

---

# Die RS232-/V.24-Schnittstelle

---

Das Herzstück der seriellen Schnittstelle im PC ist der serielle Baustein UART 8250 (*Universal Asynchronous Receiver Transmitter*). Dieser Baustein erlaubt die serielle Datenübertragung und übernimmt dabei jede von parallel nach seriell oder von seriell nach parallel Umwandlung der Daten.

Der 8250 Baustein verfügt über 10 interne Register, die es erlauben die Leitungen der seriellen Schnittstelle zu steuern, Daten zu senden und zu empfangen. Der größte Teil dieser Register wird bei der Initialisierung des Bausteins verwendet, während bei der Datenübertragung meistens nur ein bis zwei Register verwendet werden.

Auf die Register des 8250 Bausteins wird mit den Assembler-Befehlen OUT und IN zugegriffen.

Nach der Initialisierung der Schnittstelle, werden die Daten, die dem 8250 über das Senderegister (*Transmitter Holding Register*) übergeben werden, automatisch seriell übertragen so daß sich das Hauptprogramm anderen Aufgaben widmen kann. Es muß lediglich das Statusregister (*Line Status Register*) überprüfen bevor es ein weiteres Zeichen an das Senderegister übergeben kann.

## Die BIOS-Schnittstelle

---

Das BIOS des PC verfügt über Funktionen, die eine serielle Datenübertragung unterstützen. Diese Funktionen können verwendet werden um - ohne großen Aufwand - die serielle Schnittstelle zu initialisieren und Daten zu übertragen.

Diese BIOS-Funktionen werden mittels eines Software-Interrupt (INT14h) aufgerufen. Beim Einsprung muß das Register AH die Funktionsnummer und DX die serielle Schnittstelle enthalten. Eine detaillierte Beschreibung kann in der entsprechenden BIOS-Dokumentation gefunden werden.

Für einfache Anwendungen dürften diese BIOS-Funktionen ausreichend sein, doch sie sind nicht flexibel und schnell genug, um komplexe Aufgaben zu überwältigen. Deshalb muß ein Systemprogrammierer sich detaillierter mit der Programmierung des UART-Bausteins beschäftigen, wenn er in seiner Anwendung eine Datenübertragung im Hintergrund realisieren möchte, die interruptgetrieben, mit maximaler Geschwindigkeit und Sicherheit, komplizierte Flußsteuerung und unter verschiedenen Protokollen laufen soll.

## Programmierung des 8250-Bausteins

---

Der 8250-Baustein enthält 10 interne Register, die ihn auf den ersten Blick recht kompliziert erscheinen lassen. Diese Komplexität verschwindet aber sobald man sich diese Register aus der Nähe betrachtet. Außerdem werden nicht alle Register in jeder Situation verwendet.

Die folgende Tabelle listet diese Register und ihre Adresse auf.

Register	Adresse (Offset)
Transmitter Holding Register (THR)	0 <sup>1)</sup>
Receive Data Register (RDR)	0 <sup>1)</sup>
Baud Rate Divisor LSB	0 <sup>2)</sup>
Baud Rate Divisor MSB	1 <sup>2)</sup>
Interrupt Enable Register (IER)	1 <sup>1)</sup>
Interrupt Identification Register (IIR)	2
Line Control Register (LCR)	3
Modem Control Register (MCR)	4
Line Status Register (LSR)	5
Modem Status Register (MSR)	6

1) Bit 7 im Line Control Register = 0

2) Bit 7 im Line Control Register = 1

## Initialisierung

---

Die folgende 5 Register müssen initialisiert werden, bevor die serielle Schnittstelle ansprechbar ist. Diese Register sind:

- Baud Rate Divisor LSB
- Baud Rate Divisor MSB
- Line Control Register
- Modem Control Register
- Interrupt Enable Register

Es ist darauf zu achten, daß Bit 7 (DLAB) des *Line Control Registers* auf 1 steht, um die *Baud Rate Divisor Register* zu adressieren. In allen anderen Fällen ist dieses Bit = 0.

