

5. Serielle Schnittstelle

5.1 Vergleich der seriellen und parallelen Datenübertragung

Man unterscheidet zwischen paralleler und serieller Datenübertragung.

- Bei der **parallelen Datenübertragung** werden jeweils n Datenbit gleichzeitig (parallel) über n Datenleitungen übertragen.

Vorteil: Hohe Übertragungsgeschwindigkeit.

- Bei der **seriellen Datenübertragung** werden die n Datenbit nacheinander (seriell) über nur eine Datenleitung übertragen.

Vorteil: Einsparungen von Übertragungsleitungen.

5.2 Serielle Datenübertragung

5.2.1 Einleitung

Anwendungen:

- serielle Schnittstelle
- USB
- Firewire

Versuch: Serielle Datenübertragung mit Hilfe einer Taschenlampe

Messergebnisse:

ohne Takt: gesendete Daten: 10100111

empfangene Daten: 10101

Beobachtung:

Ohne Takt können aufeinanderfolgende identische Bits nicht voneinander unterschieden werden.

mit Takt: gesendete Daten: 10100111

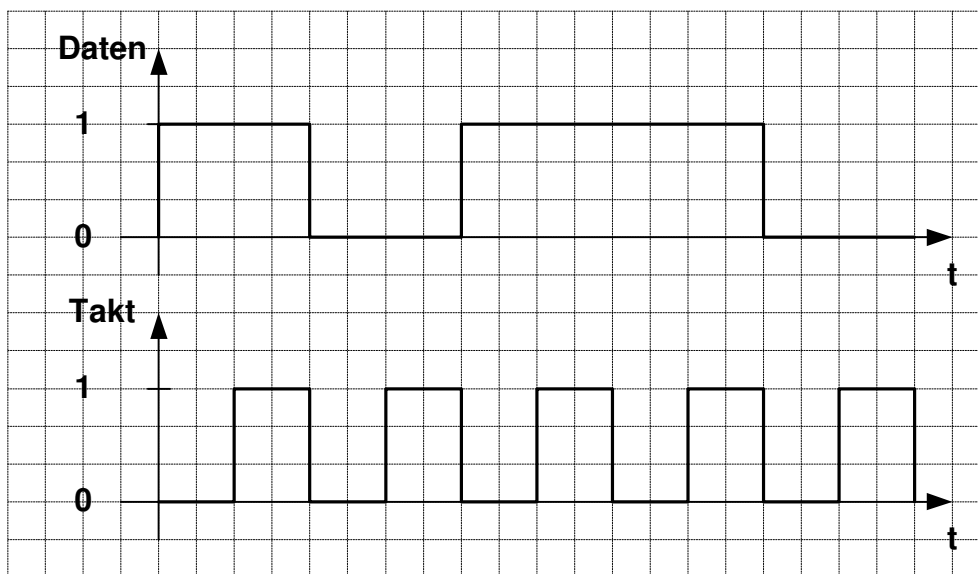
empfangene Daten: 10100111

5.2.2 Synchrone und asynchrone Datenübertragung

Aus dem Versuch wird klar, dass eine Datenübertragung nur erfolgreich sein kann, wenn die Zeitpunkte festliegen an denen der Sender seine Daten senden kann und der Empfänger den Datenstrom auslesen darf. Man sagt auch es muss eine **Synchronisation** zwischen dem Sender und dem Empfänger erfolgen.

Synchrone Datenübertragung:

Die einfachste Art den Sender und den Empfänger zu synchronisieren, ist es mit den Daten auf einer separaten Leitung ein Taktsignal zu übertragen.



Beim Empfänger legen die Flanken des Taktsignals die Zeitpunkte fest, an denen der Datenstrom ausgelesen werden muss. Beim Sender legen die Flanken die Zeitpunkte fest, an denen die einzelnen Bits gesendet werden müssen. Das heißt:

Die synchrone Datenübertragung darf nicht zu einem beliebigen Zeitpunkt beginnen.

Genau dieser Punkt führt aber bei verschiedenen Anwendungen zu Problemen.

So erfolgt zum Beispiel die Tastatureingabe auf einem PC selten genau im Takt (synchron) mit einem Übertragungstakt. Mit der Tastatureingabe fallen aber Daten an, die zu der Hauptplatine übertragen werden müssen.

Man benötigt also eine Übertragungsart die es ermöglicht die Datenübertragung zu einem beliebigen Zeitpunkt zu starten. Man nennt diese Übertragungsart **asynchrone Datenübertragung**.

