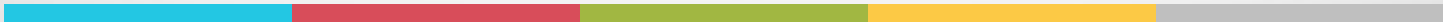
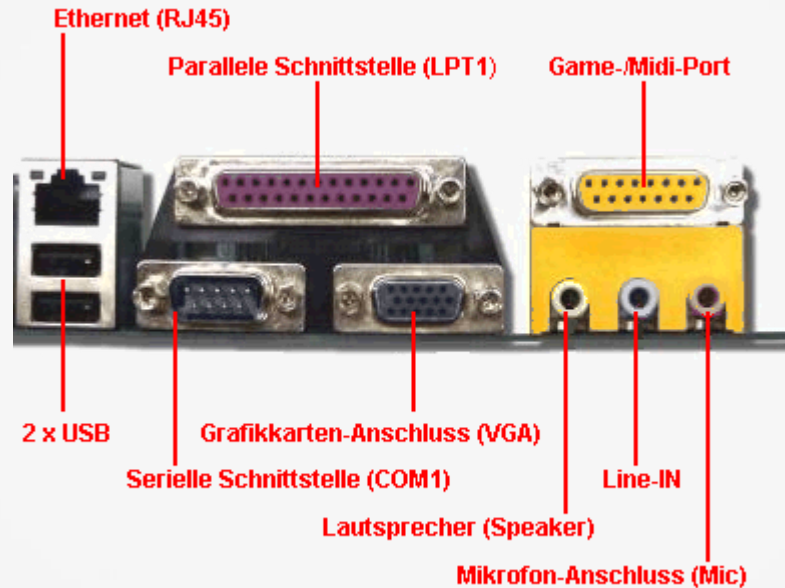


# **Serielle Kommunikation** - Grundlagen



# Übersicht Schnittstellen

---



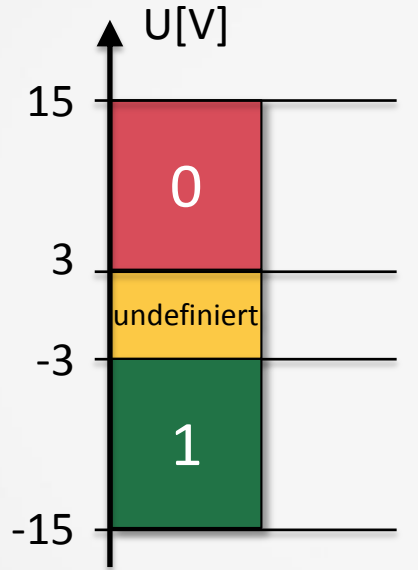
# Serielle Datenübertragung

---

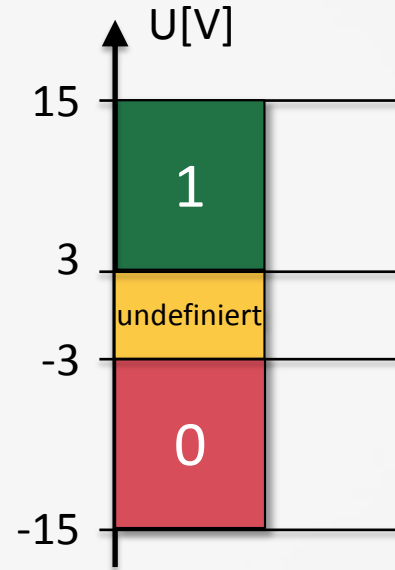
- ✓ Seriell: Daten werden nacheinander übertragen
- ✓ Bekannte serielle Schnittstellen:
  - ✓ USB
  - ✓ Ethernet
  - ✓ SDATA(Seriell Advanced Technology Attachment)
- ✓ Codierung kommt meistens der ASCII-Code zum Einsatz
- ✓ Die RS-232 Schnittstelle arbeitet (bit-)seriell: Sendet die Daten nacheinander auf der entsprechenden Datenleitung.
- ✓ Um die Daten weiterzuverarbeiten werden UARTs (Universal Asynchronous Receiver Transmitter) eingesetzt

# Spannungen bei RS-232

---



Datenleitungen  
(negative Logik)



