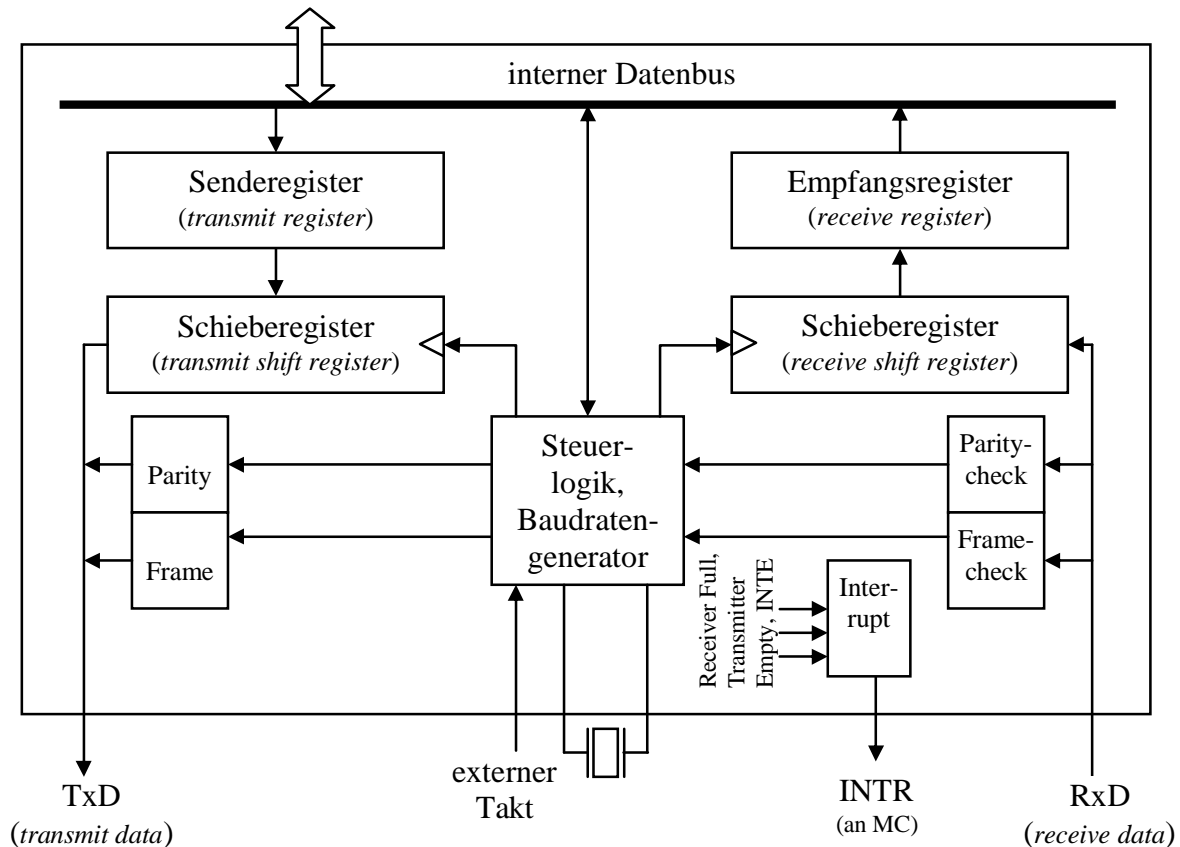


## Ausführungseinheit für serielle Schnittstellen



Da die Daten im Mikrocomputer im Allgemeinen in paralleler Form vorliegen, benötigen serielle Schnittstellen Seriell-/Parallel-Umwandler als Kernkomponenten. Diese Aufgabe übernehmen Schieberegister für die beiden Datenflussrichtungen "Senden" und "Empfangen". Übliche Längen orientieren sich noch heute an den Datenbreiten für Textzeichen, die im Fernschreiber 5 Bit betragen haben, für Computerterminals später 7 Bit. Zur Sicherung der Übertragung wurde ein Paritätsbit angefügt, weshalb auch 8 Bit möglich sind. Mit zunehmender Internationalisierung durch die weite Verbreitung von Personal Computer musste das Parity Bit für einen erweiterten ASCII Code mit 256 gültigen Zeichen herangezogen werden. Außerdem wurde es zunehmend notwendig, Binärdaten zu übertragen (Ablösung der Fax-Geräte durch Bildversand zwischen PC), die nativ in 8-Bit Bytes vorliegen. Daher haben viele Realisierungen serieller Schnittstellen, die sich zur Datenfernübertragung eignen, auch die Möglichkeit, 9 Bit lange Datenwörter zu verarbeiten (8 Bit + Parity).