Asynchrone Datenübertragung:

Die asynchrone Datenübertragung erfolgt fast ausschließlich nach dem Start-Stop-Verfahren.

Die asynchrone Datenübertragung kann zu einem beliebigen Zeitpunkt gestartet werden.

Hierbei haben sowohl der Sender als auch der Empfänger einen eigenen Taktgenerator, die auf dieselbe, vereinbarte Frequenz eingestellt sein müssen. Während den Sendepausen befindet sich die Übertragungsleitung im sogenannten Ruhezustand, das heißt es wird konstant logisch 1 übertragen.

Sollen Daten übertragen werden, wird zunächst ein Startbit (logisch 0) geschickt. Die Flanke zwischen dem Ruhezustand und dem Startbit startet den Taktgenerator beim Empfänger. Für die Dauer der Übertragung der folgenden Bits ist der Sender und der Empfänger synchronisiert.

Dem Startbit folgen die Datenbits. Je nach Voreinstellung können 5, 6, 7 oder 8 Datenbits übertragen werden. Zu beachten ist, dass laut Norm **das niederwertigste Bit immer zuerst übertragen wird**.

Bemerkung: Da meistens mit 7 oder 8 Datenbits gearbeitet wird, spricht man bei einer Serie von Datenbits auch oft von einem "Zeichen".

Den Datenbits kann je nach Einstellung ein Paritätsbit folgen. Das Paritätsbit dient zur groben Fehlererkennung. Man hat die Möglichkeit eine gerade oder eine ungerade Parität einzustellen.

gerade Parität: Das Paritätsbit wird so gesetzt, dass die Summe der Einsen aus Datenbits und Paritätsbit gerade ist.

ungerade Parität: Das Paritätsbit wird so gesetzt, dass die Summe der Einsen aus Datenbits und Paritätsbit ungerade ist.

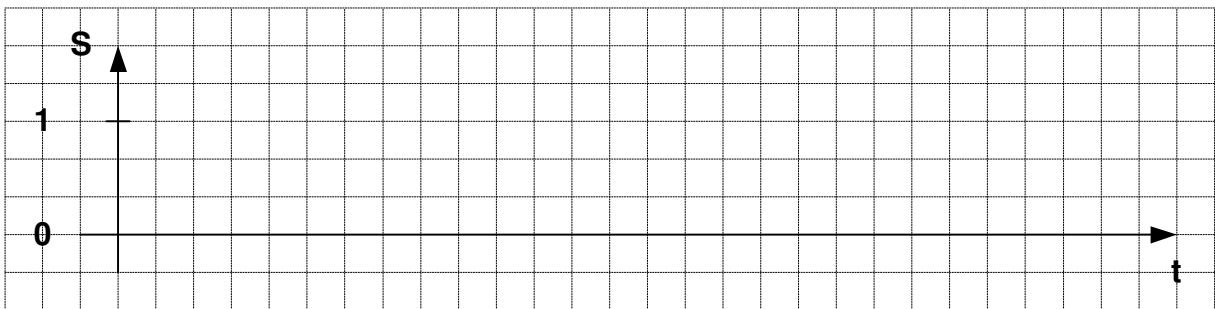
Mit Hilfe des Paritätsbits kann man eine ungerade Anzahl an Fehlern in den Datenbits plus Paritätsbit erkennen.

Abschließend werden je nach Einstellung ein oder zwei Stoppbits (logisch 1) gesendet. Das Stoppbit ist notwendig für den Fall, dass zwischen der Übertragung von zwei Zeichen kein Ruhezustand auftritt. Nur durch mindestens ein Stoppbit ist eine abfallende Flanke zwischen dem Stoppbit und dem nächsten Startbit garantiert.

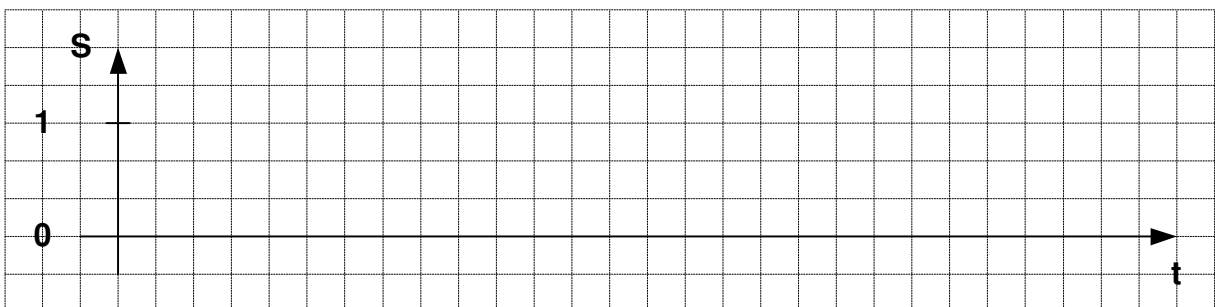
Alle Bits zwischen dem Anfang des Startbits und dem Ende des letzten Stoppbits bezeichnet man als Zeichenrahmen [engl. serial data unit (SDU)]

