



University of Applied Sciences

HOCHSCHULE  
EMDEN·LEER

## **Praktikum Digitaltechnik**

# **Serielle Datenübertragung am Beispiel der V.24/RS-232-Schnittstelle**

**von Prof. Dr.-Ing. Dirk Rabe**

Gruppe:

Teilnehmer:

Vortestat:

Testat:

Benutzte Geräte:

# 1 Einleitung und Überblick

Im Rahmen dieses Versuchs sollen Sie

- die Grundprinzipien von Übertragungs-Protokolle kennenlernen und
- am Beispiel der V24-Schnittstelle ein Protokoll analysieren.

Die Versuchseinleitung ist wie folgt gegliedert:

- Überblick über Unterscheidungskriterien von Daten-Übertragungsprotokollen,
- die V.24-Schnittstelle,
- Versuchsvorbereitung (Abgabe zu Beginn des Praktikumversuchs),
- Versuchsanleitung,
- Hinweise zum Versuchsprotokoll.

Die Informationen wurden aus den im Literaturverzeichnis angegebenen Quellen entnommen.

## 2 Überblick Daten-Übertragungsprotokolle

Bei der Leitungs-gebundenen Verschaltung von Baugruppen (z.B. Mikrocontroller, Drucker, Festplatten, Halbleiter-Speicher, Sensoren, PCs, digitale Baugruppen innerhalb eines Mikrocontroller-Chips usw.) ist ein Datenaustausch zwischen diesen Baugruppen erforderlich. Um diese Baugruppen in unterschiedlichen Systemen einsetzen zu können ist eine exakte Definition des Protokolls und der elektrischen Parameter erforderlich. Es existiert eine Vielzahl von genormten Übertragungsprotokollen, die sich durch folgende Charakteristiken unterscheiden:

1) parallele oder serielle Datenübertragung:

Bei der seriellen Datenübertragung existiert nur eine Leitung (1 Draht) zur Datenübertragung (ggf. pro Übertragungsrichtung). Über diese Leitung werden sequentiell sämtliche Informations-/Zeichenbits übertragen.

Bei der parallelen Datenübertragung werden mehrere Datenbits über mehrere physikalische Datenleitungen parallel übertragen. Typischerweise entspricht die Anzahl der Datenleitungen der Informations-/Zeichenbit-Breite (z.B. 8/16/32 Bits).

Es gibt auch Mischformen aus serieller und paralleler Datenübertragung. Hier werden z.B. bei einer Zeichenbit-Breite von 8 Bits die Daten über 4 Datenleitungen in 2 Takten übertragen.

Die Entscheidungskriterien sind hier: Übertragungsgeschwindigkeit (parallele Übertragung performanter), Hardware-Aufwand (Anzahl Leitungen bei serieller Übertragung geringer).

2) weitere Steuerleitungen:

Zur Kommunikation werden Protokoll-spezifisch weitere Leitungen eingesetzt (z.B. Signalisierung von neuen Datenpaketen, Mitteilung der Empfangsbereitschaft, Adressbus, Steuerinformationen).

3) Anzahl der an einen Bus angeschlossenen Baugruppen:

Wenn an einem Bus nur 2 Teilnehmer angeschlossen sind, ist der Empfänger von Daten eindeutig. Bei mehr als 2 Teilnehmern muss der Empfänger von Daten (wenn es nicht alle

Teilnehmer sein sollen) ausgewählt werden. Dies kann durch zusätzliche Adressierungs- und Auswahlleitungen erfolgen oder durch Übertragung einer Adresse über die Datenleitung bzw. die Datenleitungen in einem definierten Protokoll.

- 4) Voll- und Halb-duplex-Übertragung:  
Bei der Voll-duplex-Übertragung existieren separate Leitungen und es können gleichzeitig Daten gesendet und empfangen werden.  
Bei der Halb-duplex-Übertragung kann bzw. wird nicht zeitgleich gesendet und empfangen werden. Typischerweise sind hier dann auch keine getrennten Datenleitungen für das Senden und Empfangen vorhanden.
- 5) Elektrische Signal-Codierung:  
Für die Kodierung von binären Werten durch elektrische Größen zur Datenübertragung kann ein Stromfluß oder Spannungswerte mit unterschiedlichen Schwellwerten verwendet werden.
- 6) Energieversorgung über das Buskabel:  
Über das Buskabel ist ggf. auch die Energieversorgung der Komponente möglich, so dass u.U. keine separate Energieversorgung notwendig ist.
- 7) Leitungslängen:  
Insbesondere durch die elektrischen Parameter, die verwendeten Leitungen, die Strom-/Spannungspegel und die Übertragungsgeschwindigkeiten sind die maximalen Leitungslängen begrenzt.
- 8) Bus-Topologie:  
Bei mehr als 2 Teilnehmern müssen alle Teilnehmer mit dem Bus verbunden werden. Dies kann in einer Ring-, Stern oder Baumstruktur erfolgen.
- 9) Synchron- und asynchrone Datenübertragung:  
Bei der synchronen Übertragung wird nach einem festen Taktraster - ohne Lücken - gearbeitet. Eine neue Übertragung beginnt immer in einem klar definierten Takt.  
Bei der asynchronen Datenübertragung wird das Taktsignal beim Empfänger und Sender getrennt generiert. Während der Übertragung eines Zeichens müssen Empfänger und Sender synchron arbeiten. Hierfür ist es erforderlich, dass die Taktrate (Baudrate) bei beiden Kommunikationsstellen gleich ist. Bei asynchronen Protokollen sind typischerweise die zu übertragenden Daten in speziell kodierete Start- und Stop-Bit-Rahmen eingebettet.
- 10) (digitales) Busprotokoll:  
Über das Protokoll müssen verschiedene Aspekte definiert werden:
  - Synchronisation: Wie kann sich ein (neuer) Teilnehmer auf das Protokoll synchronisieren (Start-/Stop-Bits, Pausen...)
  - Adressierung: Wie wird ein Zielmodul angesprochen zum Daten-Empfang oder Daten-Senden?
  - Master/Slave-Betrieb: Wird die Busbenutzung von einem (Single-Master-Betrieb) oder mehreren Mastern (Multi-Master-Betrieb) gesteuert. Alle anderen Baugruppen antworten nur wenn Sie von einem Master angesprochen werden?
  - Arbitrierung (bei mehreren Teilnehmern): Wie wird entschieden welcher Teilnehmer den Daten-Übertragungskanal benutzen darf bei gleichzeitigen „Nutzungswünschen“?
  - Fehlererkennung: Bei der Übertragung können Fehler (z.B. durch Störungen) auftreten. Hierfür werden oftmals redundante Daten übertragen (Parity-Bits oder Signaturen). Die korrekte oder fehlerhafte Datenübertragung kann durch ein Handshake vom Empfänger

signalisiert werden (separate Leitung oder über eine Antwort auf der(den) Datenleitung(en)). Im Protokoll kann auch definiert sein wie auf einen Fehler reagiert werden soll (z.B. automatische nochmalige Datenübertragung).

