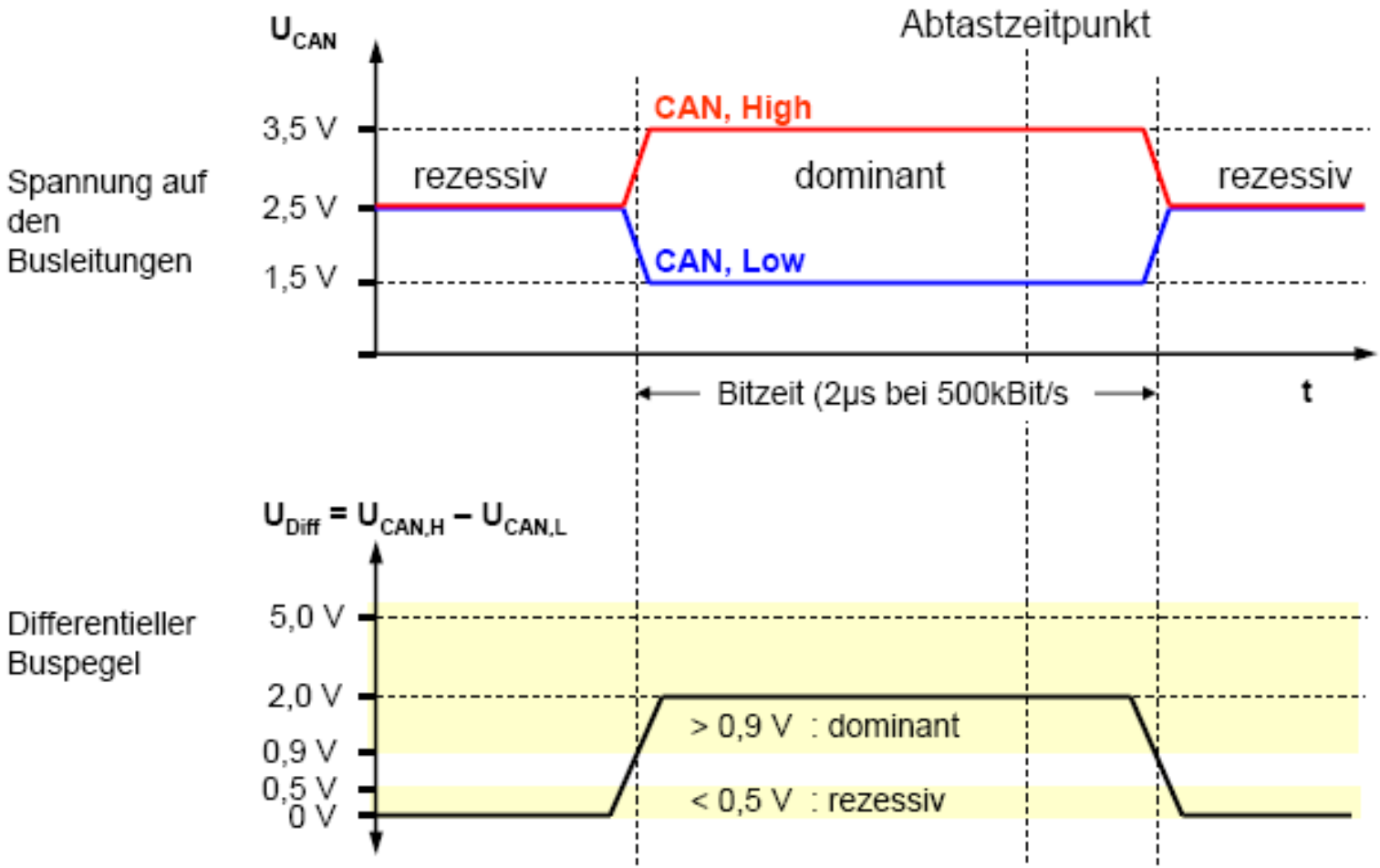
CAN - Datenrate



OSI Layer 1

CAN

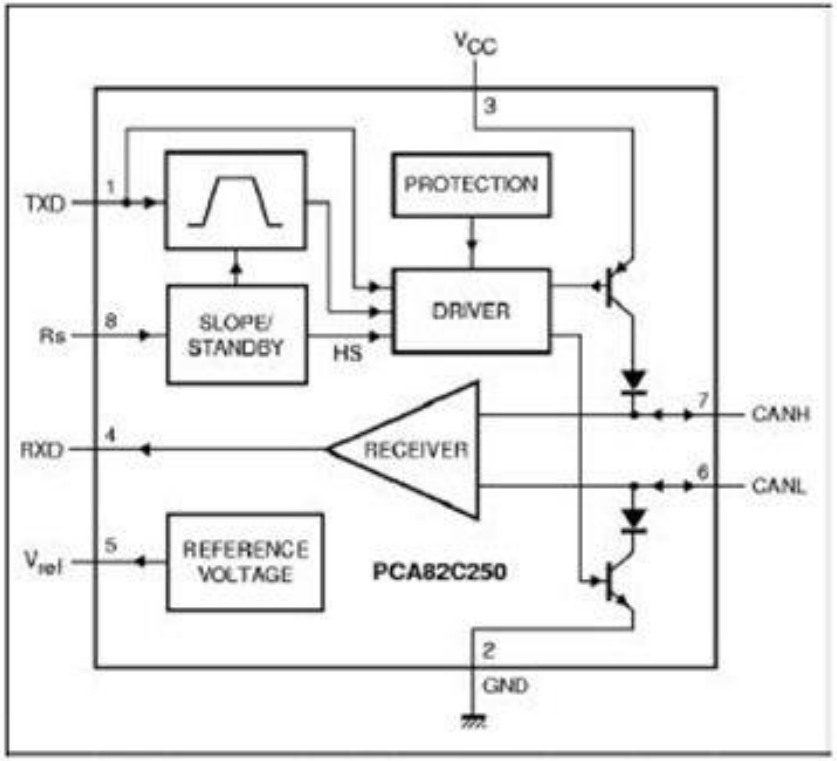
Bus-Pegel (High-Speed) ISO11898



OSI Layer 1

CAN

High-Speed Transceiver

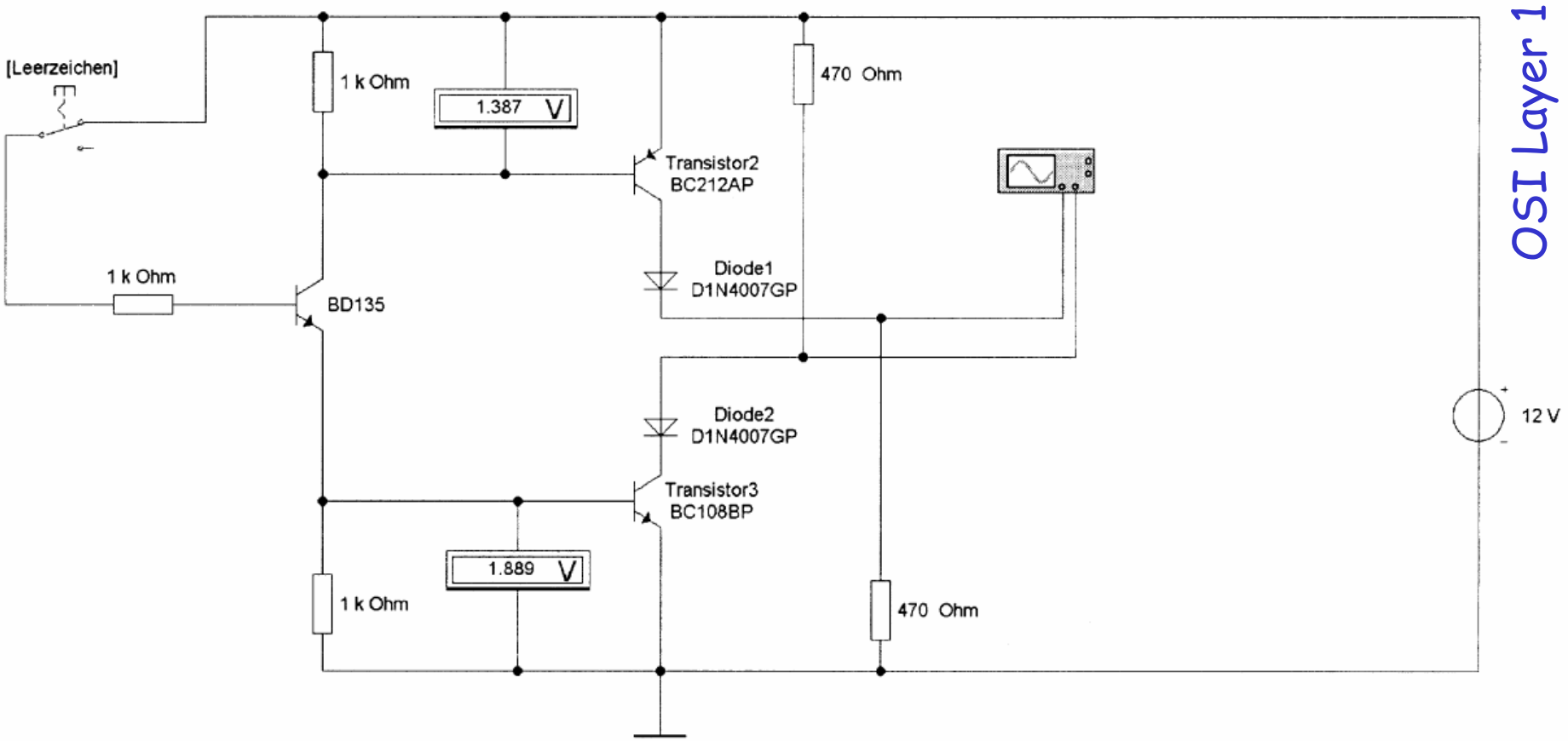


logischer Pegel	Zustand	CAN-H	CAN-L	Differenzpegel
0	dominant	Transistor durchgeschaltet und zieht Pegel auf VCC	Transistor durchgeschaltet und zieht Pegel auf GND	> 1,5 Volt
1 (hochohmig)	rezessiv	Transistor sperrt	Transistor sperrt	0 Volt

OSI Layer 1

CAN

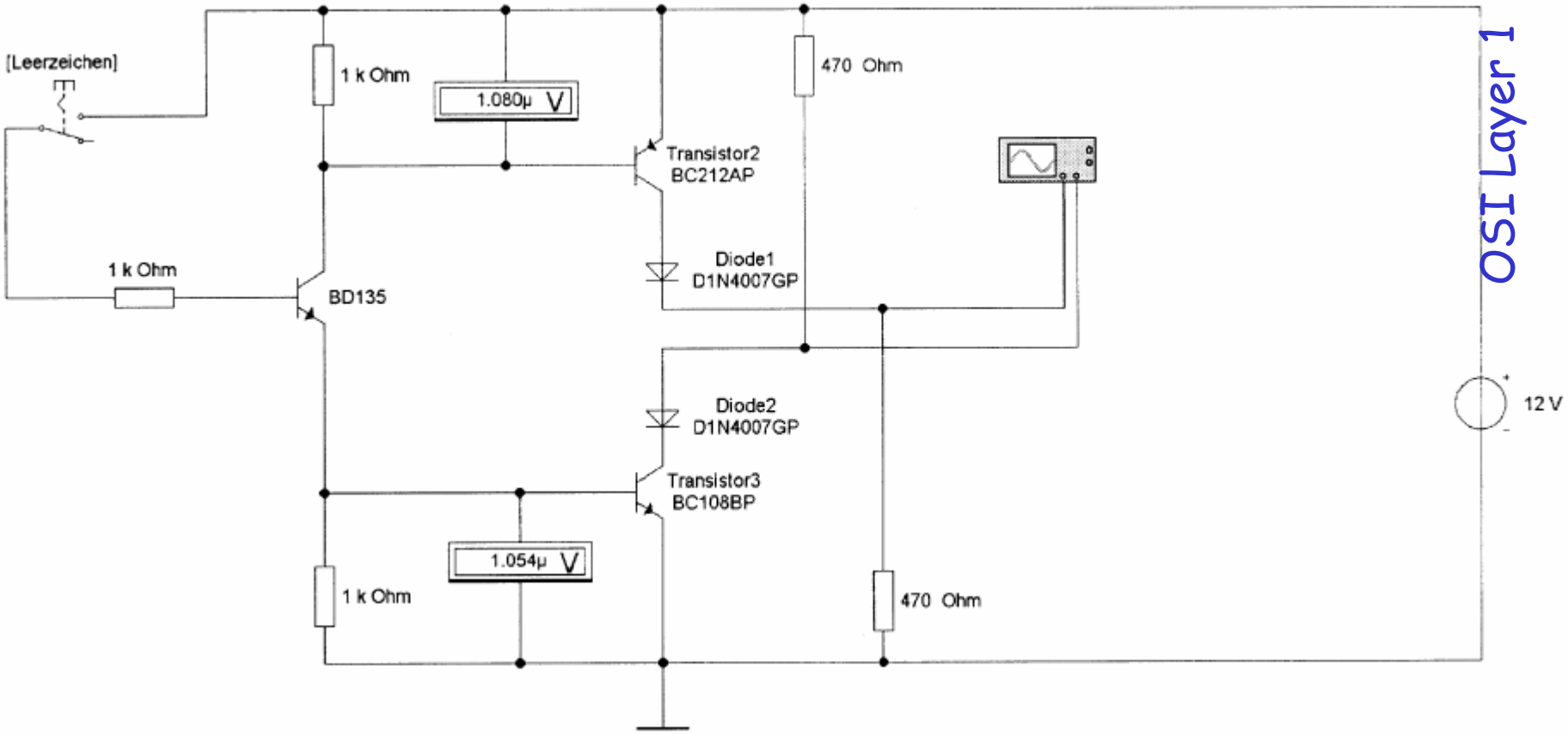
Simulation Pegel-erzeugung(1)



OSI Layer 1

CAN

Simulation Pegel-erzeugung(2)



http://www.nibis.de/nli1/bbs-cms/Berufsbereiche/Fahrzeugtechnik/Unterrichtsmaterialien/Kooperation/BeleuchtungGolfV/PDF/4.1-Signalerzeugung-01-loe.pdf

