

02 Serielle Schnittstelle

1 Ziel

Die asynchrone serielle Schnittstelle wird im professionellen Bereich nach wie vor häufig verwendet, um über eine Konsole Geräte (z.B. Router oder Embedded Controller) zu konfigurieren.

- Bestimmung der wichtigsten Parameter und Einrichten einer asynchronen seriellen Schnittstelle.
- Einarbeitung in die Bedienung eines Oszilloskops.

2 Vorbereitung

- Studieren Sie den **Anhang A** zur asynchronen seriellen Schnittstelle.

Q01 Welche Spannung entspricht bei der RS232-Schnittstelle einer logischen 1 resp. einer logischen 0?

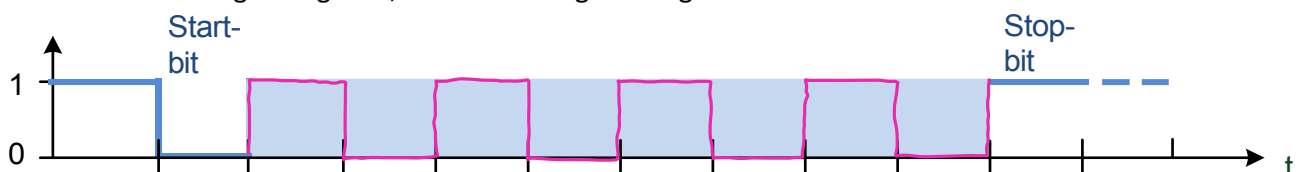
0: +3 ... +15 Volt, 1: -3 ... -15 Volt

Im Folgenden wird eine asynchrone serielle Schnittstelle mit 115'200 Baud; 8 Daten-Bit, ohne Parity und einem Stop-Bit verwendet (siehe auch Abbildung 6):

Q02 Wie lange ist die Zeit für die Übertragung eines Bits?

$$\text{Bitzeit} = \frac{1}{\text{Baudrate}} \Rightarrow \frac{1}{115200} \approx 8.68 \mu\text{s} \quad (\text{Mikrosekunden})$$

- Zeichnen Sie eine Signalfolge ein, bei der sich logische Signale 1 und 0 abwechseln.



Q03 Welche Frequenz hat das Rechtecksignal ausgedrückt durch T und als numerischer Wert in Hertz?

$$f = \frac{1}{\text{Bitzeit}} \Rightarrow \frac{1}{8.68 \cdot 10^{-6} \text{ s}} \approx 115'207 \text{ Hz} \quad \frac{115'207}{2} = 57,6 \text{ kHz}$$

Q04 Welchem Zeichen entspricht diese Signalfolge?

10101010 \Rightarrow 01010101 \Rightarrow "U" (LSB first)

- Wählen Sie in der ASCII-Tabelle ein anderes, beliebiges Zeichen aus.

Q05 bestimmen Sie den Binärwert und zeichnen Sie das Zeitdiagramm der Übertragung in Abbildung 1 ein.

Zeichen: x

ASCII-Wert: 01111000

Sendefolge:

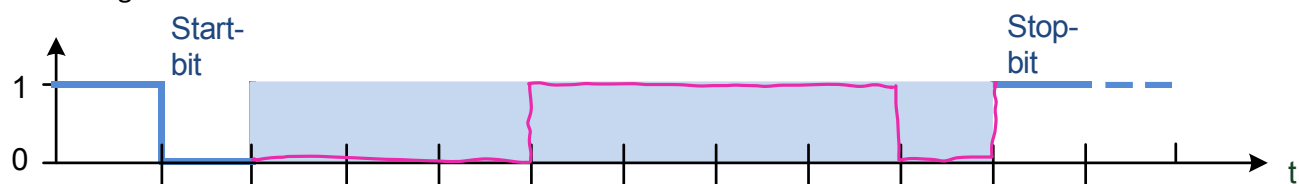


Abbildung 1

Abbildung 2 zeigt für die minimale und maximale Senderfrequenz die Grenzsituationen, wo ein idealer Empfänger gerade noch richtig funktioniert oder eben nicht mehr.

Q06 Wie gross darf der Fehler des Senders sein? Formulieren Sie die Bedingung. (Es ist keine Rechnung gefragt!)

