

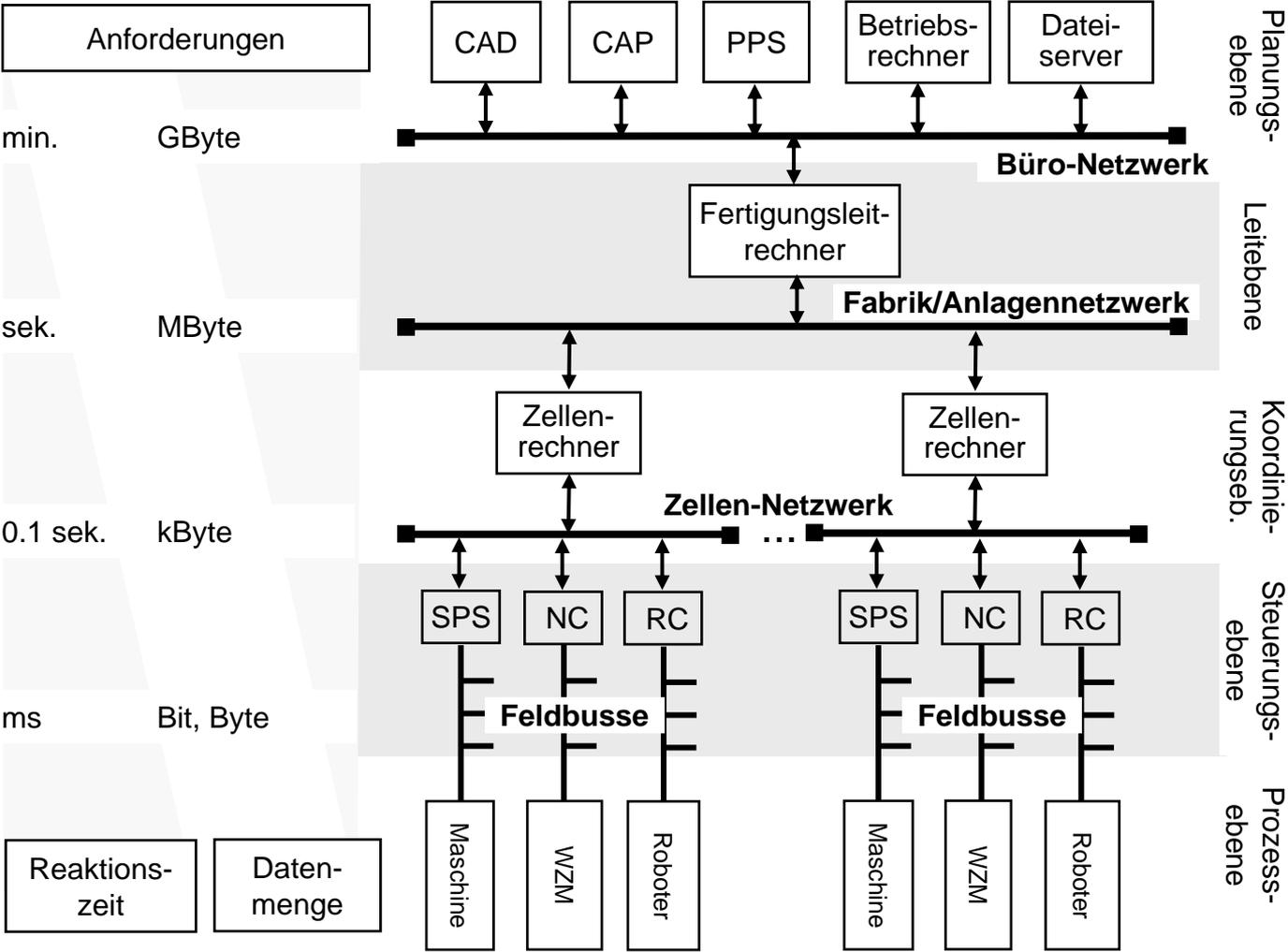
# Kapitel 5

## Echtzeitkommunikation

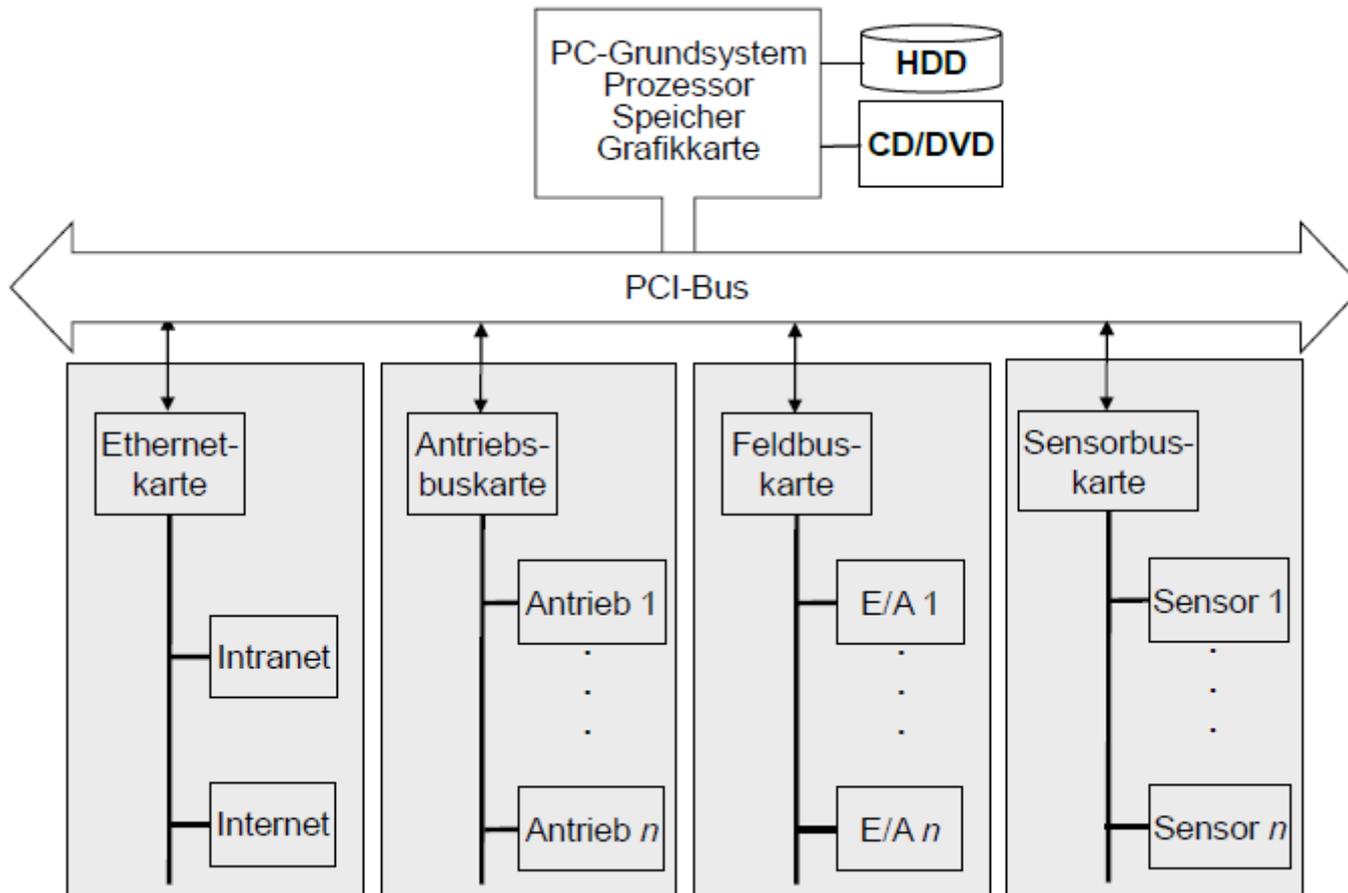
# Glass Production Facility (Fieldbus Systems)