### 3 Die V.24-Schnittstelle

Der Inhalt dieses Kapitels ist teilweise direkt aus einer Diplomarbeit zu diesem Thema entnommen worden [2].

Für die Datenübertragung zwischen dem Parallelkonverter und dem Datenendgerät wird die V.24 Schnittstelle eingesetzt. Diese Schnittstelle ist die alte US Fernschreiberschnittstelle und diente ursprünglich der Datenfernübertragung über ein Modem.

Modems (auch Akustikkoppler genannt) sind Modulations- und Demodulationsgeräte, mit denen Daten über das Fernmeldenetz der Bundespost weltweit übertragen werden können. Die Daten werden dabei vom Modem durch Frequenzumtastung in den Tonfrequenzbereich gewandelt.

Gegenüber der direkten Verbindung zwischen Computern und Peripheriegeräten ist diese Anwendung heutzutage in den Hintergrund getreten. In einem PC ohne Postzulassung ist diese Schnittstelle oft nicht vollständig realisiert und wird RS232C Schnittstelle genannt.

Auch wenn heutzutage USB-Schnittstellen die gebräuchlicheren PC-Schnittstellen sind, so stellt die V.24 bzw. RS232C Schnittstelle eine einfache und gebräuchliche Mirko-Controller-Schnittstelle dar.

Die V.24-Schnittstelle ist eine serielle Schnittstelle, d.h. nacheinander werden die Daten Bit für Bit übertragen (bitseriell). Die Übertragung der Daten findet in negativer Logik statt.

Jedoch besitzt die V.24-Schnittstelle parallele Steuerleitungen, die in positiver Logik ausgeführt sind. Dadurch ist ein paralleles Handshake durchführbar.

#### 3.1 Die Datenübertragung über die V.24 Schnittstelle

Ein zentrales Problem bei der seriellen Datenübertragung ist die Synchronisation zwischen Sender und Empfänger. Deshalb unterteilt man die serielle Bitfolge in einzelne Blöcke. Diese Datenblöcke werden zusammen mit einem header (Nachrichtenkopf) und einem trailer (Nachspann) in einem Senderahmen (siehe Abbildung 1) übertragen.

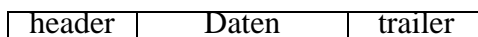


Abbildung 1: Senderahmen einer serieller Übertragung

Eine Betriebsart der seriellen Datenübertragung ist die synchrone Übertragung. Da sie hier nicht angewendet wird, soll sie nur kurz erläutert werden. Sie dient der schnellen Datenübertragung über das Postnetz.

Bei der synchronen Übertragung wird beim Übertragungsbeginn nach einer Pause eine bestimmte Bitfolge (Synchronisationswort) zur Synchronisation als header eingefügt. Auf diese Weise kann der Empfänger den Beginn eines Datenblockes erkennen. Die Daten werden als ein zusammenhängendes Paket gesendet. Wenn keine Daten vorliegen, werden nur Syn-

chronworte gesendet. Auf diese Weise kann sich der Taktgenerator im Empfänger immer auf auf das ankommende Signal synchronisieren.

Die Einsynchronisation des Taktgenerators im Empfänger ist relativ aufwendig zu realisieren.

Bei der asynchronen Übertragung wird als header ein Startbit, als trailer ein (optionales) Paritäts- und Stopbit eingesetzt. Wenn keine Daten zu übertragen sind, werden nur Stopbits gesendet. Ändert die Datenleitung ihren Zustand (Startbit), erkennt der Empfänger, dass nun Daten übertragen werden. Bei der Dekodierung im Empfänger wird üblicherweise mit dem 4 bis 16 fachen Takt asynchron abgetastet.

Dabei wartet der Empfänger auf den Übergang vom Stop- zum Startbit und tastet danach solange ab, bis er die halbe Bitzeit erreicht hat (also in der Mitte). Anschließend wird jedes weitere Zeichen nach der ganzen Bitzeit abgetastet und weiterverarbeitet.

Sende und Empfangstakt sind also nicht synchronisiert, sondern der Empfangstakt wird nur ungefähr ( $\pm 3\%$ ) auf ein vielfaches der Sendefrequenz eingestellt. Aus diesem Grunde können nur kurze Datenblöcke (ein Zeichen) zwischen zwei Synchronisationszeichen übertragen werden! Diese Betriebsart gelangt in fast allen EDV Peripherieschnittstellen zur Anwendung und wird deshalb auch hier eingesetzt. In Abbildung 2 ist ein Senderahmen dargestellt.

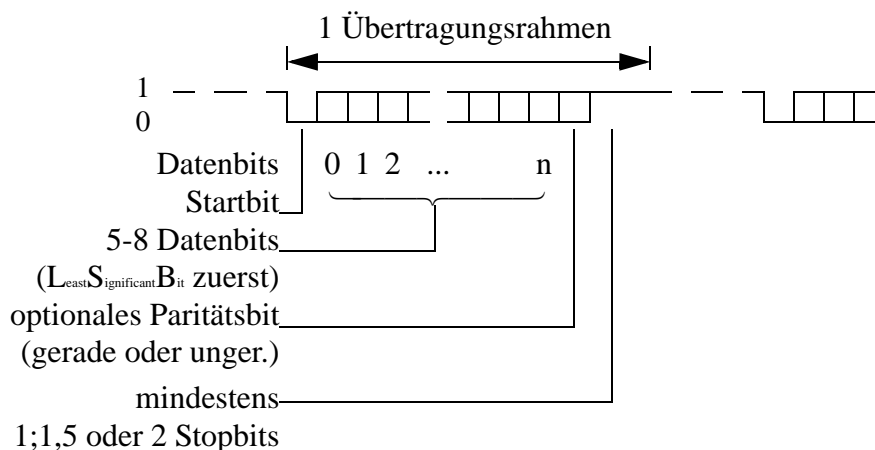


Abbildung 2: Übertragungsrahmen im asynchronen Betrieb

Die Zeichenlänge ist bei der V.24 Übertragung veränderbar. Es können 5 bis 8 Datenbits übertragen werden, üblich sind jedoch 7 oder 8 Bits. Die minimale Anzahl der Stopbits kann 1 oder 1,5 oder 2 betragen (wenn im Anschluss keine neuen Daten gesendet werden sollen, können beliebig viele weitere Stop-Bits folgen). Bei 8 Datenbits ist jedoch für einige Geräte nur ein Stopbit erlaubt. Den Datenbits kann ein gerades, ein ungerades oder kein Paritätsbit angefügt werden. Ein gerades Paritybit ist dann "1", wenn die Summe aller übertragenen Datenbits ungerade ist (also wird die Anzahl der 1-Bits durch das Paritybit auf eine gerade Anzahl 1-Bits ergänzt).