$\frac{1}{2} T$ auf $9\frac{1}{2} T$ (in beide Richtungen)

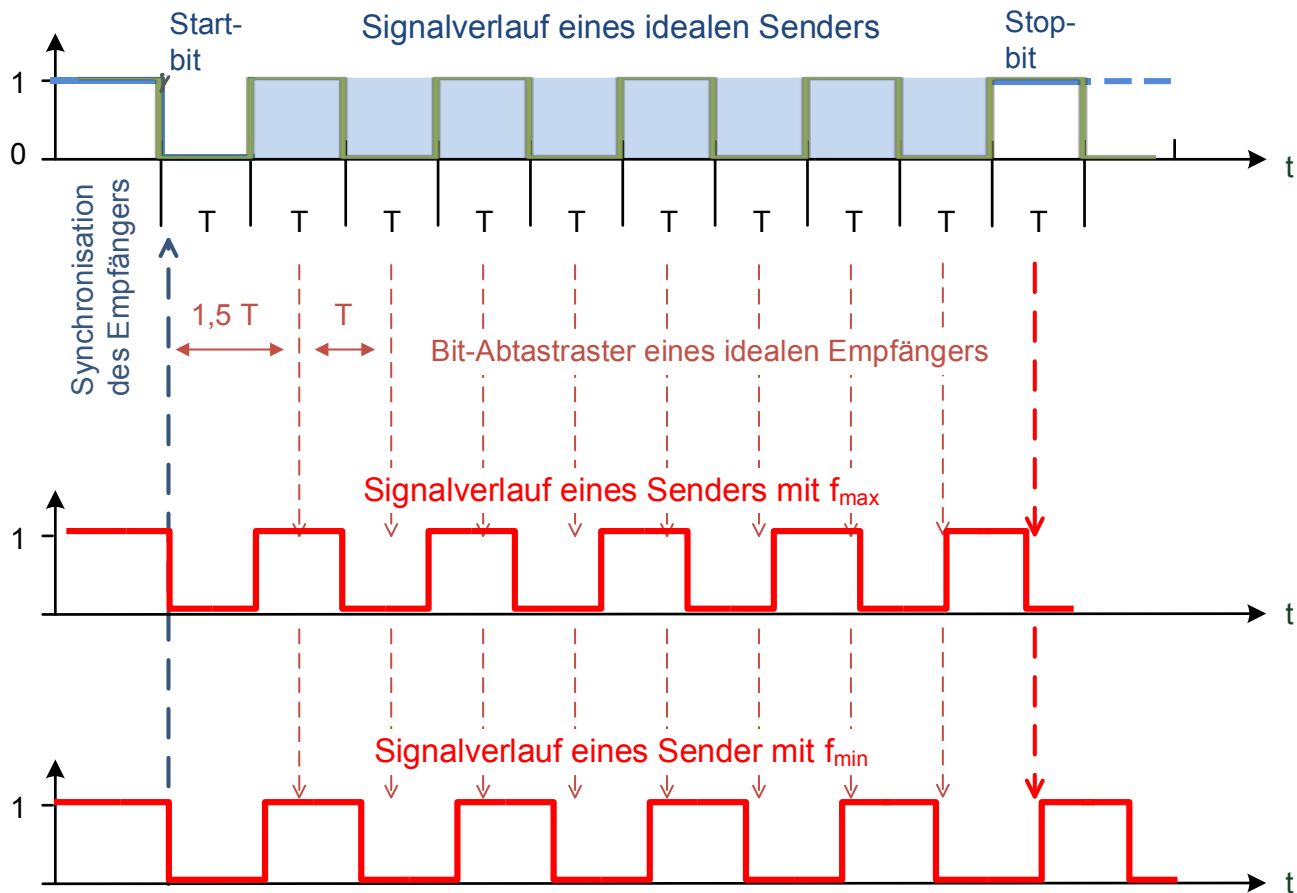


Abbildung 2

Berechnen Sie die minimalen und maximalen Senderfrequenzen und tragen Sie diese in Tabelle 1 ein.

$$9\frac{1}{2} T = 57,6 T$$

Berechnung: Formel	f [kHz]	Messung ¹	Die Toleranz der Messung ist: ¹
$f_{\max} = \frac{10}{9,5}$	60,6 kHz	59,4 kHz	1,2 kHz
$f_{\min} = \frac{9}{9,5}$	54,5 kHz	54,3 kHz	0,2 kHz

Tabelle 1

Zeigen Sie Ihre Vorbereitungen dem Laborbetreuer.



¹ Wird im Abschnitt 3 «Messung von Toleranzwerten mit Funktionsgenerator» bestimmt.

3 Messung von Toleranzwerten mit Funktionsgenerator

- Schliessen Sie den Funktionsgenerator an das Oszilloskop an (Abbildung 3).
- Stellen Sie mit Hilfe des Oszilloskops ein Rechtecksignal ein, das die richtigen Signal-Pegel für eine RS232-Schnittstelle und die richtige Frequenz für 115'200 Baud hat (siehe Abschnitt 2).