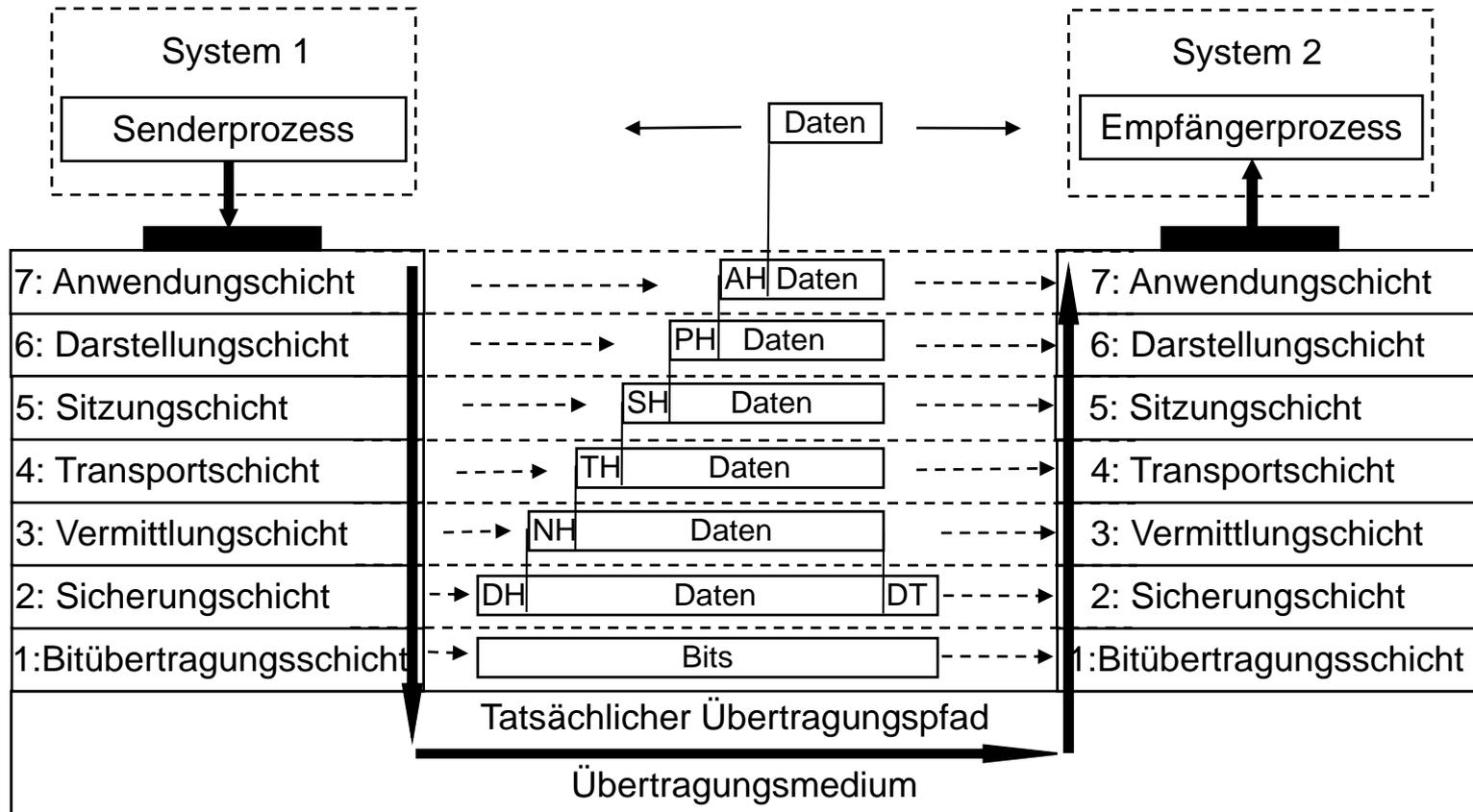
# Kommunikation in Fertigungsbetrieben

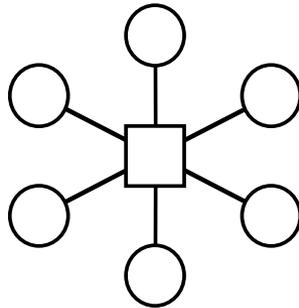


# IPC-basiertes Automatisierungssystem

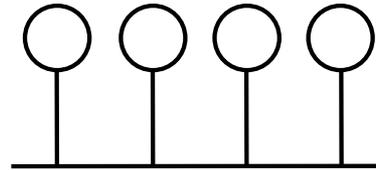


# ISO/OSI Referenzmodell der Kommunikation

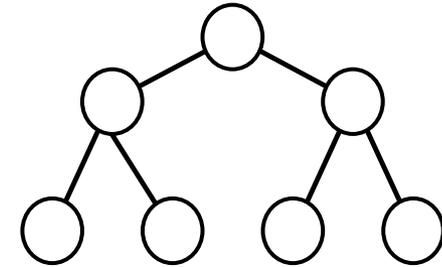




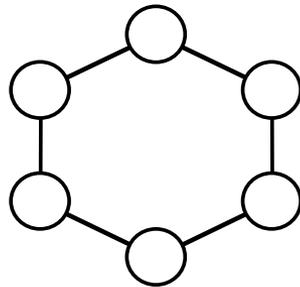
a)



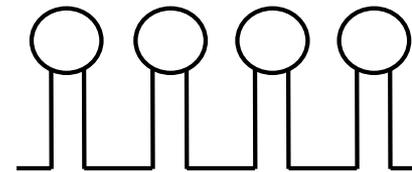
c)



e)



b)



d)

# Zugriffsverfahren auf Echtzeit-Kommunikationssysteme

Zugriffsverfahren auf Echtzeit-Kommunikationssysteme

Polling (zyklisches Abfragen)

CSMA/CD (Carrier Sense Multiple Access/ Collision Detection)

CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance)

Token-Passing

TDMA (Time Division Multiple Access)

# Geräte zum Aufbau von Netzwerken

|                                  | Koaxial-Kabel,<br>Twisted-Pair,<br>Glasfaser | Repeater | Hub | Switch | Bridge | Router | Gateway |
|----------------------------------|----------------------------------------------|----------|-----|--------|--------|--------|---------|
| <b>7: Anwendungsschicht</b>      |                                              |          |     |        |        |        | X       |
| <b>6: Darstellungsschicht</b>    |                                              |          |     |        |        |        | X       |
| <b>5: Sitzungsschicht</b>        |                                              |          |     |        |        |        | X       |
| <b>4: Transportschicht</b>       |                                              |          |     | (X)    |        |        | X       |
| <b>3: Vermittlungsschicht</b>    |                                              |          |     | (X)    |        | X      | X       |
| <b>2: Sicherungsschicht</b>      |                                              |          |     | X      | X      | X      | X       |
| <b>1: Bitübertragungsschicht</b> |                                              | X        | X   | X      | X      | X      | X       |
| <b>Übertragungsmedium</b>        | X                                            | X        | X   | X      | X      | X      | X       |

# Mögliche digitale Signalformen

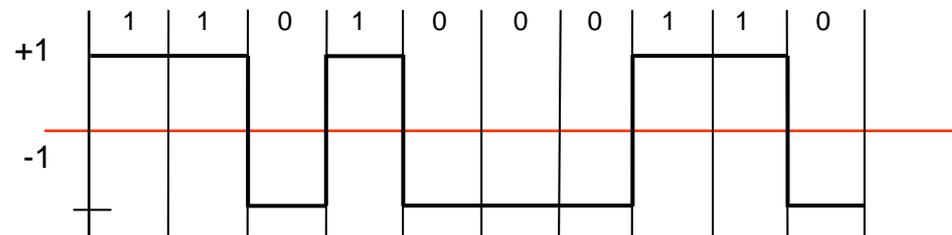
Information

1 1 0 1 0 0 0 1 1 0

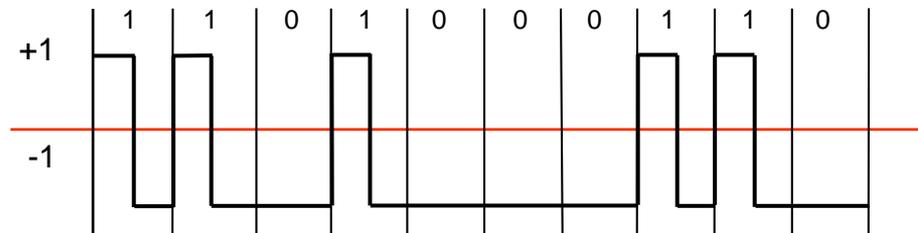
Grundtakt



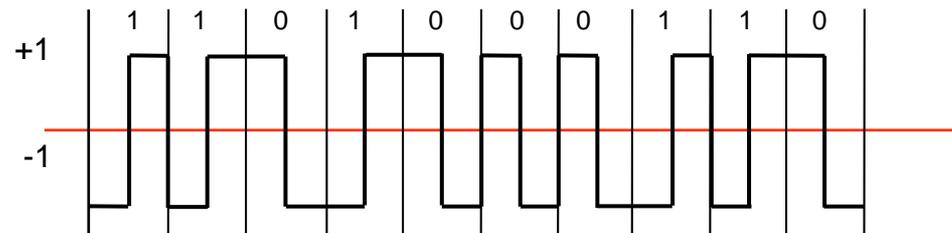
No Return to Zero (NRZ)