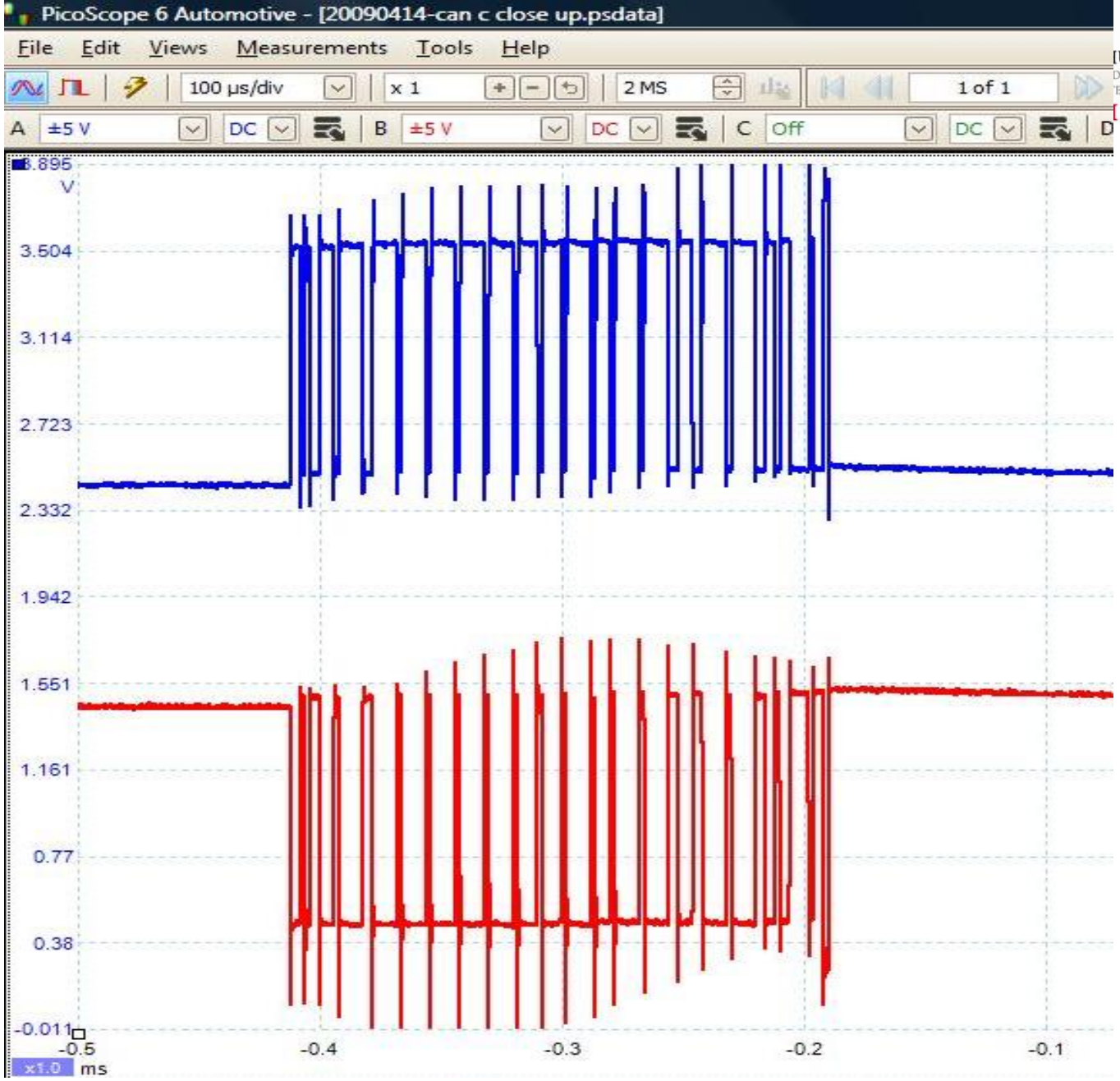
CAN

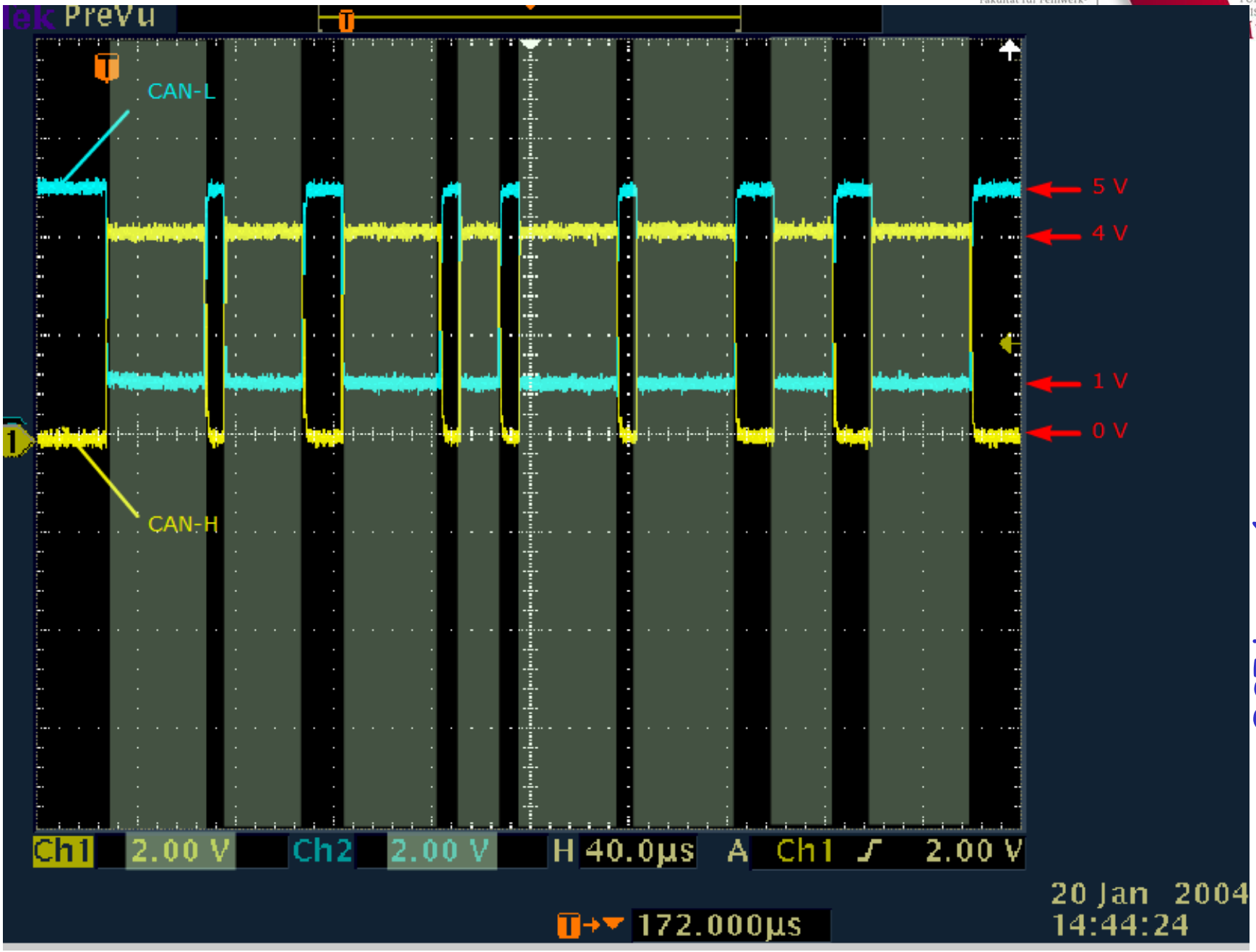
Screenshot

CAN-High

OSI Layer 1

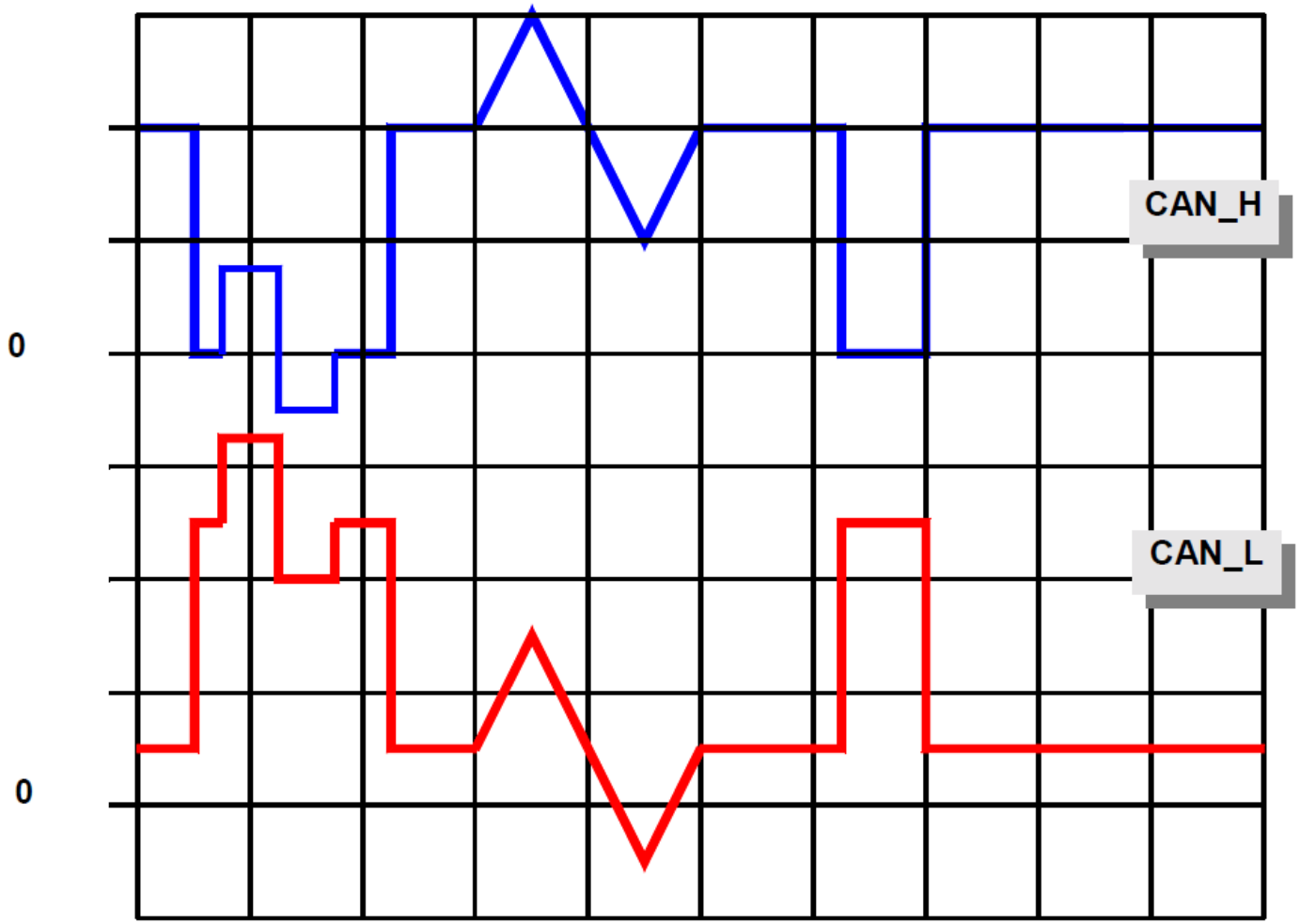
CAN-Low




Tek PreVu
CAN-L
CAN-H
5 V
4 V
1 V
0 V
Ch1 2.00 V
Ch2 2.00 V
H 40.0µs
A Ch1 2.00 V
172.000µs
20 Jan 2004
14:44:24