## 3.2 Technische Spezifikationen der V.24 Schnittstelle

In der Norm für die V.24 Schnittstelle (DIN 66020, 66022, CCIT V.24, ISO 2110) ist der HIGH Pegel als eine Spannung von +3V bis +15V definiert. Die Spannung des LOW Pegels

ist mit -3V bis -15V festgelegt. Die Zuordnung der Spannungspegel zu den logischen Zuständen sind in Abbildung 3 dargestellt.

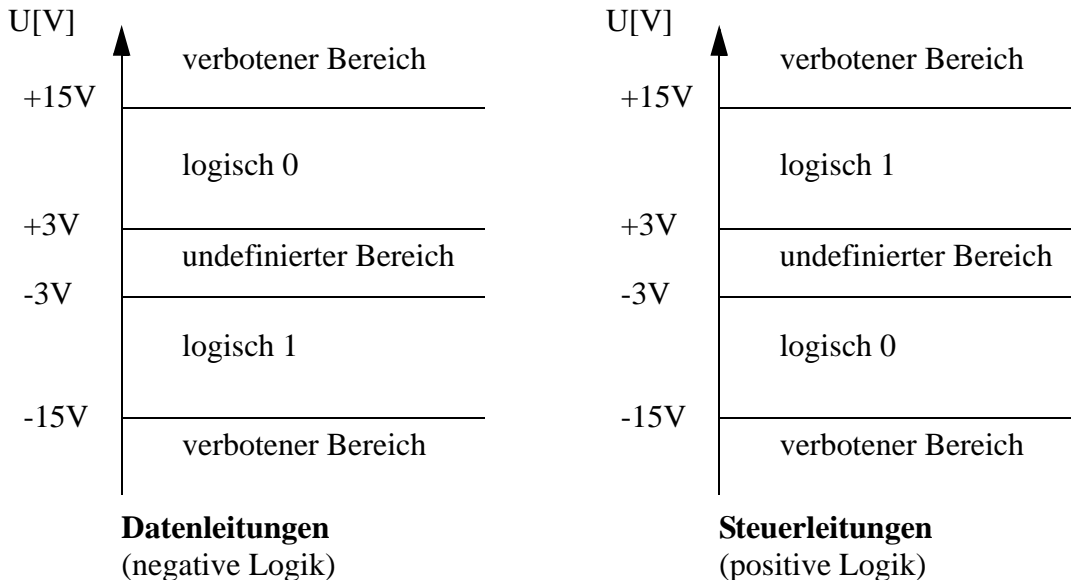


Abbildung 3: Spannungspegel der V.24-Schnittstelle

Die folgende Auflistung bietet eine Übersicht über die elektrischen Spezifikationen der V.24 Schnittstelle.

- Treiberausgang
  - die maximale Leerlaufspannung darf  $\pm 25V$  nicht überschreiten.
  - bei Belastung eines Ausgangs mit  $3k\Omega$  bis  $7k\Omega$  muss die Spannung am Stecker  $\pm(5$  bis  $15)V$  betragen.
  - der maximale Kurzschlußstrom zwischen zwei beliebigen Signalen darf  $500mA$  nicht überschreiten.
  - die Signalleitungen müssen bei abgeschalteter Betriebsspannung einen Innenwiderstand von mindestens  $300\Omega$  haben.
- Empfängereingang
  - die Empfängerschwelspannungen betragen  $\pm(3$  bis  $25)V$
  - die im Leerlauf erzeugte Gegenspannung darf  $\pm 2V$  nicht überschreiten.
  - der Innenwiderstand muss in einem Bereich von  $3$  bis  $7k\Omega$  liegen.
  - die Eingangskapazität darf  $2500pF$  nicht überschreiten.
  - der Eingangswiderstand darf nicht induktiv sein.
- Anstiegsgeschwindigkeit
  - bei einer Belastung mit  $C$   $2500pF$  und  $R=3k\Omega$  bis  $7k\Omega$  darf die maximale Flankensteilheit  $30V/\mu s$  nicht überschritten werden.
  - die minimale Flankensteilheit:
    - der Steuer und Meldeleitungen für das Durchlaufen des undefinierten Bereichs von  $+3V$  bis  $3V$  beträgt  $1ms$ ;
    - für die Daten und Taktleitungen beträgt sie ebenfalls  $1ms$  oder  $3\%$  der Baudzeit, es gilt der geringste Wert.

Die Treiber der V.24 Schnittstelle müssen Kurzschluss und gegenspannungsfest sein. Dadurch können diese nicht durch eine falsche Pinbelegung der Schnittstelle zerstört werden.

### 3.3 Klassen von V.24-Geräten

Traditionell wurde die V.24 Schnittstelle zur Datenfernübertragung verwendet. Die Betrachtungsrichtung der Daten- und Steuersignale spielt hier eine Rolle. Man kann Geräte mit V.24-Schnittstellen in zwei Klassen einteilen:

1. DCE = Data Communication Equipment = Datenübertragungseinrichtung (DÜE)
2. DTE = Data Terminal Equipment = Datenendeinrichtung (DEE)

Datenendeinrichtungen sind letztlich die Ziel- und Ursprungspunkte von Daten. Dies können z.B. PCs, Drucker oder beliebige sonstige Endgeräte sein. Datenübertragungseinrichtungen sind Geräte, die nur zur Weiterübertragung der Daten dienen - z.B. Modems.

Diese beiden Rollen sind deshalb wichtig, weil die Verbindung der Daten- und Steuersignale unterschiedlich erfolgen muss. Z.B. muss der Datenausgang eines Endgeräts (TxD - transmitted data) mit dem Dateneingang eines anderen Endgeräts (RxD - received data) verbunden werden. Bei einer Übertragungseinrichtung hingegen fungiert der TxD-Pin als Dateneingang, weil das Übertragungsgerät dort die Sendedaten eines Endgeräts erwartet (Frage der Sichtweise). D.h., dass bei der direkten Verbindung zweier Endgeräte die Leitungen für TxD und RxD gekreuzt werden müssen.

In diesem Abschnitt sollten zunächst einmal die beiden unterschiedlichen Geräteklassen eingeführt werden und die Grundproblematik erklärt werden. Die tatsächlichen Verschaltungen der entsprechenden Klassen miteinander wird erst nach der Einführung der V.24-Signale und der Handshake-Verfahren behandelt.