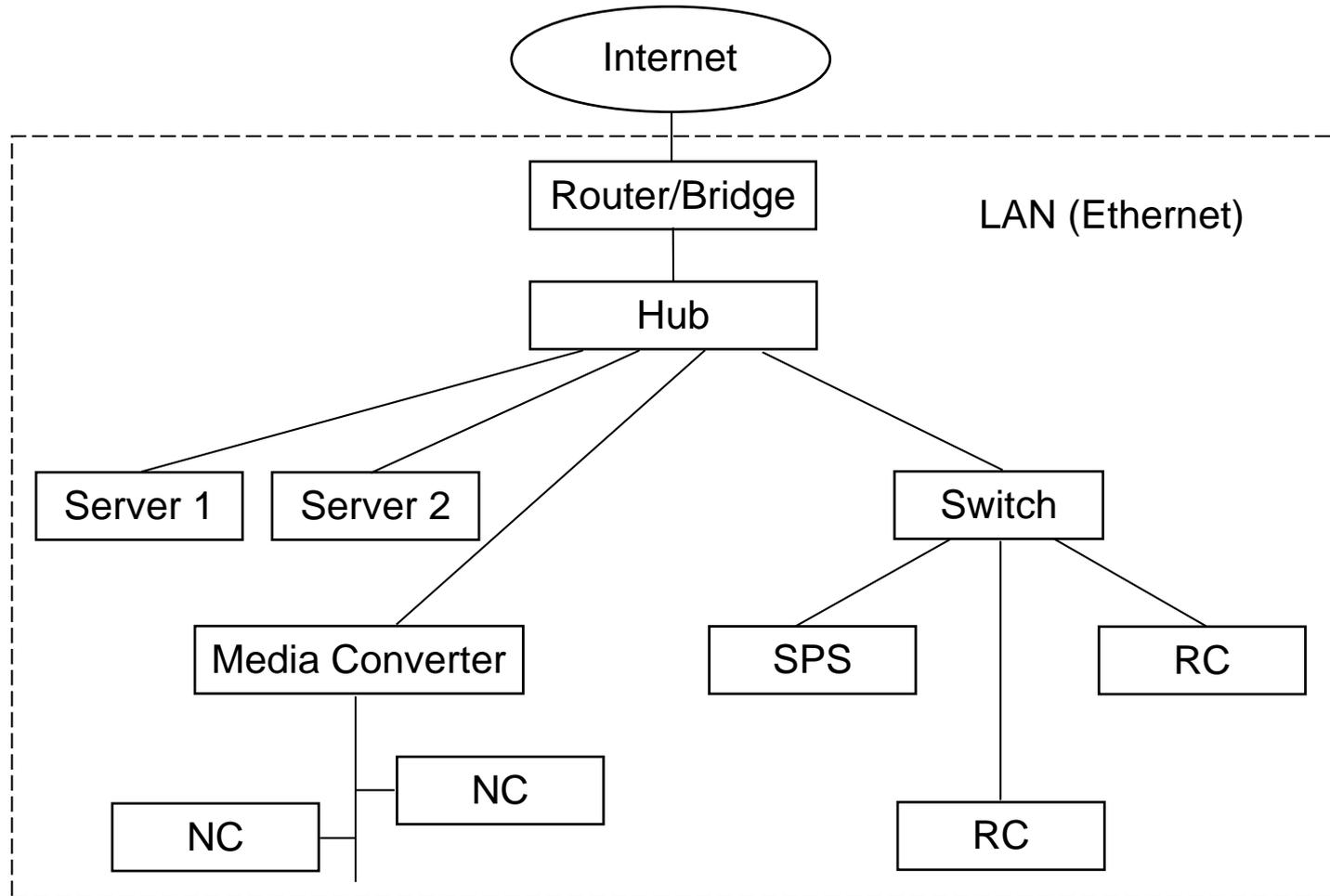
Return to Zero (RZ)



Manchester  
(NRZ **xor** Takt)

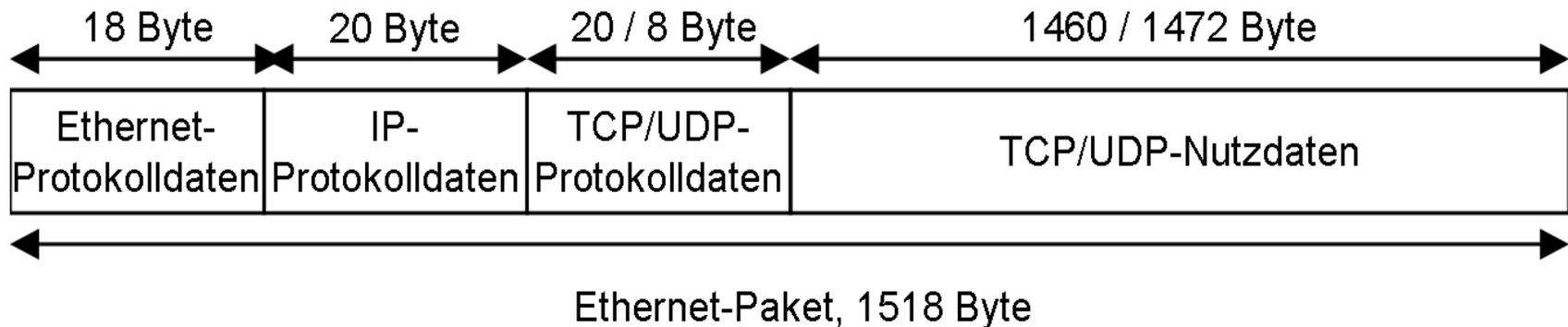


# Ethernet-basiertes lokales Netz



| OSI – Schicht      | Norm                                        | Protokoll                                      |
|--------------------|---------------------------------------------|------------------------------------------------|
| 7 : Anwendung      | JTM ISO 8832<br>FTAM ISO 8571<br>MHS X.400  | HTTP, FTP, SNMP, RPC,<br>SMTP, TELNET, NFS,... |
| 6 : Präsentation   | ISO 8823                                    | ASN.1, XDR                                     |
| 5 : Sitzung        | ISO 8327                                    | X.225                                          |
| 4 : Transport      | ISO 8073, ISO 8602                          | TCP, UDP, SPX                                  |
| 3 : Vermittlung    | ISO 8208, ISO 8473<br>X.25                  | IP, IPX, ICMP, ARP,<br>X.25                    |
| 2 : Sicherung      | IEEE802.2 (LLC )<br>IEEE802.3 (MAC)         | CSMA/CD, Token Ring,<br>ATM, X.25...           |
| 1 : Bitübertragung | ISO 8802<br>IEEE802.3, IEEE802.4...<br>X.21 | Protokollnorm Medium                           |

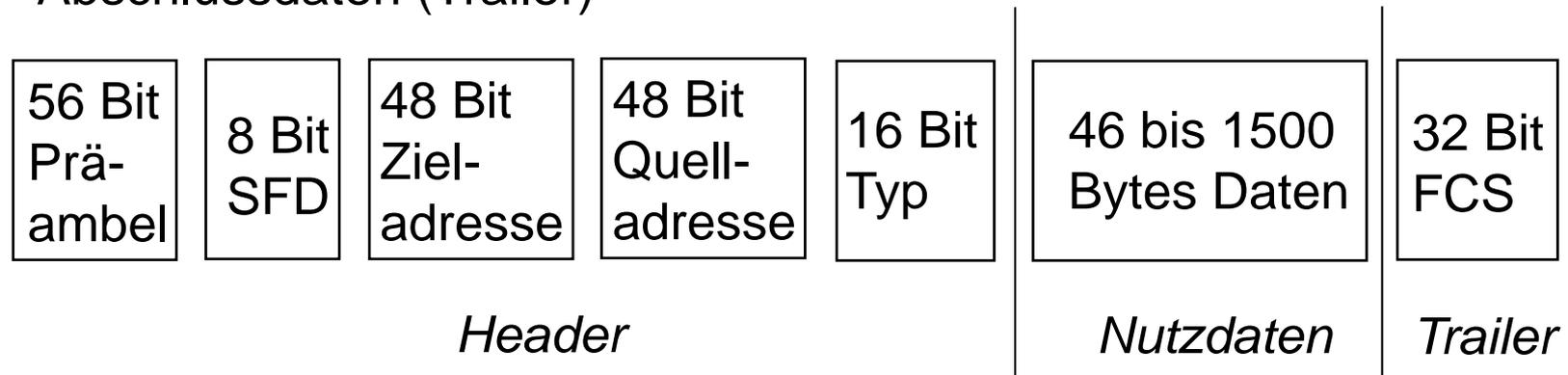
# Protokolldaten und Paketlängen bei TCP/UDP