OSI Layer 1

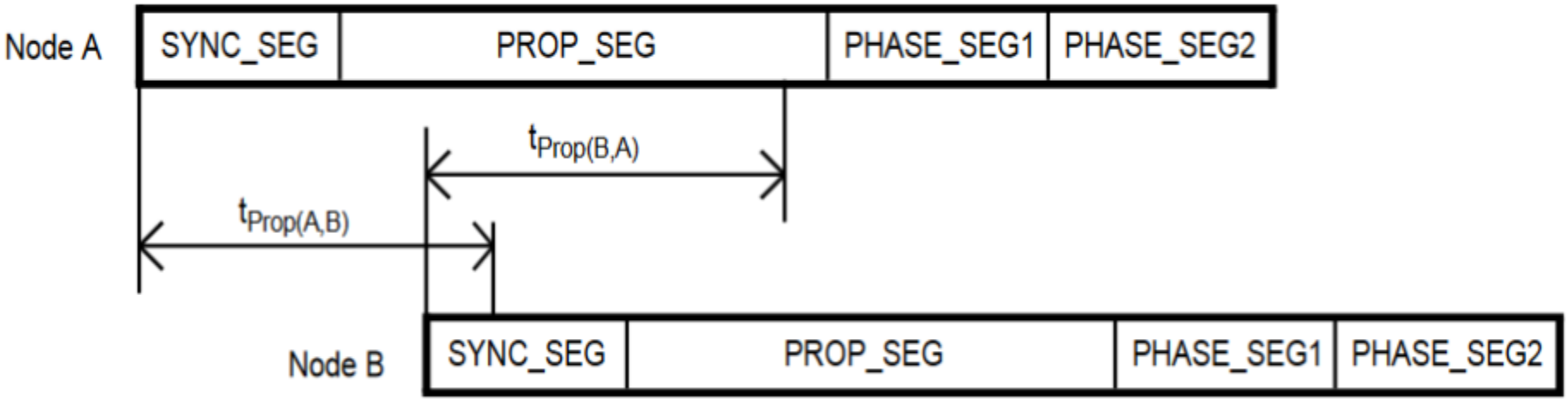
CAN



OSI Layer 1

CAN – Bit Timing

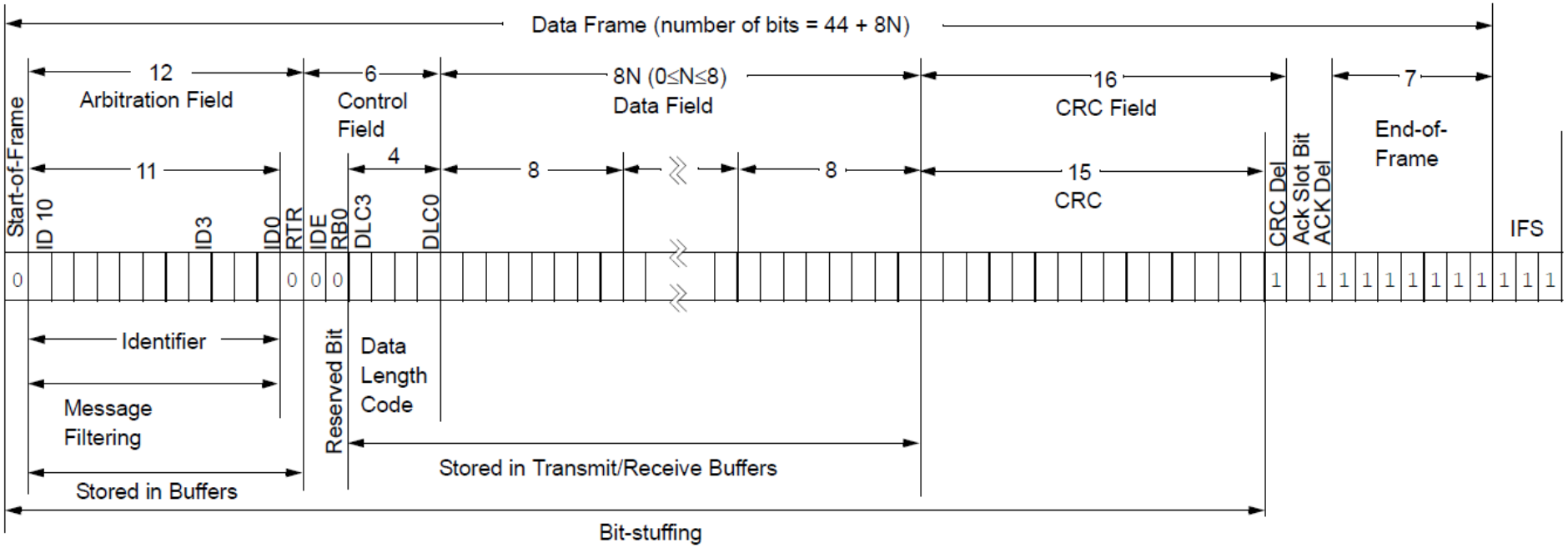
OSI Layer 1



CAN

Datentelegramm CAN 2.0A

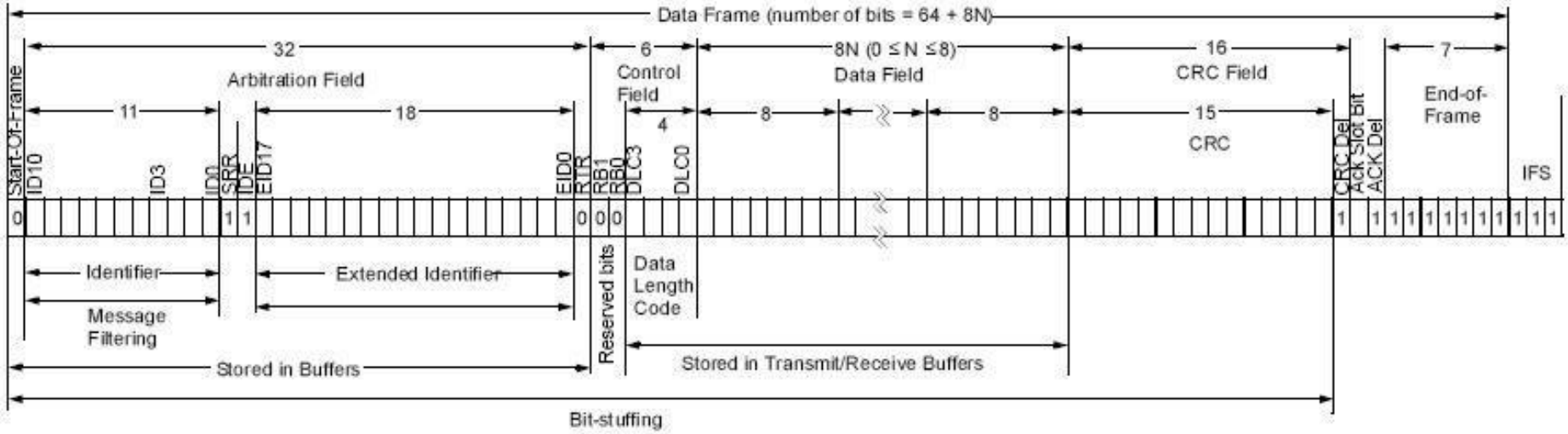
OSI Layer 2



CAN

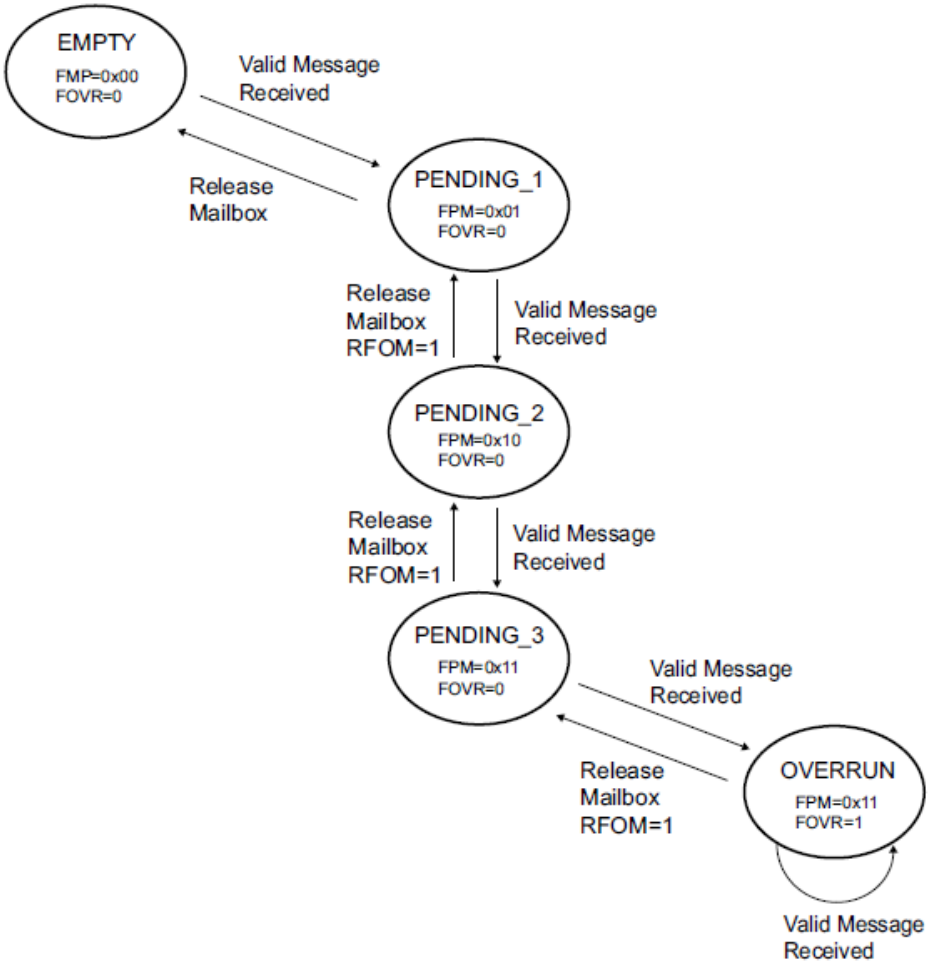
Datentelegramm CAN 2.0B

OSI Layer 2



CAN

OSI Layer 2



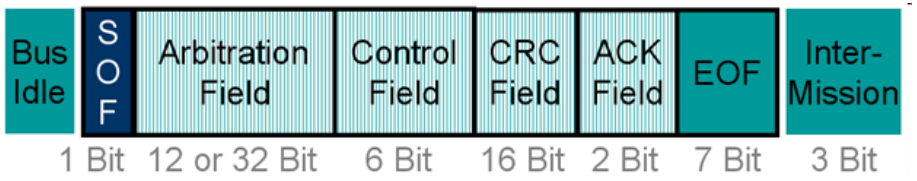
Overload Frame

CAN

Remote Frame

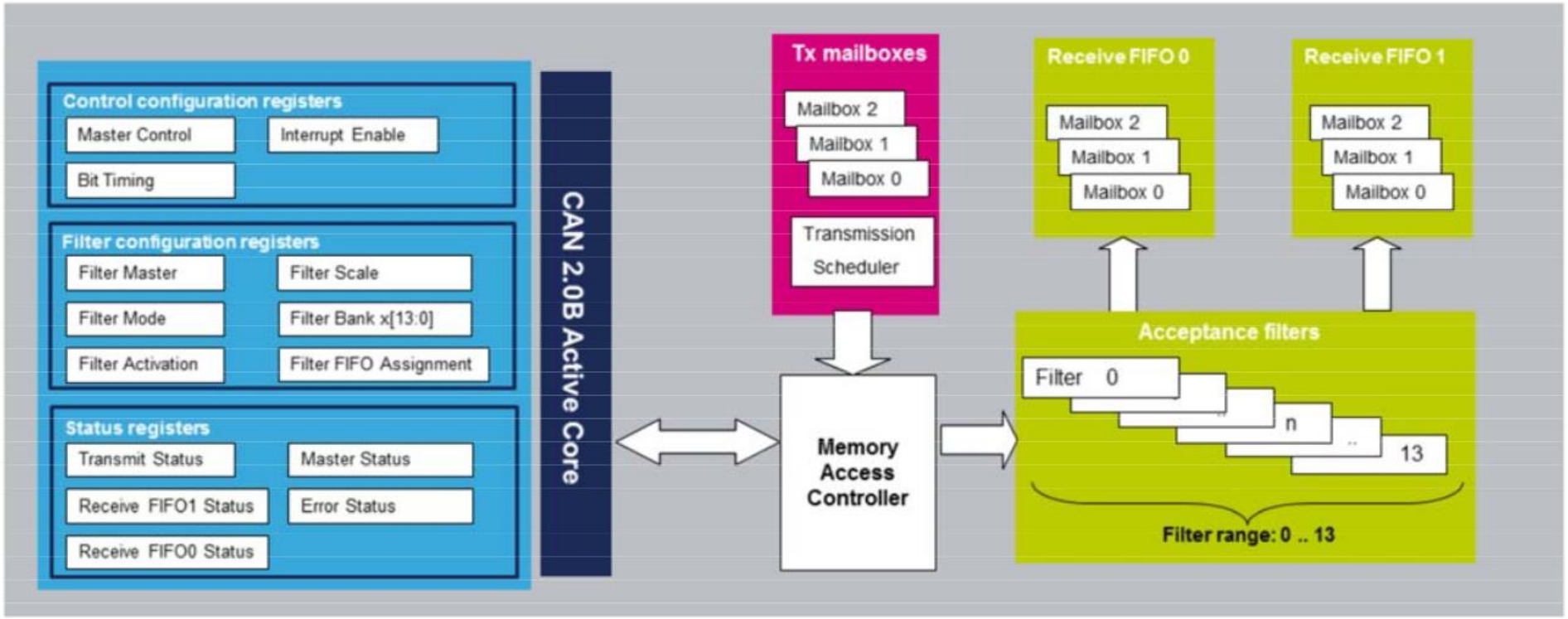
- Producer schickt Frame mit RTR-Bit = rezessiv an einen Consumer mit entsprechendem Identifier bzw. IDMask
- Der oder die Consumer mit der entsprechenden Identifier/IDMask Kombination antworten mit einem Daten-Frame

OSI Layer 2



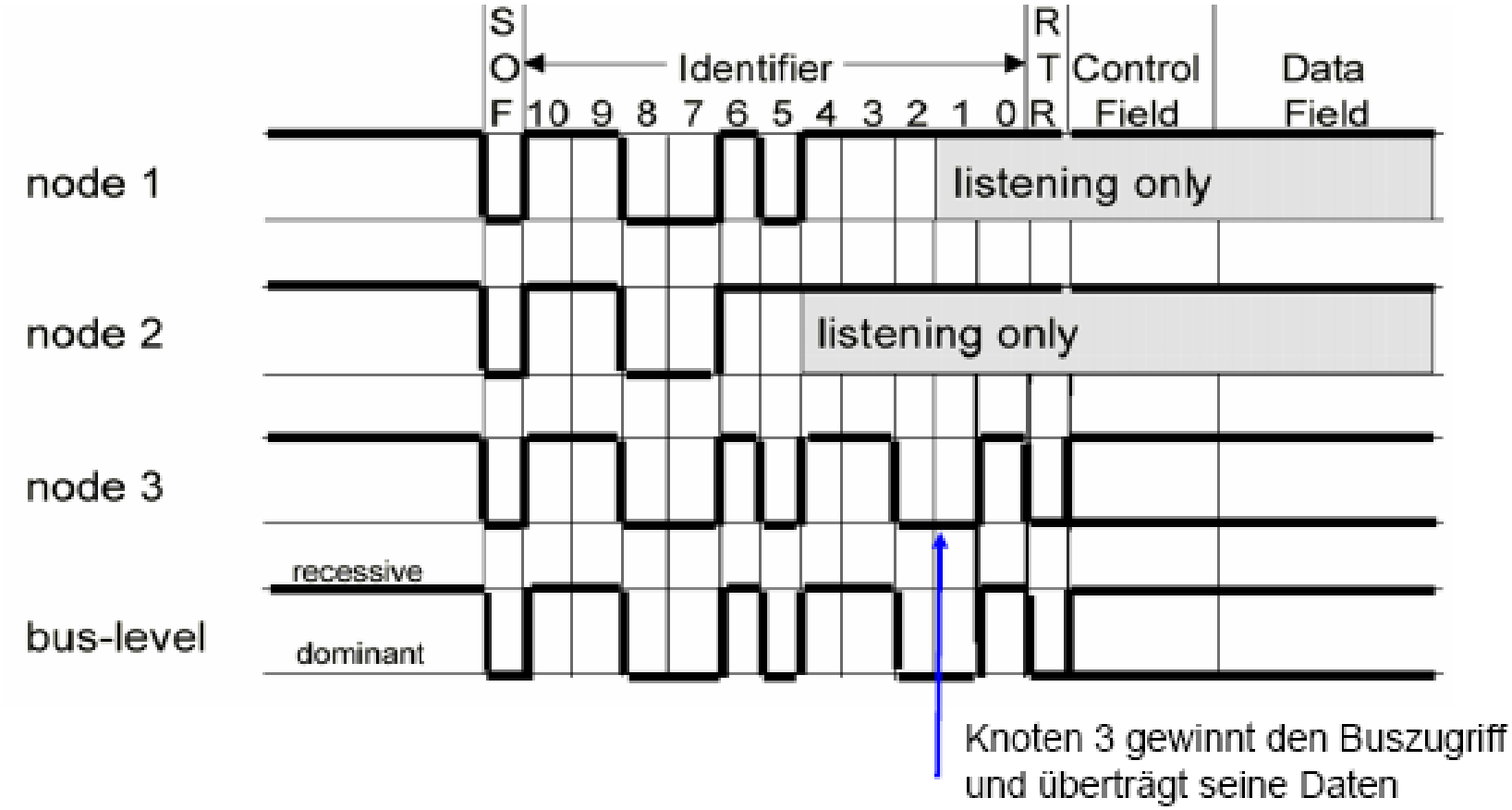
CAN

Akzeptanzfilterung



CAN

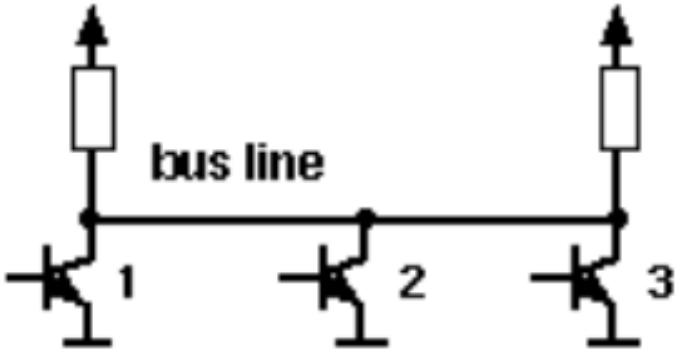
Arbitrierung Zugriffskontrolle

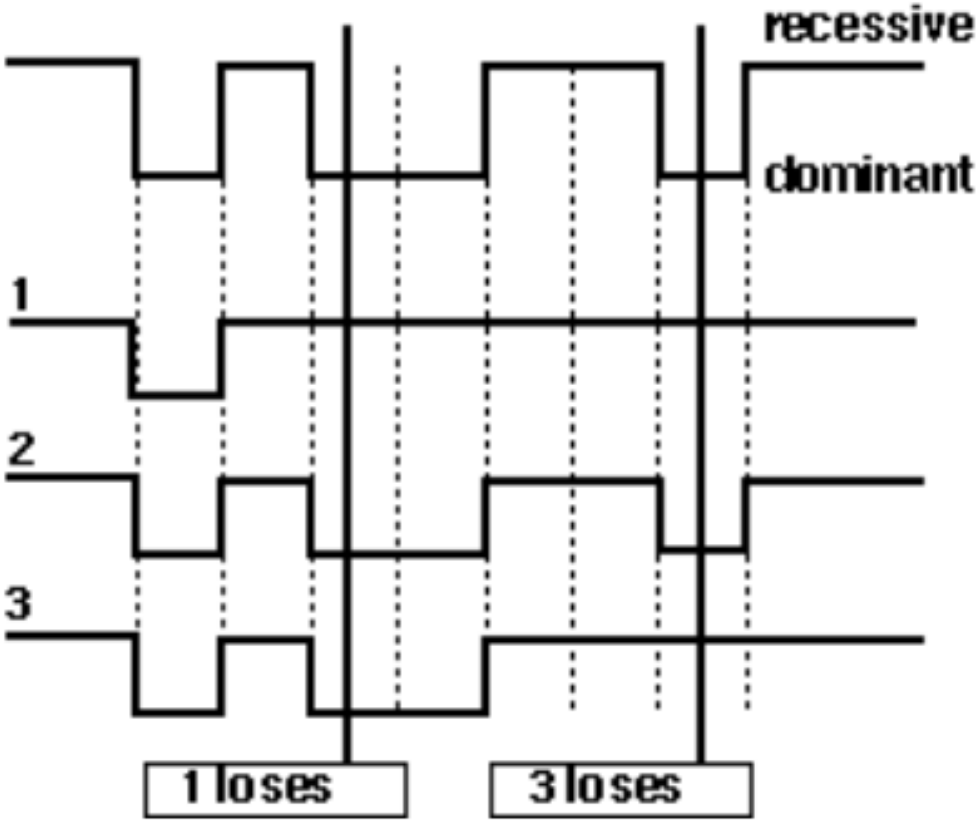


OSI Layer 2

CAN

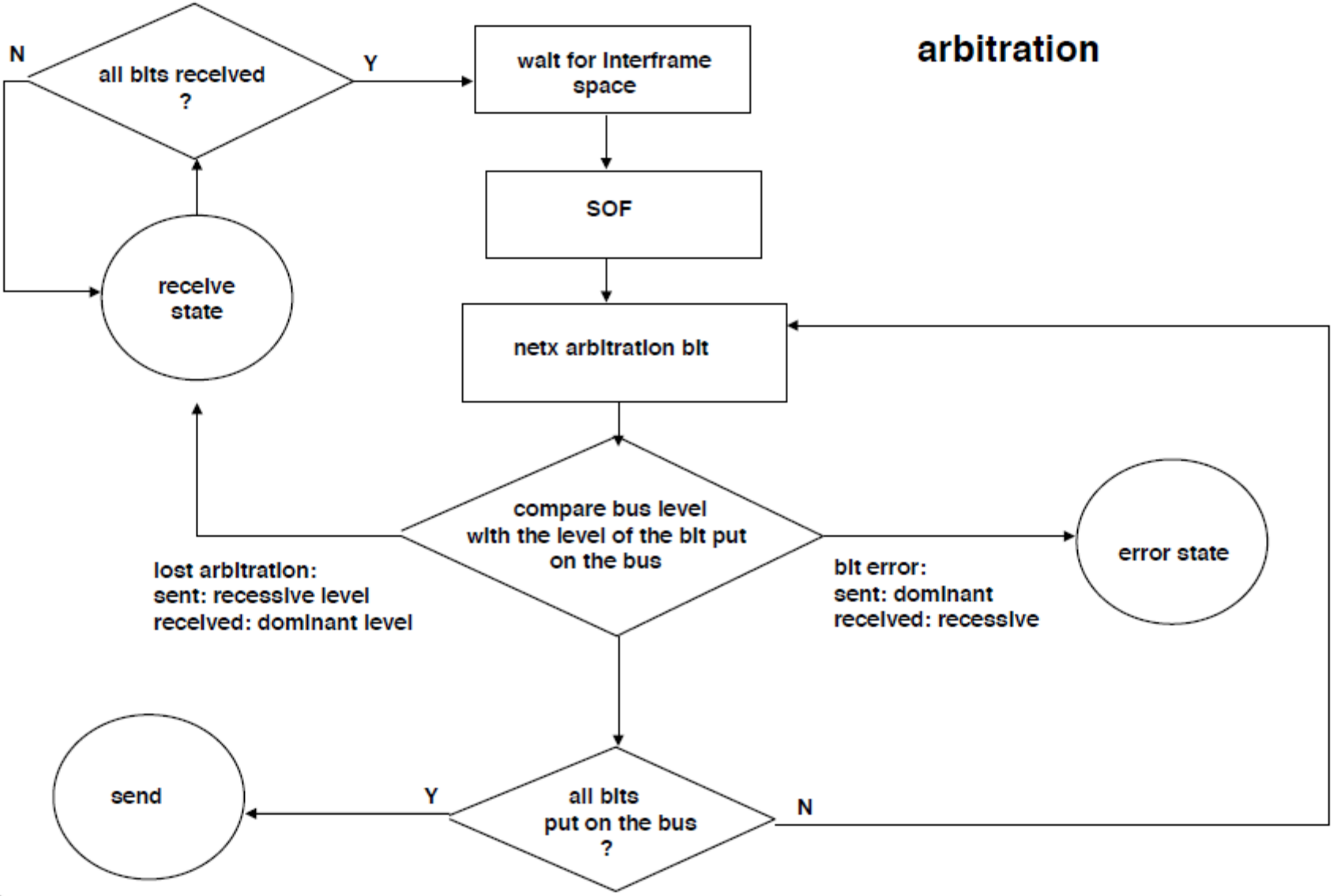
Arbitrierung Zugriffskontrolle





OSI Layer 2

CAN


arbitration
all bits received ?
N
Y
wait for interframe space
SOF
receive state
netx arbitration bit
compare bus level with the level of the bit put on the bus
lost arbitration:
sent: recessive level
received: dominant level
bit error:
sent: dominant
received: recessive
error state
send
Y
all bits put on the bus ?
N

