

1 Einführung

05.09.2025

Aufgabe 1.1: Serielle Kommunikation

1. Beantworte folgende Fragen schriftlich:

- An welcher Stelle spielt die serielle Kommunikation heutzutage eine Rolle?
- Erkläre die Begriffe: Startbit, Datenbit, Stoppbit.
- Ein PC sendet den Buchstaben 'A' mit 1 Startbit, 8 Datenbits, 1 Stoppbit. Skizziere das resultierende Bitmuster
- Welche Parameter müssen Sender und Empfänger bei RS232 vorab gemeinsam einstellen?
- Warum reicht ein einziger Draht für die Übertragung?

2. Aufgabe 2 – Serielle Kommunikation (RS232) in Java mit Hilfe eines Emulators

- Warum brauchen Sender und Empfänger die gleiche Baudrate?
- Was passiert, wenn der Sender schneller schreibt als der Empfänger lesen kann? Welche Lösungen gibt es für dieses Problem?

Lösung 1.1: Serielle Kommunikation

1. Beantworte folgende Fragen schriftlich:

- Serielle Kommunikation wird heute noch in eingebetteten Systemen, Industrieanlagen, Messgeräten und auch beim Serverzugriff (Konsolenport) genutzt, da sie einfach, robust und für kurze Distanzen ausreichend ist.
- **Startbit:** signalisiert Beginn eines Zeichens (logisch 0) und dient zur Synchronisation. **Datenbits:** eigentliche Nutzinformation, meist 8 Bit, LSB zuerst. **Stopbit:** beendet die Übertragung (logisch 1), Leitung geht in Idle-Zustand.
- Beispiel: Der Buchstabe A hat den ASCII-Wert $0x41 = 01000001$. Übertragung (LSB zuerst) mit 1 Startbit und 1 Stopbit:

$\underbrace{1}_{\text{Idle}} \quad \underbrace{0}_{\text{Start}} \quad 10000010 \quad \underbrace{1}_{\text{Stop}} \quad 1$

- Sender und Empfänger müssen sich bei RS232 vorab einigen auf: Baudrate, Datenbits, Parität, Stoppbits, sowie ggf. Art der Flusskontrolle.
- Ein einzelner Draht genügt pro Richtung, weil Bits nacheinander mit fester Baudrate gesendet werden; Start- und Stoppbits übernehmen die Synchronisation. Für echte Vollduplex-Kommunikation sind jedoch zwei Leitungen (Tx/Rx) plus Masse üblich.

2. Aufgabe 2 – Serielle Kommunikation (RS232) in Java mit Hilfe eines Emulators

- Beide Seiten brauchen dieselbe Baudrate, da es kein separates Taktsignal gibt. Unterschiedliche Baudraten führen zu falscher Bitinterpretation.
- Wenn der Sender schneller schreibt als der Empfänger liest, läuft dessen Puffer über und Daten gehen verloren. Lösungen: Hardware-Flowcontrol (RTS/CTS), Software-Flowcontrol (XON/XOFF), größere FIFO-Puffer oder Protokolle mit Bestätigung (ACK/NACK).

```

1  import com.fazecast.jSerialComm.SerialPort;
2  public class Sender {
3      public static void main(String[] args) throws Exception {
4          SerialPort sp = SerialPort.getCommPort("COM5");
5          sp.setBaudRate(9600);
6          sp.openPort();
7          sp.getOutputStream().write("Hallo, COM6\n".getBytes());
8          sp.closePort();
9      }
10 }
```