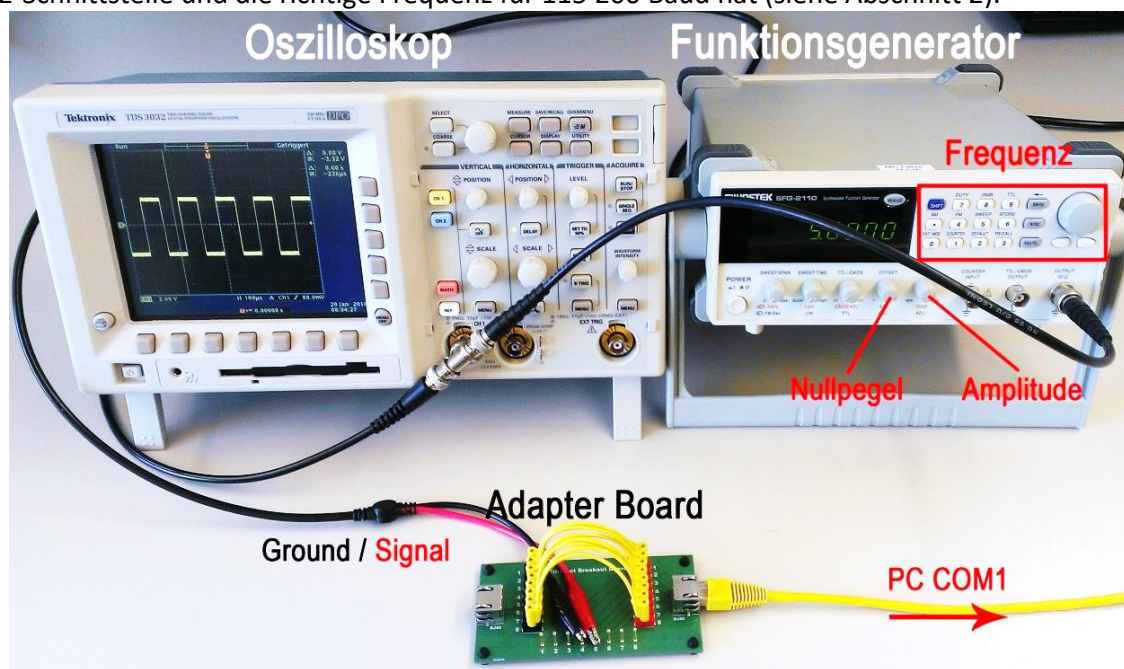


Abbildung 3

- Führen Sie Signal und Ground des Funktionsgenerators über ein Adapter-Board auf die COM1-Schnittstelle eines PCs. Auf dem Adapter-Board sind folgende Signale vorhanden (Bezeichnungen sind aus Sicht des PCs zu lesen):

Pin	Abkürzung	Beschreibung	Anschlussklemme des Kabels
1	DSR	Data Set Ready	
2	DCD	Data Carrier Detect	
3	DTR	Data Terminal Ready	
4	GND	Signal Ground	Schwarz (Ground)
5	RD	Received Data	Rot (Signal)
6	TD	Transmitted Data	
7	CTS	Clear To Send	
8	RTS	Request To Send	

- Starten Sie einen PC unter Linux und das Terminal-Emulations-Programm `putty`. Überprüfen Sie die Einstellungen (COM1, 115'200 Baud, 8 Daten-Bit, keine Parity-Bit, 1 Stop-Bit, keine Flow-Control).
- Variieren Sie am Drehknopf des Funktionsgenerators die Frequenz, bis der Empfang nicht mehr richtig funktioniert. Tragen Sie minimalen und maximalen Senderfrequenzen in Tabelle 1 ein (3 Stellen).
- Vergleichen Sie in Tabelle 1 die Werten jeweils für f_{\max} und f_{\min} . und tragen Sie ein, ob die Frequenztoleranz der Messung gegenüber der Berechnung grösser oder kleiner ist.

Q07 Was gibt es für die Abweichungen plausible Erklärungen (insbesondere für die von f_{\min})?

• Diverse Toleranzen: Frequenzgenerator, Schnittstelle zum PC

• Error Code Connecting ??

• Tastet etwas später ab als in der Mitte

- Verkleinern Sie am Drehknopf die Amplitude bis der Empfang nicht mehr richtig funktioniert. Bestimmen Sie den Grenzwert und vergleichen Sie diesen mit dem Standard.