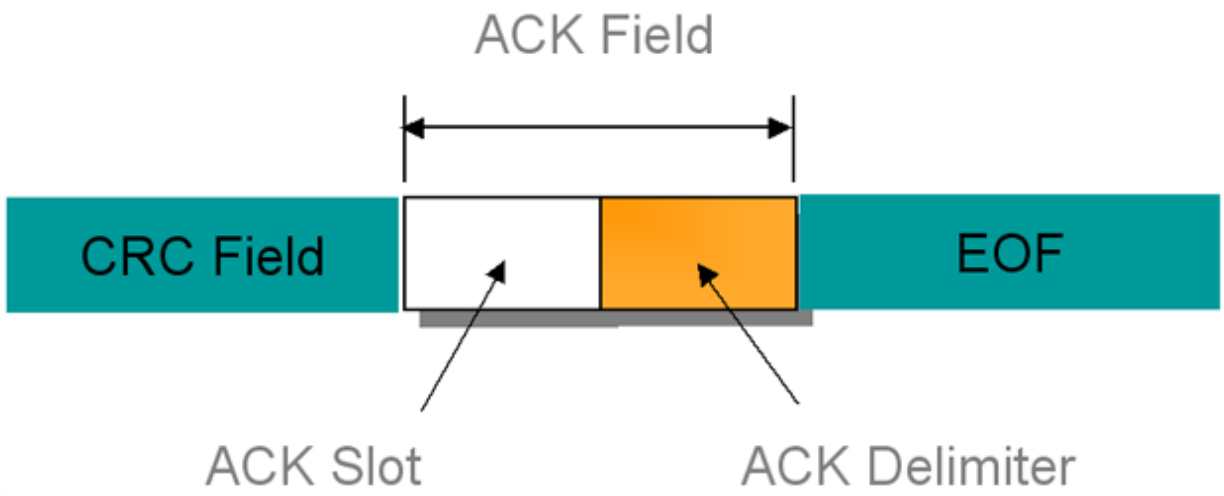
CAN

Acknowledge

OSI Layer 2

Das ACK Feld besteht aus zwei Bits

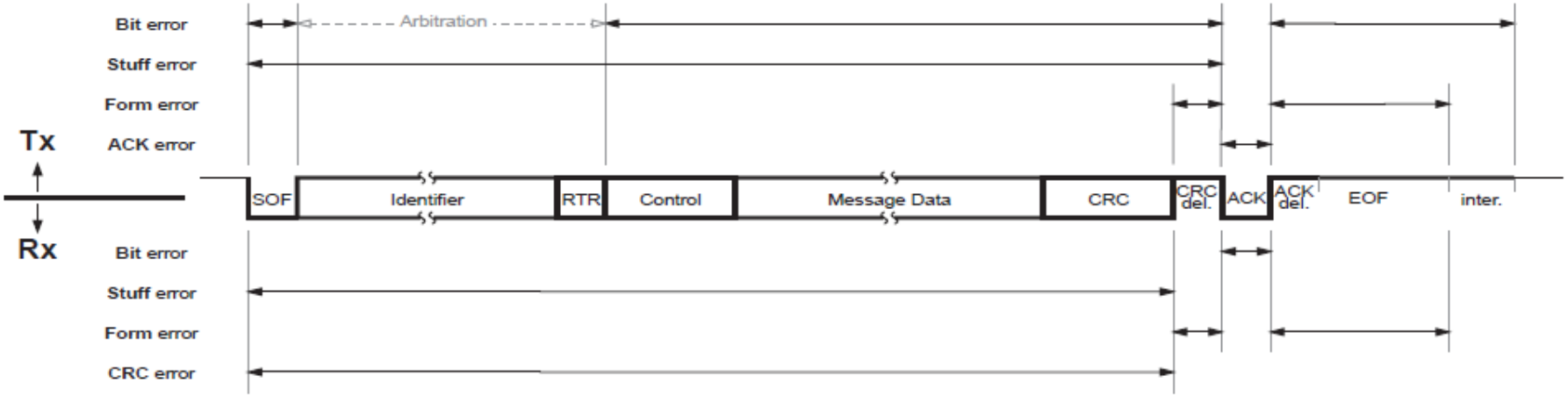
- **ACK Slot**
- **ACK Delimiter**



CAN

Fehlerbehandlung

Figure 19-13. Error Detection Procedures in a Data Frame



OSI Layer 2

CAN

Fehlerbehandlung(2)

LPC17xx

OSI Layer 2

Table 320. CAN Interrupt and Capture Register (CAN1ICR - address 0x4004 400C, CAN2ICR - address 0x4004 800C) bit description

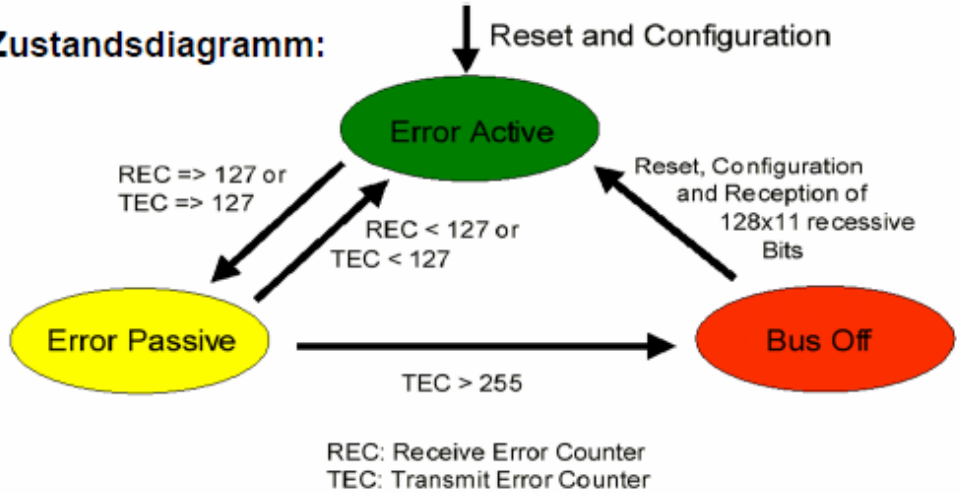
Bit	Symbol	Value	Function	Reset Value	RM Set
20:16	ERRBIT 4:0[3]		Error Code Capture: when the CAN controller detects a bus error, the location of the error within the frame is captured in this field. The value reflects an internal state variable, and as a result is not very linear:	0	X
		00011	Start of Frame		
		00010	ID28 ... ID21		
		00110	ID20 ... ID18		
		00100	SRTR Bit		
		00101	IDE bit		
		00111	ID17 ... 13		
		01111	ID12 ... ID5		
		01110	ID4 ... ID0		
		01100	RTR Bit		
		01101	Reserved Bit 1		
		01001	Reserved Bit 0		
		01011	Data Length Code		
		01010	Data Field		
		01000	CRC Sequence		
		11000	CRC Delimiter		
		11001	Acknowledge Slot		
		11011	Acknowledge Delimiter		
		11010	End of Frame		
		10010	Intermission		
		10001	Active Error Flag		
		10110	Passive Error Flag		
		10011	Tolerate Dominant Bits		
		10111	Error Delimiter		
		11100	Overload flag		
21	ERRDIR		When the CAN controller detects a bus error, the direction of the current bit is captured in this bit.	0	X
		0	Error occurred during transmitting.		
		1	Error occurred during receiving.		
23:22	ERRC1:0		When the CAN controller detects a bus error, the type of error is captured in this field:	0	X
		00	Bit error		
		01	Form error		
		10	Stuff error		
		11	Other error		

CAN

Zustandsdiagramm Fehlerbehandlung(3)

OSI Layer 2

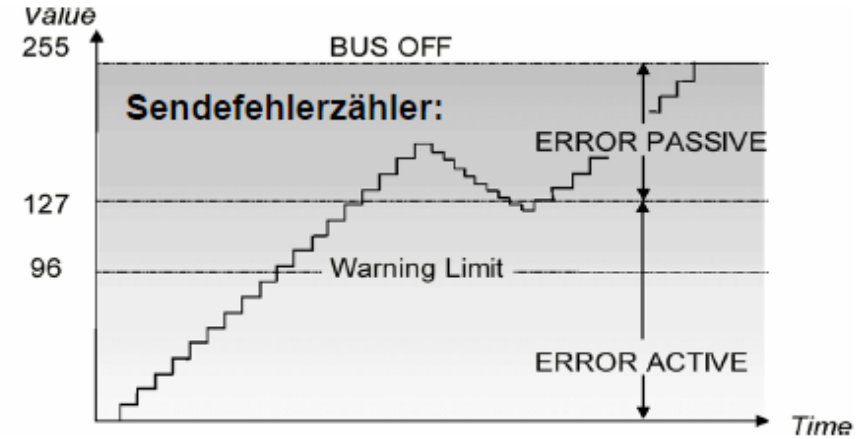
Zustandsdiagramm:

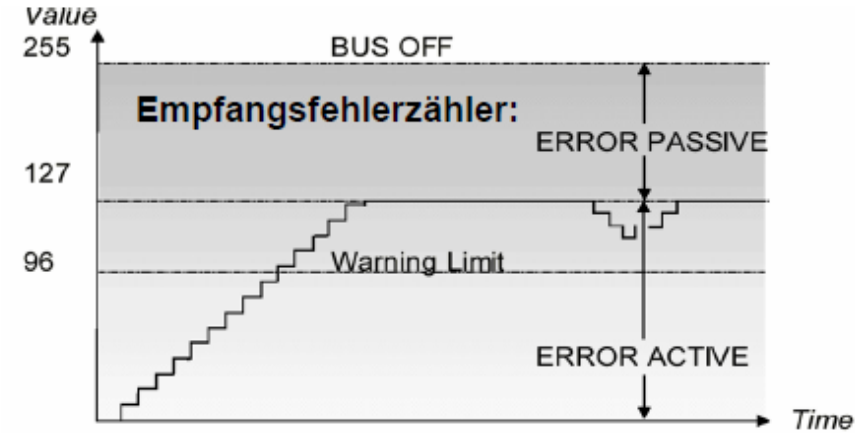


- *Error active:* Knoten nimmt weiter an Kommunikation teil, sendet aber ACTIVE ERROR FLAG
  Andere Teilnehmer erfahren so, dass eine Störung vorlag.

- *Error passive:* Knoten nimmt weiter an Kommunikation teil, unterdrückt aber ACTIVE ERROR FLAG
  Hiermit wird sichergestellt, dass ein gestörter Teilnehmer nur für begrenzte Zeit Telegramme blockieren kann.

- *Bus Off:* Ausgangstreiber abgeschaltet
  Teilnehmer ist so stark gestört, dass er vom Bus isoliert wird.

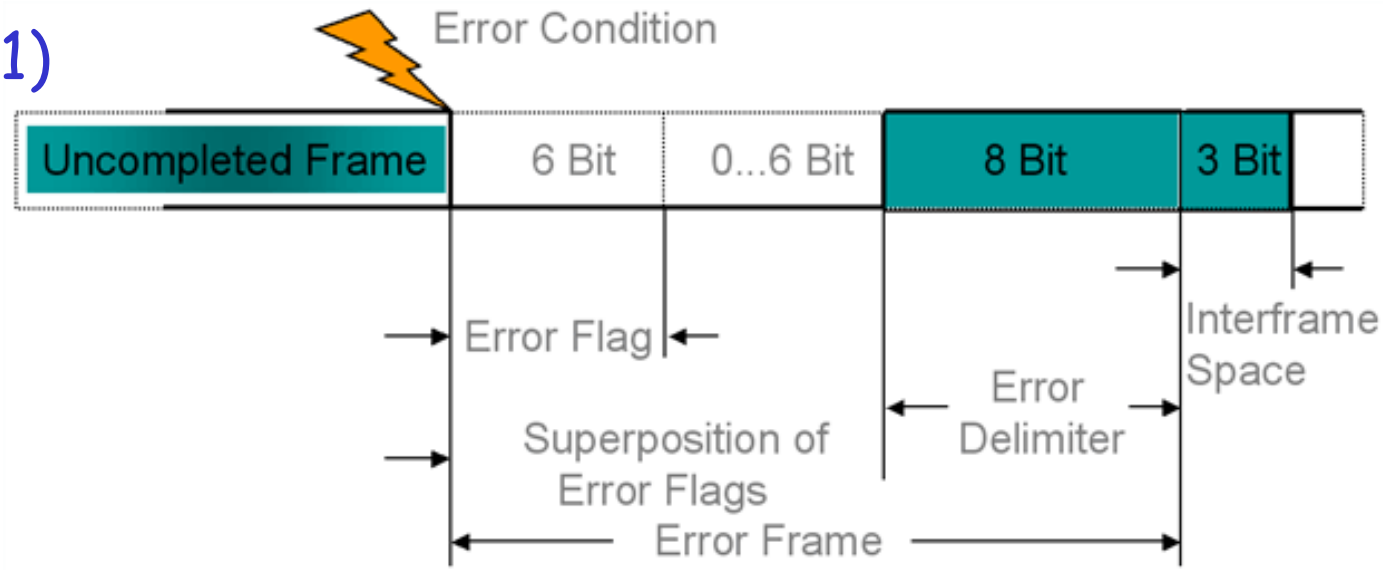




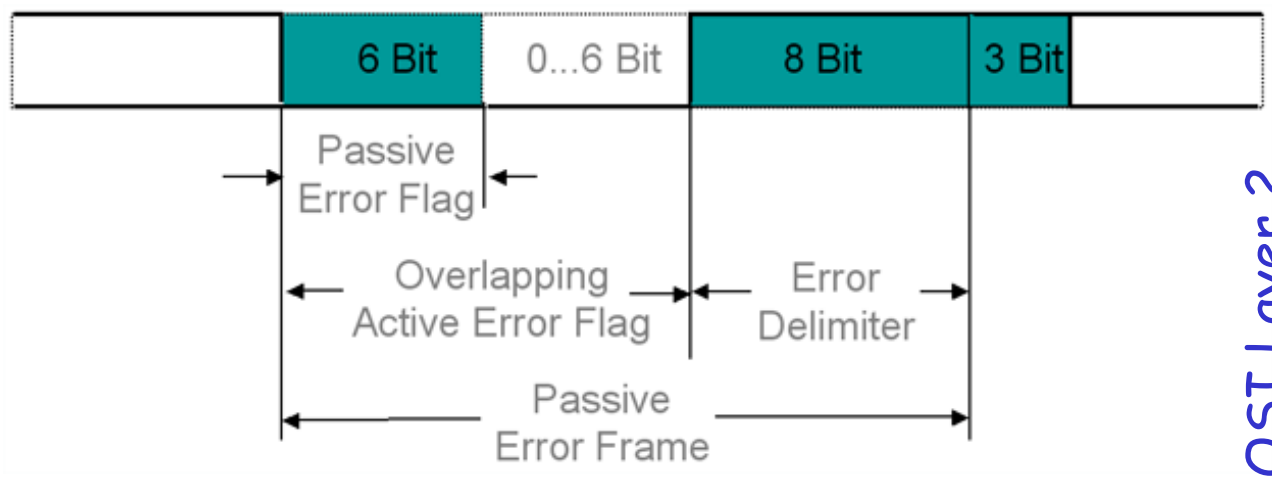
CAN

ErrorFlag(1)