Für den weiteren Verlauf der Datenübertragung werden die einzelnen Daten über das *Transmitter Holding Register* gesendet und über das *Receiver Data Register* empfangen. Das *Line Status Register* teilt dem Anwendungsprogramm mit, ob ein Zeichen gesendet werden kann oder ob ein Zeichen empfangen wurde und abholbereit liegt.

Um ein Zeichen zu senden wird, im *Line Status Register* das Bit 5 überprüft. Ist Bit 5 = 1, ist das *Transmitter Holding Register* leer, d. h., der 8250 ist bereit ein weiteres Zeichen aufzunehmen und zu senden. Ob ein Zeichen abholbereit vorliegt wird über das Bit 0 des *Line Status Registers* herausgefunden. Dafür muß Bit 0 = 1 sein.

## Baud Rate Divisor

---

Die Übertragungsgeschwindigkeit der seriellen Schnittstelle wird über die beiden *Baud Rate Register* eingestellt. Dabei wird der Teilerwert (1 Word) über die ganzzahlige Division

Teiler = 115200 / Baudrate

ermittelt und der LSB-Teil des Teilerwerts im *Baud Rate Register* LSB übertragen und der MSB-Teil zum *Baud Rate Register* MSB.

Für die Initialisierung auf 9600 Baud ergibt die obige Formel

$$115200 / 9600 = 12 = 0\text{Ch.}$$

Damit wird das *Baud Rate Register* LSB mit 0Ch initialisiert und das *Baud Rate Register* MSB mit 00h.

Wie schon erwähnt, muß für die Initialisierung der *Baud Rate Register* Bit 7 (DLAB) des *Line Control Registers* auf 1 gesetzt werden.

**Beispiel:**

```
MOV AL,80h           ; Bit 7 Maske
MOV DX,IOPort+3     ; Line Control Register
OUT DX,AL           ; Bit 7 setzen

MOV DX,IOPort       ; Baud Rate Register LSB
MOV AX,Teiler       ; Teilerwert
OUT DX,AL           ; LSB
MOV DX,IOPort+1     ; Baud Rate Register MSB
MOV AL,AH
OUT DX,AL           ; MSB
```

---

## Line Control Register

---

Nachdem die Einstellung der Übertragungsgeschwindigkeit erfolgt ist, muß über das *Line Control Register* die Einstellung des Protokolls (Zeichenlänge, Anzahl der Stopbits, Parität usw.) erfolgen.

Diesmal muß Bit 7 = 0 (DLAB) sein, um nicht an das *Baud Rate Register* MSB zu adressieren.