Aufgaben zum Start-Stop-Verfahren:

1. Zeichne und beschrifte den Zeichenrahmen für die Übertragung des Zeichens "+" mit 7 Datenbits, ungerader Parität und zwei Stoppbit.



2. Gib alle Einstellungen an die sowohl beim Sender als auch beim Empfänger gleich eingestellt werden müssen, damit eine Datenübertragung nach dem Start-Stop-Verfahren stattfinden kann.
3. Zeichne den Zeichenrahmen für die Übertragung des Zeichens "!" mit 8 Datenbits, gerader Parität und einem Stoppbit.



ASCII-Code-Tabelle

| Code | Zeichen | Code | Zeichen | Code | Zeichen | Code | Zeichen |
|----------|------------|----------|-----------|----------|---------|----------|------------|
| 000 0000 | <i>NUL</i> | 010 0000 | <i>SP</i> | 100 0000 | @ | 110 0000 | ` |
| 000 0001 | <i>SOH</i> | 010 0001 | ! | 100 0001 | A | 110 0001 | a |
| 000 0010 | <i>STX</i> | 010 0010 | " | 100 0010 | B | 110 0010 | b |
| 000 0011 | <i>ETX</i> | 010 0011 | # | 100 0011 | C | 110 0011 | c |
| 000 0100 | <i>EOT</i> | 010 0100 | \$ | 100 0100 | D | 110 0100 | d |
| 000 0101 | <i>ENQ</i> | 010 0101 | % | 100 0101 | E | 110 0101 | e |
| 000 0110 | <i>ACK</i> | 010 0110 | & | 100 0110 | F | 110 0110 | f |
| 000 0111 | <i>BEL</i> | 010 0111 | ' | 100 0111 | G | 110 0111 | g |
| 000 1000 | <i>BS</i> | 010 1000 | (| 100 1000 | H | 110 1000 | h |
| 000 1001 | <i>HT</i> | 010 1001 |) | 100 1001 | I | 110 1001 | i |
| 000 1010 | <i>LF</i> | 010 1010 | * | 100 1010 | J | 110 1010 | h |
| 000 1011 | <i>VT</i> | 010 1011 | + | 100 1011 | K | 110 1011 | k |
| 000 1100 | <i>FF</i> | 010 1100 | , | 100 1100 | L | 110 1100 | l |
| 000 1101 | <i>CR</i> | 010 1101 | - | 100 1101 | M | 110 1101 | m |
| 000 1110 | <i>SO</i> | 010 1110 | . | 100 1110 | N | 110 1110 | n |
| 000 1111 | <i>SI</i> | 010 1111 | / | 100 1111 | O | 110 1111 | o |
| 001 0000 | <i>DLE</i> | 011 0000 | 0 | 101 0000 | P | 111 0000 | p |
| 001 0001 | <i>DC1</i> | 011 0001 | 1 | 101 0001 | Q | 111 0001 | q |
| 001 0010 | <i>DC2</i> | 011 0010 | 2 | 101 0010 | R | 111 0010 | r |
| 001 0011 | <i>DC3</i> | 011 0011 | 3 | 101 0011 | S | 111 0011 | s |
| 001 0100 | <i>DC4</i> | 011 0100 | 4 | 101 0100 | T | 111 0100 | t |
| 001 0101 | <i>NAK</i> | 011 0101 | 5 | 101 0101 | U | 111 0101 | u |
| 001 0110 | <i>SYN</i> | 011 0110 | 6 | 101 0110 | V | 111 0110 | v |
| 001 0111 | <i>ETB</i> | 011 0111 | 7 | 101 0111 | W | 111 0111 | w |
| 001 1000 | <i>CAN</i> | 011 1000 | 8 | 101 1000 | X | 111 1000 | x |
| 001 1001 | <i>EM</i> | 011 1001 | 9 | 101 1001 | Y | 111 1001 | y |
| 001 1010 | <i>SUB</i> | 011 1010 | : | 101 1010 | Z | 111 1010 | z |
| 001 1011 | <i>ESC</i> | 011 1011 | ; | 101 1011 | [| 111 1011 | { |
| 001 1100 | <i>FS</i> | 011 1100 | < | 101 1100 | \ | 111 1100 | |
| 001 1101 | <i>GS</i> | 011 1101 | = | 101 1101 |] | 111 1101 | } |
| 001 1110 | <i>RS</i> | 011 1110 | > | 101 1110 | ^ | 111 1110 | ~ |
| 001 1111 | <i>US</i> | 011 1111 | ? | 101 1111 | _ | 111 1111 | <i>DEL</i> |