### 3.4 Die wichtigsten Anschlussleitungen der V.24 Schnittstelle

Die Informationen werden über die Datenleitungen gesendet. Diese sind in Tabelle 1 dargestellt.

Abkürzung	Bedeutung englisch	Bedeutung deutsch	Pin-Nr. <sup>‡</sup>
RxD	Received Data	Empfangsdaten (aus Sicht des Endgeräts - DEE)	2
TxD	Transmitted Data	Sendedaten (aus Sicht des Endgeräts - DEE)	3

Tabelle 1: Datenleitungen

<sup>‡</sup> alle Pin-Nummern beziehen sich auf einen 9-poligen Stecker

Die am häufigsten verwendeten Steuerleitungen sind in Tabelle 2 dargestellt.

Um einen Bezugspegel für alle Signale zu haben ist eine Masseleitung erforderlich (siehe Tabelle 3).

Abkürzung	Bedeutung englisch	Bedeutung deutsch	Pin-Nr.
RTS	Request to send	Sendeanforderung vom Endgeräts (DEE) an das Übertragungsgerät (DÜE)	7
CTS	Clear to Send	Empfangsbereitschaft des Übertragungsgeräts (DÜE) wird dem Endgerät (DEE) angezeigt (Endgerät darf senden)	8
DSR	Data set ready	Betriebsbereitschaft des Übertragungsgeräts (DÜE) - heißt aber nicht, dass das Gerät auch empfangsbereit ist (siehe CTS)	6
DTR	Data Terminal Ready	Betriebsbereitschaft des Endgeräts (DEE)	4
DCD	Data Carrier Detect	Datenträgersignal wurde vom Übertragungsgerät (DÜE) erfasst	1
RI	Ring Indicator	Übertragungsgerät (DÜE) signalisiert ankommenden Anruf	9

Tabelle 2: Steuerleitungen

Abkürzung	Bedeutung englisch	Bedeutung deutsch	Pin-Nr.
SGND	Signal Ground	Betriebserde	5

Tabelle 3: sonstige Leitungen

Die Bedeutung der Signale kann durch den logischen Ablauf der Interfacesignale während einer Übertragung über eine Fernleitung im Halbduplex-Betrieb am besten verstanden werden (siehe Abbildung 4):

- 1) Zunächst zeigt die DÜE A über einen High-Pegel auf DSR (Data Set Ready) die Betriebsbereitschaft an. DEE A zeigt (bereits vorher oder als Reaktion auf DSR) der DÜE A über einen High-Pegel auf DTR (Data Terminal Ready), dass das Endgerät (ebenfalls) bereit ist. Entsprechend läuft die Kommunikation zwischen DÜE B und DEE B.
- 2) DEE A kündigt mit RTS (Request to Send) an, dass über DÜE A und DÜE B Daten zu DEE B übertragen werden sollen. Diesen Request gibt DÜE B als Empfangs-Request über DCD (Data Carrier Detect) weiter und DÜE A quittiert den Request über CTS (Clear to send).
- 3) Nun werden die Daten von der DEE A über die DÜE A und die DÜE B an die DEE B gesendet.



- 4) Sobald alle Daten gesendet sind, signalisiert die DEE A der DÜE A mit der Deaktivierung von RTS (Request to Send), dass alle Daten gesendet wurden. Die DÜE A quittiert die Deaktivierung von RTS mit der Deaktivierung von CTS (Clear to Send). Gleichzeitig deaktiviert DÜE B die Datenempfangsbereitschaft von DEE B durch die Deaktivierung des DCD-Signals.
- 5) Die Datensendung von DEE B nach DEE A erfolgt entsprechend anders herum.

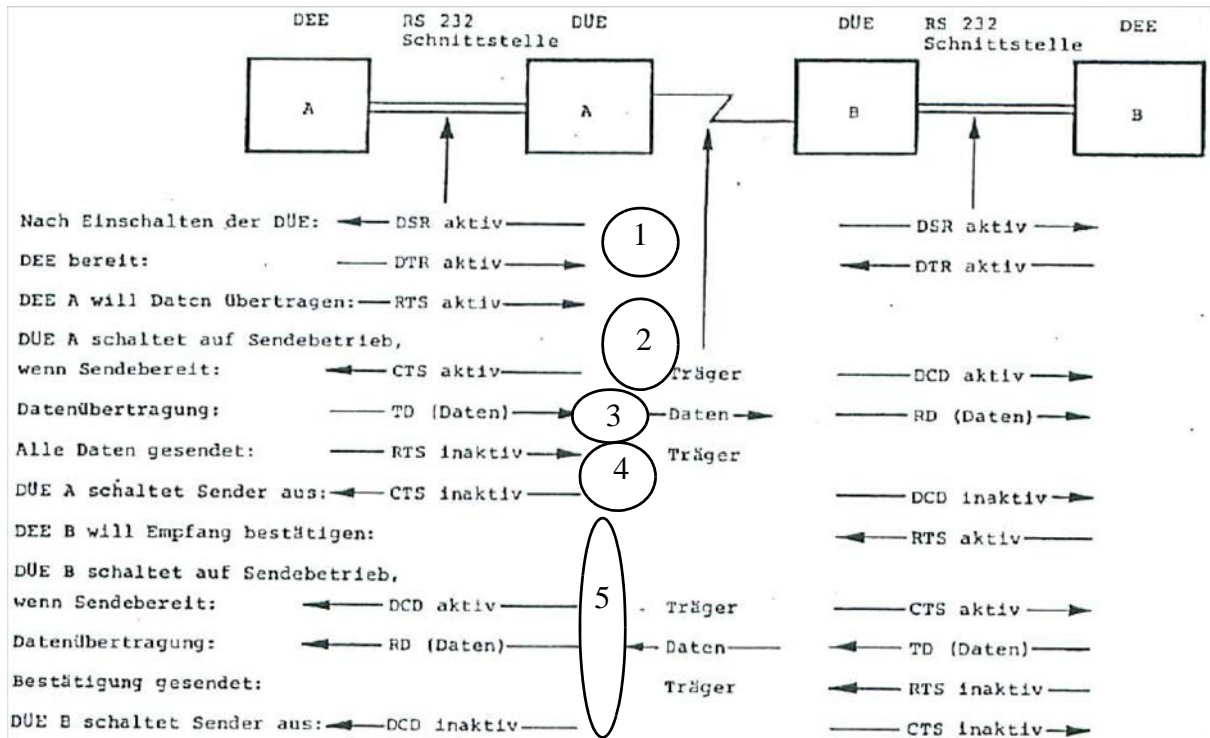


Abbildung 4: Beispiel für den logischen Ablauf der Interfacesignale während einer Übertragung im Halbduplex-Verkehr (Wechsel-Verkehr bei 2 Draht-Verbindung) über eine Standleitung

Mit dieser Anordnung wäre auch ein Vollduplex-Verkehr möglich wenn die Datenträger zwischen den DÜE A und DÜE B dies erlauben. Die V.24- bzw. RS232-Schnittstelle zwischen DÜE und DEE sind diesbezüglich nicht limitiert.