$\approx 1,85 V$

Q08 Genügt der Empfänger den Spezifikationen für eine RS232-Schnittstelle? (Begründung)

ja, der spezifizierte Bereich 3-15 V



Zeigen Sie die Resultate dem Laborbetreuer.

4 Messungen an einer Konsolenverbindung

- Entfernen Sie den Funktionsgenerator. Verbinden Sie stattdessen den Konsolen-Eingang (ttyS0) einer Embedded Linux Box (ELB) über das Adapter-Board mit der COM1-Schnittstelle des PCs (Abbildung 4. und starten Sie eine Konsolenverbindung (**putty**) zur ELB.

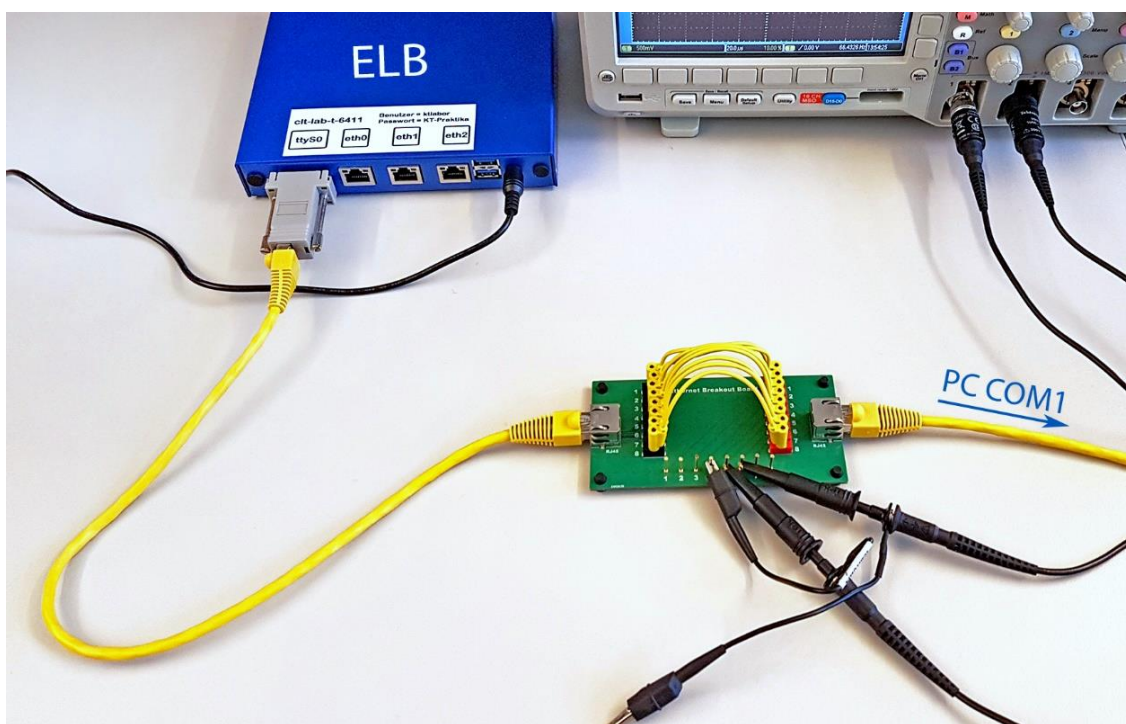


Abbildung 4

- Verwenden Sie für den Anschluss des Oszilloskops Proben. Verbinden Sie einen Erdungsanschluss mit Pin 4, Kanal 1 mit Pin 6 (TD) und Kanal 2 mit Pin 5 (RD).
- Passen Sie Oszilloskop-Einstellungen an: Verwenden Sie «Normal» Triggerung auf Kanal 1, damit das Bild bis zum nächsten Trigger-Ereignis stehen bleibt. Schalten Sie im «Horizontal Menu» den «Delay» aus und setzen Sie die «Horizontal Position» auf 10%, damit möglichst viel vom Signal angezeigt wird.
- Damit wir keine störenden Ausgaben der Shell bekommen, starten Sie auf der ELB ein Programm, das nichts macht, ausser Zeichen von der Konsole entgegenzunehmen:
 - `cat > /dev/null`

- Drücken Sie auf der Tastatur das Zeichen, das Sie in der Vorbereitung für die Abbildung 1 gewählt hatten. Vergleichen Sie die von Ihnen gezeichnete Kurvenform mit der vom Oszilloskop angezeigten.

Q09 Gibt es Abweichungen? Falls ja, worin bestand der Fehler?

Signal ist invertiert, und leicht abgerundet

Q10 Welche Oszilloskop-Einstellung bewirkt, so dass das Bild unserer gewohnten Darstellung entspricht?