Die oben skizzierte Ausführungseinheit zeigt die wichtigsten Elemente einer Schnittstelle nach der EIA Norm RS-232, die in Europa weitgehend identisch in der CCITT (heute ITU) Norm V.24 beschrieben ist. Sie ist schon sehr lange im Gebrauch und hat noch immer Bedeutung, entweder direkt als Debug- und Wartungsschnittstelle in Informationstechnischen Geräten oder in abgewandelter Form zur robusten Steuerung und Überwachung von Endgeräten aller Art über mittlere Distanzen von einigen 100m. Für den ursprünglichen Zweck der Weitverkehrs- oder Terminalverbindungen hat sie inzwischen keine nennenswerte Bedeutung mehr.

## **Funktionsweise**

Die Ausführungseinheit hat getrennte Schieberegister für Empfangen und Senden. Die Schieberegister sind vom Mikrocomputer durch je ein Pufferregister entkoppelt, für den Programmierer sind dies die Empfangs- und Senderegister. Die Schieberegister selbst sind i.A. nicht vom Prozessor aus zugänglich. Die serielle Übertragung erfolgt so, dass das niederwertigste Bit eines zu übertragenden Wortes zeitlich zuerst auf die Leitung gelegt wird. Die Repräsentation einer binären Stelle auf der Datenleitung wird in der seriellen Übertragungstechnik als "**Symbol**" bezeichnet. In den Normen sind keine weiteren Vereinbarungen zum logischen Umgang mit den Daten auf der Protokoll Ebene fixiert. Datenaustauschprotokolle müssen per Software separat implementiert werden.

### Sende- und Empfangstakt

Ein Teil der Steuerlogik betrifft die Generierung eines gemeinsamen Takts für Sender und Empfänger. Die Symbolraten auf der Leitung sind von diesem Takt abgeleitet. Eine Kommunikation zwischen entfernten Geräten kann nur stattfinden, wenn die Symbolrate des sendenden Teilnehmers identisch mit der des empfangenden Teilnehmers ist. Im einfachsten Fall wird der Takt auf einer eigenen Leitung mitübertragen und so eine starre Kopplung erstellt. Diese Art der Übertragung wird als "**synchron**" bezeichnet. Sie stellt geringe Anforderungen an die Qualität des Takts, führt aber zu störanfälligen Systemen. Schon bei 20000 Baud sind nur noch 10 – 20 m Distanz überbrückbar. Sie wird daher hauptsächlich für die Anbindung ortsnaher Hardware (Sensoren, Wechseldatenträger, etc.) in Massenprodukten eingesetzt.

Ebenfalls als synchron wird ein Verfahren bezeichnet, bei dem vor jedem Datenblock ein eigener Synchronisationsblock (die Präambel) gesendet wird. Auf eine physikalische Taktleitung wird dann verzichtet, im Empfänger muss ein elektrisch justierbarer Taktgenerator für einen ausreichend synchronen Empfangstakt sorgen.