5.3 V.24 / RS-232-C – Schnittstelle

Zwei europäische Normen definieren eine serielle Schnittstelle.

- Die CCITT V.24 beschreibt die Funktion von verschiedenen Signalen.
- Die CCITT V.28 legt die Signalpegel fest.

Vereinfachend spricht man oft nur von der V.24-Norm.

Die amerikanische Norm EIA-232-D (Nachfolger der RS-232-C) geht über die europäischen Normen hinaus und enthält zusätzlich:

- Spezifizierung eines SUB-D Trapezsteckverbinders
- Zuordnung der Signale zu den Kontaktstiften

Die amerikanische Norm ist die weltweit am meisten benutzte Norm.

5.3.1 Schnittstellensignale

Die wichtigsten der in den Normen festgelegten Schnittstellensignale sind in folgender Tabelle zusammengefasst:

| Signal-kürzel | Signalname | Beschreibung | Art | Pin auf SUB-D 25 | Pin auf SUB-D 9 |
|---------------|---------------------|------------------------------|-------------|------------------|-----------------|
| TxD | Transmit Data | Sendeleitung | Datenleit. | 2 | 3 |
| RxD | Receive Data | Empfangsleitung | Datenleit. | 3 | 2 |
| RTS | Request To Send | Ausgabe Sendeanfrage | Steuerleit. | 4 | 7 |
| CTS | Clear To Send | Empfang Sendefreigabe | Meldeleit. | 5 | 8 |
| DSR | Data Set Ready | Empfang Betriebsbereitschaft | Meldeleit. | 6 | 6 |
| GND | Signal Ground | Betriebserde | Erdleitung | 7 | 5 |
| DTR | Data Terminal Ready | Ausgabe Betriebsbereitschaft | Steuerleit. | 20 | 4 |

Steuerleitungen bezeichnen Leitungen über die Signale zu der Gegenstelle geschickt werden um den Datenfluss zu steuern.

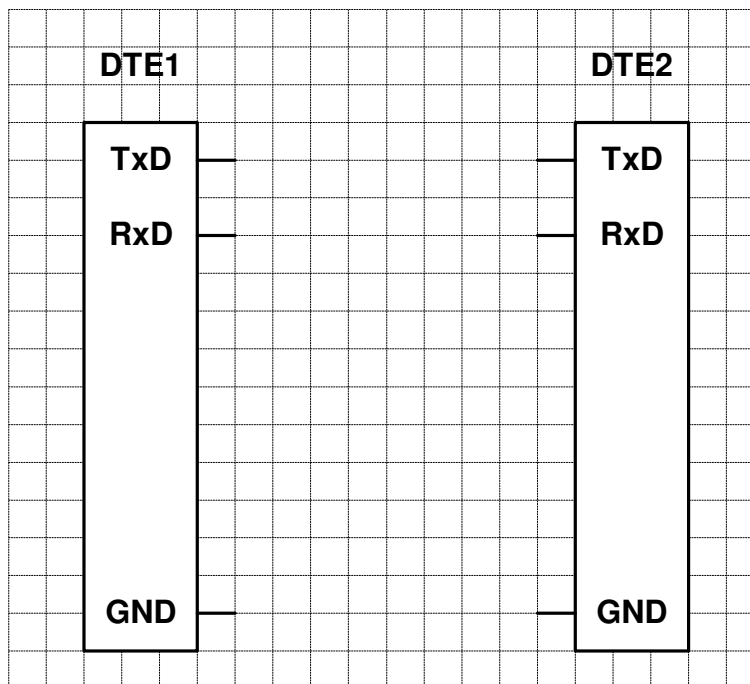
Meldeleitungen bezeichnen Leitungen auf denen Signale von der Gegenstelle empfangen werden um den Datenfluss zu steuern.