Invert

Q11 Vergleichen Sie die Signalpegel des PC (Kanal 1) und der Embedded Linux Box (Kanal 2). $K1 : 1.1 \text{ V}$

Amplitude: Kanal 2 < Kanal 1, extrem verschoben $K2 : 0.8 \text{ V}$

Q12 Wie gross ist der Delay zwischen TD/RD und wodurch entsteht er?

ca. 1,2 - 1,4 ms

Q13 Sehen Sie bei der Übertragung von «normalen» Zeichen weitere Unterschiede auf den TD/RD-Leitungen?

K2 Amplitude schwankt, & Delay

Q14 Vergleichen Sie TD und RD beim Drücken der «Enter»-Taste. Erklären Sie, warum diese nicht gleich sind.

K2 produziert ein in die Länge gezogen

"Carriage return, KT>" wird dargestellt

(mit richtigen Einstellungen ist zus. "new line")

Zeigen Sie Ihre Vorbereitungen dem Laborbetreuer.



5 Weitere Ideen (eigene sind erwünscht):

- Verbinden Sie zwei Konsolen über zwei COM-Schnittstellen (putty zweimal starten, kann auch auf demselben PC sein). Sie benötigen eine Null-Modem-Verbindung (Abbildung 10), die Sie mit Hilfe des Adapter-Boards erstellen.
- «Chatten» Sie über die Leitung und probieren Sie verschiedene Steuerzeichen (CR, LF, BS etc.) aus.
- Stellen Sie die Bitrate um (putty Terminal-Einstellungen).
- Untersuchen Sie die Flow-Control-Mechanismen z.B.
 - Öffnen Sie z.B. auf dem Adapter-Board alle Leitungen ausser 4, 5 und 6. Überprüfen Sie die Funktion mit und ohne Flow-Control (siehe Terminal-Einstellungen).
 - Testen Sie Soft-Flow-Control (mit Steuer-Zeichen)
 - Testen Sie Hard-Flow-Control (mit Signal-Leitungen)

Anhang A: Serielle asynchrone Übertragung

A.1 Typische Einsatzgebiete

Typische Einsatzgebiete für serielle asynchrone Schnittstellen sind:

- Anschluss von Modems und ISDN-Adapter an eine DTE
- Anschluss von Terminals an einen Rechner
- Einfache PC-PC-Verbindungen
- Feldbusse in der Automatisierungstechnik
- Anschluss von Endgeräten an den PC (Drucker, Digitalkameras usw.)
- Verbinden von Systemeinheiten in Embedded Systems wie Mikrokontroller, Modems, Drucker, GPS usw.

A.2 Herkunft

Serielle asynchrone Schnittstellen wurden ursprünglich entwickelt, um Datenübertragungseinrichtungen (z.B. Modems) an Datenendeinrichtungen (z.B. Terminals, PCs) anzuschliessen. Darum sind die Eigenschaften der Schnittstellen auf diese Anwendung ausgelegt.

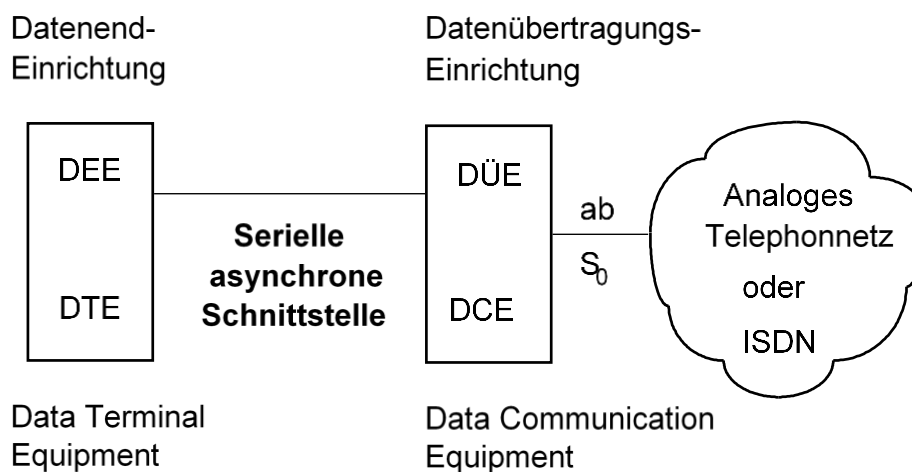


Abbildung 5

Mit der starken Verbreitung von PC und Embedded Systemen wurden solche serielle asynchrone Schnittstellen häufig für andere Zwecke, wie z.B. zum Verbinden zweier Rechner und der Anschaltung anderer Systemeinheiten benutzt.