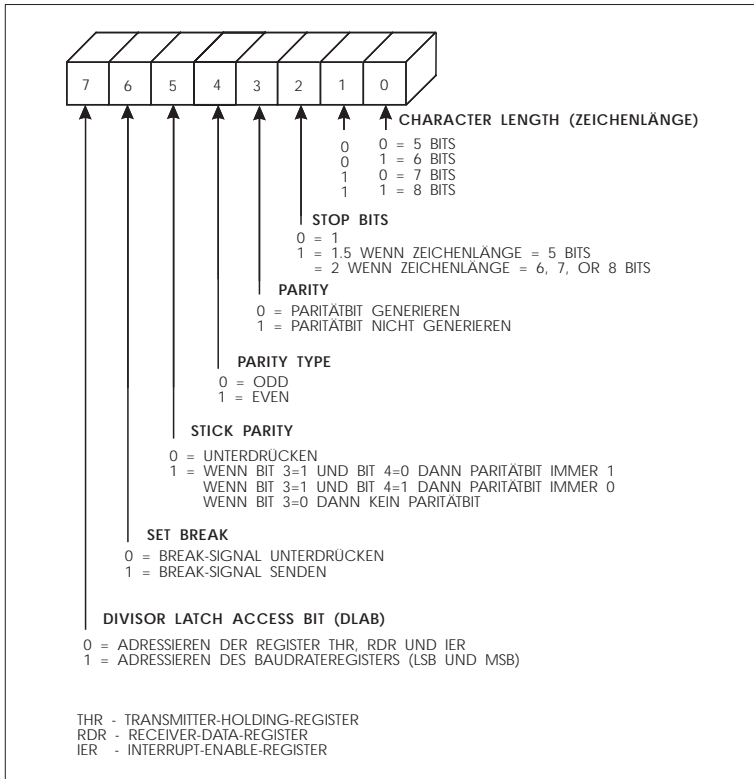


Abb. 1-1 Line Control Register

**Beispiel:**

```

MOV AL,00000011b ; Bitmaske für 8 Daten-
                  ; bits,
                  ; 1 Stopbits
                  ; keine Parität

MOV DX,IOWReg+3 ; Line Control Register
OUT DX,AL ; initialisieren

```

## Modem Control Register

Das *Modem Control Register* dient zur Steuerung der Modemleitungen. Dieses Register wird in der nicht interruptfähigen Arbeitsweise mit 03h initialisiert (DTR=1 und RTS=1), was für die meisten Fälle richtig ist und wenn kein Modem angeschlossen ist, auch nicht weiter stört.

Wenn die serielle Schnittstelle im Interruptbetrieb arbeiten soll (Kommunikation im Hintergrund), ist Bit 3 (OUT2) auf 1 zu setzen. Im Loop-Modus ist dagegen Bit 4 auf 1 zu setzen.

Im Interruptbetrieb muß für die Schnittstelle auch eine IRQ-Leitung existieren, über die der Interrupt ausgelöst werden kann. Zusätzlich muß der 8259 Interrupt Controller so eingestellt werden (entsprechendes Bit auf 0 setzen), daß über die betreffende IRQ-Leitung ein Interrupt ausgelöst werden kann.

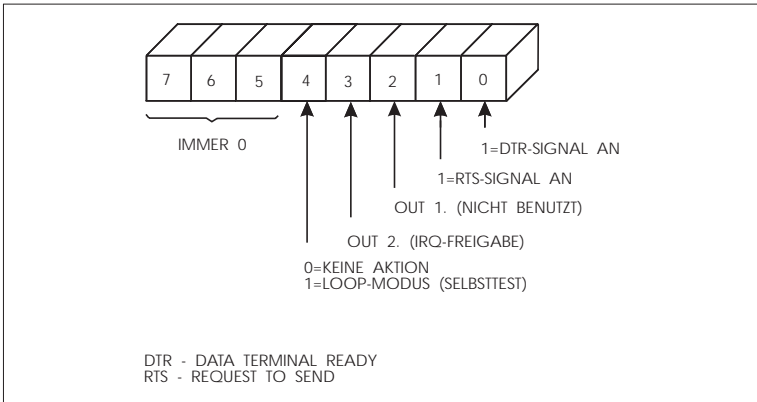


Abb. 1-2 Modem Control Register

Im Loop-Modus (Testbetrieb) wird jedes gesendete Zeichen wieder von der Schnittstelle empfangen. Damit lässt sich die Arbeitsweise der Schnittstelle testen.

### Beispiel:

```
MOV AL,00001011b ; Bitmaske für DTR, RTS,  
                  ; OUT2  
MOV DX,IOReg+4   ; Modem Control Register  
OUT DX,AL        ; initialisieren
```

## Interrupt Enable Register

Der 8250-Baustein verfügt über mehrere Interruptquellen, kann aber von sich aus nur einen Interrupt auslösen.