- Aktiv



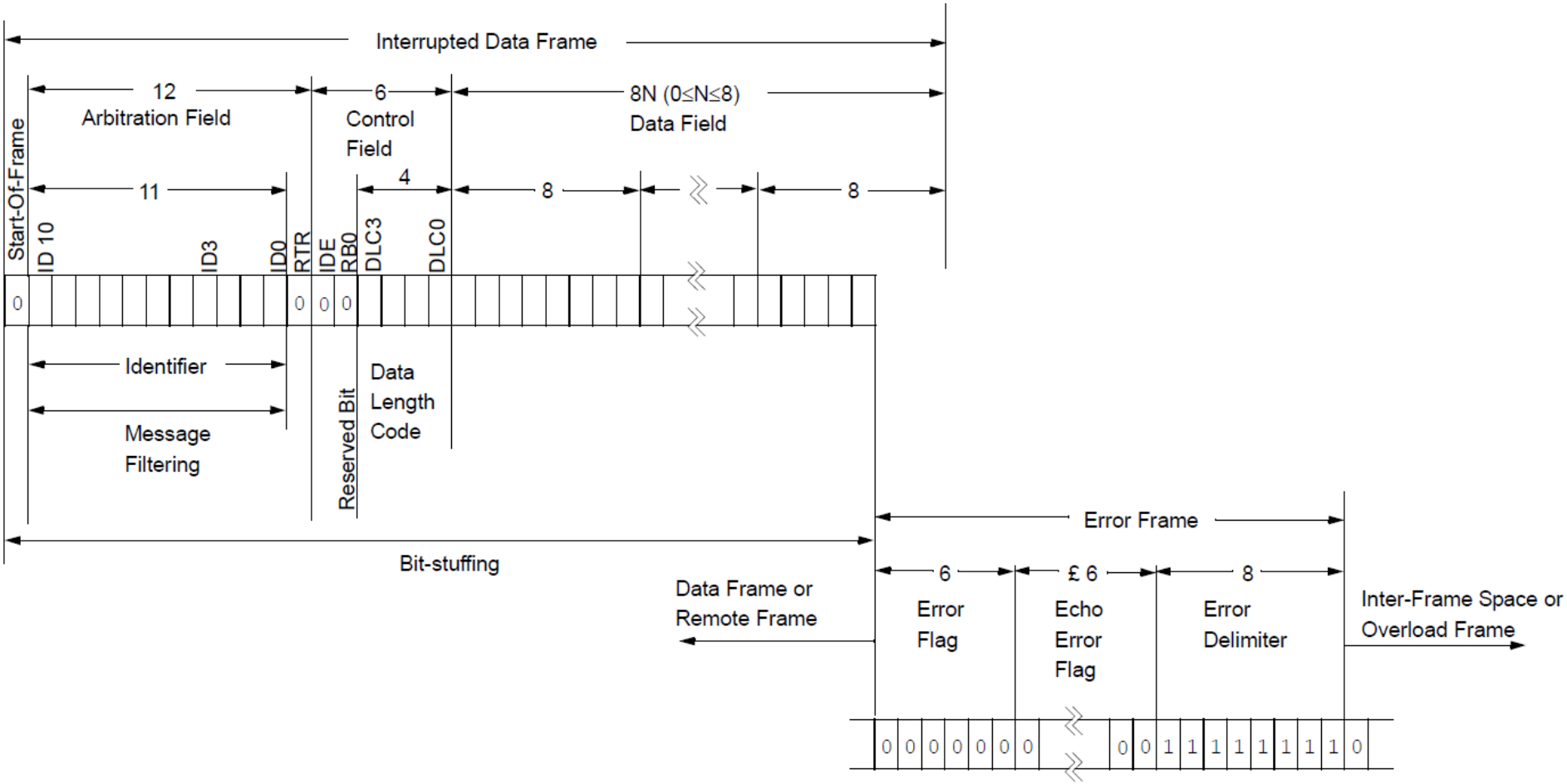
- Passiv



Quelle: softing.com

OSI Layer 2

CAN



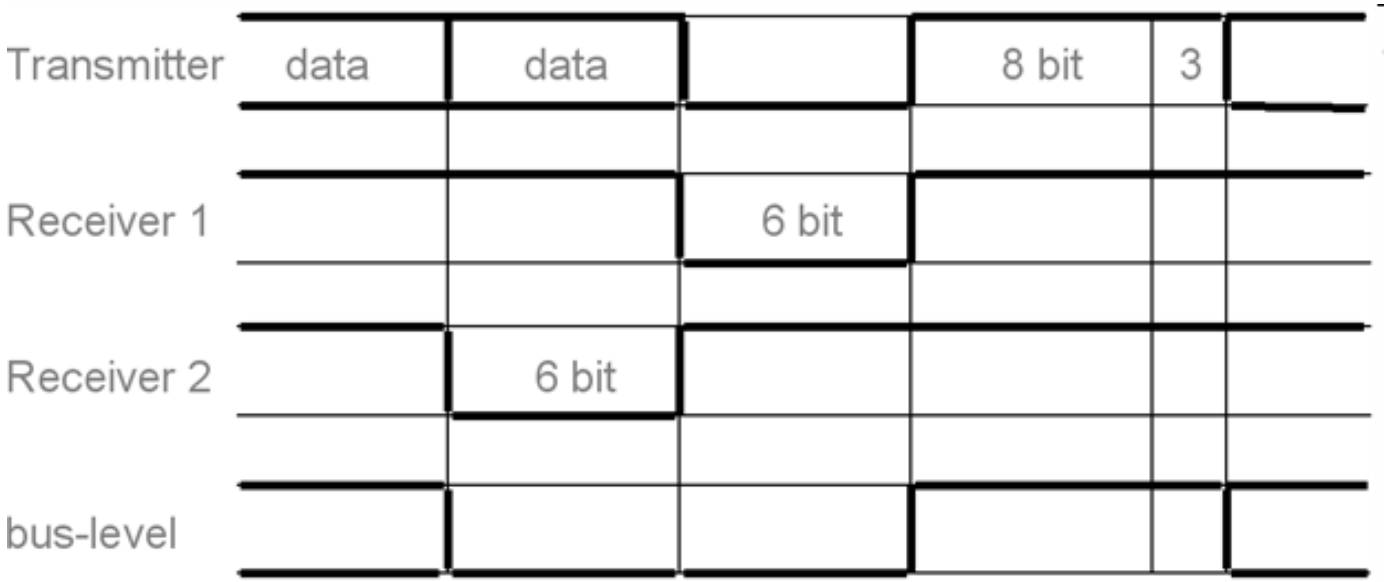
Quelle: softing.com

ISO

CAN

ErrorFlag (2)

1. Was bedeutet Aktiv und passiv?
2. Welcher Knoten kann einen Error Frame senden?
3. Was bedeutet ***Error Flag*** bzw. ***Error Delimiter?***



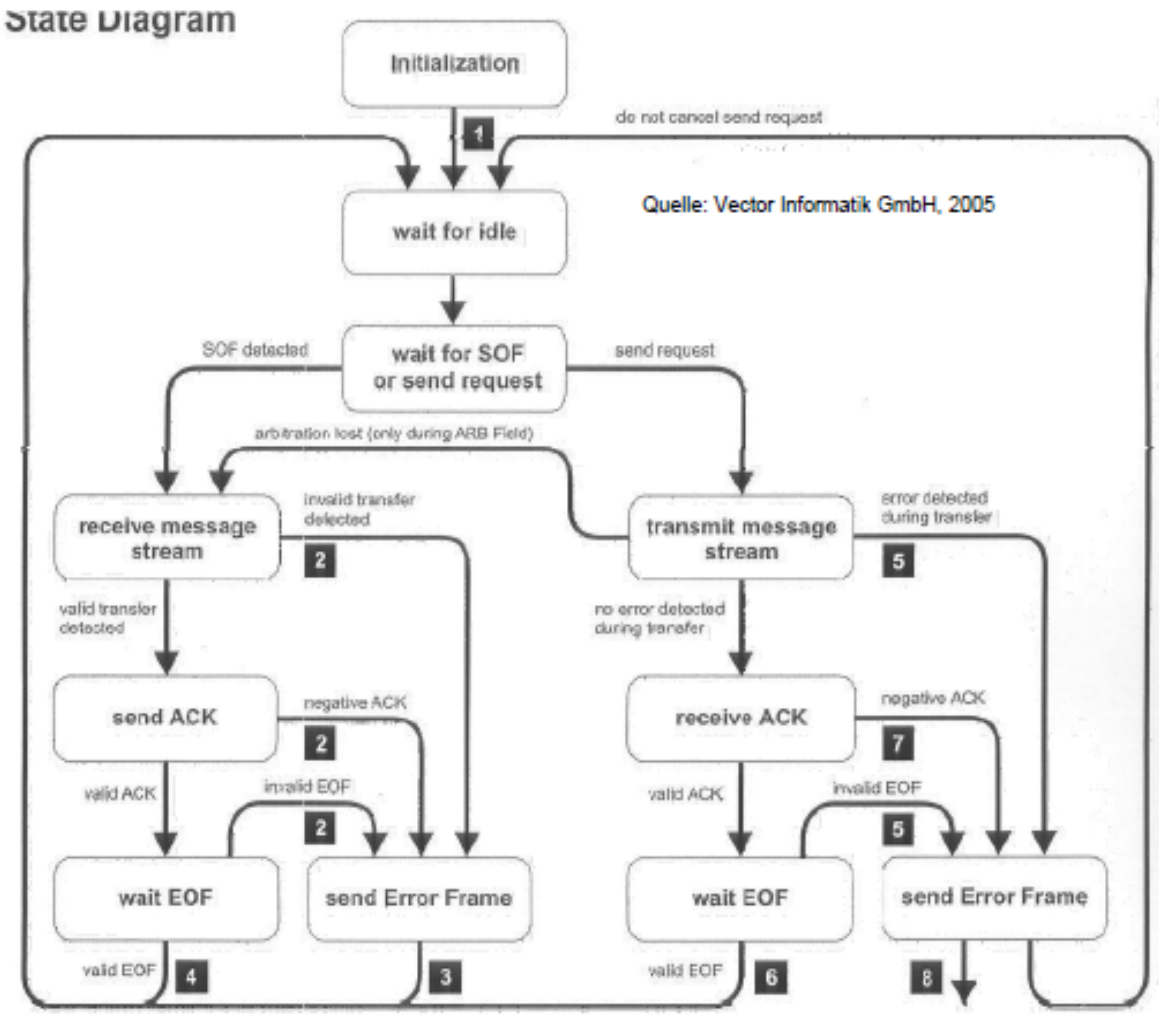
The Receiver 2 detects an error and makes it public to the other nodes.

OSI Layer 2

CAN

Zustandsdiagramm eines CAN-Controllers

OSI Layer 2



1 – Initialisierung RX/TX-Fehlerzähler = 0

2 – Inkrementierung RX-Fehlerzähler um Eins bei fehlerbehaftetem Empfang

3 – Inkrementierung RX-Fehlerzähler um Acht wenn der Controller als erster aller Teilnehmer einen Errorframe sendet

4 – Nach fehlerfreiem Datenempfang dekrementieren des RX-Fehlerzählers um Eins

5 – Inkrementierung TX-Fehlerzähler um Acht, wenn ein Fehler beim Senden einer eigenen Nachricht erkannt wird

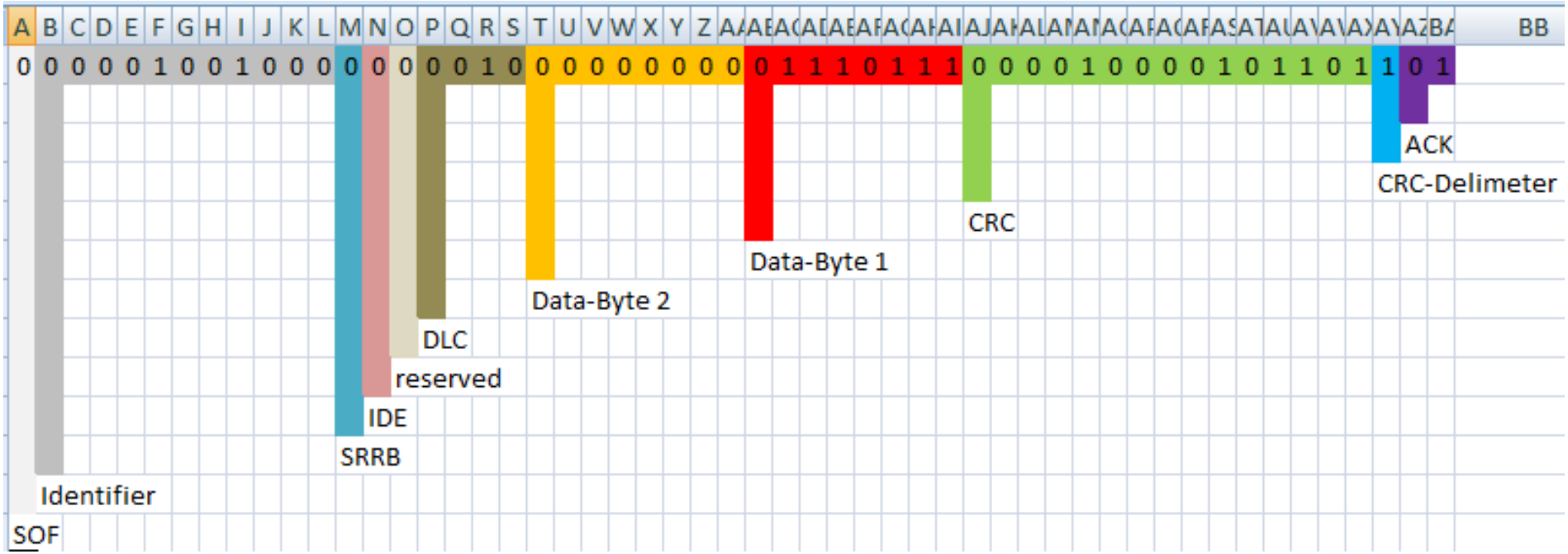
6 – Dekrementieren des TX-Fehlerzählers nach erfolgreichem Senden einer Nachricht

7 – TX-Fehlerzähler um Acht inkrementieren solange TX-Fehlerzähler kleiner als 127 ist

8 – Ist der Wert des TX-Fehlerzählers >255 wird der CAN-Controller vom Bus genommen

CAN

CRC-Check CAN-Bus



OSI Layer 2

Quelle: http://www.mikrocontroller.net/wikifiles

```
10010000000010000000000011101110000000000000000
1100010110011001|||||||||||||||||||||||||||||||| XOR
----------------||||||||||||||||||||||||||||||||
01010101100100010||||||||||||||||||||||||||||||
 1100010110011001||||||||||||||||||||||||||||| XOR
 ----------------|||||||||||||||||||||||||||||
 01101110101110110|||||||||||||||||||||||||||
  1100010110011001|||||||||||||||||||||||||| XOR
  ----------------||||||||||||||||||||||||||
  0001100011101111000||||||||||||||||||||||
     1100010110011001||||||||||||||||||||| XOR
     ----------------|||||||||||||||||||||
     0000001011100001001110||||||||||||||||
           1100010110011001|||||||||||||||| XOR
           ----------------||||||||||||||||
           01111101110101111|||||||||||||||
            1100010110011001||||||||||||||| XOR
            ----------------|||||||||||||||
            001111100011011011||||||||||||
              1100010110011001|||||||||||| XOR
              ----------------||||||||||||
              001111010100001000|||||||||||
                1100010110011001||||||||||| XOR
                ----------------|||||||||||
                 001100001001000100||||||||
                   1100010110011001|||||||| XOR
                   ----------------||||||||
                   000001111101110100000||||
                        1100010110011001|||| XOR
                        ----------------||||
                        001111100011100100||
                          1100010110011001|| XOR
                          ----------------||
                          00111101011110100|
                            1100010110011001 XOR
                            ----------------
                            0000010000101101 das ist die CRC Checksumme
```

OSI Layer 2

CAN

CRC-Check CAN-Bus

Quelle: http://www.mikrocontroller.net/wikifiles

CAN

Modulo 2 Division

Call the uppermost c+1 bits of the message the remainder
Beginning with the most significant bit in the original message and for each bit position that follows, look at the c+1 bit remainder:

- If the most significant bit of the remainder is a one, the divisor is said to divide into it. If that happens (just as in any other long division) it is necessary to indicate a successful division in the appropriate bit position in the quotient and to compute the new remainder. In the case of modulo-2 binary division, we simply:
  Set the appropriate bit in the quotient to a one, and XOR the remainder with the divisor and store the result back into the remainder
- Otherwise (if the first bit is not a one):
  Set the appropriate bit in the quotient to a zero, and XOR the remainder with zero (no effect)

Left-shift the remainder, shifting in the next bit of the message. The bit that's shifted out will always be a zero, so no information is lost.