**Protokolldaten = Header und Trailer**

# Ethernet-Frame (Ebene 2 im ISO/OSI Modell)

- **Ethernet-Frame** besteht aus Kopfdaten (Header), Nutzdaten und Abschlussdaten (Trailer)



- Präambel mit wechselnder Folge 1010 etc. zur Synchronisierung
- Start Frame Delimiter (SFD) kennzeichnet Beginn des Adressfelds
- Zieladressfeld umfasst 48 Bit
- Quelladressfeld umfasst 48 Bit
- Typfeld gibt Länge des Datenfelds an
- Datenfeld muss mindestens 46 Bytes umfassen, bei kürzeren Nachrichten Auffüllen mit Nullen; maximale Länge 1500 Bytes
- Frame Check Sequence mit CRC-Prüfsumme als Abschluss

# TCP/IP-Frame

|                                        |                                       |                           |                      |                                         |                                    |
|----------------------------------------|---------------------------------------|---------------------------|----------------------|-----------------------------------------|------------------------------------|
| Version/<br>Header<br>Length<br>1 Byte | Type of<br>Service<br>(TOS)<br>1 Byte | Total<br>Length<br>2 Byte | Identifier<br>2 Byte | Flags /<br>Fragment<br>Offset<br>2 Byte | Time to<br>Live (TTL)<br>1 Byte... |
|----------------------------------------|---------------------------------------|---------------------------|----------------------|-----------------------------------------|------------------------------------|

|     |                    |                              |                                |                                     |                     |                                          |
|-----|--------------------|------------------------------|--------------------------------|-------------------------------------|---------------------|------------------------------------------|
| ... | Protocol<br>1 Byte | Header<br>Checksum<br>2 Byte | Source IP<br>Address<br>4 Byte | Destination<br>IP Address<br>4 Byte | Options/<br>Padding | Data (in Ether-<br>net bis 1480<br>Byte) |
|-----|--------------------|------------------------------|--------------------------------|-------------------------------------|---------------------|------------------------------------------|

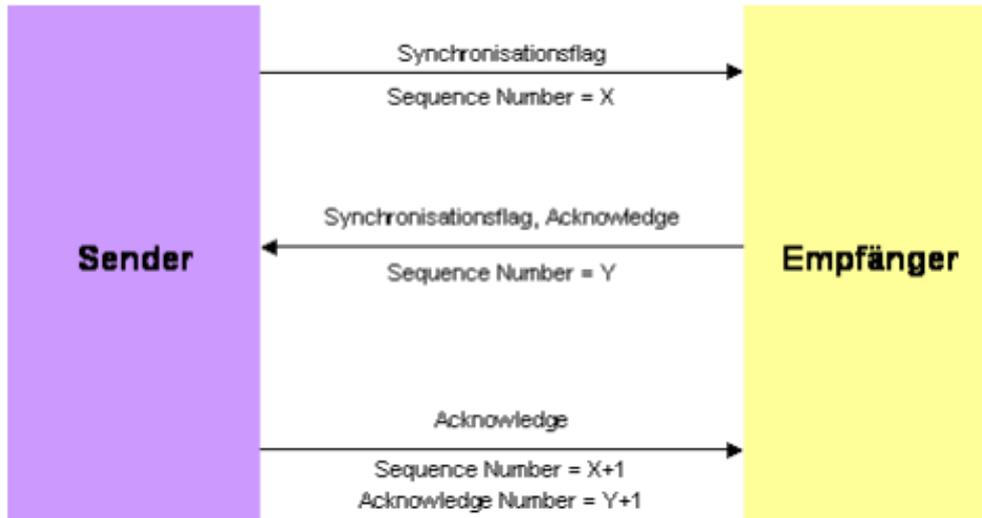
IP-Frame (Ebene 3 im ISO/OSI Modell)

|                          |                           |                              |                                   |                                             |
|--------------------------|---------------------------|------------------------------|-----------------------------------|---------------------------------------------|
| Source<br>Port<br>2 Byte | Destin.<br>Port<br>2 Byte | Sequence<br>Number<br>4 Byte | Acknowled-<br>ge Number<br>4 Byte | Data Offset/<br>Control Flags<br>2 Byte ... |
|--------------------------|---------------------------|------------------------------|-----------------------------------|---------------------------------------------|

|     |                  |                            |                             |                               |                         |
|-----|------------------|----------------------------|-----------------------------|-------------------------------|-------------------------|
| ... | Window<br>2 Byte | Control<br>(CRC)<br>2 Byte | Urgent<br>Pointer<br>2 Byte | Options/<br>Padding<br>4 Byte | Data<br>bis<br>64 kByte |
|-----|------------------|----------------------------|-----------------------------|-------------------------------|-------------------------|

TCP-Frame (Ebene 4 im ISO/OSI Modell)

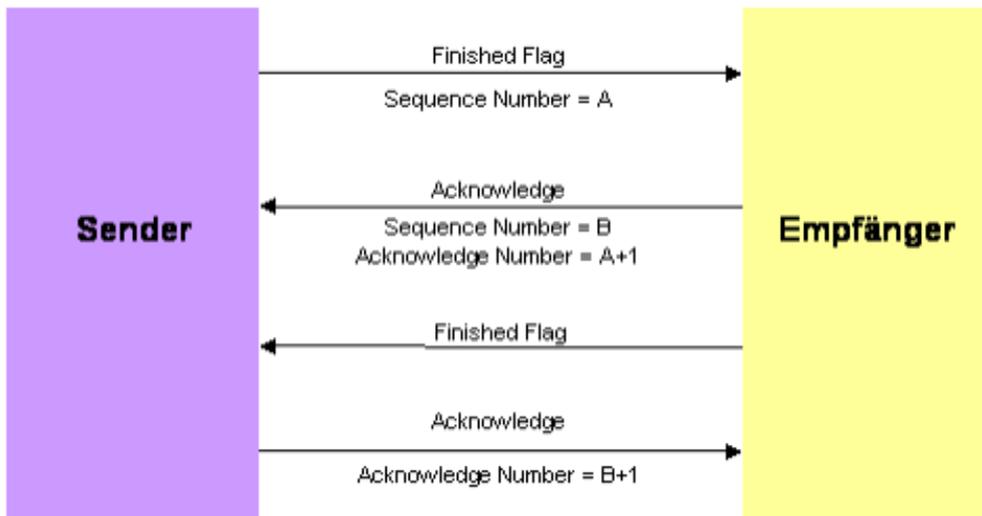
# Auf- und Abbau von TCP-Verbindungen



**1. Schritt:** Der Sender sendet ein Paket mit gesetztem Synchronisationsflag und der Sequence Number X.

**1. Schritt:** Der Empfänger bestätigt den Erhalt und sendet seinerseits ein Paket mit Synchronisationsflag und der eigenen Sequence Number Y.

**1. Schritt:** Der Sender bestätigt das Paket und sendet ein Paket mit Sequence Number X+1 und Acknowledge Number Y+1.



**1. Schritt:** Der Sender sendet ein Paket mit gesetztem Finish Flag und der Sequence Number A.

**1. Schritt:** Der Empfänger bestätigt den Erhalt und sendet ein Paket mit Acknowledge Flag, der eigenen Sequence Number B und der Acknowledge Number A+1.

**1. Schritt:** Der Empfänger sendet ein weiteres Paket mit dem Finished Flag.

**1. Schritt:** Der Sender bestätigt das Paket und sendet ein Paket mit Acknowledge Number B+1.

# UDP-Frame (auch: Datagramm)

|                       |                        |                       |               |      |
|-----------------------|------------------------|-----------------------|---------------|------|
| Source Port<br>2 Byte | Destin. Port<br>2 Byte | Data Length<br>2 Byte | CRC<br>2 Byte | Data |
|-----------------------|------------------------|-----------------------|---------------|------|

Die Bedeutung der einzelnen Felder ist:

## Quellportnummer

bezeichnet die Portnummer des Anwenderschichtprogrammes (Rechners), von dem die UDP-Message abgeschickt wurde.

## Zielportnummer

bezeichnet die Portnummer des Empfängerprogramms (Rechners), auf der Anwendungsschicht.