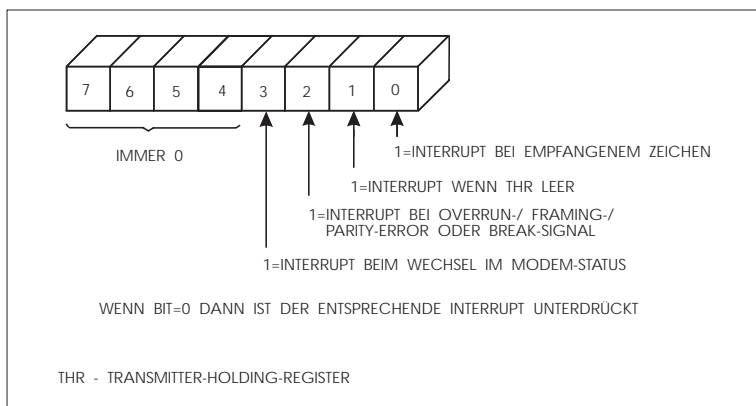


Abb. 1-3 Interrupt Enable Register

Wenn OUT2 im *Modem Control Register* auf 1 gesetzt wurde, dann können über das *Interrupt Enable Register* die Ereignisse eingestellt werden, die zu einer Unterbrechung führen dürfen.



Die möglichen Ereignisse sind:

- Interrupt nach Empfang eines Zeichens
- Interrupt wenn *Transmitter Holding Register* leer
- Interrupt, wenn Fehler bei der Übertragung oder wenn ein Break-Signal erkannt wird
- Interrupt nach einer Änderung im Status der Modemleitungen.

**Beispiel:**

```
MOV  AL,00001111b  ; Bitmaske für alle  
                        ; Interruptquellen  
MOV  DX,IOWReg+1   ; Interrupt Enable  
                        ; Register  
OUT  DX,AL         ; initialisieren
```

## Datenübertragung mit dem 8250

---

Während der Datenübertragung sind drei Register von großer Wichtigkeit:

- Line Status Register
- Transmitter Holding Register
- Receiver Data Register

Die einzelnen Daten werden über das *Transmitter Holding Register* gesendet und über das *Receiver Data Register* empfangen.

Um ein Zeichen zu senden, muß im *Line Status Register* das Bit 5 überprüft werden. Ist Bit 5 = 1, ist das *Transmitter Holding Register* leer und der 8250 ist bereit ein weiteres Zeichen

aufzunehmen und zu senden. Solange ein Zeichen gesendet wird, hat Bit 5 den Wert 0.

Um ein Zeichen zu empfangen, muß Bit 0 des *Line Status Registers* überprüft werden. Ist dieses Bit eine 1, dann wurde ein Zeichen empfangen und es ist abholbereit. Das Bit 0 wird zurückgesetzt, wenn das empfangene Zeichen aus dem *Receiver Data Register* ausgelesen wurde.

Ein Übertragungsfehler ist aufgetreten, wenn Bit 1, 2 oder 4 des *Line Status Registers* den Wert 1 haben. Ein erkanntes Break-Signal wird über Bit 4 mitgeteilt. Mögliche Fehler während der Datenübertragung sind:

- Rahmenfehler (Framing Error) wenn z. B. falsche Wortlänge, falsche Anzahl von Datenbits oder Stopbits benutzt wird.
- Paritätsfehler (Parity Error) wenn die Einstellung der Parität nicht übereinstimmt
- Überlauffehler (Overrun Error) wenn empfangene Zeichen durch neu empfangene Zeichen überschrieben worden sind