The final value of the remainder is the CRC of the given message.

Quelle: Embedded systems, Michael Barr

Fehlererkennung: CRC

- Cyclic Redundancy Check (CRC)
  - Polynom G(x) vom Grad r
  - Rahmen M(x) mit m Bits
  - r Nullbits an niederwertigstes Bit des Rahmens anhängen
    - entspricht Polynom $x^r \cdot M(x)$, bestehend aus m+r Bit
  - teile $x^r \cdot M(x)$ durch G(x) (modulo 2)
  - subtrahiere *Divisionsrest* von $x^r \cdot M(x)$ (modulo 2)
  - Ergebnis T(x) ist Frame mit CRC.

M(x)=1101011011

G(x)=x^4+x+1=10011

```
11010110110000
10011
 10011
 10011
  000010110
      10011
       010100
        10011
         01110=Divisionsrest

11010110110000
         01110
11010110111110=T(x)
```

Senden

Quelle: Datenkommunikation, Hannes Federrath

CAN

Fehlererkennung:CRC-Prüfung

- Cyclic Redundancy Check (CRC)
  - Polynom G(x) vom Grad r
  - Empfänger dividiert empfangenes Frame durch G(x)
    - Divisionsrest == 0: OK
    - sonst: FEHLER

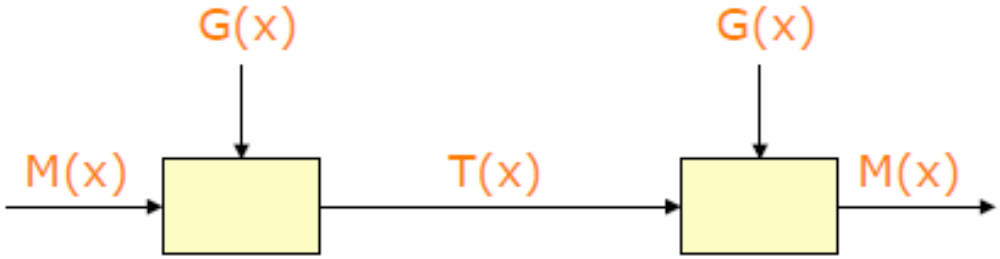
      ↓

      Nachrichtenwiederholung

M(x)=1101011011

$G(x)=x^4+x+1=10011$

```
T(x)=11010110111110
     10011
      10011
      10011
      0000010111
           10011
           0010011
             10011
             000000-->OK
```



Empfangen

Quelle: Datenkommunikation, Hannes Federrath

Fehlererkennung: CRC: G(x)

- Typische Generatorpolynome:
  - CRC-12: $x^{12} + x^{11} + x^3 + x^2 + x + 1$
    - geeignet für Zeichenlängen von 6 Bit
  - CRC-16: $x^{16} + x^{15} + x^2 + x + 1$
    - geeignet für Zeichenlängen von 8 Bit
  - CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$
    - geeignet für Zeichenlängen von 8 Bit

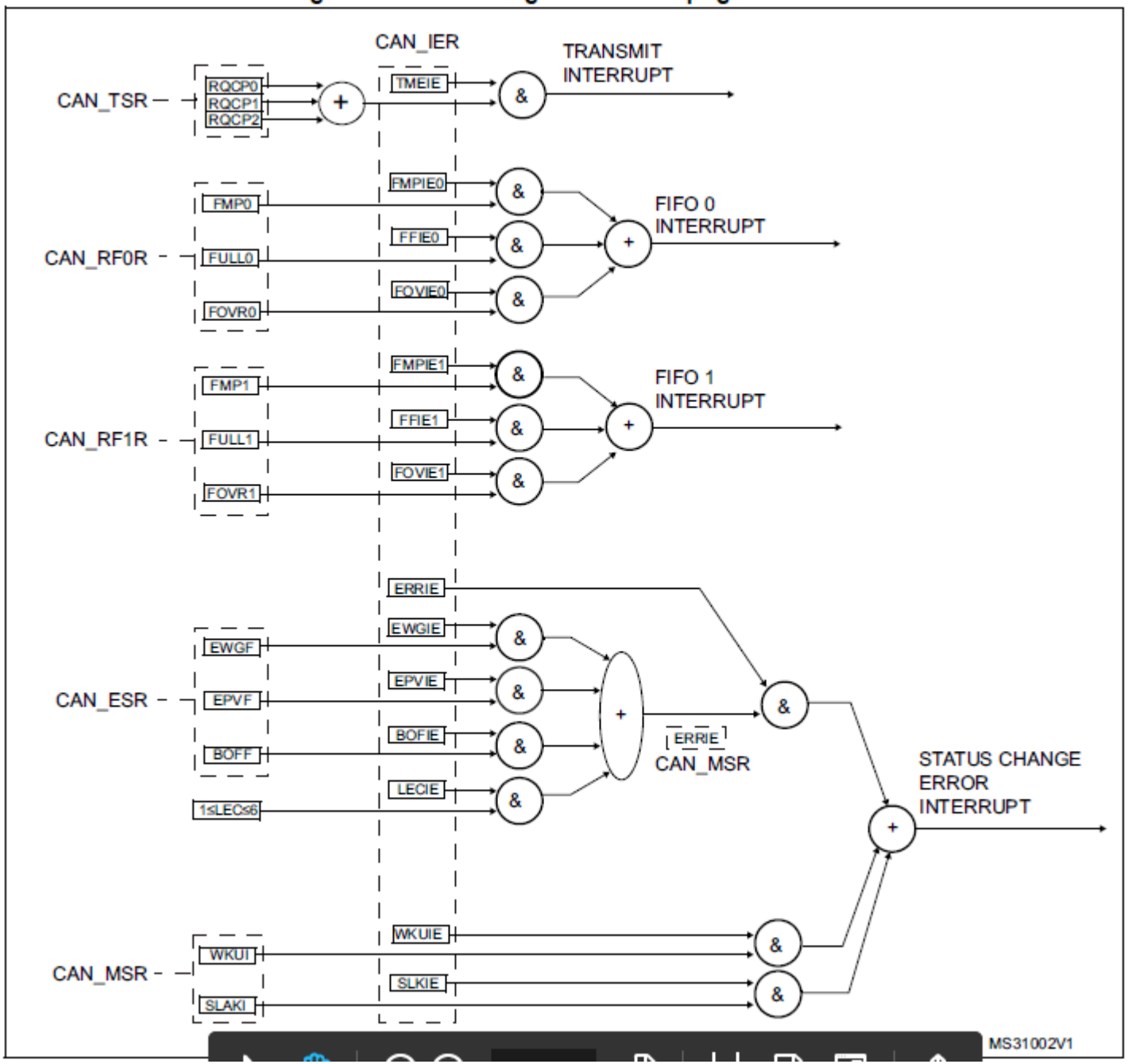
CCITT: Comite Consulatif International Telegraphique et Telephonique, später ITU-T (International Telecommunications Union)

- Leistungsfähigkeit von CRC mit G(x) vom Grad r
  - Einzelfehlererkennung
  - Erkennung aller Doppelfehler mit einem Abstand von k Bit, falls G(x) nicht den Faktor (x^k+1) enthält (k=2...Framelänge)
  - Erkennung aller Fehler mit ungerader Anzahl Fehlerbits, da G(x) den Faktor (x+1) enthält
  - Erkennung von Burst-Fehlern <= r Bit

Quelle: Datenkommunikation, Hannes Federrath

CAN-Interrupts (STM32)

Figure 321. Event flags and interrupt generation



CAN-Interrupts STM32)

CAN error status register (CAN_ESR)

Address offset: 0x18
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REC[7:0]								TEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEC[2:0]			Res.	BOFF	EPVF	EWGF
									rw	rw	rw		r	r	r

Bits 6:4 **LEC[2:0]**: Last error code
This field is set by hardware and holds a code which indicates the error condition of the last error detected on the CAN bus. If a message has been transferred (reception or transmission) without error, this field will be cleared to '0'.
The LEC[2:0] bits can be set to value 0b111 by software. They are updated by hardware to indicate the current communication status.
000: No Error
001: Stuff Error
010: Form Error
011: Acknowledgment Error
100: Bit recessive Error
101: Bit dominant Error
110: CRC Error
111: Set by software

Bit 3 Reserved, must be kept at reset value.

Bit 2 **BOFF**: Bus-off flag
This bit is set by hardware when it enters the bus-off state. The bus-off state is entered on TEC overflow, greater than 255, refer to *Section 29.7.6 on page 829*.

Bit 1 **EPVF**: Error passive flag
This bit is set by hardware when the Error Passive limit has been reached (Receive Error Counter or Transmit Error Counter>127).

CAN

Transportprotokoll ISO 15765-2 (1)

- **Single Frame:** das zu übertragende Telegramm besteht aus bis zu 6 bzw. 7 Bytes (Normal Addressing) Nutzdaten, nicht segmentierte Übertragung.
- **First Frame:** erster Frame des Senders, wenn mehr als 6 bzw. 7 Bytes Daten segmentiert übertragen werden müssen. Der Frame enthält die gesamte Länge des Telegramms.
- **Consecutive Frame:** Überträgt die weiteren Nutzdaten bei Segmentierung.
- **Flow Control Frame:** Antwort des Empfängers, der die Modalitäten für die Übertragung der weiteren Consecutive Frames festlegt.

CAN

Transportprotokoll ISO 15765-2 (2)

CAN-TP Header

Bit offset	0 .. 3	4 .. 7	8 .. 15	16..23	24..
Single	0	size (0..7)	Data A	Data B	Data C
First	1	size (7..4095)		Data A	Data B
Consecutive	2	index (0..15)	Data A	Data B	Data C
Flow	3	FC flag (0,1,2)	Block size	ST	

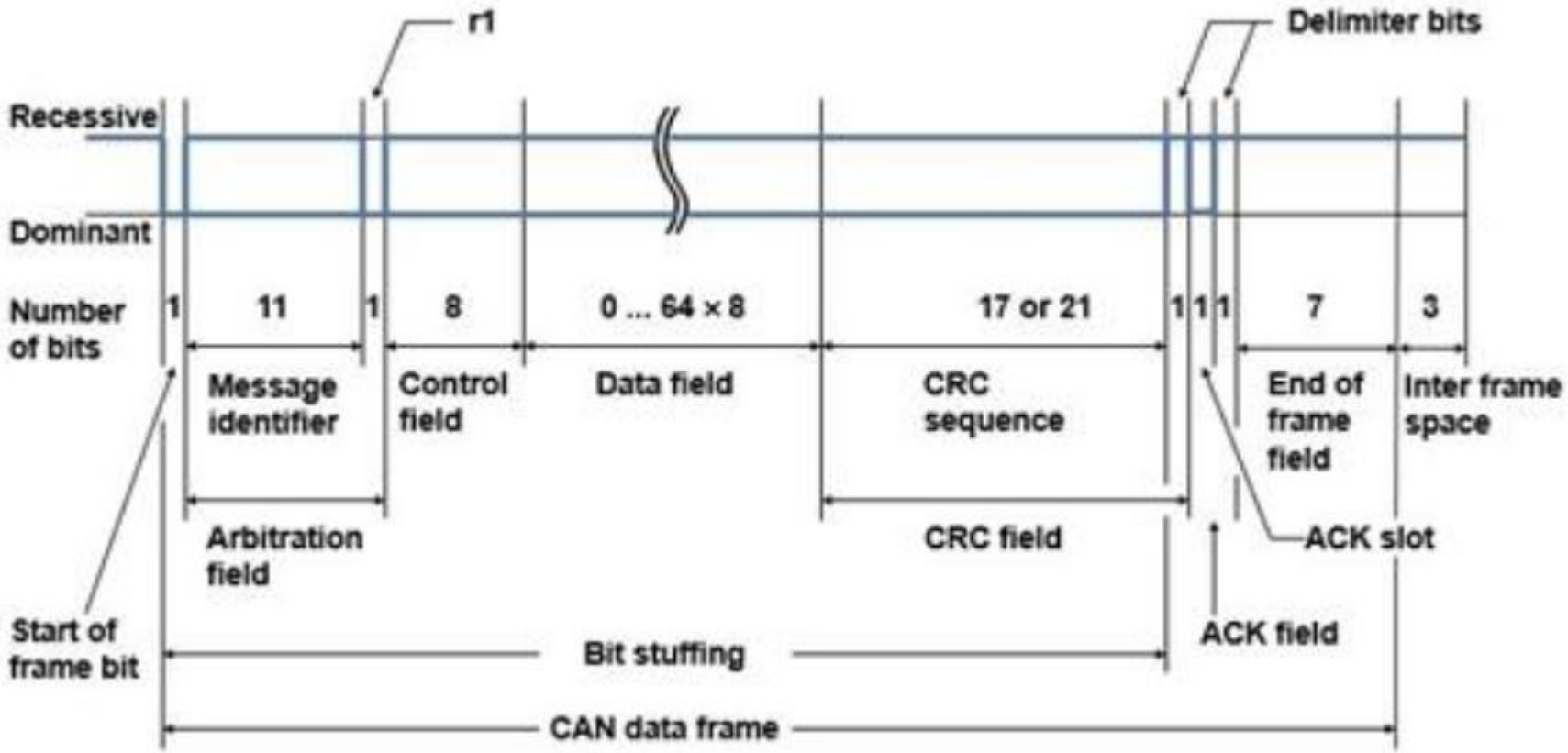
Flow Control

Bit offset	0 .. 3	4 .. 7	8 .. 15	16..23
Description	type	if the transfer is allowed	Block Size	Separation Time (ST), minimum delay time between frames (end of one frame and the beginning of the other)
Single	type = 3	(0 = Clear To Send, 1 = Wait, 2 = Overflow/abort)	0 = remaining "frames" to be sent without flow control or delay	<=127, delay between frames. Please mind it is hex. 0x14 means 20ms.
Single	type = 3	(0 = Clear To Send, 1 = Wait, 2 = Overflow/abort)	> 0 send number of "frames" before waiting for the next flow control frame	0xF1 to 0xF9, 100 to 900 microseconds.