## Länge

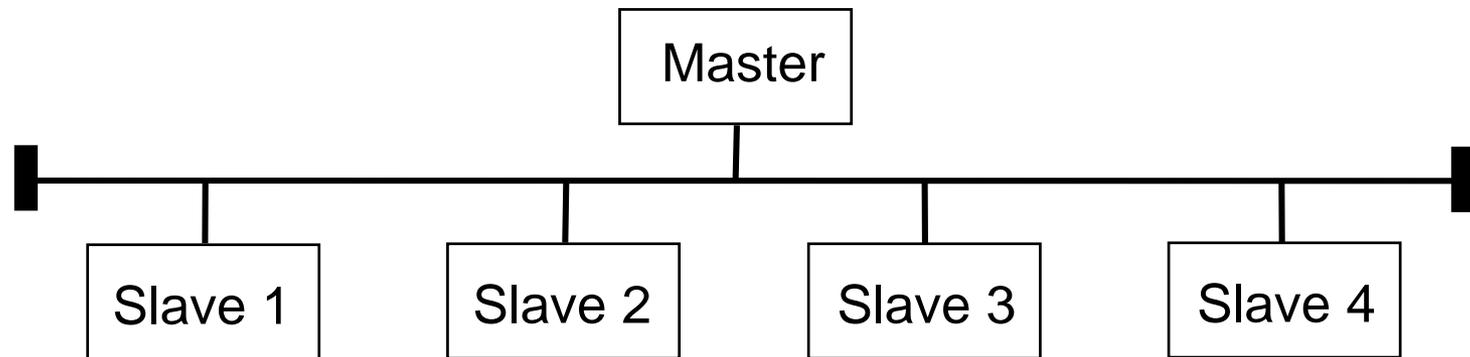
Hier steht die Länge der gesamten UDP-Message (incl. Header)

## Prüfsumme

Eine Prüfsumme über das Datenfeld

## Üblicher Einsatz des PROFIBUS - DP:

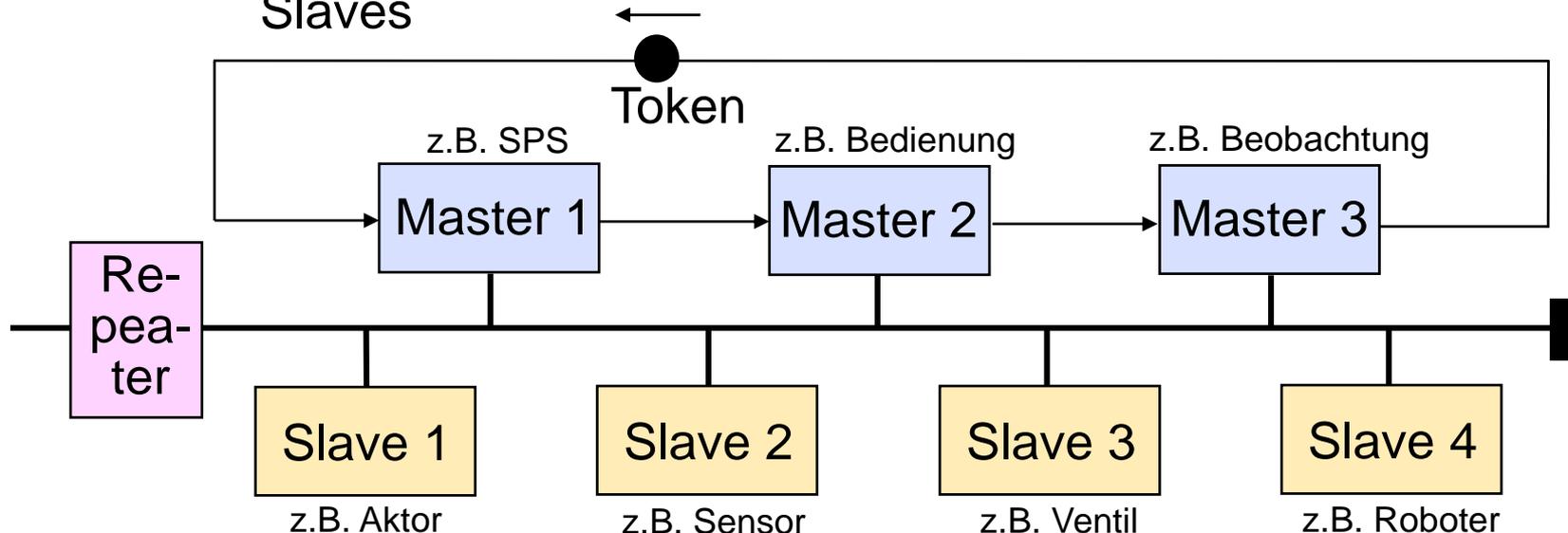
Einsatz nur eines Masters (*Mono-Master-System*)



Slaves z.B.:

Aktor, Sensor, Ventil, Antrieb, Bedienen, Schweißsteuerung, Roboter

- Hybrides Buszugriffsverfahren:
  - ⇒ Token-Passing unter den aktiven Stationen (Multi-Master)
  - ⇒ Polling zwischen einem Master und den dazugehörigen Slaves



- PROFIBUS-Telegramm: max. 244 Datenbytes pro Telegramm

|            |              |               |               |            |           |           |
|------------|--------------|---------------|---------------|------------|-----------|-----------|
| Start-byte | Ziel-adresse | Quell-adresse | Kontroll-byte | Daten-feld | Prüf-byte | Ende-byte |
|------------|--------------|---------------|---------------|------------|-----------|-----------|

# Telegrammtypen bei PROFIBUS

Typ 1:

|       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|
| SB    | ZA    | QA    | FC    | PB    | EB    |
| 8 Bit |

Typ 2:

|       |       |       |       |       |       |       |           |       |       |
|-------|-------|-------|-------|-------|-------|-------|-----------|-------|-------|
| SB    | Lä    | LäW   | SB    | ZA    | QA    | FC    | Dat ...   | PB    | EB    |
| 8 Bit | 0-1952Bit | 8 Bit | 8 Bit |

Typ 3:

|       |       |       |       |               |       |       |
|-------|-------|-------|-------|---------------|-------|-------|
| SB    | ZA    | QA    | FC    | Dat0 ... Dat7 | PB    | EB    |
| 8 Bit | 8 Bit | 8 Bit | 8 Bit | 64 Bit        | 8 Bit | 8 Bit |

Typ 4:

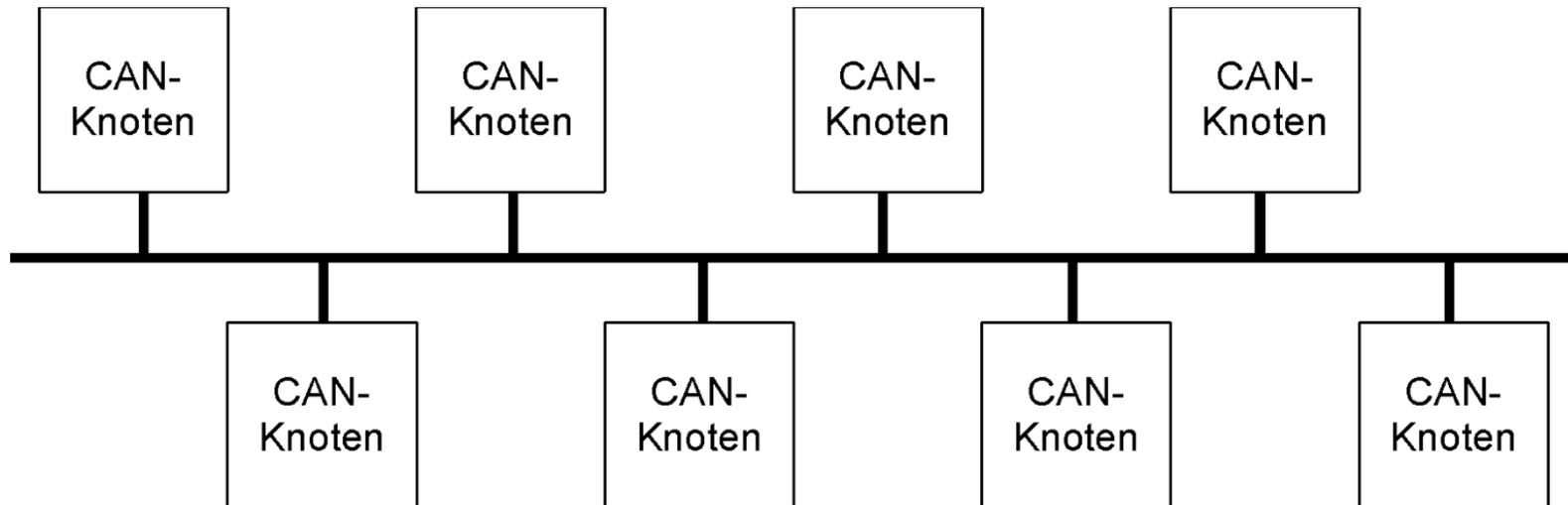
|       |       |       |
|-------|-------|-------|
| SB    | ZA    | QA    |
| 8 Bit | 8 Bit | 8 Bit |

| Abk. | Bedeutung                                          |
|------|----------------------------------------------------|
| SB   | Startbyte                                          |
| Lä   | Länge der Nutzdaten                                |
| LäW  | Wiederholung der Länge                             |
| ZA   | Zieladresse                                        |
| QA   | Quelladresse                                       |
| FC   | Funktionscode                                      |
| Dat  | Nutzdatenbytes (bis zu 244 bei Typ 2, 8 bei Typ 3) |
| PB   | Prüfbyte                                           |
| EB   | Endbyte                                            |