5.3.2 Die DTE-DTE Verbindungen (Null-Modem-Verbindungen)

Es gibt mehrere Möglichkeiten zwei Datenendeinrichtungen (DTE) miteinander über die serielle Schnittstelle zu verbinden.

Die einfachste Variante besteht aus einer Verbindung zwischen der Sendeleitung des 1. Gerätes und der Empfangsleitung des 2. Gerätes (und umgekehrt). Die TxD- und die RxD-Leitung sind also zu kreuzen. Für die Stromrückführung ist außerdem eine Verbindung der Signalmassen nötig.

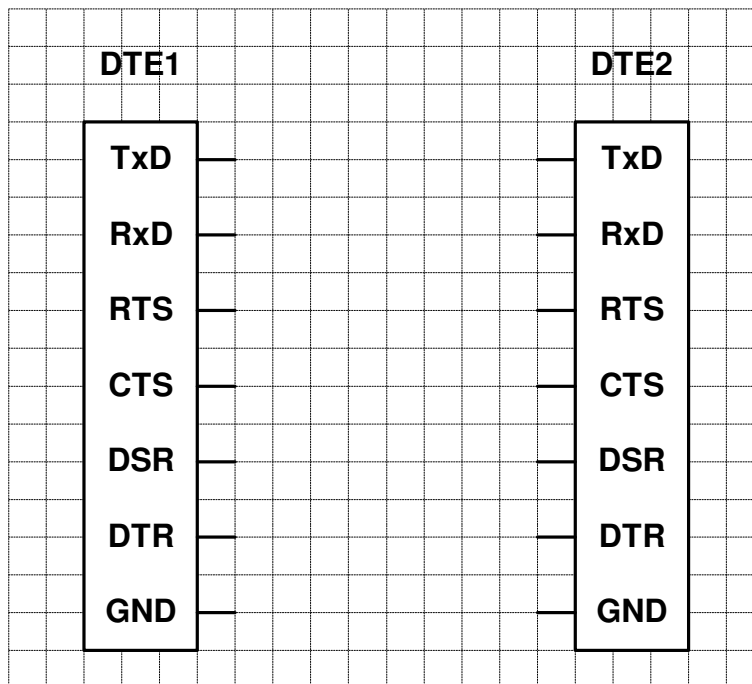
Minimalausführung der Null-Modem-Verbindung



Um die Datenübertragung zwischendurch stoppen zu können, kann eine sogenannte Datenflusskontrolle (engl.: handshaking oder flow control) eingeführt werden. Diese erfolgt dann mit den über Kreuz verbundenen RTS- und CTS-Leitungen. Der Sender fragt mit einem Request To Send (RTS) die Erlaubnis an Daten senden zu dürfen. Über die CTS-Leitung erhält er vom Empfänger die Sendefreigabe.

Eine weitere Erhöhung der Betriebssicherheit ergibt sich durch die Verwendung der Leitungen DTR und DSR. Über diese Leitungen signalisieren sich die Datenendeinrichtungen gegenseitig ihre Betriebsbereitschaft.

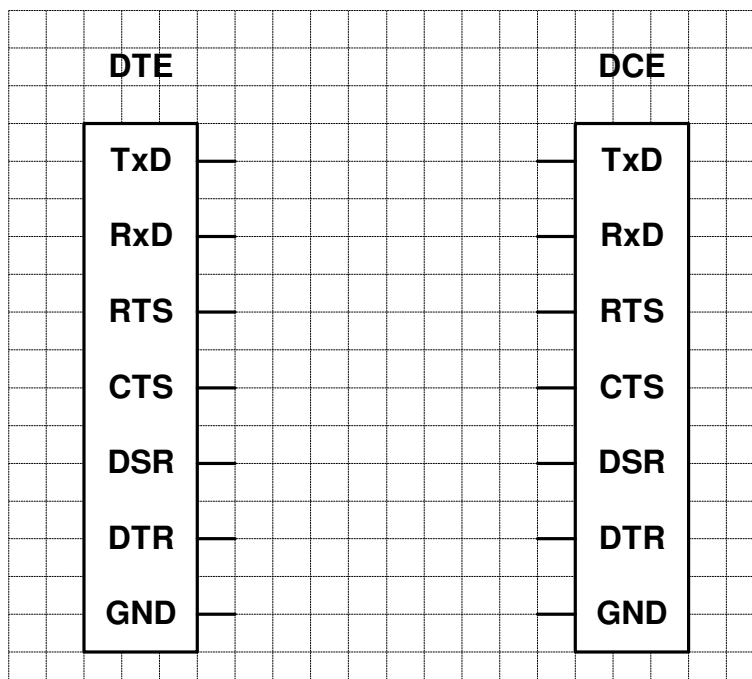
Null-Modem-Verbindung mit Handshake und Betriebsbereitschaftsmeldung



5.3.3 Die DTE-DCE Verbindung

Soll eine Datenendeinrichtung (DTE) mit einer Datenübertragungseinrichtung (DCE), z.B. einem Modem, verbunden werden so werden die Leitungen entsprechend der Normen nicht mehr gekreuzt.