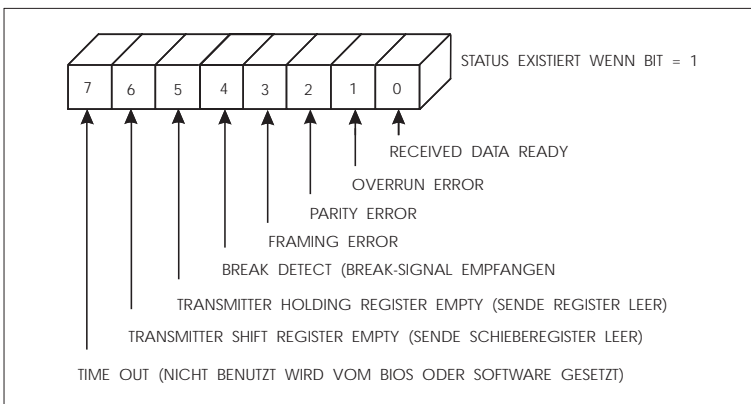


Abb. 1-4 Line Status Register

Über das *Modem Status Register* läßt sich der Status der Modemleitungen ermitteln. Die Bits 0..3 zeigen den Deltazustand der Leitungen CTS, DSR, RI und DCD an und die Bits 4 bis 7 enthalten den aktuellen Stand der Leitungen CTS, DSR, RI und DCD.

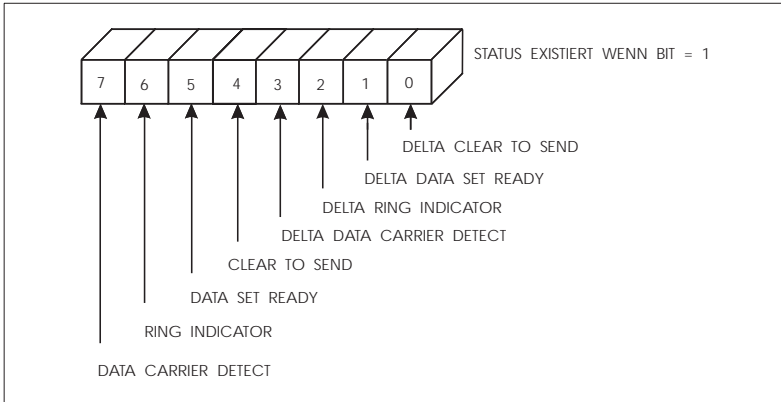


Abb. 1-5 Modem Status Register

## Modemsignale

### Vom Computer zum Modem

**DTR:** Teilt dem Modem mit, daß der Computer eingeschaltet und bereit ist.

**RTS:** Teilt dem Modem mit, daß das Programm im Computer Daten übermitteln will.

### Vom Modem zum Computer

**DSR:** Das Modem teilt dem Computer mit, daß es eingeschaltet und bereit ist.

**CTS:** Das Modem teilt dem Computer mit, daß es bereit ist Daten zu empfangen.

**DCD:** Das Modem teilt dem Computer mit, daß es eine Verbindung mit einem entfernten Modem aufgebaut hat.

**RI:** Das Modem teilt dem Computer mit, daß es einen ankommenden Ruf auf der Telefonleitung erkannt hat.

## Der Interruptbetrieb des 8250

Der Interruptbetrieb des 8250 wird selektiert, wenn die Anwendung nicht die Zeit hat die Schnittstelle zu bedienen (Schnittstellenstatus abfragen usw.) und die Datenübertragung deshalb im Hintergrund erfolgen muß.

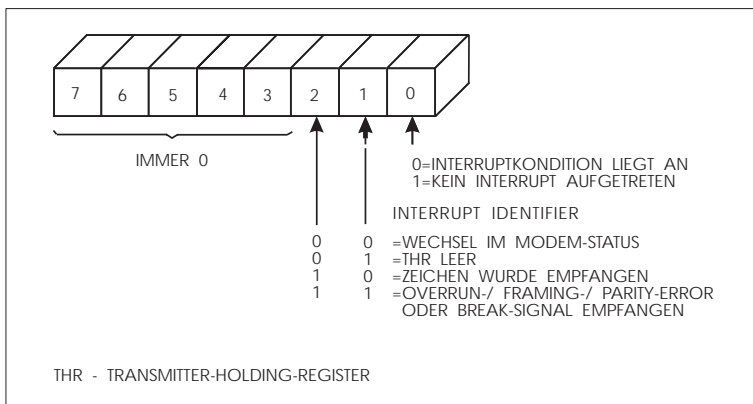


Abb. 1-6 Interrupt Identification Register

In diesem Fall wird der Interruptbetrieb des 8250-Bausteins über die entsprechende Register eingestellt und die Kommunikation erfolgt im Hintergrund. Der 8250-Baustein generiert eine Unterbrechung, wenn das Ereignis auf der seriellen Schnittstelle vom *Interrupt Enable Register* überwacht wird. Diese Unterbrechung muß dann von einer Interrupt-