Stellt man vor jedes einzelne zu sendende Zeichen eine Synchronisationsinformation, bezeichnet man dieses Verfahren als "**asynchron**". Eine Taktleitung zwischen Teilnehmern ist nicht vorhanden. Die Synchronisationsinformation besteht aus einem zu Beginn einer Zeichenübertragung eingefügten **Startbit**. Das Startbit hat eine zum Ruhepegel auf der Leitung inverse Polarität. Aus der Signalfanke und der Baudrate errechnet der Empfänger einen Versatz von einer halben Bitzeit und liest dann alle nachfolgenden Bits des Zeichens möglichst "in der Mitte" ein. Nach den Zeichendaten folgen noch ein oder mehrere **Stoppbits**, die einen Zeitbedarf zwischen den Zeichen – insbesondere beim Empfänger – berücksichtigen sollen. Die Stoppbits haben bereits die Polarität der Sendepause zwischen 2 Zeichen. Da das Verfahren verlangt, dass die Stopp Polarität zwischen den Zeichen tatsächlich erreicht wird, muss mindestens 1 Stoppbit übertragen werden<sup>3</sup>. Bei diesem Verfahren muss die Baudrate von Sender und Empfänger für die Dauer der Übertragung abgesprochen sein. Die Genauigkeit, mit der sie einzuhalten ist, hängt von der Forderung, möglichst exakt in der Mitte einer Bitzeit lesen zu wollen, ab.

Rein elektronische Lösungen arbeiten fast ausschließlich mit 1 Startbit und 1 Stoppbit. Der gesamte Block aus Zeichen, Start- und Stoppbits wird als Rahmen ("**Frame**") bezeichnet.

Mit dem asynchronen Verfahren ist bis zu 1 km Entfernung überbrückbar. Die Nutzdatenrate ist, technisch bedingt, immer kleiner als die Baudrate (beim synchronen Verfahren kann sie der Baudrate entsprechen, da kein oder relativ wenig Frame notwendig ist). In der Vergangenheit wurde

---

<sup>3</sup> Halbe Stoppbits verschieben den nachfolgenden Zeichenrahmen um ½ Taktperiode und erleichtern das Einlesen.

daher schon früh mit Datenstrom Komprimierung gearbeitet und die Nutzdatenrate in Bit pro Sekunde (bps) in den Vordergrund gestellt, die dann auch größer als die Baudrate werden kann. Die Hardware für asynchrone Übertragung wird als **UART** bezeichnet (Universal Asynchronous Receiver and Transmitter), wenn die Hardware zusätzlich synchrone Übertragung ermöglicht, heißt sie **USART** (Universal Synchronous and Asynchronous Receiver and Transmitter).

### Pegel und Fehlerbedingungen

Die Signale der RS-232 Übertragung haben Pegel von +3V bis +15V für das Symbol "**Space**" und -3V bis -15V für das Symbol "**Mark**" (ITU V.28). Sendepause, bzw. Stopp Polarität ist Mark (merke: idle mark → bei der Stromschleife wurde im Ruhezustand/Idle Stromfluss/Mark/Niederspannung vereinbart, um Leitungsfehler sofort erkennen zu können). Im Zeichen wird Mark als binäre 1 gespeichert, Space als binäre 0. Die symmetrischen Spannungspegel waren bei langen Übertragungswegen von Vorteil, wenn zur Signalverbesserung Induktivitäten eingebaut werden mussten. Je geringer der Gleichspannungsanteil eines Signals ist, desto weniger ohmsche Verluste treten im Mittel auf.

Dadurch, dass 0V kein erlaubter Symbolpegel ist, können Leitungsunterbrechungen auch im Fall der Spannungspegel erkannt werden.

Das zur Sicherung und Verifikation benutzte Paritäts-Bit generiert die Hardware selbständig beim Senden und fügt es vor dem MSB der Zeicheninformation bei, falls gewünscht. Es gehört zu den Nutzdaten, nicht zum Frame. Wenn vorhanden, kann beim Empfang geprüft werden, ob das Parity Bit noch zum Datenwort passt. Fehler werden in den Statusregistern der Hardware festgehalten (Parity Error).

Im asynchronen Fall werden auch die Start- und Stoppbits beim Senden zugefügt. Beim Empfangen, am Ende des Frames, werden die Stoppbits überprüft. Bei einer Diskrepanz wird ein Fehlerflag im Statusregister gesetzt (Framing Error).

Eine Sendepause wird als "**Break**" bezeichnet und bei bestimmten Übertragungsprotokollen als spezielle Bedingung ausgewertet. Sie ist definiert als Lücke im Mark-Pegel für mehr als eine komplette Frame-Länge und wird im Statusregister angezeigt (Break Interrupt).