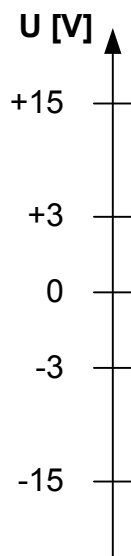
DTE-DCE-Verbindung



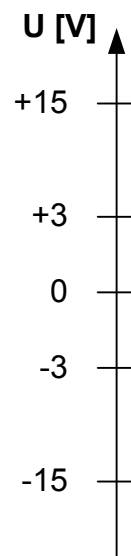
5.4 Spannungspegel an der serielle Schnittstelle

Um den Einfluss von Störungen zu verringern und problemlos Übertragungsstrecken bis etwa 20m zu ermöglichen, haben die seriellen Schnittstellensignale nicht die in der Digitaltechnik üblichen TTL-Spannungspegel. Die V.28-Norm definiert die Spannungspegel für die serielle Schnittstelle. Dabei gelten unterschiedliche Pegel für die Datenleitungen und die Steuer-/Meldeleitungen.

Spannungspegel Datenleitungen



Spannungspegel Steuer- / Meldeleitungen

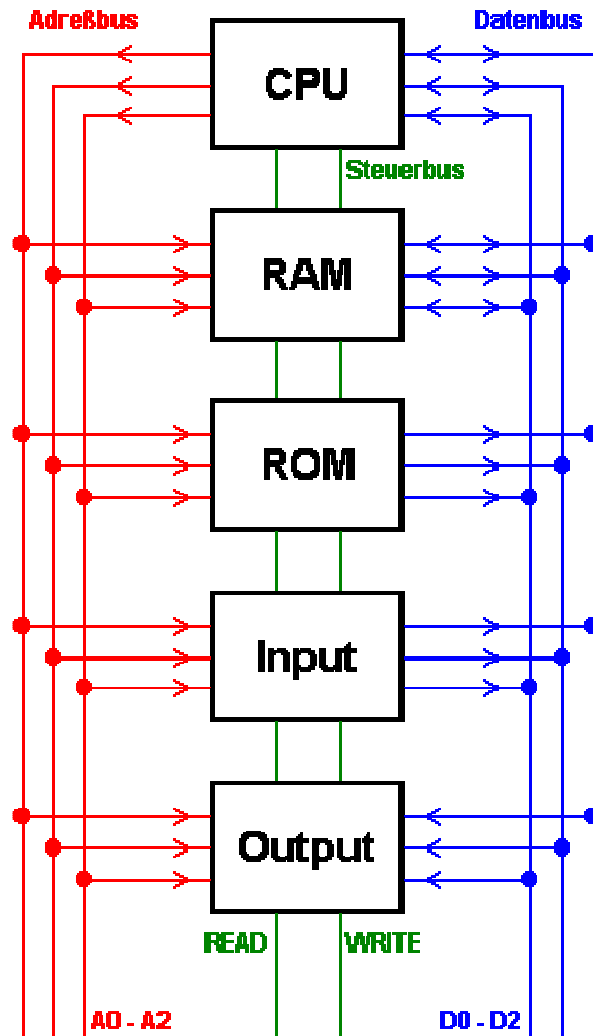


5.5 Serieller Schnittstellenbaustein 8250A

5.5.1 Systembus eines PCs

(Quelle: <http://www.elektronik-kompodium.de/sites/com/0309221.htm>)

Der Systembus eines Mikroprozessorsystems teilt sich in Datenbus, Adressbus und Steuerbus auf.



Datenbus:

Zur Übertragung von Daten zwischen Prozessor, Arbeitsspeicher und Peripherie steht der Datenbus zu Verfügung. Die Anzahl der Datenbusleitungen bestimmt die Anzahl der Byte, die pro Takt übertragen werden können.