Routine bearbeitet werden. Abhängig von der Einstellung des *Interrupt Enable Registers* muß die Interrupt-Routine eine oder mehrere Aufgaben bewältigen können.

Da mehrere Ereignisse nur einen Hardware-Interrupt auslösen, kann die Interrupt-Routine über das *Interrupt Identification Register* Auskunft darüber erhalten, welches Ereignis die Unterbrechung ausgelöst hat und abhängig davon den entsprechenden Prozeß starten. Wenn mehrere Ereignisse gleichzeitig auftreten, müssen sie von der Interrupt-Routine nacheinander abgearbeitet werden. In diesem Fall entscheidet die Priorität des Interrupttyps welches Ereignis als erstes behandelt wird. Bit 0 bleibt solange 0 bis alle Ereignisse, die die Unterbrechung verursacht haben, abgearbeitet werden.

Im *Interrupt Identification Register* sind nur die ersten drei Bits von Bedeutung.

- Bit 0 hat den Wert 0, wenn ein Interrupt anliegt.
- Bit 1 und 2 liefern den Interrupt-Code, wenn Bit 0=0

Der Interrupt-Code liefert Information über den Typ der Unterbrechung und damit über die Priorität, mit der das Ereignis behandelt wird. Jeder Interrupt Typ hat eine eigene Priorität. Der Interrupt-Code 11 hat die erste Priorität, der Interrupt-Code 10 die zweite, 01 die dritte und 00 die vierte Priorität.

Die Interrupt-Codes haben folgenden Interrupttyp:

Code	Priorität	Typ
11	1	Fehler in der Übertragung oder Break-Signal empfangen
10	2	Empfangene Daten abholbereit
01	3	Senderegister leer ( <i>Transmitter Holding Register</i> )
00	4	Änderung im Status der Modemleitungen

Um den Interruptbetrieb zu ermöglichen, muß die Initialisierung mit den folgenden 6 Schritten erfolgen.

1. *Baud Rate Divisor* LSB und MSB initialisieren
2. Im *Line Control Register* die Übertragungscharakteristigen einstellen (Datenbits, Stopbits usw.)
3. Das *Modem Control Register* so einstellen, daß der Interrupt-Betrieb möglich ist (OUT2=1)
4. Im *Interrupt Enable Register* die Ereignisse definieren, die überwacht werden sollen
5. Interrupt-Routine auf den entsprechenden Interruptvektor installieren
6. Im 8259 Interrupt Controller die entsprechende IRQ-Leitung freigeben (z.B. Bit 4 = 0, für IRQ4 und COM1)

Angenommen das *Interrupt Enable Register* wurde auf den Interrupttyp 10 eingestellt und die Leitung OUT2 des *Modem Control Registers* sowie der 8259 IC aktiviert, dann kann die folgende Interrupt-Routine, nach einer Unterbrechung von der seriellen Schnittstelle, das empfangene Zeichen in einen dafür vorgesehenen Puffer ablegen.