### 3.5 Wahl des Handshakeverfahrens

Über ein Handshake-Verfahren soll sicher gestellt werden, dass keine Daten dadurch verloren gehen können, dass z.B. der Empfänger nicht empfangsbereit ist (z.B. Buffer voll). Das im Versuch verwendete Terminal-Programm erlaubt es aber auch ohne Handshakes zu arbeiten. Hier werden die Handshake-Signale einfach komplett ignoriert.

Aber auch wenn im Terminal-Programm Hardware-Handshakes aktiviert sind, so können die Handshakes durch entsprechende Verschaltung der entsprechenden Signale deaktiviert werden, indem sämtliche Eingänge auf 1 gelegt werden.

Bei der V.24-Übertragung gibt es drei verschiedene Handshakeprotokolle.

### 3.5.1 RDY/BSY-Prozedur

Die Ready/Busy-Prozedur ist die einfachste und in der Praxis wohl am häufigsten verwendete Prozedur. Sie ist sozusagen eine "Hardwareprozedur", d.h. der Anwender braucht keine eigene Software dazu erstellen. Die Intelligenz für die ordnungsgemäße Kommunikation zweier Geräte befindet sich schon in diesen Geräten. Zur Signalisierung des Status "Ready"(bereit zum Datenempfang) oder "Busy" (Gerät ist beschäftigt und kann keine Daten empfangen) wird im einfachsten Fall nur eine Leitung benötigt, nämlich DTR (Data Terminal Ready). Ist diese Leitung positiv, so besteht Empfangsbereitschaft (beim Endgerät - DEE). Negative Spannung hingegen signalisiert den Busy-Status.

Die Empfangsbereitschaft beim Übertragungsgerät (DÜE - z.B. Modem) wird dem DEE durch das Signal DSR (Data Set Ready) signalisiert.

Entsprechend tauschen das Übertragungsgerät und das Endgerät über DTR (DEE -> DÜE) und DSR (DÜE -> DEE) gegenseitig die Betriebsbereitschaft aus.

Außerdem dienen RTS (Request to Send - DEE -> DÜE) und CTS (Clear to Send - DÜE -> DEE) dazu den Datensende-Request des Endgeräts anzukündigen und vom Übertragungsgerät zu bestätigen.

Das Datensenden vom Übertragungsgerät zum Datenendgerät wird durch das Signal DCD (Data Carrier Detect) angekündigt. Ein Handshake von der DEE zur DÜE existiert hier nicht. Es wird davon ausgegangen, dass das DEE empfangsbereit ist wenn DTR (Data Terminal Ready) aktiv ist.

### 3.5.2 ETX/ACK-Prozedur

Bei dieser Prozedur werden die ASCII-Zeichen ETX (03hex - End of Text) und ACK (06hex - Acknowledge) angewandt. Ist das Endgerät (DEE) bereit, Daten entgegenzunehmen, so wird DTR positiv und das Endgerät sendet das ACK-Zeichen an die Übertragungseinheit (DÜE). Diese sendet die Daten, die mit einem ETX-Zeichen abgeschlossen werden. Erkennt das Endgerät unter den Daten das ETX-Zeichen, so sendet es wieder den ACK-Code zur Übertragungseinheit und signalisiert damit, dass das nächste Datenpaket gesendet werden kann. Der ETX-Code muss im Datenfluß der Übertragungseinheit entsprechend der Pufferkapazität des Endgerätes eingebracht werden.

Diese Handshake-Form ist für die Druckerkommunikation geeignet, um dem Drucker anzuzeigen wann der Datendruck von übertragenen Daten gestartet werden kann (z.B. am Seiten- oder Zeilenende...).

Dieses Handshake wird vom Datensender gesteuert (ETX senden wenn Datenblock übertragen wurde und auf ACK warten). Es muss also sicher gestellt werden, dass der Empfänger die gesendeten Daten aufnehmen kann.

### 3.5.3 XON/XOFF-Prozedur

Bei dieser Prozedur kommen die Steuerzeichen XON und XOFF zur Anwendung (ASCII-Zeichen DC1 und DC3). Das Endgerät sendet bei Empfangsbereitschaft (Ready) einen XON-Code (11 hex), im anderen Fall (Busy) einen XOFF-Code (13 hex).

Während des Empfangs von Daten kann nun auf der Sendeleitung XOFF gesendet werden, um dem Sender zu signalisieren, dass man keine weiteren Daten mehr empfangen kann. Wenn der Empfänger wieder empfangsbereit ist, so sendet er XON. Bei dieser Übertragungsart kann also der Empfänger - ähnlich wie beim Hardware-Handshake - die Übertragung kontrollieren. Hierfür sind nur 3 Leitungen erforderlich: TxD, RxD und SGND.

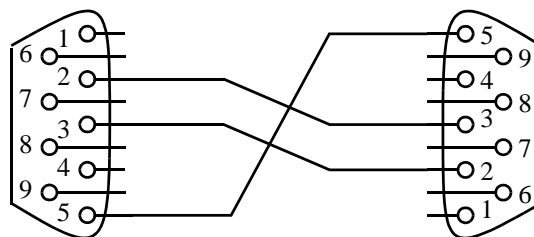
### 3.6 Nullmodem-Verbindung

Wie in Abschnitt 3.3 erklärt, ist die Bedeutung der Signal-Leitungen aus der Sicht der Datenendeinrichtungen definiert (DEE). Das heißt z.B., dass der Datenausgang einer Datenendeinrichtung (DEE) der TxD-Pin (Transmitted Data) ist. Bei der Datenübertragungseinrichtung (DÜE) stellt der TxD-Pin entsprechend den Dateneingang dar. Damit sind die Kabel für die Verbindung einer Datenendeinrichtung (DEE) mit einer Datenübertragungseinrichtung (DÜE) ein zu eins Verbindungen (Pin x der DEE ist mit Pin x der DÜE verbunden).