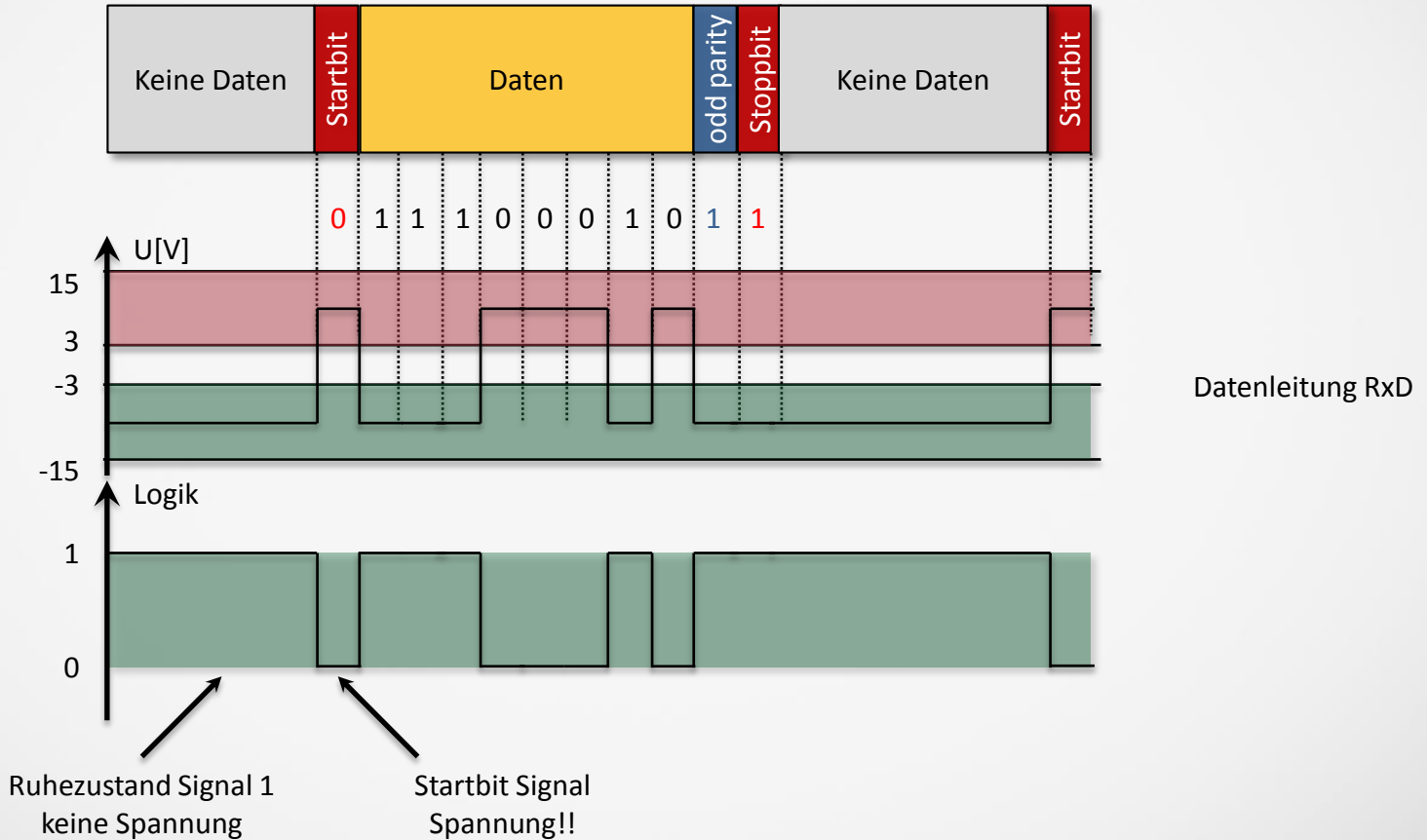
Steuerleitungen  
(positive Logik)