A.3 Das Funktionsprinzip im Detail

Die Daten werden seriell pro Byte in der folgenden Reihenfolge übertragen: das niederwertigste Daten-Bit (LSB = Least Significant Bit) zuerst, das höchstwertige Daten-Bit (MSB = Most Significant Bit) zuletzt. Die Übertragung wird als asynchron bezeichnet, weil kein Takt für die Bitsynchronisation übertragen wird.

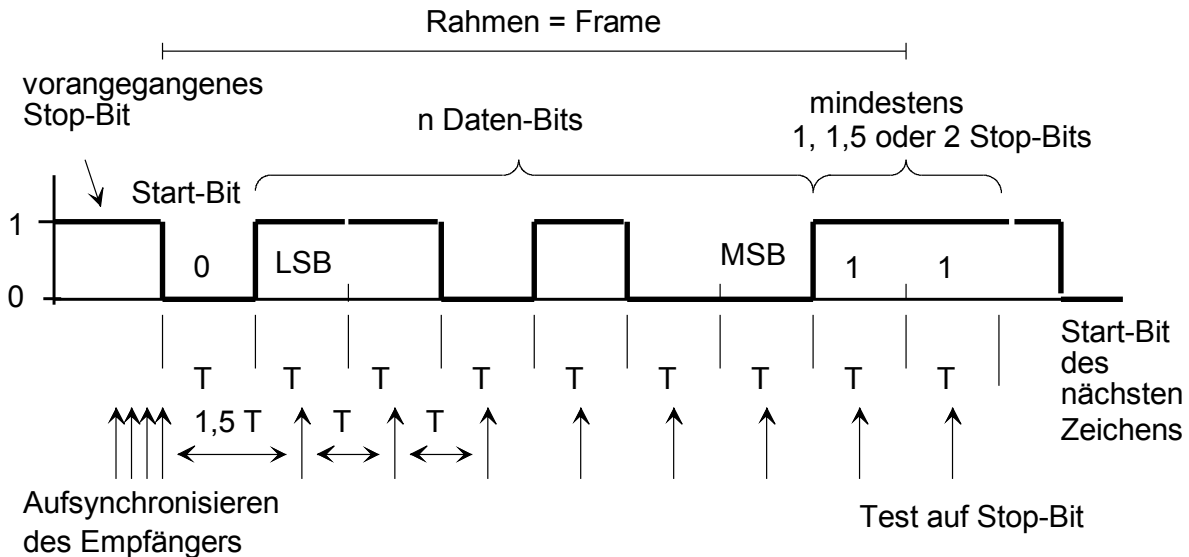


Abbildung 6

Der Sender und der Empfänger besitzen eigene, unabhängige Taktquellen mit annähernd gleicher Frequenz. Jedes übertragene Zeichen wird separat mit Hilfe eines Start- und Stop-Bits synchronisiert. Für jedes Zeichen sendet der Sender ein Start-Bit, n Daten-Bits und zum Abschluss 1-2 Stop-Bits.

Am Anfang jedes Zeichens synchronisiert sich der Empfänger auf den Sender. Dazu wird die Leitung mit einer höheren Frequenz als der Bitrate abgetastet, bis die fallende Flanke des Start-Bits erkannt wird. Dann werden die n Daten-Bits asynchron transferiert, d.h. der Empfänger muss die Sendefrequenz kennen. Kleine Taktdifferenzen zwischen Sender und Empfänger werden durch Stop-Bits aufgefangen. Zwischen Sender und Empfänger müssen Taktfrequenz, Anzahl übermittelte Daten-Bits und Anzahl Stop-Bits abgemacht sein.

Die n Daten-Bits bestehen aus Nutzdaten und in vielen Fällen aus einem zusätzlichen Parity-Bit für die Erkennung von einfachen Übertragungsfehlern. Der Sender bildet eine Parity über die Nutzdaten (Even oder Odd Parity) und überträgt dieses zusätzliche Bit als letztes Bit. Der Empfänger überprüft das Parity-Bit der empfangenen Nutzdaten mit dem selbst gebildeten Parity-Bit. Zwischen Sender und Empfänger muss natürlich abgemacht sein, ob mit geradem oder ungeradem Parity-Bit gearbeitet wird.