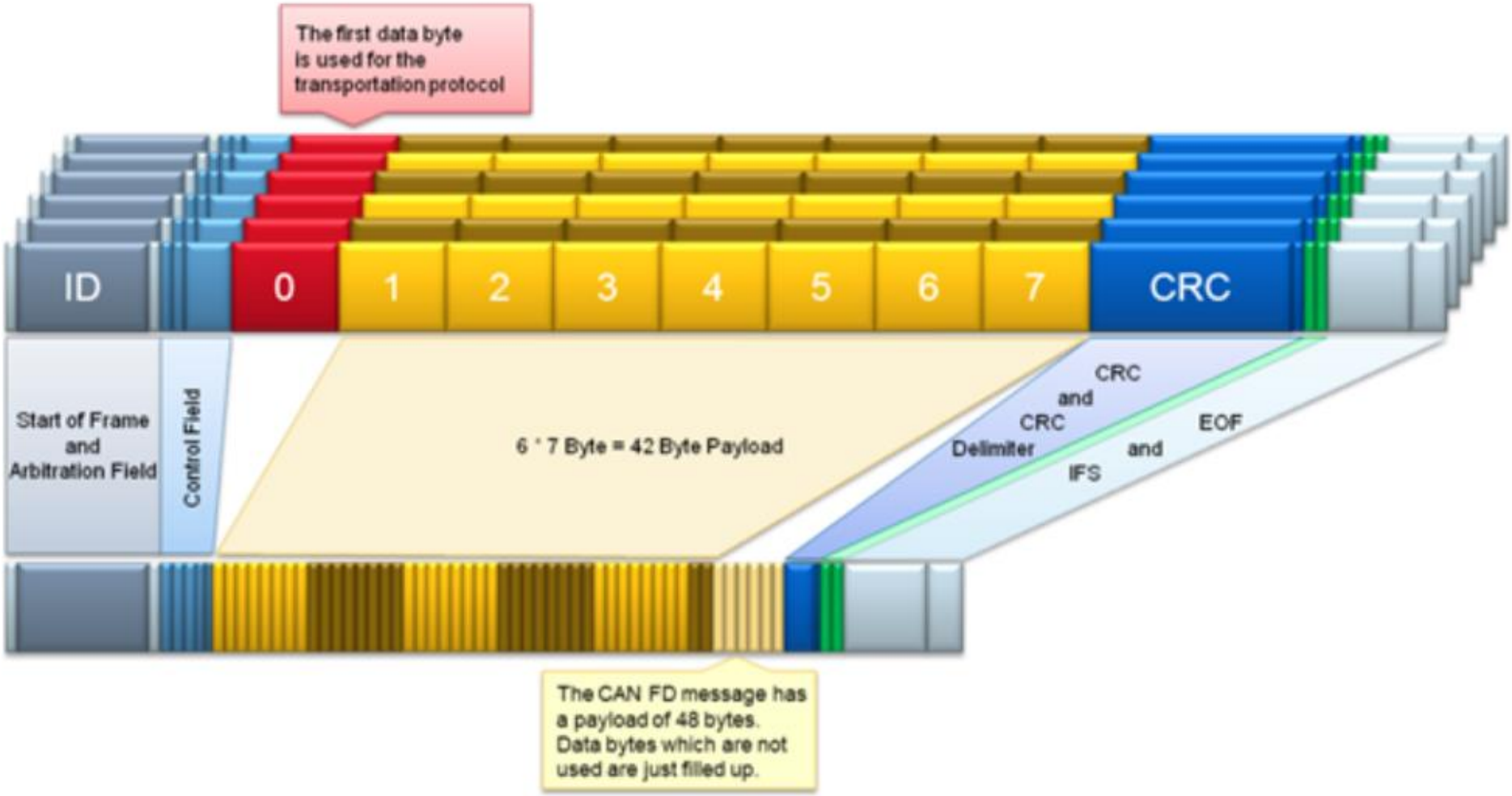
http://en.wikipedia.org/wiki/ISO_15765-2

CAN-FD

CAN-Flexible Datarate

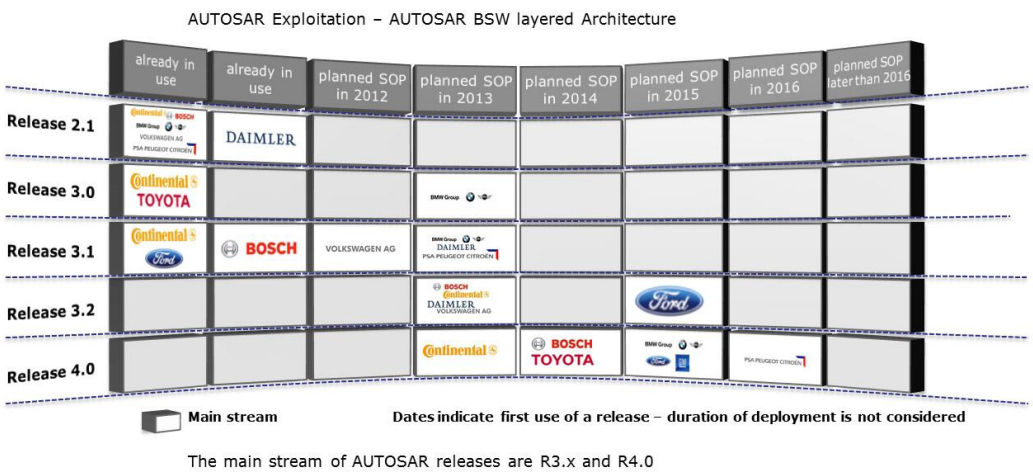


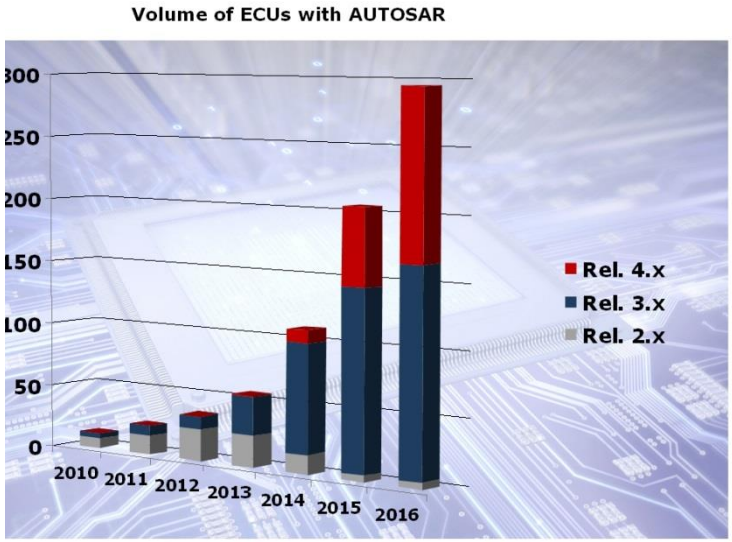
CAN-FD


The first data byte is used for the transportation protocol
ID
0
1
2
3
4
5
6
7
CRC
Start of Frame and Arbitration Field
Control Field
6 * 7 Byte = 42 Byte Payload
CRC and CRC Delimiter
EOF and IFS
The CAN FD message has a payload of 48 bytes. Data bytes which are not used are just filled up.

Software in ECUs

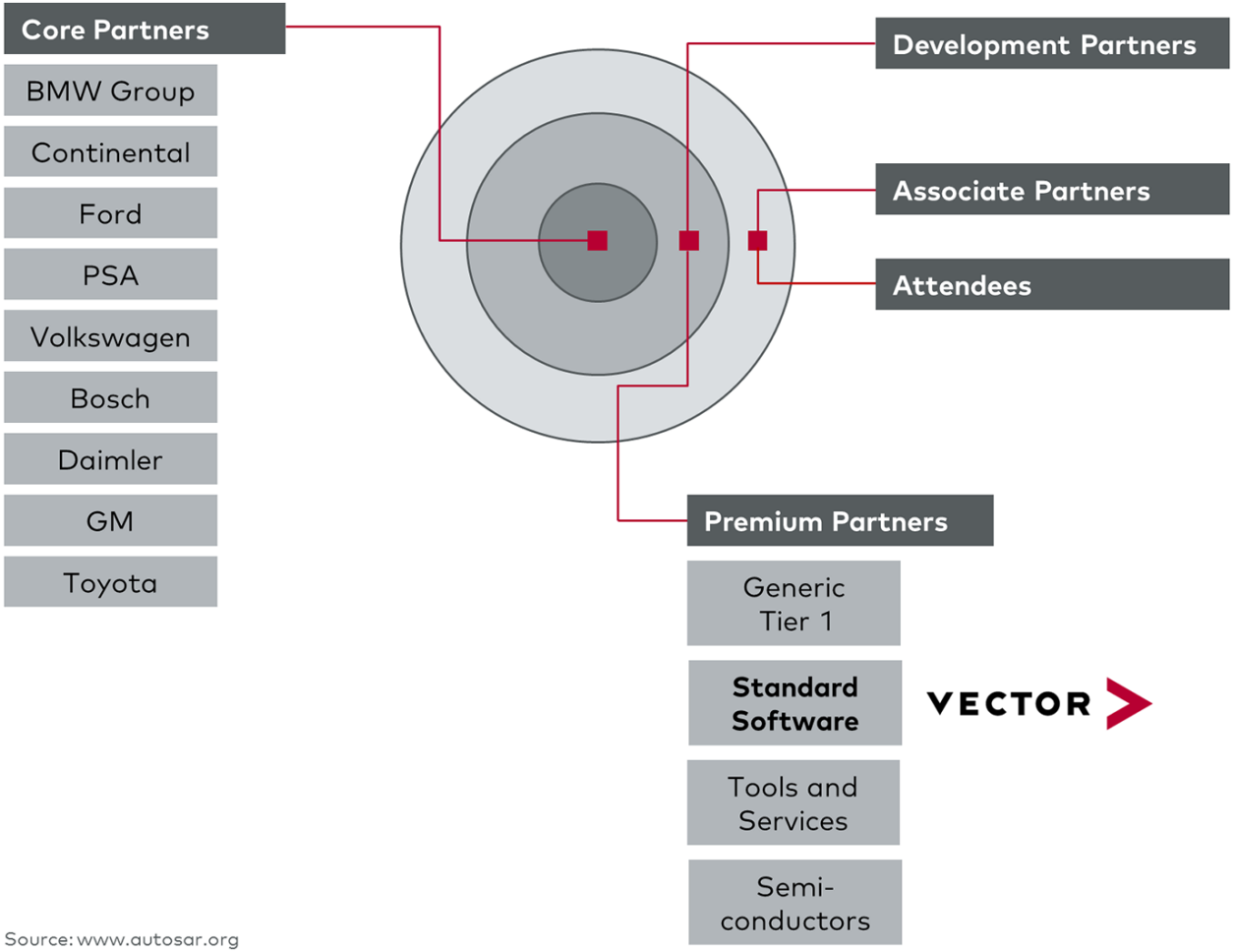
AUTOSAR AUTomotive Open System ARchitecture





Software in ECUs

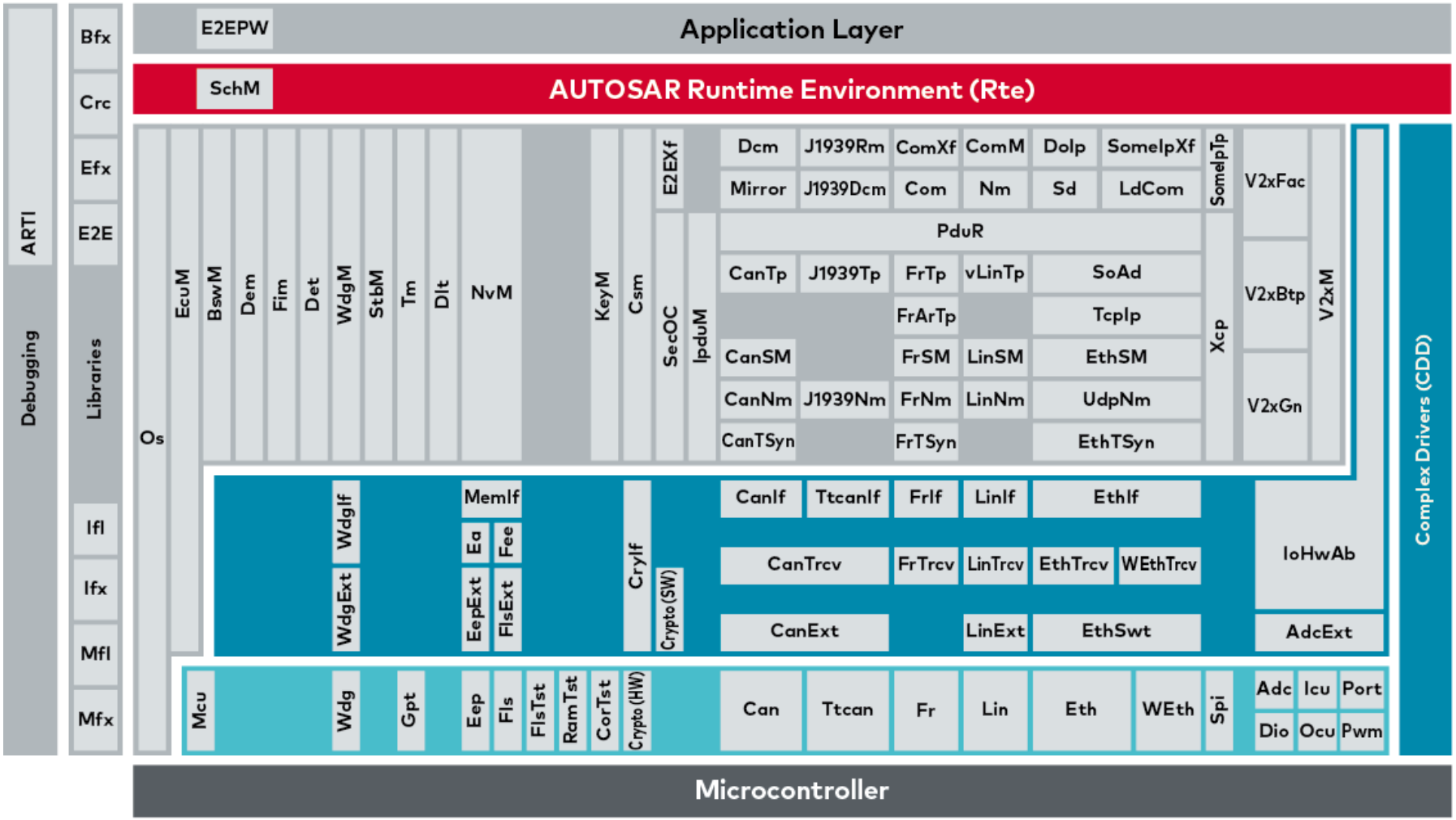
The AUTOSAR Development Partnership



Source: www.autosar.org

AUTOSAR
AUTomotive
Open System
ARchitecture

Software in ECUs



Software in ECUs

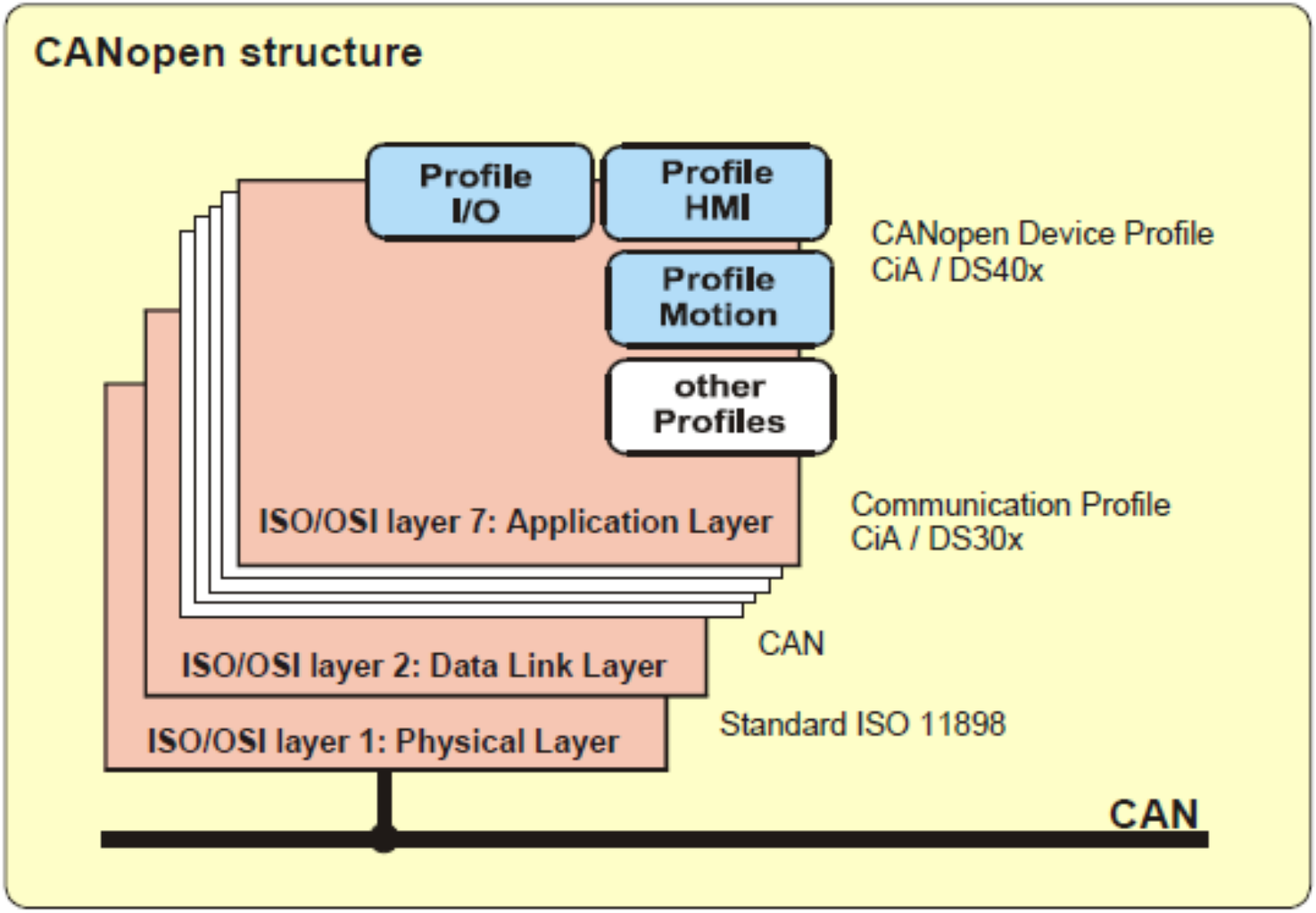
AUTOSAR AUTomotive Open System Architecture