# Eigenschaften des PROFIBUS

|                                        |                                                                     |
|----------------------------------------|---------------------------------------------------------------------|
| Topologie                              | Linie mit Stichleitungen, abgeschlossen an beiden Enden             |
| Buslänge                               | max. 1,2 km ohne Repeater,<br>max. 10 km mit Repeater               |
| Übertragungsmedium                     | zweiadrig, verdrillt, abgeschirmt, seltener:<br>LWL                 |
| Anzahl Nutzdatenbytes<br>pro Telegramm | 0-244                                                               |
| Anzahl E/A- Stationen                  | max. 32 ohne Repeater,<br>max. 126 mit Repeater                     |
| Bitkodierung                           | NRZ-Kodierung bei FMS und DP<br>MBP (Manchester Bus Powered) bei PA |
| Übertragungsrate                       | 9,6 kbit/sec bis 12 Mbit/sec                                        |
| Übertragungssicherheit                 | CRC-Check (mit Hamming-Distanz 4)                                   |
| Buszugriffsverfahren                   | Polling + Token-Passing zwischen Mastern                            |
| Busverwaltung                          | Multimaster, Monomaster                                             |

# Topologie bei CAN (Linien- bzw. Bustopologie)



# Aufbau eines CAN Telegramms

## Aufbau eines CAN-2.0a Data Frame bzw. Remote Frame

|                     |                           |                    |                          |                               |
|---------------------|---------------------------|--------------------|--------------------------|-------------------------------|
| Startbit<br>(1 Bit) | Identifizier<br>(11 Bits) | RTR-Bit<br>(1 Bit) | Kontrollfeld<br>(6 Bits) | Datenfeld<br>(0..8 Bytes) ... |
|---------------------|---------------------------|--------------------|--------------------------|-------------------------------|

|     |                          |                     |                    |                     |                      |                       |
|-----|--------------------------|---------------------|--------------------|---------------------|----------------------|-----------------------|
| ... | CRC-Sequenz<br>(15 Bits) | CRC-Ende<br>(1 Bit) | ACK-BIT<br>(1 Bit) | ACK-Ende<br>(1 Bit) | Endefeld<br>(7 Bits) | Trennfeld<br>(3 Bits) |
|-----|--------------------------|---------------------|--------------------|---------------------|----------------------|-----------------------|

## Aufbau eines CAN-2.0b Data Frame bzw. Remote Frame

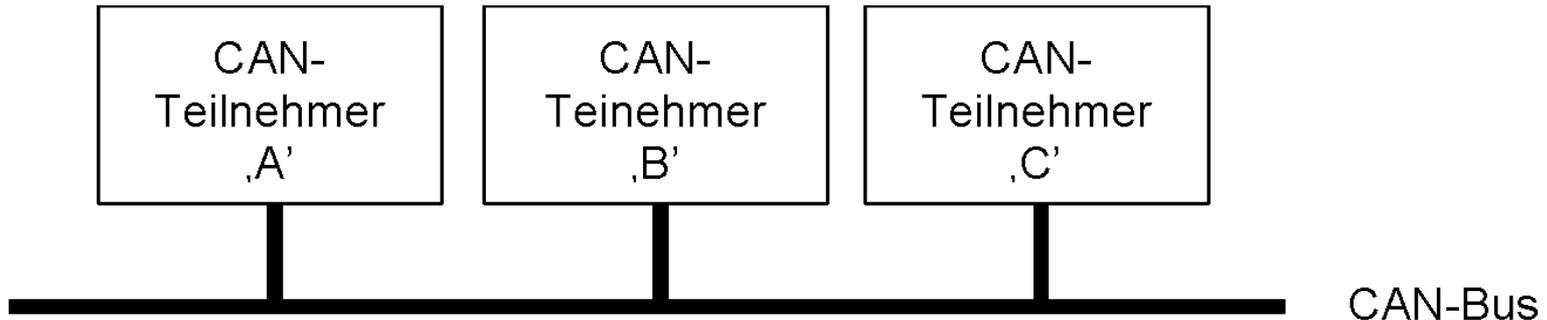
|                     |                           |                    |                    |                           |                    |                          |                               |
|---------------------|---------------------------|--------------------|--------------------|---------------------------|--------------------|--------------------------|-------------------------------|
| Startbit<br>(1 Bit) | Identifizier<br>(11 Bits) | SRR-Bit<br>(1 Bit) | IDE-Bit<br>(1 Bit) | Identifizier<br>(18 Bits) | RTR-Bit<br>(1 Bit) | Kontrollfeld<br>(6 Bits) | Datenfeld<br>(0..8 Bytes) ... |
|---------------------|---------------------------|--------------------|--------------------|---------------------------|--------------------|--------------------------|-------------------------------|

|     |                          |                     |                    |                     |                      |                       |
|-----|--------------------------|---------------------|--------------------|---------------------|----------------------|-----------------------|
| ... | CRC-Sequenz<br>(15 Bits) | CRC-Ende<br>(1 Bit) | ACK-BIT<br>(1 Bit) | ACK-Ende<br>(1 Bit) | Endefeld<br>(7 Bits) | Trennfeld<br>(3 Bits) |
|-----|--------------------------|---------------------|--------------------|---------------------|----------------------|-----------------------|

4 verschiedene Telegrammtypen:

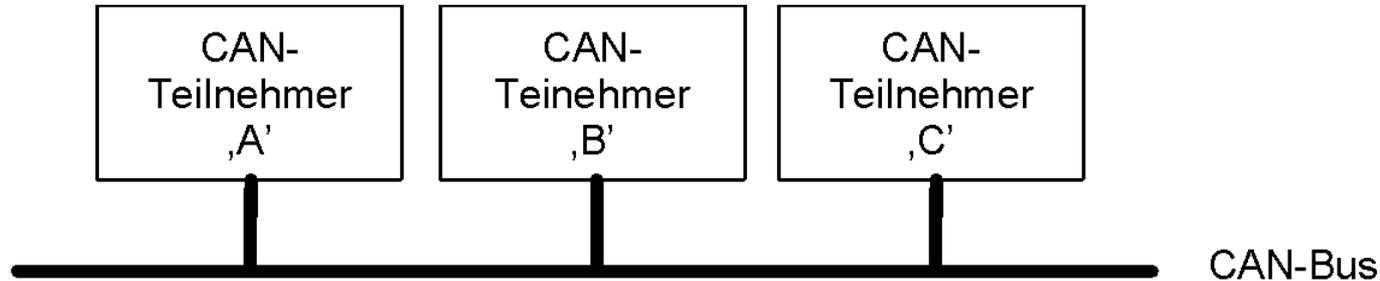
- ⇒ **Data Frame:** Zur Datenübertragung
- ⇒ **Remote Frame:** Sendeaufforderung an andere Teilnehmer
- ⇒ **Error Frame:** Meldung von Fehler an andere Teilnehmer
- ⇒ **Overload Frame:** Signalisation der aktuellen Nicht-Bereitschaft

# Dominante 0 und rezessive 1 bei der CAN Übertragung



| Fall                                       | 1 | 2 | 3 | 4 |
|--------------------------------------------|---|---|---|---|
| A sendet                                   | 0 | 0 | 1 | 1 |
| B sendet                                   | 0 | 1 | 0 | 1 |
| C empfängt<br>(resultierender<br>Buspegel) | 0 | 0 | 0 | 1 |

# Arbitrierung bei CAN



| Bit-Nr =                                                            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|----|----|
| A: Sendewunsch mit<br>Identifizier = $42_{10}$<br>= $00000101010_2$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1  | 0  |
| B: Sendewunsch mit<br>Identifizier = $24_{10}$<br>= $00000011000_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0  | 0  |
| C: empfängt<br>(resultierender Identifizier)                        | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0  | 0  |