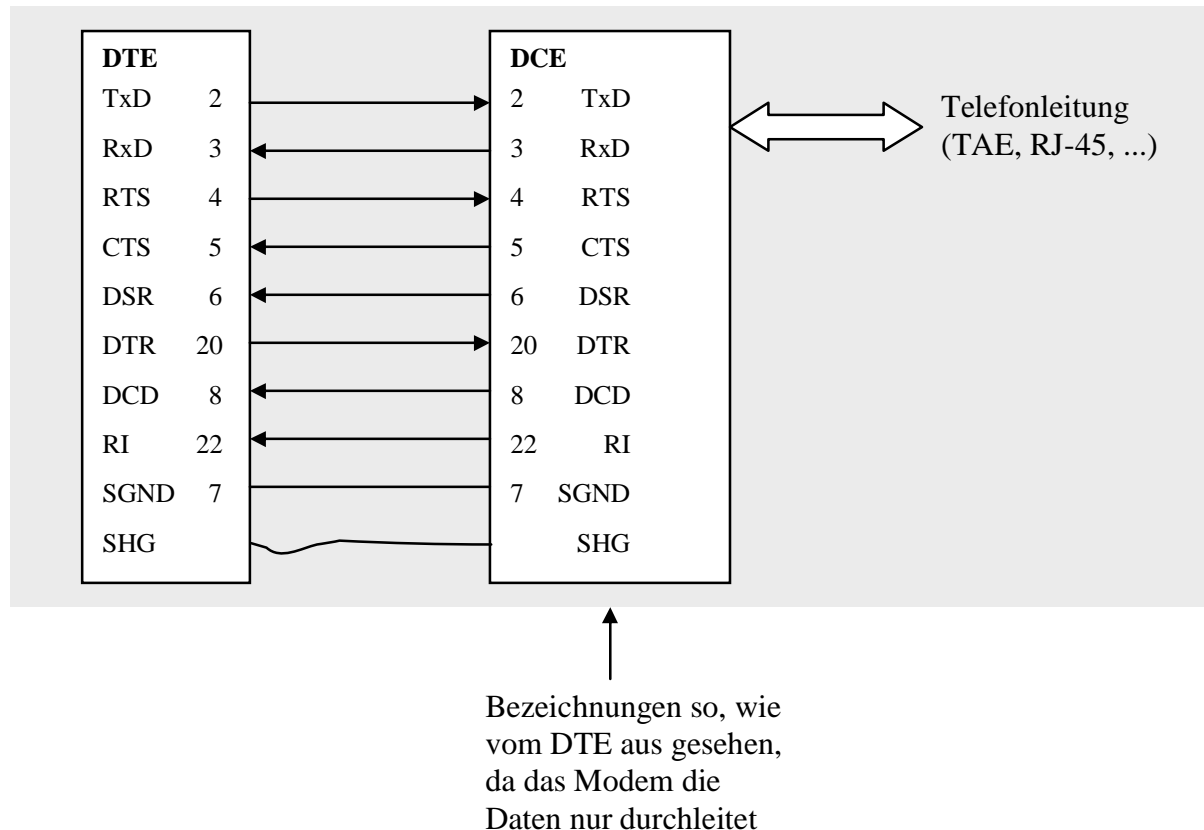
Üblich bleiben Fehlermeldungen so lange im Statusregister eingetragen, bis sie entweder vom Programm gelesen oder explizit zurückgesetzt wurden.

## ***Datenflussteuerung und Datenübertragungsprotokolle***

Wie bei jeder Kommunikation muss der Datenaustausch synchronisiert werden, wenn nicht sichergestellt ist, dass der Empfänger jederzeit schneller Daten aufnehmen kann, als der Sender sie liefert. Für Applikationen ist es oft zusätzlich notwendig, dass der Verbindungsaufbau und -abbau bestimmten Regeln gehorcht. Grundsätzlich können Protokolle als Zusatzinformation im Datenkanal oder, unabhängig von den Daten, als Hardware Signale auf separaten Leitungen realisiert werden.