Beispiel:

```
void DoorReceiverRightRear (Rte_Instance self)
{
  DoorStatusType statusRightRear;
  Rte_Read_RDoorRightRear_status (self, &statusRightRear);
  if(statusRightRear != DoorClosed) {
    Rte_IWrite_DoorReceiverRightRear_PHorn_cmd (self,HornOn);}
  else {
    Rte_IWrite_DoorReceiverRightRear_PHorn_cmd (self,HornOff);}
}
```

Effiziente Entwicklung von AUTOSAR-Komponenten mit domänenspezifischen Programmiersprachen
Dr. Frank Höwing
Lineas Automotive GmbH

CANOpen


CANopen structure
Profile I/O
Profile HMI
Profile Motion
other Profiles
CANopen Device Profile
CiA / DS40x
ISO/OSI layer 7: Application Layer
Communication Profile
CiA / DS30x
ISO/OSI layer 2: Data Link Layer
CAN
ISO/OSI layer 1: Physical Layer
Standard ISO 11898
CAN

CANOpen

Geräteprofile (CiA)	für die Gerätetypen
DS 401	Digitale und analoge E/A
DS 402	Antriebe
DS 403	Bedienen und Beobachten
DS 404	Sensoren / Regler
DS 405	Programmierbare Geräte
DS 406	Encoder
DS …	(weitere Geräteprofile)

CANOpen

Objektverzeichnis

Prinzipielle Vergabe der Objekt-Index-Nummern

Object Index (hex)	Object
0000	Not used
0001 - 001F	Static Data Types
0020 - 003F	Complex Data Types
0040 - 005F	Manufacturer Specific Complex Data Types
0060 - 007F	Device Profile Specific Static Data Types
0080 - 009F	Device Profile Specific Complex Data Types
00A0 - 0FFF	Reserved for further use
1000 - 1FFF	Communication Profile Area
2000 - 5FFF	Manufacturer Specific Profile Area
6000 - 9FFF	Standardized Device Profile Area
A000 - FFFF	Reserved for further use

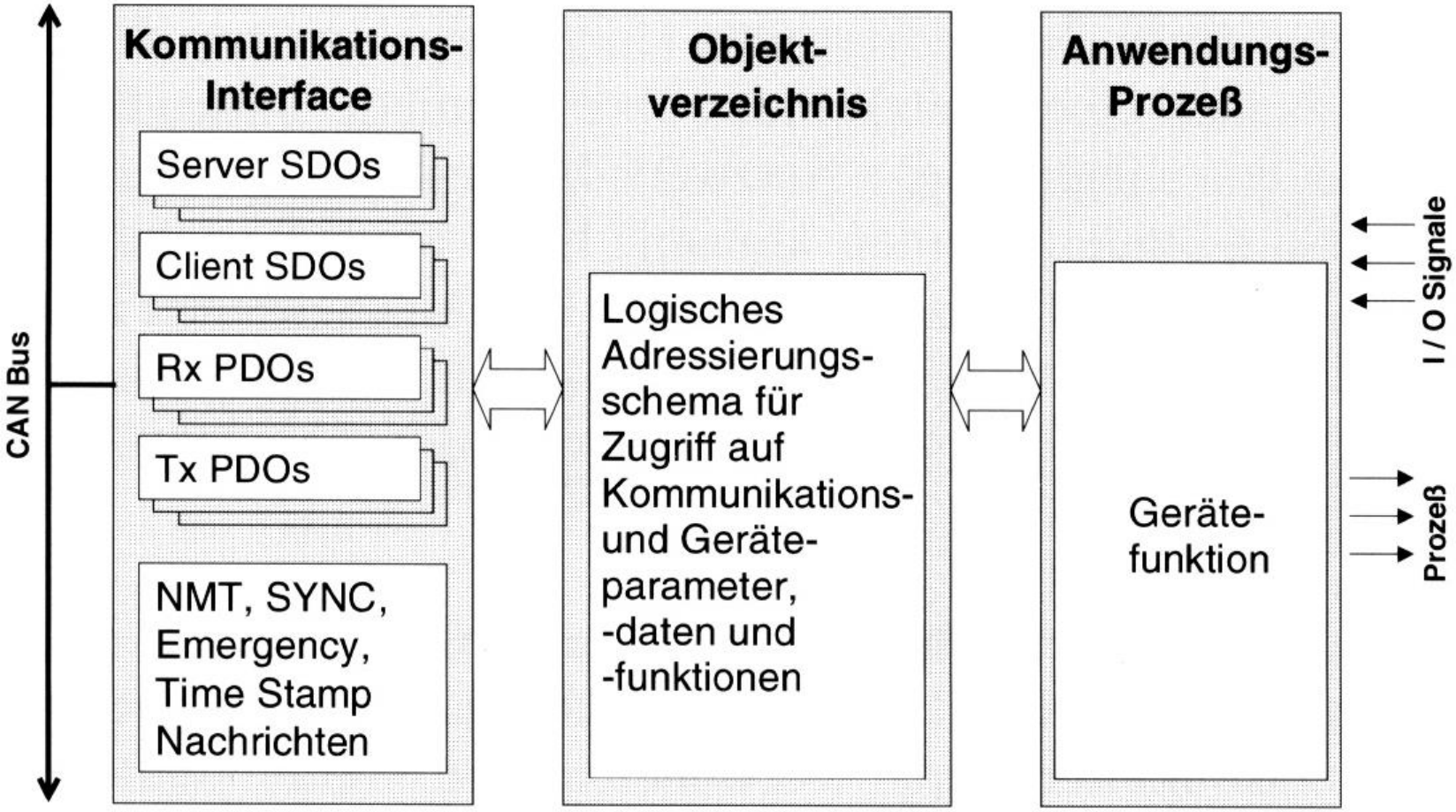
CANOpen

Objektverzeichnis(2)

Beispiel für ein Objektverzeichnis: Ausschnitt aus dem Verzeichnis des CANopen Chip CO4011

Index	Sub-Index	Name	Acc.	PDO-Mapping
0005	-	Dummy 8	ro	Yes
0006	-	Dummy 16	ro	Yes
100B	-	Node ID	ro	-
100C	-	Guard Time	rw	-
100D	-	Life Time Factor	rw	-
100E	-	COB-ID Guard	rw	-
1010	-	Store Parameters *1)	wo	-
1011	-	Reload Default Parameter *1)	wo	-
1014	-	COB ID Emergency	rw	-
1015	-	Inhibit Time Emergency	rw	-
1017	-	Producer Heartbeat Time	rw	-
1018		Identity Object		
	0	(no. of subentries)	ro	-
	1	Vendor ID	ro	-
	2	Product Code	ro	-
	3	Revision Number	ro	-

CANOpen


CAN Bus
Kommunikations-Interface
Server SDOs
Client SDOs
Rx PDOs
Tx PDOs
NMT, SYNC, Emergency, Time Stamp Nachrichten
Objekt-verzeichnis
Logisches Adressierungs-schema für Zugriff auf Kommunikations- und Geräte-parameter, -daten und -funktionen
Anwendungs-Prozeß
Geräte-funktion
I / O Signale
Prozeß

CANOpen

Default Identifier

Identifier 11-Bit (binär)	Identifier (dezimal)	Identifier (hexadezimal)	Funktion
00000000000	0	0	Netzwerkmanagement
00010000000	128	80h	Synchronisation
0001xxxxxxxx	129 - 255	81h - FFh	Emergency
0011xxxxxxxx	385 - 511	181h - 1FFh	PDO1 (tx)
0100xxxxxxxx	513 - 639	201h - 27Fh	PDO1 (rx)
0101xxxxxxxx	641 - 767	281h - 2FFh	PDO2 (tx)
0110xxxxxxxx	769 - 895	301h - 37Fh	PDO2 (rx)
0111xxxxxxxx	897 - 1023	381h - 3FFh	PDO3 (tx)
1000xxxxxxxx	1025 - 1151	401h - 47Fh	PDO3 (rx)
1001xxxxxxxx	1153 - 1279	481h - 4FFh	PDO4 (tx)
1010xxxxxxxx	1281 - 1407	501h - 57Fh	PDO4 (rx)
1011xxxxxxxx	1409 - 1535	581h - 5FFh	SDO senden
1100xxxxxxxx	1537 - 1663	601h - 67Fh	SDO empfangen
1110xxxxxxxx	1793 - 1919	701h - 77Fh	NMT Error Controll
xxxxxxxx = Knotennummer 1 - 127			

CANOpen

Service-Daten-Objekte

Service Data Objekt SDO

	Byte 0	Byte 1-3: Multiplexer		Byte 4-7: Data
Initiate Command	**Domain Protocol**	**16 bit Index**	**8 bit Subindex**	**1- 4 byte bit parameter data**

- Upload
- Download
- Data Length
- Expedited Transfer
- Segmented Transfer
- Abort
- Toggle Bit

16 bit Index	Subindex	discription	value (parameter)
1000H	00H	Device Type	70191H
...	...	...	...
1018H	00H	Identity Object	03H
	01H	Vendor ID	
	02H	Product Code	value (parameter)
	03H	Revision Number	00H ... FFFFFFFFH
...	...	...	...
6002H	00H	Chance Polarity	value (parameter)
...	...	...	...

Object dictionary

A SDOs offers the access to the object dictionary

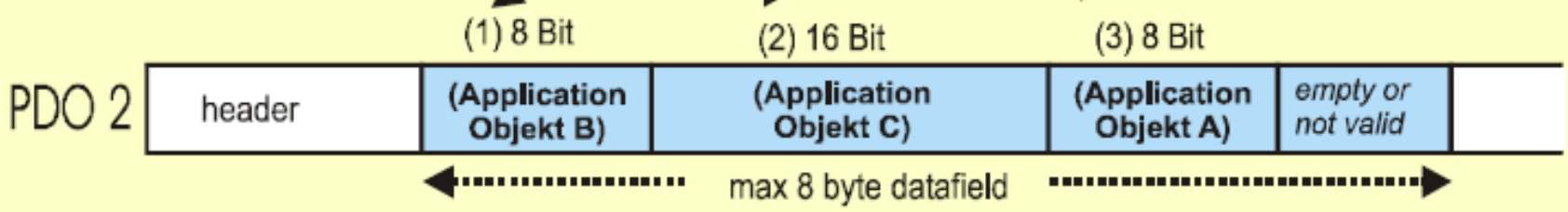
CANOpen

Prozess-Daten-Objekte

PDO Mapping

Object Dictionary		
...	...	
xxxxh	xxh	(Application Object A)
yyyyh	yyh	(Application Object B)
zzzzh	zzh	(Application Object C)
...	...	
nnnnh	nnh	(Mapping Object 1)
ooooh	ooh	(Mapping Object 2)
...	...	
		(Mapping Object x)
...	...	

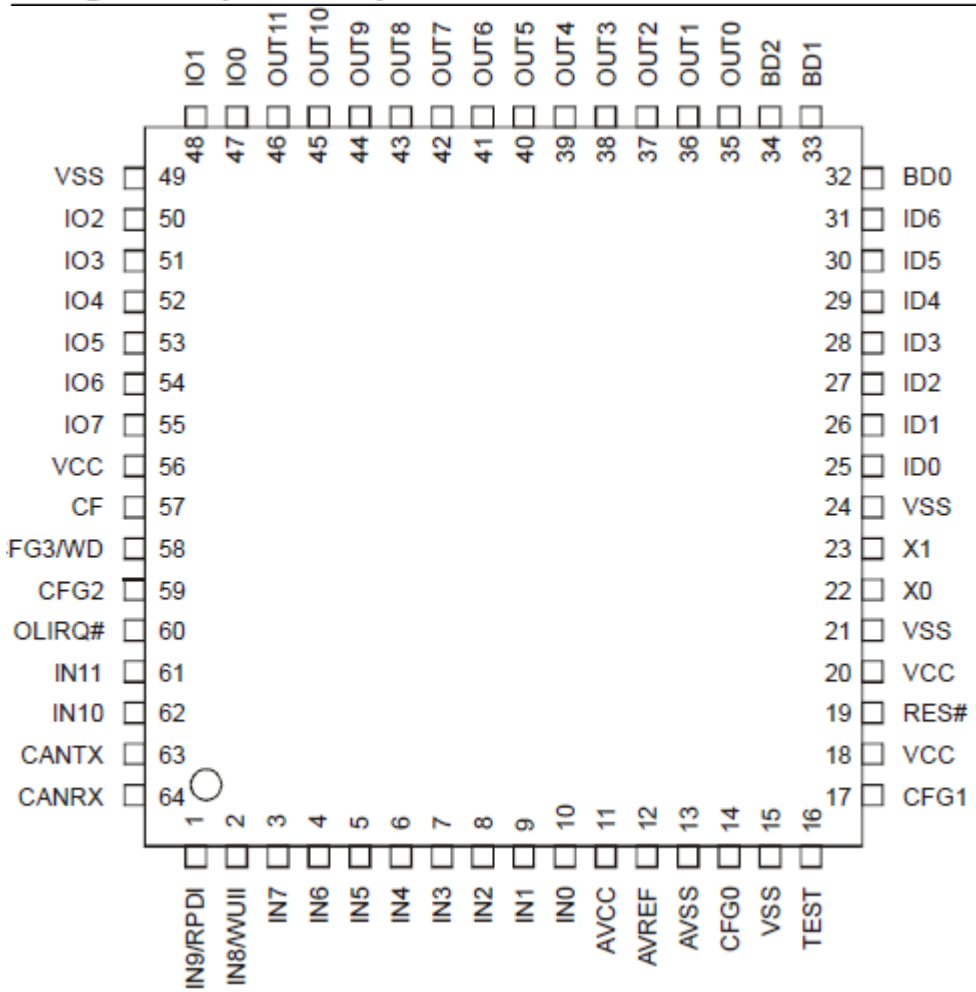
Mapping Object 2			
0	3	(3 mapped objects)	
1	yyyyh	yyh	8
2	zzzzh	zzh	16
3	xxxxh	xxh	8



CANOpen

CO4011A-FL

Single Chip CANopen Controller for Remote I/O



DS401: Digital Output Objects

Index	Sub-Index	Name	Acc.
5200	-	Reset Output on Error Option	rw
6200	0 to n	Write Output 8 Bit	rw
6202	0 to n	Change Polarity Output 8 bit	rw
6206	0 to n	Error Mode Output	rw
6207	0 to n	Error State Output	rw
6300	0 to n	Write digital output 16 bit	wo
6320	0 to n	Write digital output 32 bit	wo

Beispiel CO4011
CANOpen Controller