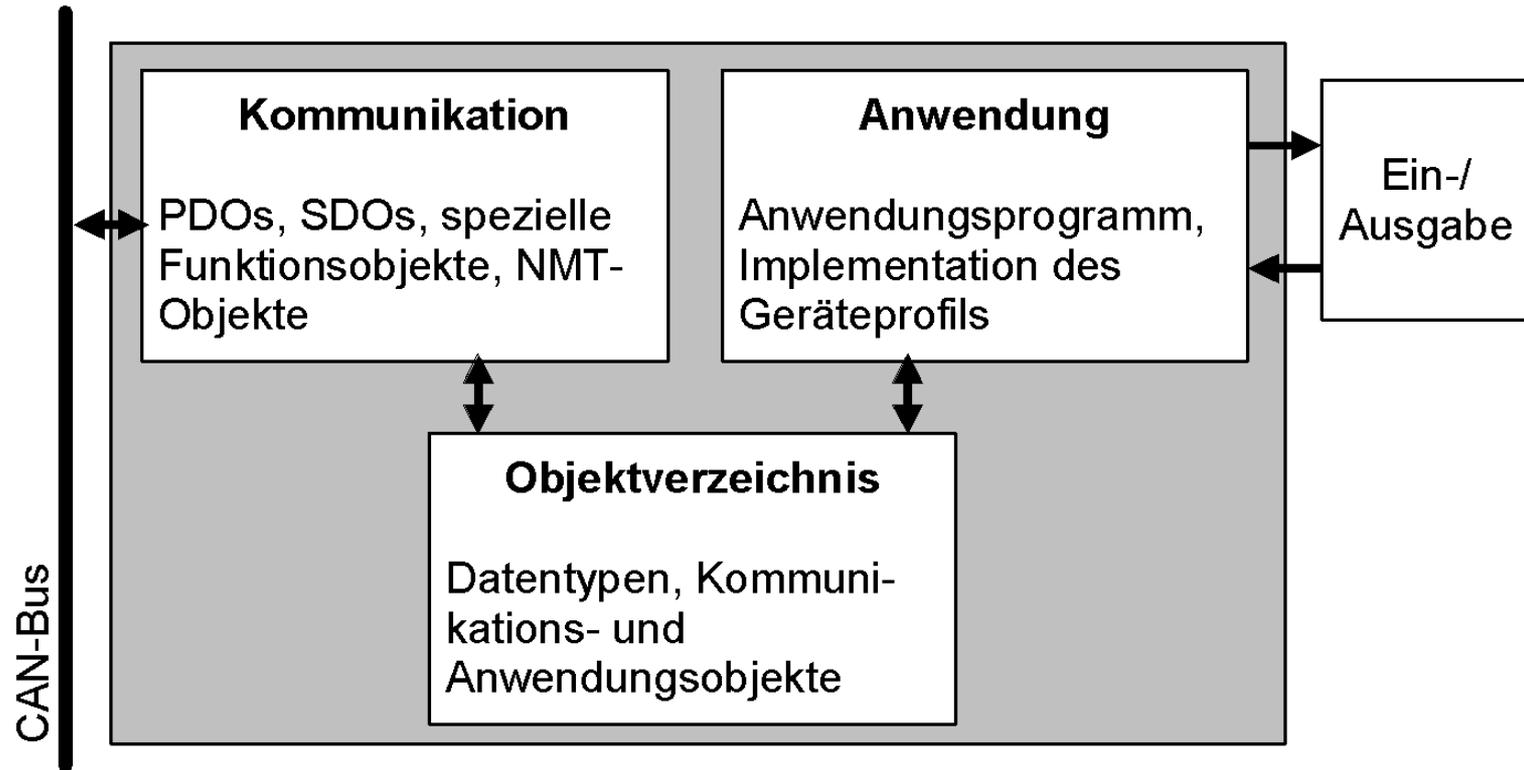
Bemerkung: Zum Zeitpunkt  $t=6$  erkennt A die höhere Priorität von B und zieht sich vom Senden zurück, danach sendet **nur** noch B.

# Eigenschaften des CAN-Bus (CAN-Basisprotokoll)

|                                     |                                                                                                                                          |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Topologie                           | Linie mit Stichleitungen, abgeschlossen an beiden Enden                                                                                  |
| Buslänge                            | 5 km bei 10 kbit/sec<br>25 m bei 1 Mbit/sec                                                                                              |
| Übertragungsmedium                  | zweiadrig, verdrillt, abgeschirmt, seltener: LWL                                                                                         |
| Anzahl Nutzdatenbytes pro Telegramm | 0 - 8                                                                                                                                    |
| Anzahl E/A Stationen                | Nur beschränkt durch die Treiberbausteine der CAN-Transceiver, nicht durch das Protokoll. Üblich: 30, mehr mit Repeatern/Spezialtreibern |
| Bitkodierung                        | NRZ-Kodierung mit dominanter 0 und rezessiver 1                                                                                          |
| Übertragungsrate                    | 10 kbit/sec bis 1 Mbit/sec                                                                                                               |
| Übertragungssicherheit              | CRC-Check (mit Hamming-Distanz 6)                                                                                                        |
| Buszugriffsverfahren                | Polling- oder ereignisgesteuerter Betrieb möglich (CSMA/CA: bitweise, nicht zerstörende Arbitrierung)                                    |
| Busverwaltung                       | Multimaster: Alle Teilnehmer sind gleichberechtigt, prioritätsgesteuert über Identifier                                                  |

# Schematische Darstellung des CANopen Gerätemodells

HLP CANopen realisiert Ebene 3 - 7



PDO: Prozessdatenobjekte für DÜ von Prozessdaten

SDO: Servicedatenobjekte - Zugriff auf Objektverzeichnis z.B. für Konfiguration, Parametrisierung

NMT: Netzwerkmanagement

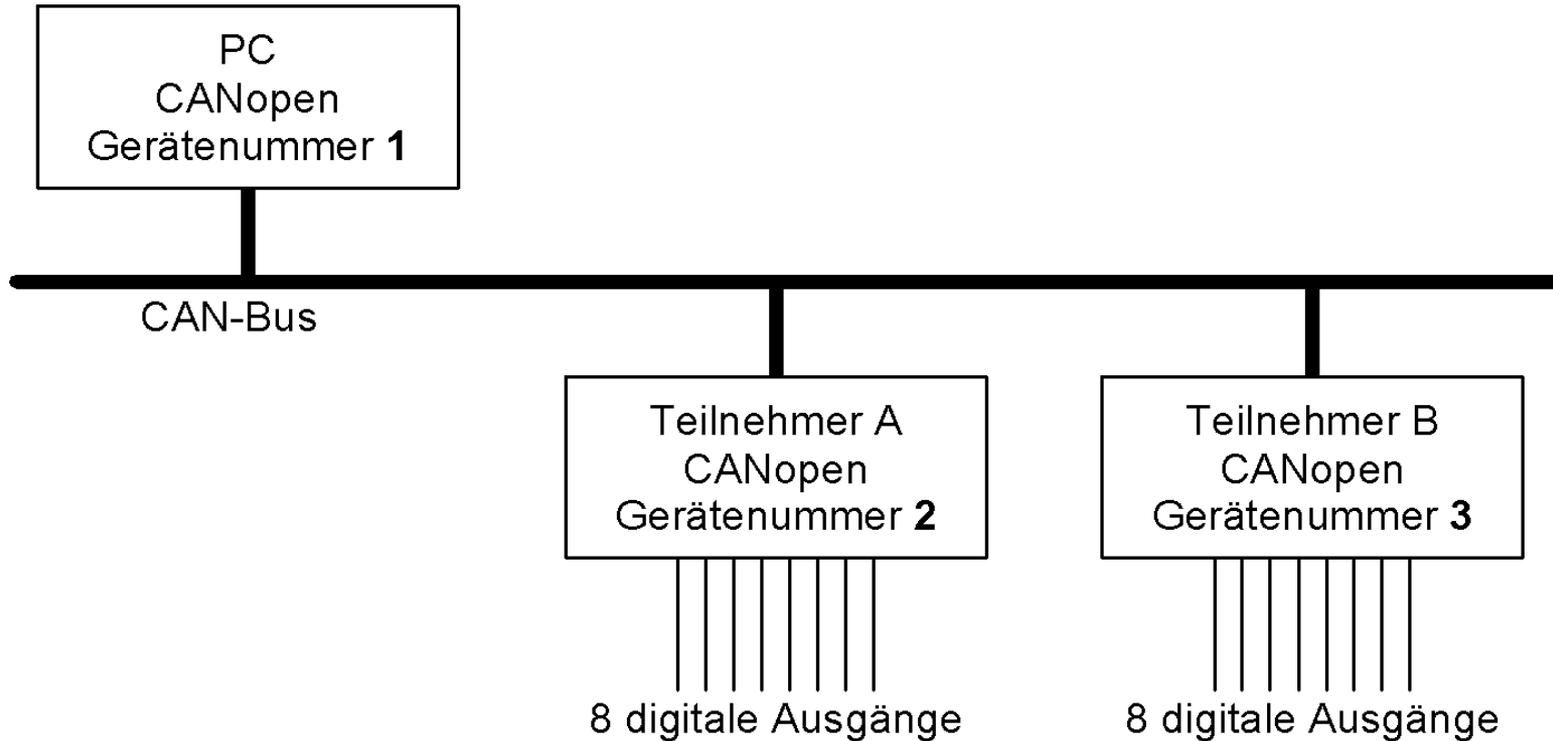
- Standardisierter Befehlssatz für die Datenübertragung in einer Domäne
- Standardisierter Ablauf der Übertragung
- Standardisierte Festlegung der Darstellung, Bedeutung der Daten, ...
- CANopen-Geräteprofil für Ein-/Ausgabe-Module (CiA DSP-401)
- CANopen-Geräteprofil für Antriebe (CiA DSP-402)
- CANopen-Geräteprofil für Encoder (CiA DSP-406)
- CANopen-Geräteprofil für Mensch-Maschine-Schnittstellen (CiA WD-403),
- CANopen-Geräteprofil für Messwertaufnehmer und Regler (CiA WD-404)
- CANopen-Geräteprofil für IEC-1131-kompatible Steuerungen (CiA WD-405).