A.4 Die Schnittstelle RS232D

EIA-RS232 bezeichnet Normen der EIA (Electronic Industries Association). RS bedeutet Recommended Standard. Sie beinhalten die mechanischen, die elektrischen und die funktionalen Eigenschaften. D bezeichnet die Version. Die EIA-RS232 Schnittstelle ist elektrisch identisch mit der ITU-T-Norm V.28.

a) Elektrische Eigenschaften

Bei RS232D werden für die Datenleitungen Spannungspegel von +3...+15 Volt dem logisch Null und -3...-15 Volt dem logisch Eins zugeordnet. Für die Steuer- und Meldeleitungen ist diese Zuordnung umgekehrt. Die symmetrischen Spannungspegel bewirken, dass der DC-Signalanteil klein gehalten wird. Die hohe Pegeldifferenz ergibt einen grossen Störabstand. Auf diese Art werden

Leitungslängen von 10-100 m und Übertragungsraten von 1-20 KBit/s erreicht (offiziell 15 m, 20 KBit/s).

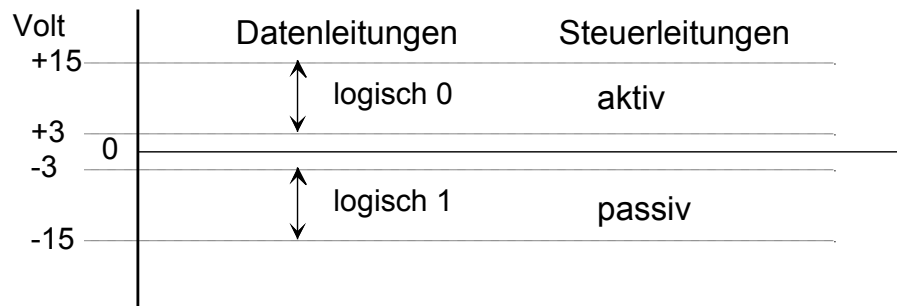


Abbildung 7

Ein Nachteil ist, dass spezielle Treiber- und Empfänger-Bausteine und die Spannungen +12 und -12 Volt verwendet werden müssen. Es gibt allerdings Bausteine, welche die notwendigen Spannungen intern aus den 5 Volt generieren (z.B. MAX 232).

b) Mechanische Eigenschaften

In der Norm RS232D wird ein 25poliger Cannon-Stecker verwendet. In der Praxis wird jedoch meist ein 9poliger Cannon-Stecker (Subminiatur D male) verwendet, der mit den PCs als defakto Standard eingeführt wurde.

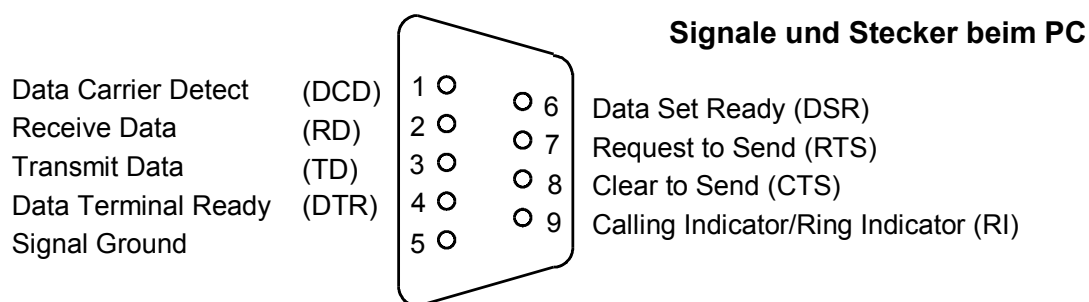


Abbildung 8

c) Funktionale Eigenschaften der Schnittstellen

In der Praxis werden meistens nur zwischen drei und neun dieser Signale verwendet. In der folgenden Tabelle sind die neun wichtigsten dieser Signale dargestellt:

Übertragungsleitungen:

Transmit Data (TD):	Datensendeleitung von der DTE zur DCE.
Receive Data (RD):	Datenempfangsleitung von der DCE zur DTE.

Flow-Control-Signale:

Request to Send (RTS):	Anforderung des DTE an das DCE sich für den Sendebetrieb vorzubereiten.
Clear to Send (CTS):	Mit diesem Signal zeigt das DCE an, dass der Sender eingeschaltet ist und dass sie bereit ist, Daten zu senden.

Modem-Control-Signale:

Data Terminal Ready (DTR):	Zeigt an, dass das DTE betriebsbereit ist.
Data Set Ready (DSR):	Zeigt an, dass das DCE betriebsbereit ist.
Data Carrier Detect (DCD):	Mit diesem Signal zeigt das DCE an, dass ein Trägersignal empfangen wird.

Ring Indicator (RI):

Meldung des DCE an die DTE, dass ein Rufsignal anliegt.