**Beispiel:**

```

TITLE COMISR
LOCALS

DATA    SEGMENT      WORD PUBLIC

        ASSUME DS:DATA

        WrPtr    dw 0
        Buffer    db 32 dup (?)

DATA    ENDS

IOReg   EQU    3F8h           ; COM1

CODE    SEGMENT      BYTE PUBLIC

        ASSUME      CS:CODE

ISR     PROC FAR
        PUSH  AX           ; Register sichern
        PUSH  BX
        PUSH  DS

        MOV   AX,SEG Data  ; Datensegment des
                           ; Empfangspuffers
        MOV   DS,AX        ; als neues DS setzen
        MOV   BX,WrPtr     ; Schreibzeiger holen
    
```

```

MOV    DX,IOReg      ; Portadresse der se-
                    ; riellen Schnitt-
                    ; stelle
IN     AL,DX         ; Zeichen abholen

MOV    Buffer[BX],AL ; Zeichen im Puffer
                    ; ablegen
INC    BX            ; Schreibzähler wei-
                    ; terschalten

MOV    AL,020h      ; EOI Wert laden
OUT    20h,AL       ; EOI an 8259 senden

POP    DS            ; Register wiederher-
                    ; stellen

POP    BX
POP    AX

IRET                    ; Zurück zum unter-
                    ; brochenen Programm

ISR    ENDP

CODE   ENDS
       END

```

Diese Interrupt-Routine ist in ihrer Arbeitsweise recht einfach, da sie auch nur ein Beispiel darstellt, wie eine Interrupt-Routine die Unterbrechungen des 8250-Bausteins bearbeiten kann. Sie kann aber ohne weiteres erweitert werden um den Überlauf des Puffers abzufangen, Flußsteuerung durchzuführen, weitere Ereignisse bearbeiten zu können, variable Sende- und Empfangspuffer...

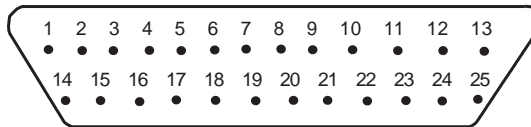


# Pinbelegung der RS232-/V.24-Schnittstelle

---

## DB 25 Anschlußstecker

---

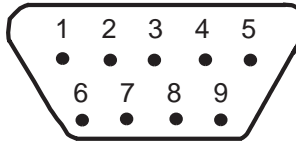


Die Pinbelegung des Anschlußsteckers DB25 ist folgende:

Pin	DB25
1	
2	TXD
3	RXD
4	RTS
5	CTS
6	DSR
7	GND
8	DCD
9	
20	DTR
22	RI

## DB 9 Anschlußstecker

---

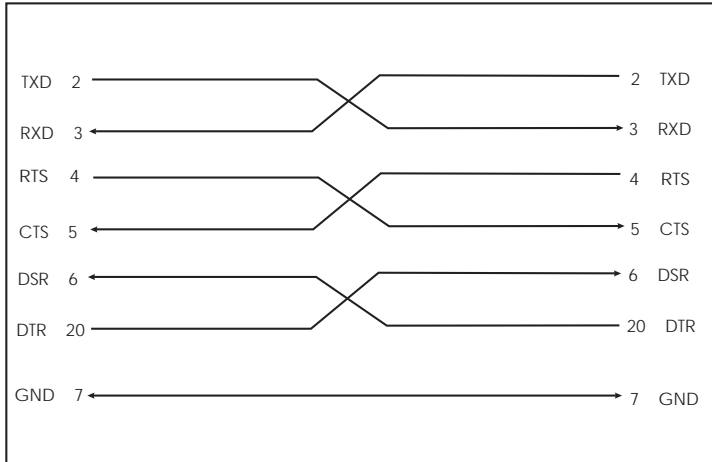


Die Pinbelegung des Anschlußsteckers DB9 ist folgende:

Pin	DB9
1	DCD
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

Die verwendeten Abkürzungen sind im Glossar erklärt.

## NULL-Modem-Verbindung mit DB25



### 3-Draht NULL-Modem mit DB25

---

**Wichtig:** Wenn nur eine 3-Draht-Verbindung verwendet wird, sollten die Modemleitungen, wie abgebildet, kurzgeschlossen werden, sonst werden häufig durch Störsignale blinde Interrupts für diese Leitungen ausgelöst und damit unnötige Last erzeugt, die auch den Datendurchsatz negativ beeinflussen kann.