Alle Baugruppen, die Daten an den Datenbus abgeben können (Prozessor, Speicher, I/O) sind mit dem Datenbus verbunden. Das Ziel der Daten wird über die Adresse auf dem Adressbus vorgegeben.

Adressbus:

Die meisten Komponenten eines PC enthalten Speicherzellen. Jede Speicherzelle muss eine eindeutige Adresse haben. Über den Adressbus gibt die CPU die Ziel-Adresse der auf dem Datenbus befindlichen Daten aus.

Die Anzahl der Adressleitungen bestimmt die maximale Anzahl an zu adressierbaren Speicherzellen.

$$\text{maximale Anzahl der adressierbaren Speicherzellen} = 2^{\text{Anzahl der Adressleitungen}}$$

Aufgabe zum Adressbus:

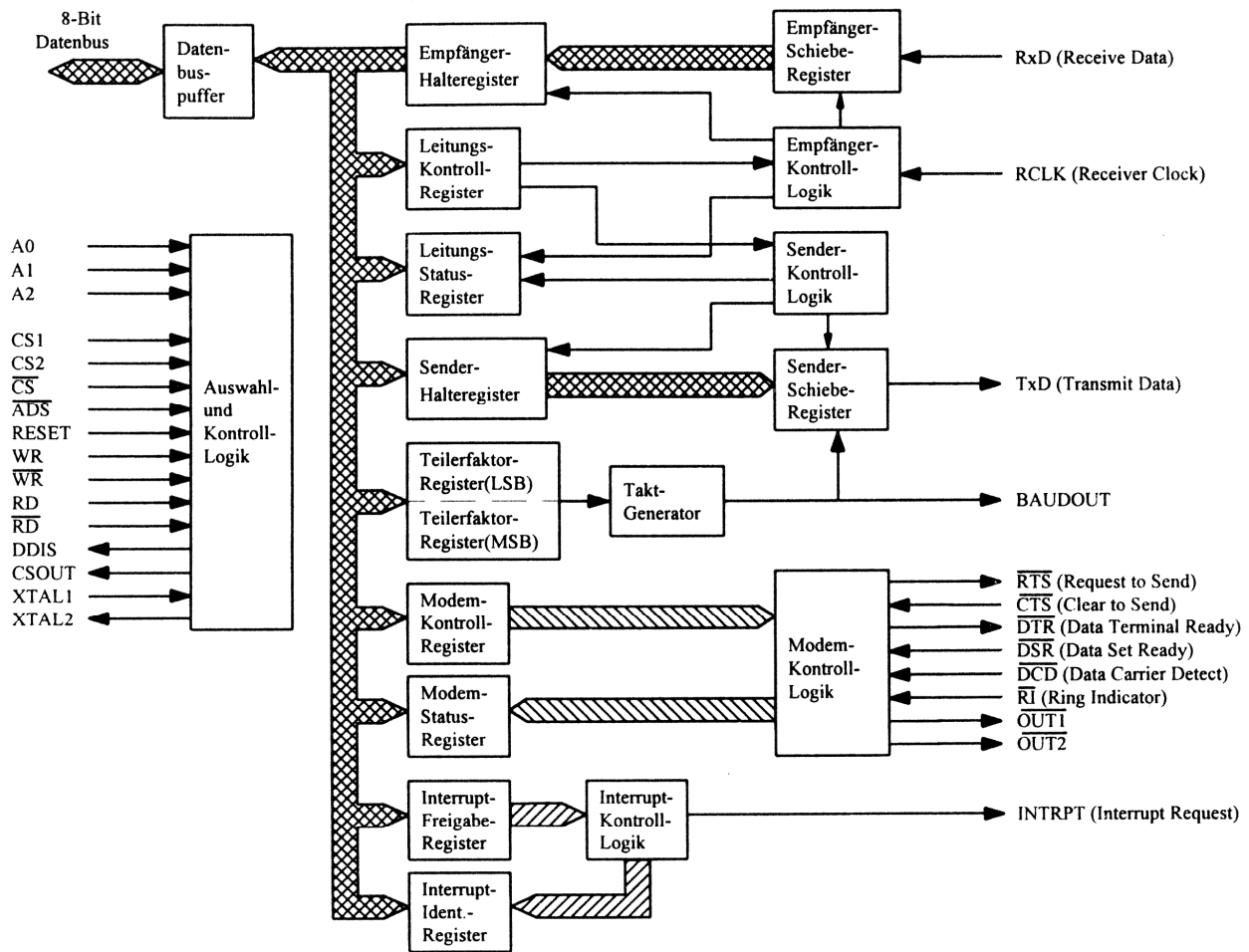
Der Intel Itanium Prozessor hat einen 44-Bit Adressbus. Wie viel Giga-Byte Speicher sind somit adressierbar, wenn jede Speicherzelle ein Byte speichert?