# Zuordnung von Objekten, Funktionscodes und Identifiern bei CANopen

Identifizier = Teilnehmer Nummer (die 7 niederwertigen Bits des ID)  
+ Funktionscode (die 4 höherwertigen Bits des ID)

| Objekt     | Funktions-Code (binär) | Resultierender Identifizier (dezimal) | Typ          |
|------------|------------------------|---------------------------------------|--------------|
| NMT        | 0000                   | 0                                     | Broadcast    |
| SYNC       | 0001                   | 128                                   | Broadcast    |
| TIME-STAMP | 0010                   | 256                                   | Broadcast    |
| EMERGENCY  | 0001                   | 129 - 255                             | Peer-to-Peer |
| PDO1 (tx)  | 0011                   | 385 - 511                             | Peer-to-Peer |
| PDO1 (rx)  | 0100                   | 513 - 639                             | Peer-to-Peer |
| PDO2 (tx)  | 0101                   | 641 - 767                             | Peer-to-Peer |
| PDO2 (rx)  | 0110                   | 769 - 895                             | Peer-to-Peer |
| SDO (tx)   | 1011                   | 1409 - 1535                           | Peer-to-Peer |
| SDO (rx)   | 1100                   | 1537 - 1663                           | Peer-to-Peer |
| Nodeguard  | 1110                   | 1793 - 1919                           | Peer-to-Peer |

# Schematischer Aufbau eines einfachen CANopen Bussystems



# NMT Objekt (Starten) PC → Teilnehmer 2

|                      |                    |   |   |                                           |   |   |   |   |   |   |                   |            |     |
|----------------------|--------------------|---|---|-------------------------------------------|---|---|---|---|---|---|-------------------|------------|-----|
| CAN-<br>Bezeichnung  | Identifizier       |   |   |                                           |   |   |   |   |   |   | Kontroll-<br>feld | ...        |     |
| CANopen<br>Bedeutung | Funktions-<br>Code |   |   | Gerätenummer<br>(bei NMT nicht verwendet) |   |   |   |   |   |   | Daten-<br>länge   | ...        |     |
| Feldlänge            | (4 Bit)            |   |   | (7 Bit)                                   |   |   |   |   |   |   | (4 Bit)           | ...        |     |
| Wert                 | 0                  | 0 | 0 | 0                                         | 0 | 0 | 0 | 0 | 0 | 0 | 0                 | <b>0x2</b> | ... |

|     |              |                   |   |   |   |   |   |   |
|-----|--------------|-------------------|---|---|---|---|---|---|
| ... | Datenbytes   |                   |   |   |   |   |   |   |
| ... | NMT-<br>Code | Geräte-<br>nummer | - | - | - | - | - | - |
| ... | (1 Byte)     | (1 Byte)          | - | - | - | - | - | - |
| ... | <b>0x01</b>  | <b>0x02</b>       | - | - | - | - | - | - |

NMT-Code 01 bedeutet Starten

# NMT Objekt (Starten) PC → Teilnehmer 3

|                      |                    |   |   |                                           |   |   |   |   |   |   |                   |     |     |
|----------------------|--------------------|---|---|-------------------------------------------|---|---|---|---|---|---|-------------------|-----|-----|
| CAN-<br>Bezeichnung  | Identifizier       |   |   |                                           |   |   |   |   |   |   | Kontroll-<br>feld | ... |     |
| CANopen<br>Bedeutung | Funktions-<br>Code |   |   | Gerätenummer<br>(bei NMT nicht verwendet) |   |   |   |   |   |   | Daten-<br>länge   | ... |     |
| Feldlänge            | (4 Bit)            |   |   | (7 Bit)                                   |   |   |   |   |   |   | (4 Bit)           | ... |     |
| Wert                 | 0                  | 0 | 0 | 0                                         | 0 | 0 | 0 | 0 | 0 | 0 | 0                 | 0x2 | ... |

|     |              |                   |   |   |   |   |   |   |
|-----|--------------|-------------------|---|---|---|---|---|---|
| ... | Datenbytes   |                   |   |   |   |   |   |   |
| ... | NMT-<br>Code | Geräte-<br>nummer | - | - | - | - | - | - |
| ... | (1 Byte)     | (1 Byte)          | - | - | - | - | - | - |
| ... | <b>0x01</b>  | <b>0x03</b>       | - | - | - | - | - | - |

# PDO (Setzen von Ausgängen) PC → Teilnehmer 2

|                   |                |   |   |   |              |   |   |   |   |   |              |     |     |
|-------------------|----------------|---|---|---|--------------|---|---|---|---|---|--------------|-----|-----|
| CAN-Bezeichnung   | Identifizier   |   |   |   |              |   |   |   |   |   | Kontrollfeld | ... |     |
| CANopen Bedeutung | Funktions-Code |   |   |   | Gerätenummer |   |   |   |   |   | Datenlänge   | ... |     |
| Feldlänge         | (4 Bit)        |   |   |   | (7 Bit)      |   |   |   |   |   | (4 Bit)      | ... |     |
| Wert              | 0              | 1 | 0 | 0 | 0            | 0 | 0 | 0 | 0 | 1 | 0            | 0x1 | ... |

|     |                     |   |   |   |   |   |   |   |
|-----|---------------------|---|---|---|---|---|---|---|
| ... | Datenbytes          |   |   |   |   |   |   |   |
| ... | 8 digitale Ausgänge | - | - | - | - | - | - | - |
| ... | (1 Byte)            | - | - | - | - | - | - | - |
| ... | <b>0xff</b>         | - | - | - | - | - | - | - |

# PDO (Setzen von Ausgängen) PC → Teilnehmer 3

|                      |                    |   |   |              |   |   |   |   |   |   |                   |     |     |
|----------------------|--------------------|---|---|--------------|---|---|---|---|---|---|-------------------|-----|-----|
| CAN-<br>Bezeichnung  | Identifizier       |   |   |              |   |   |   |   |   |   | Kontroll-<br>feld | ... |     |
| CANopen<br>Bedeutung | Funktions-<br>Code |   |   | Gerätenummer |   |   |   |   |   |   | Daten-<br>länge   | ... |     |
| Feldlänge            | (4 Bit)            |   |   | (7 Bit)      |   |   |   |   |   |   | (4 Bit)           | ... |     |
| Wert                 | 0                  | 1 | 0 | 0            | 0 | 0 | 0 | 0 | 0 | 1 | 1                 | 0x1 | ... |

|     |                        |   |   |   |   |   |   |   |
|-----|------------------------|---|---|---|---|---|---|---|
| ... | Datenbytes             |   |   |   |   |   |   |   |
| ... | 8 digitale<br>Ausgänge | - | - | - | - | - | - | - |
| ... | (1 Byte)               | - | - | - | - | - | - | - |
| ... | <b>0xf0</b>            | - | - | - | - | - | - | - |