# Prinzip der asynchronen Datenübertragung I

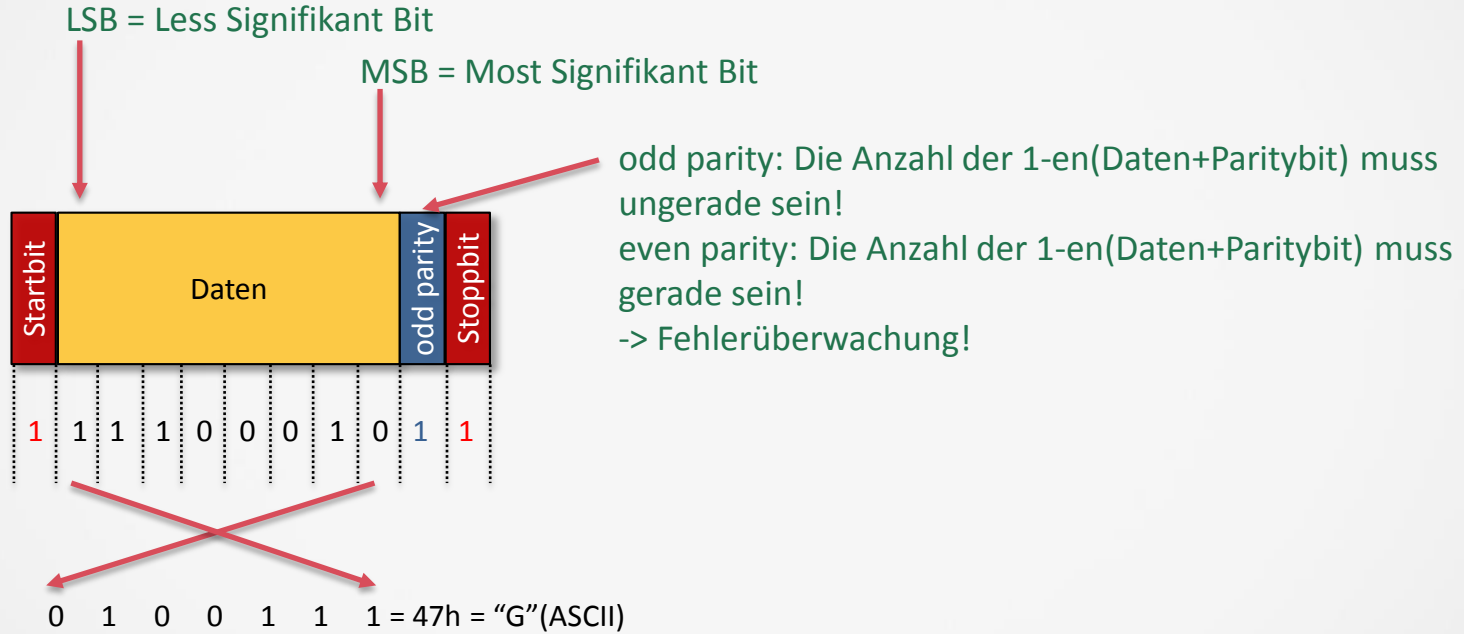
---

- ✓ Asynchron: Synchronisation bzw. Taktung muss bei der seriellen Übertragung auf der Datenleitung durch die Daten selbst erfolgen
- ✓ Die Zeit zwischen zwei Übertragungen kann beliebig lang sein kann und es gibt keine gemeinsame Taktleitung
- ✓ Einem zu übertragenden "Daten-Wort" werden deshalb Steuerinformationen gemäß dem RS-232-Protokoll voran- und nachgestellt:
  - ✓ 1 Bit: Startbit
  - ✓ 5-8 Bit: Daten
  - ✓ 1 Bit: Parity Bit
  - ✓ 1-2 Bit: Stoppbits
- ✓ Die Baudrate(Anzahl Symbol/Sekunde) muss bei Empfänger und Sender gleich sein!

# Datenübertragung bei RS-232 – Teil I



# Datenübertragung bei RS-232 – Teil II



# Datenflusskontrolle

---

- ✓ Um eine verlustfreie Datenübertragung zu ermöglichen müssen sich Sender und Empfänger darüber einigen, wann Daten gesendet werden.
  - Software-Handshake
  - Hardware-Handshake
- ✓ Verfahren:
  - XON/XOFF-Protokoll (nur Software Handshake)
  - ETX/ACK-Protokoll (Software Handshake mit einer Steuerleitung)
  - READY/BUSY-Protokoll (nur Hardware Handshake)
  - XModem-Protokoll (nur Software Handshake)