Sollen nun zwei Datenendeinrichtungen (DEE1 und DEE2) direkt über die V.24-Schnittstellen miteinander verbunden werden - also ohne Zwischenschaltung Datenübertragungseinrichtung, so spricht man von einer Nullmodem-Verbindung (also eine Verbindung mit NULL Modems). Hierbei ist nun natürlich darauf zu achten, dass die Datenleitungen gekreuzt werden müssen, damit der TxD-Pin der Datenendeinrichtung DEE1 mit dem RxD-Pin der Datenendeinrichtung DEE2 verbunden wird. Entsprechend können dann auch die Steuerleitungen miteinander verbunden werden.

Die verschiedenen Möglichkeiten werden in den folgenden Abschnitten kurz vorgestellt und in der Aufgabenstellung zur Versuchsvorbereitung (siehe Abschnitt 4) diskutiert. Die Informationen stammen primär aus [4].

#### 3.6.1 Nullmodem ohne Handshake



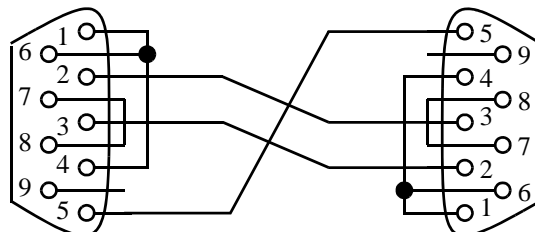
Pin DEE1		Pin DEE2
5	SGND	5
2(RxD)	Empfangsdaten ← Sendedaten	3(TxD)
3(TxD)	Sendedaten → Empfangsdaten	2(RxD)

Abbildung 5: Nullmodem ohne Handshake

Die in Abbildung 5 dargestellte Nullmodem-Realisierung erfordert minimalen Hardware-Aufwand. Man kommt mit einem 3-adrigen Kabel aus.

### 3.6.2 Nullmodem mit Rückkopplung der eigenen Handshake-Signale

Die in Abbildung 6 dargestellte Nullmodem-Realisierung erfordert nur ein 3-adriges Verbindungskabel und eine Rückkopplung der Handshake-Request-Leitungen auf die jeweiligen Acknowledge-Signale des selben Steckers. Wird damit ein wirkliches Handshake erreicht? Was bedeutet dies für den Betrieb? Was bedeutet dies für die Implementierung der V.24/RS232-Schnittstelle? (diese Fragen sind in den Aufgaben zur Versuchsvorbereitung zu beantworten).



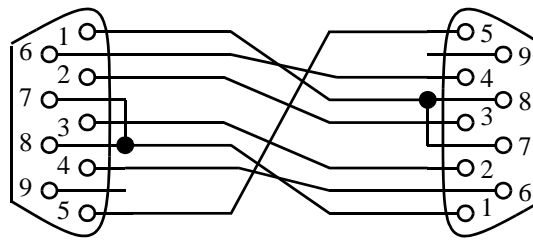
Pin DEE1		Pin DEE2
5	SGND	5
2(RxD)	Empfangsdaten ← Sendedaten	3(TxD)
3(TxD)	Sendedaten → Empfangsdaten	2(RxD)
1(DCD)+4(DTR)+6(DSR)	Ausgangssignal DataTerminalReady (praktisch immer auf logisch 1) treibt folgende Eingänge des eigenen DEE1: • DSR: DEE2 betriebsbereit (immer) • DCD: DEE2 kündigt (immer) neue Daten zum Empfang von DEE1 an	
	entsprechend vorherige Zeile	1(DCD)+4(DTR)+6(DSR)
7 (RTS) + 8 (CTS)	Ausgangssignal „Request to Send“ ist auf „Clear to Send“ zurück gekoppelt (signalisiert quasi beim Sendewunsch, dass DEE2 (immer) betriebsbereit sei)	
	entsprechend vorherige Zeile	7 (RTS) + 8 (CTS)

Abbildung 6: Nullmodem mit Rückkopplung der eigenen Handshake-Signale

### 3.6.3 Nullmodem mit teilweise realisiertem Handshake

Die in Abbildung 7 dargestellte Nullmodem-Realisierung erfordert ein 6-adriges Verbindungskabel und eine Rückkopplung der Handshake-Request-Leitungen RTS auf das Acknowledge-

Signal CTS des selben Steckers. Welche Übertragungen sind hier nicht durch ein Handshake abgesichert? (diese Frage ist in den Aufgaben zur Versuchsvorbereitung zu beantworten)



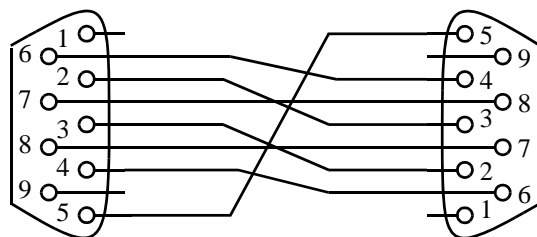
Pin DEE1		Pin DEE2
5	SGND	5
2(RxD)	Empfangsdaten ← Sendedaten	3(TxD)
3(TxD)	Sendedaten → Empfangsdaten	2(RxD)
4(DTR)	DEE1 bereit → DEE2 DSR-Pin am DEE2 zeigt an, ob DEE1 betriebsbereit ist.	6(DSR)
6(DSR)	DEE1 ← DEE2 bereit entsprechend vorherige Zeile	4(DTR)
7 (RTS) + 8 (CTS)	Ausgangssignal „Request to Send“ des DEE1 ist auf eigenen „Clear to Send“ zurück gekoppelt (signalisiert quasi beim Sendewunsch, dass DEE2 (immer) betriebsbereit sei). Dieses Signal wird gleichzeitig dafür verwendet den Sendewunsch von DEE1 auch wirklich an DEE2 weiter zu melden.	1(DCD)
1(DCD)	entsprechend vorheriger Zeile	7 (RTS) + 8 (CTS)

Abbildung 7: Nullmodem mit teilweise realisiertem Handshake

### 3.6.4 Nullmodem mit vollem Handshake

Die in Abbildung 8 dargestellte Nullmodem-Realisierung erfordert ein 7-adriges Verbindungskabel und keine Rückkopplung an dem eigenen Stecker. Welches Signal erhält hier beim Datenempfänger eine zur originalen Definition abgewandelte Bedeutung? (diese Frage ist in den Aufgaben zur Versuchsvorbereitung zu beantworten)

Dieses Nullmodem ist das gebräuchlichste Modem wenn ein Kommunikationspartner ein PC ist.