### Schnittstellenleitungen

Als Schnittstelle, die entfernte Teilnehmer wie Terminals eines Großrechners an dessen Vorrechner anbinden soll, war die RS-232 Norm insbesondere für den Anschluss des Datenendgeräts ("Data Terminal Equipment", DTE) an ein Gerät zur Anpassung an Weitverkehrsverbindungen (Modem, "Data Communication Equipment", DCE) ausgerichtet. Die Weitverkehrsverbindung war in erster Linie eine Telefonleitung. Deshalb sind die Protokollsignale der RS-232 an den Bedürfnissen von Fernmeldeeinrichtungen orientiert.



Bedeutung der Signale:

TxD	Transmit Data Sendedaten	DSR	Data Set Ready Modem betriebsbereit
RxD	Receive Data Empfangsdaten	DTR	Data Terminal Ready Endgerät betriebsbereit
RTS	Request to Send Sendeanforderung	DCD	Data Carrier Detect Gegenstelle bereit
CTS	Clear to Send Sendebereitschaft	RI	Ring Indicator ankommender Ruf
SHG	Shielded Ground Schutzerde	SGND	Signal Ground Signalerde

In der Norm ist für den Steckverbinder ein 25-poliger D-Sub Stecker (ISO 2110) vorgesehen, die Belegung ist in der Skizze eingetragen.

### Die Modem Verbindung

Der Ablauf der Modem Flusssteuerung erklärt sich aus der Benennung der Signale. Sobald Terminal und Modem Betriebsbereitschaft zeigen (DTR/DSR), kann das Terminal eine Sendeanforderung (RTS) stellen, die bei Sendebereitschaft (CTS) auch zum Datentransfer zu einer entfernten Station führt. Andernfalls soll das Terminal auf einen ankommenden Ruf reagieren und Daten entgegen nehmen. Das Signal DCD soll hauptsächlich dazu dienen, festzustellen, ob das Modem einer Gegenstelle aktionsbereit ist, bzw. zu reagieren, wenn im Übertragungsweg Probleme auftauchen und die Verbindung zur Gegenstelle unmotiviert unterbrochen wird. Im Falle des Dateneempfangs durch das Terminal muss der Empfänger den Empfang unterbrechen können, wenn sein Empfangspuffer vollzulaufen droht. Für diesen Fall wurde ein Signal RFR (Ready For Receive, V.24) definiert, das aber mit RTS denselben Stecker-Pin teilt. Ein Wegnehmen von RFR soll dem Sender signalisieren, dass das Senden vorübergehend einzustellen ist.

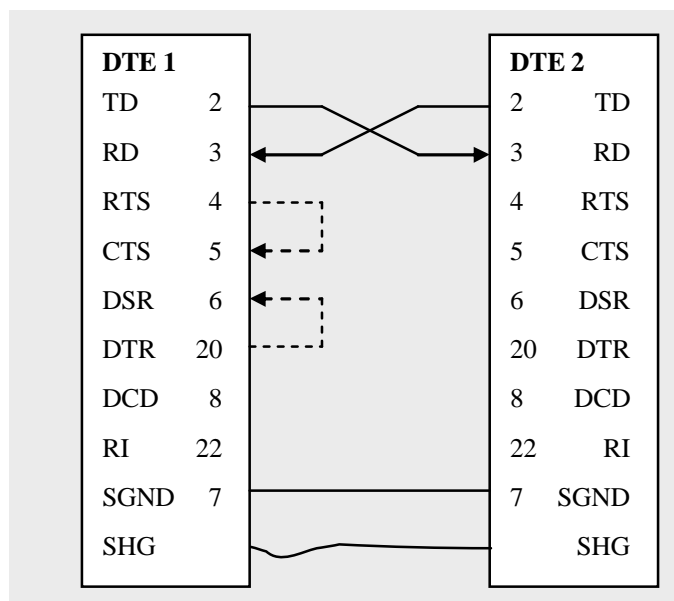
SGND trägt das Bezugspotential für alle Signale (0V). SHG ist der Anschluss für die Schirmung des RS-232 Verbindungskabels.