A.5 Verbinden von Rechnern über serielle Schnittstellen

Da diese Schnittstellen-Normen auch im Nahbereich verwendet werden, fehlt in vielen Fällen die Datenübertragungseinrichtung (z.B. ein Modem) und es werden zwei Rechner (DTE) direkt zusammengeschaltet. In diesem Fall müssen bestimmte Leitungen wie z.B. Sende- und Empfangs-Datenleitungen ausgekreuzt werden.

Beim Zusammenschalten unterschiedlicher DTEs kommt es vor, dass das eine Gerät bestimmte Signale nicht besitzt, das andere Gerät diese Signale aber prüft. In solchen Fällen werden die Signale so beschaltet, dass das prüfende Gerät einen plausiblen Zustand vorfindet.

Damit die Datenübertragung z.B. zwischen zwei PC ohne Datenverluste funktioniert, muss eine Flussteuerung realisiert werden. Dazu gibt es 2 Möglichkeiten:

1. Die Fluss-Steuerung zwischen Sender und Empfänger wird hardware-mässig über die vorgesehenen Leitungen realisiert.
2. Sender und Empfänger realisieren die Fluss-Steuerung software-mässig mit Meldungen.

a) Hardwaremässige Fluss-Steuerung

Die Signale müssen wie folgt verbunden werden (Stecker-Nummern beziehen sich auf den 9poligen Stecker des PC):

- Sende- und Empfangsleitungen müssen ausgekreuzt werden.
- Das Signal "Data Terminal Ready" wird mit den Signalen "Data Set Ready" und "Data Carrier Detect" des Partners verbunden.
- Das Signal "Request to Send" wird mit dem Signal "Clear to Send" des Partners verbunden.

Eine solche Verbindung wird als "Nullmodem" bezeichnet.

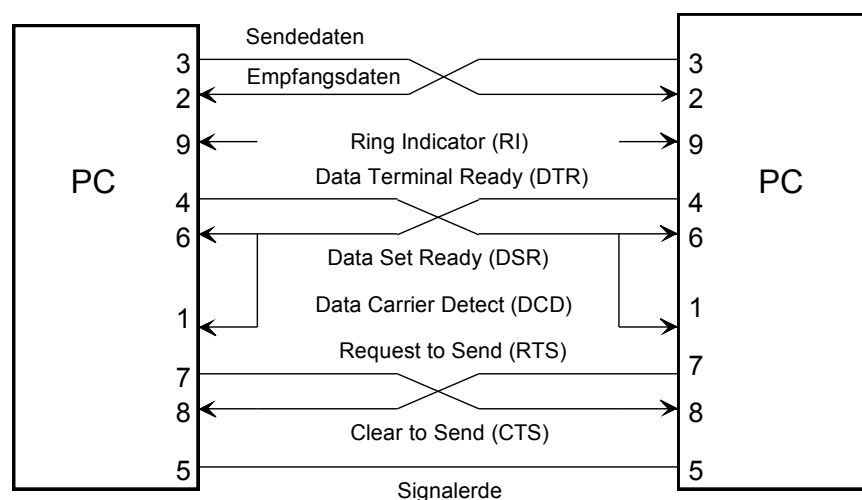


Abbildung 9

b) Software-mässige Fluss-Steuerung

Anstelle der hardwaremässigen Steuerung mit den oben beschriebenen Signalen ist es auch möglich, die Steuerung von Sender und Empfänger softwaremässig zu realisieren. Dazu werden die ASCII-Zeichen XON (<Ctrl>Q) und XOFF (<Ctrl>S) verwendet.

Mit XON (DC1, Wert 11h) meldet der Empfänger, dass er bereit ist Daten zu empfangen und mit XOFF (DC3, Wert 13h), dass er bis auf Widerruf keine weiteren Daten entgegennehmen kann.

Die Hardware-Steuerleitungen sind nicht notwendig.

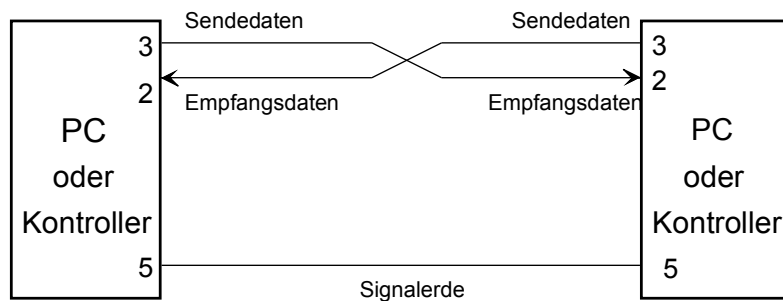


Abbildung 10