Pin DEE1		Pin DEE2
5	SGND	5
2(RxD)	Empfangsdaten ← Sendedaten	3(TxD)
3(TxD)	Sendedaten → Empfangsdaten	2(RxD)
4(DTR)	siehe Abbildung 7: „Nullmodem mit teilweise realisiertem Handshake“	6(DSR)
6(DSR)		4(DTR)
7 (RTS)	Ausgangssignal „Request to Send“ des DEE1 ist mit „Clear to Send“ von DEE2 verbunden. Warum ist dies ein Kompatibilitäts-Thema zu der Standard-Definition von CTS („Clear to Send“) beim Anschluß von der Dateneneinrichtung DEE1 an eine Datenübertragungseinrichtung (DÜE)?	8 (CTS)
8 (CTS)	entsprechend vorheriger Zeile	7 (RTS)

Abbildung 8: Nullmodem mit vollem Handshake

## 4 Versuchsvorbereitung

Gruppe:	Name(n):
---------	----------

Dieser Teil ist eigenhändig handschriftlich vor dem Praktikumstermin in den folgenden Tabellen vorzubereiten.

- 1) Wo liegt der Unterschied zwischen parallelen und seriellen Übertragungsprotokollen? Stellen Sie stichwortartig die Vor- und Nachteile einer parallelen Übertragung zusammen:

Unterschiede:

Vorteile parallele Übertragung	Nachteile parallele Übertragung:
--------------------------------	----------------------------------

- 2) Geben Sie die Charakteristika der hier vorgestellten V.24-Datenübertragung an:

parallele oder serielle Datenübertragung?

Wieviel Teilnehmer können über einen Verbindungskanal kommunizieren?

Arbeiten die Kommunikationspartner automatisch mit der selben Übertragungsrage?

Wird ein Datenübertragungsfehler eindeutig durch die Übertragung eines Paritätsbits erkannt (Begründung)?

Durch welche Spannungspegel wird die logische 0 und 1 im Protokoll definiert (Achtung: nicht für alle Signale gleich)?

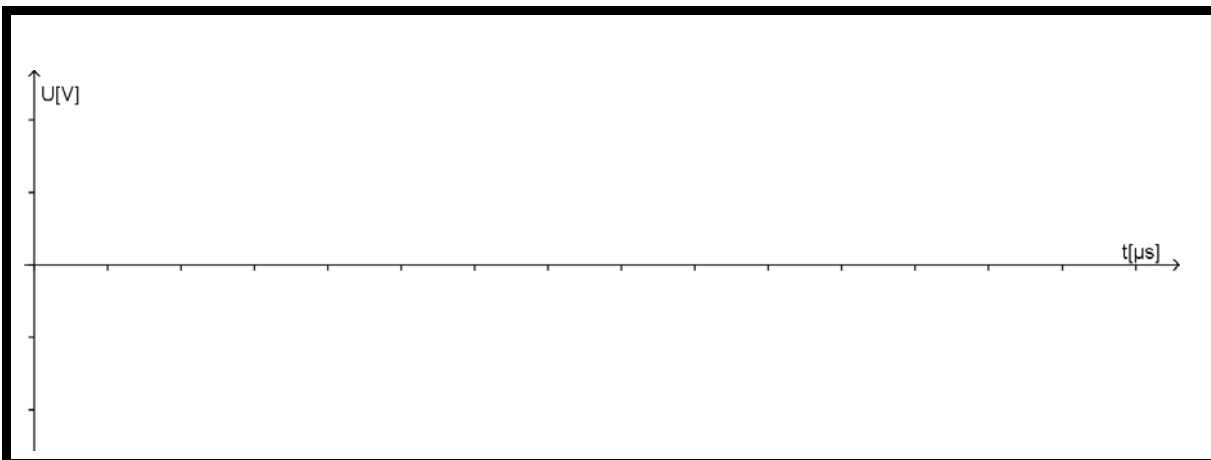
welche Signale?		
logisch 0:		
logisch 1:		

Sind im Protokoll Hardware-Handshake-Signale definiert?

Ist ein Software-Handshake über die Daten-Leitungen möglich (wenn ja - gibt es mehrere und ggf. wo unterscheiden sich diese)?

- 3) Wie lautet bei gerader Parität das Paritätsbit wenn die hexadezimal übertragenen 8-Bit-Daten 0x76 lauten?

- 4) Erstellen Sie ein Zeitdiagramm in dem Sie den Spannungsverlauf der Datenleitung TxD für die Übertragung des ersten beiden Buchstaben Ihres Vornamens skizzieren (ein Vorname reicht). Geben Sie pro Takt die Bedeutung des Spannungspegels an.  
Randbedingungen: 1 Stop-Bit, gerade Parität, 8 Datenbits, 9600 Baud (ASCII-Tabelle siehe z.B.:  
[http://de.wikipedia.org/wiki/American\\_Standard\\_Code\\_for\\_Information\\_Interchange](http://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange))



Hilfe bei der Beantwortung der folgenden Fragen finden Sie in diesem Dokument oder unter [http://www.lammertbies.nl/comm/info/RS-232\\_null\\_modem.html](http://www.lammertbies.nl/comm/info/RS-232_null_modem.html)



- 5) Beschreiben Sie Probleme, die auftreten können wenn ein Nullmodem-Kabel nach Abschnitt 3.6.1 (Nullmodem ohne Handshake) verwendet wird?  
Hinweis: Denken Sie bei der Beantwortung an 2 Endgeräte, die das Hardware-Handshakes verwenden. Wie verhalten sich die Endgeräte wenn die Handshake-Signale nicht verdrahtet sind?

- 6) Was verändert sich bei der Verwendung eines Nullmodem-Kabel nach Abschnitt 3.6.2 (Nullmodem mit Rückkopplung der eigenen Handshake-Signale)?

- 7) Was verändert sich bei der Verwendung eines Nullmodem-Kabel nach Abschnitt 3.6.3 (Nullmodem mit teilweise realisiertem Handshake)?

- 8) Wo liegen die Kompatibilitätsprobleme bezüglich des Standard-V.24 Protokolls bei der Verwendung eines Nullmodem-Kabels nach Abschnitt 3.6.4 (Nullmodem mit vollem Handshake)?