Für PC wurde der 25-polige Stecker mit kleiner werdenden Gehäuseabmessungen durch einen 9-poligen Sub-D Stecker ersetzt. Die Schirmung des Kabels benötigt keinen Steckkontakt, dadurch war dieser erhebliche Miniaturisierungsschritt möglich. Die RS-232 Signale sind jedoch nicht für Protokolle wie "Plug-and-Play" einsetzbar, weshalb die RS-232 Schnittstelle beim PC einer moderneren seriellen Schnittstelle weichen musste.

Modems sind Geräte, die per Befehlsschnittstelle viele Standard Aufgaben in Zusammenhang mit der Datenkommunikation selbst wahrnehmen können. Dazu gehören die Wahl eines Teilnehmeranschlusses, der Verbindungsabbau und –aufbau, Datensicherung und auch Komprimier- und Verschlüsselungsvorgänge. Die Befehle dazu werden vom DTE als ASCII-Text an das Modem übergeben (am Bekanntesten: der AT-Befehlssatz von Hayes Communications).

### Die NULL-Modem Verbindung und Datenflusssteuerung

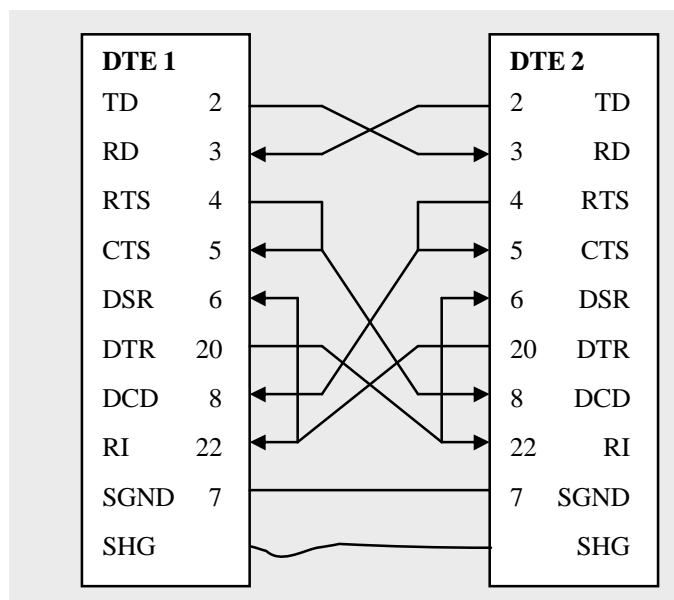
RS-232 Schnittstellen werden heute gerne zur Verbindung zweier gleichartig aufgebauter Endgeräte (PC mit Adapter an Single-Board Computer, etc.), also ohne zwischengeschaltete Modems, eingesetzt. Da kein Modem zum Einsatz kommt, werden solche Anordnungen als "NULL-Modem" Verbindung bezeichnet. Im einfachsten Fall wird nur jeweils der Empfänger (RxD) mit dem Sender (TxD) der Gegenstelle verbunden. Es entsteht die typische Verbindung mit gekreuzten Datenleitungen:



Terminal-Terminal Verbindung  
mit NULL-Modem Kabel

Prinzipiell genügt allein die Verschaltung der Datenleitungen und der Signalerde. Einige Hardware Implementierungen oder fest eingebundene Treiberprogramme können jedoch verlangen, dass DSR und CTS gültige Pegel erhalten, um die Sende- und Empfangsteile zu aktivieren (gestrichelte Verbindungen an DTE 1).

Für die Flusssteuerung mit Signalen ("**Hardware Handshake**") ist für gewöhnlich die RTS-CTS Brücke an beiden Bausteinen erforderlich und/oder eine Verbindung von DTR zu DSR des jeweils anderen Bausteins. Ein Kabel für RTS/CTS und DTR/DSR Flusssteuerung kann wie folgt aussehen:

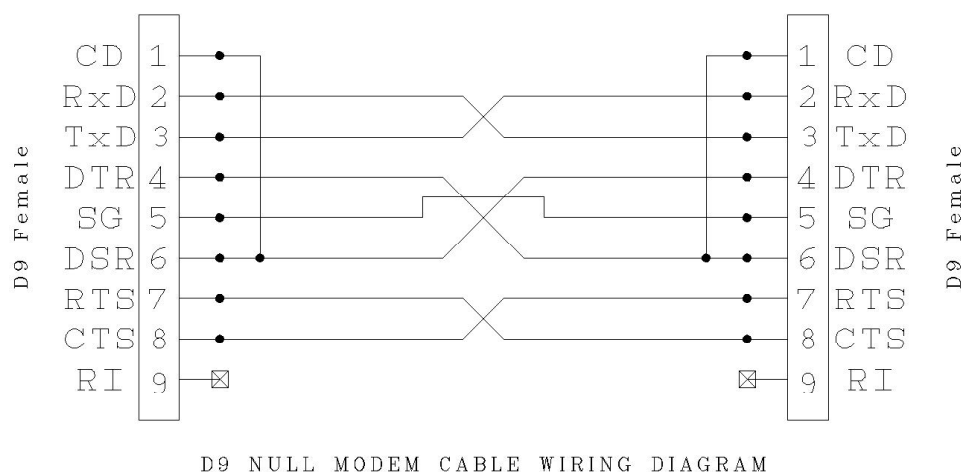


NULL-Modem Verbindung mit  
Möglichkeit zum Hardware  
Handshake (z.B. PC/Single-Board  
Computer)

RTS/CTS wirkt wieder als lokale Brücke, kann aber über DCD des jeweils anderen Rechners auch zur Flusssteuerung eingesetzt werden: ein Wegnehmen des RTS durch die Empfängerseite bewirkt, dass der Sender annimmt, der Empfänger hätte sich ausgeschaltet.

DTR/DSR wirken direkter auf den Datenfluss. Da DTR auch benutzt wird, um das Modem zurückzusetzen, kann damit leicht von Empfängerseite mit Standard Terminal Programmen der Sender zur Pause gezwungen werden. Für alle Flusssteuerungsmethoden gilt, dass Sie aufgrund der Physik der seriellen Schnittstelle auch nach anlegen der Pause-Anforderung noch mindestens 1 Zeichen austauschen, im Allgemeinen aufgrund von Programmlaufzeiten sogar deutlich mehr. Puffer für Hardware Handshake können aber dennoch kleiner gehalten werden, als solche für den SW-Handshake.

Die oben gezeigten Beispiele sind nur Vertreter einer großen Anzahl möglicher NULL-Modem Verbindungen. Häufig sind auch Leitungen mit der Kreuzverbindung von RTS und CTS zu finden. Dabei ist zu beachten, dass RTS dann im Sinne einer RFR Meldung des DTE benutzt wird. Dennoch wird diese Art der Flusssteuerung üblich mit RTS/CTS bezeichnet. Beispiel:



Um die Hardware der Schnittstelle möglichst gering zu halten, nur mit Signalerde und den beiden Datenleitungen auszukommen (häufig als "Zweidrahtverbindung" bezeichnet), wurde eine Flusssteuerung auf Basis von Steuerzeichen im Sendekanal des Datenempfängers definiert ("**Software Handshake**"). Sollte der Empfänger kurzfristig nicht mehr in der Lage sein, Daten abzunehmen, sendet er auf seinem RxD Kanal ein spezielles ASCII Zeichen ("XOFF"  $\triangleq$  DC3), auf das hin der Sender seine Übertragung aussetzt. Ist er wieder in der Lage, neue Daten aufzunehmen, sendet er ein weiteres Zeichen ("XON"  $\triangleq$  DC1), das dem Sender die Empfangsbereitschaft des Empfängers mitteilt. Nach den Steuerzeichen wird das Verfahren auch als "**XON/XOFF-Protokoll**" bezeichnet. Die Namen der Zeichen sind nicht im ASCII Code Standard definiert, vielmehr sind die Codes aus dem allgemeinen Teil von 4 "device controls" entnommen. Mit der Computer Tastatur kann man die Codes durch drücken von Strg-S(top = XOFF) und Strg-Q(uit = XON) erzeugen<sup>4</sup>.