Testat (diese Zettel sind mit unterschriebenem Testat der Versuchsausarbeitung beizulegen):

## 5 Versuchsanleitung

In dem Versuch soll die Kommunikation über einen PC unter Verwendung eines Terminal-Programms hergestellt und analysiert werden. Als Terminal-Programm wird folgendes verwendet: <https://sites.google.com/site/terminalbpp/> (siehe auch Abb. 9)

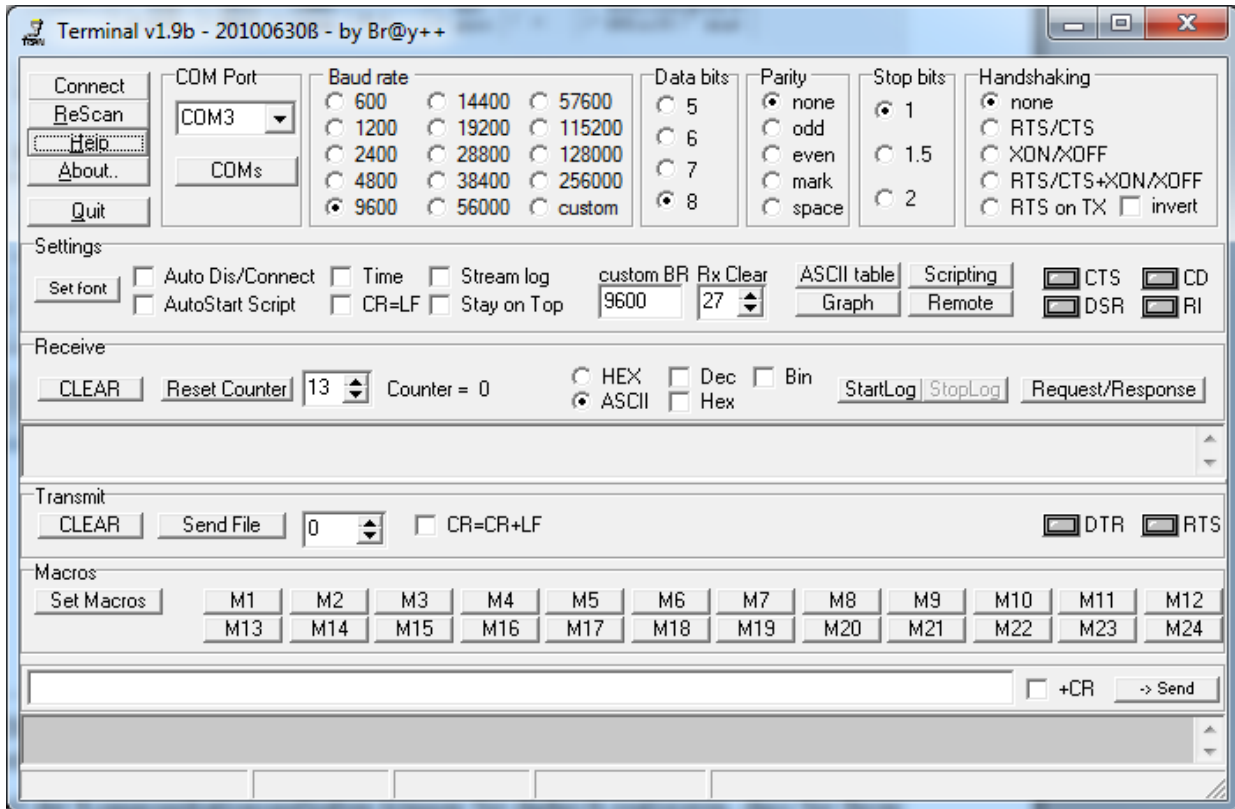


Abbildung 9: Oberfläche des Terminal-Programms

Der 9-polige RS232-Stecker ist auf ein Steckboard geführt. Auf dem Steckboard können Sie per Bananenstecker Verbindungen herstellen.

Den größten Teil der Kommunikationsaufgaben können Sie dadurch realisieren, dass Sie Ihren eigenen PC als Datensender und Datenempfänger konfigurieren. Die Ergebnisse der Aufgaben sind zu oszillographieren und die Bilder im Versuchsprotokoll zu dokumentieren.

- 1) Überprüfen Sie, ob die Stecker-Pins Ein- oder Ausgänge sind und die Ausgänge die beschriebenen Funktionen erfüllen (Tipp: DTR und RTS können Sie durch Maus-Klick aktivieren oder deaktivieren).
- 2) Konfigurieren Sie das Terminal-Programm so, dass keine Handshakes implementiert sind. Verdrahten Sie das Nullmodem so, dass der eingegebene Text auch empfangen wird. Verwenden Sie hier unterschiedliche Paritätseinstellungen.
- 3) Konfigurieren Sie das Terminal-Programm so, dass ein Hardware-Handshake über RTS und CTS implementiert ist (zunächst ohne Verdrahtungsänderungen). Wie verhält sich der Empfänger und wie können Sie dieses Verhalten korrigieren?

Können Sie nun die Anzahl der Stop-Bits kontrollieren (wie und Bild)?

Anmerkung: RTS und DTR können per Mausklick auf 1- bzw. 0-Pegel im Terminal-Programm gesetzt werden.

- 4) Konfigurieren Sie das Terminal-Programm so, dass ein Hardware-Handshake über RTS/CTS und DTR/DTS implementiert ist. Verdrahten Sie das Nullmodem so, dass der eingegebene Text auch empfangen wird. Welche Handshake-Signale müssen verdrahtet werden?
- 5) Überprüfen Sie jeweils pro Leitung (Daten- und Steuerleitungen) in welchem Bereich der „unerlaubte Spannungsbereich liegt.  
Tipp: Die Spannung können Sie durch eine Ohmsche Last nach Masse verringern.
- 6) Stellen Sie eine Kommunikation mit dem Rechner einer anderen Versuchsgruppe her. Verwenden Sie zunächst die Konfiguration wie in der vorherigen Aufgabe. Anschließend setzen Sie ein Software-Handshake um (für die Übertragung von Steuercodes siehe help-Menü des Terminal-Programms)!

## 6 Hinweise zum Versuchsprotokoll

Es ist jeweils die Verbindung auf dem Steckboard pro Aufgabe zu dokumentieren. Außerdem sind die Oszilloskop-Bilder **bitweise** zu dokumentieren (sämtliche übertragene Bits sind im Bild zu kennzeichnen - Start-, Stop-, Paritätsbits sowie die einzelnen Bits der übertragenen Buchstaben).

## 7 Literatur

- [1] Urbanski, Woitowitz – Digitaltechnik (Springer Verlag – ISBN-10 3-540-40180-6)
- [2] Temmo Weerts - Diplomarbeit „Universelle Messdatenerfassung für handelsübliche DOS-PC über die serielle Schnittstelle" (1992)
- [3] RS232 Specifications and standard  
[http://www.lammertbies.nl/comm/info/RS-232\\_specs.html](http://www.lammertbies.nl/comm/info/RS-232_specs.html)
- [4] RS232 serial null modem cable wiring  
[http://www.lammertbies.nl/comm/info/RS-232\\_null\\_modem.html](http://www.lammertbies.nl/comm/info/RS-232_null_modem.html)
- [5] Karlowsky - V.24 Versuchsanleitung des Labors für Digitaltechnik an der FH OOW
- [6] <http://de.wikipedia.org/wiki/RS-232>