Auf der Senderseite folgt das Protokoll der Regel:

- der Sender prüft nach jedem gesendeten Zeichen oder per Interrupt seinen Empfangskanal

<sup>4</sup> nach Digital Equipment Corporation, Hersteller einiger Standard Datensichtgeräte (VT-100, etc.)

- wird XOFF (DC3) empfangen, setzt der Sender die Datenübertragung mit dem nächsten Zeichen aus
  - im XOFF Zustand prüft der Sender weiter den Empfangskanal, zusätzliche XOFF Meldungen vom Empfänger werden ignoriert
  - wird XON empfangen, setzt der Sender die Datenübertragung mit dem nächsten zu sendenden Zeichen fort
- im XON Zustand werden weitere XON Meldungen ignoriert, der Vorgang beginnt von vorne

Das Ignorieren von XON und XOFF in den jeweiligen Senderzuständen ist notwendig, da manche Empfänger aus Sicherheitsgründen so lange ihr Steuersignal senden, bis sie eine Reaktion des Senders bei ihnen feststellen können. Das kann länger als die Zeit für 1 Zeichenübertragung sein.

Zeichen, die der Sender noch abgibt, nachdem der Empfänger XOFF gemeldet hat, werden im Allgemeinen nicht wiederholt. Für sie muss noch Platz im Empfangspuffer vorhanden sein. Daher besitzen Empfangspuffer 2 Werte für ihren Füllgrad, bei denen XON, bzw. XOFF, gesendet wird:

- die "High-Water Mark" (obere Grenze) → aussenden von XOFF
- die "Low-Water Mark" (untere Grenze) → aussenden von XON

Die Low-Water Mark soll gewährleisten, dass das empfangende Gerät (z.B. ein Drucker) ohne Unterbrechung eingegangene Daten verarbeiten kann. Besonders für die Steuerung mechanischer Aktoren kann dies wichtig sein.

Auf der Empfängerseite sieht die Regel wie folgt aus:

- der Empfänger übernimmt ankommende Zeichen und legt sie im Empfangspuffer ab
- wird die High-Water Mark erreicht, sendet der Empfänger XOFF aus (weitere XOFF können folgen, mindestens bis der Empfänger die Break-Bedingung erkennt)
- der Empfangspuffer Füllgrad wird auf die Low-Water Mark geprüft
- wird die Low-Water Mark erreicht, sendet der Empfänger XON aus (weitere XON können folgen, bis Zeichen im Empfangsregister ankommen); dann beginnt der Vorgang von vorne

### Punkt-zu-Punkt Verbindungen: Voll- und Halbduplex / Protokolle

Die RS-232 Spezifikation berücksichtigt zu einem Zeitpunkt nur die Verbindung zwischen 2 Teilnehmern (Duplex Verbindung). Beim Datenaustausch wird noch unterschieden, ob beide gleichzeitig Daten übertragen dürfen ("**Vollduplex**", FDX) oder ob abwechselnd ("**Halbduplex**", HDX), nacheinander.

Während im Vollduplex Betrieb beide Übertragungsleitungen benötigt werden, kommt das Halbduplex Verfahren mit einer einzigen aus ("Eindrahtverbindung", Signalerde obligatorisch). Dennoch ist der Datenaustausch in beiden Richtungen möglich. Im Schnittstellenbaustein werden für Halbduplex RxD und TxD miteinander verbunden und am TxD-Anschluss nach außen geführt. Die Kommunikationsrichtung muss durch ein den Handshake Protokollen überlagertes Protokoll im Datentransfer ausgehandelt werden. Beispiele bekannter Protokolle sind KERMIT (Columbia University) und XMODEM (Ward Christensen), die für Dateitransfer und BBS Zugänge benutzt wurden. Beide versenden (schwach) gesicherte Datenpakete (96 bzw. 128 Byte Nutzdaten), die mit Nummern versehen sind und 1 oder mehrere ASCII Zeichen für Steuer- und Kontrollinformationen mitführen. Jedes versandte Paket muss der Empfänger mit ACK quittieren, d.h. im Halbduplex Betrieb kann nach jedem Paket ein Umschalten der Datenrichtung erfolgen.