Steuerbus:

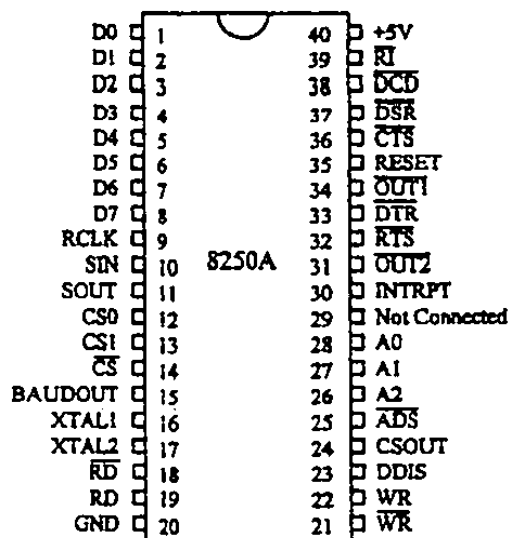
Der Steuerbus dient der Übertragung von Befehlen an die einzelnen Komponenten. Auf diese Weise wird den Komponenten mitgeteilt, was sie zu tun haben z. B. die Daten vom Datenbus aufnehmen oder Daten auf den Datenbus bereitstellen.

5.5.2 Anschluss des 8250A am Systembus

Blockschaltbild des 8250A:

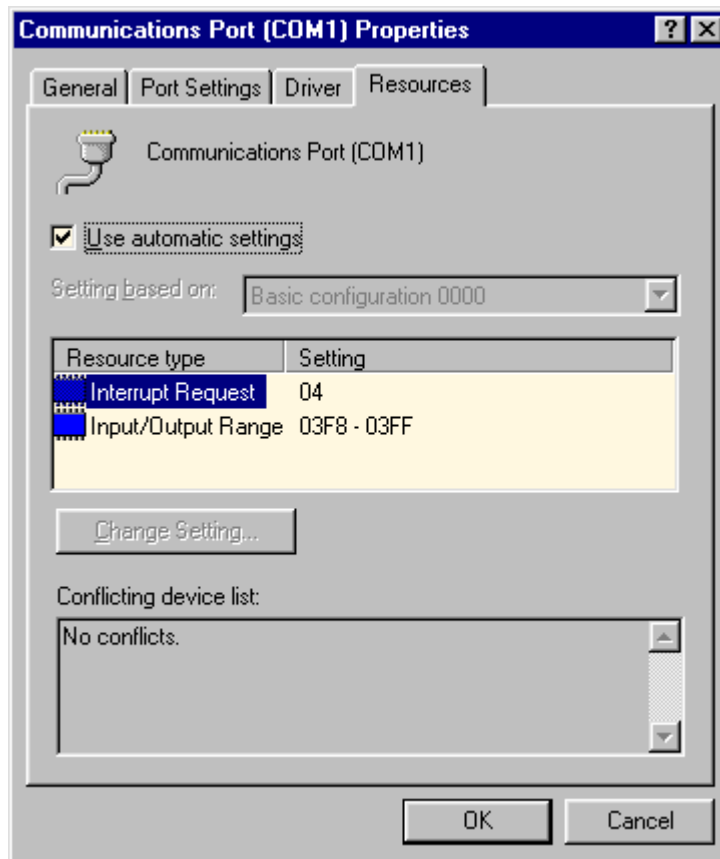


Pinbelegung des 8250A:



In dem 8250A befinden sich eine Reihe von Registern (=8-Bit Speicherzellen) die über die Datenleitungen D0 bis D7 ausgelesen und/oder beschrieben werden können.

Jedem Register ist eine Adresse zugewiesen, die z.B. im Gerätemanager von Windows nachgeschlagen werden können (Start/Systemsteuerung/Gerätemanager/Ports/COM1).



Mit den drei Adressleitungen A0 bis A2 ist der 8250A an die niederwertigsten Leitungen des Adressbus angeschlossen. Über den \overline{CS} -Eingang kann der IC aktiviert werden. CS steht für Chip Select.

Info:

Der Strich über dem CS will sagen, dass auf diesem Eingang mit invertierter Logik gearbeitet wird. Das heißt um den IC zu aktivieren muss logische Null angelegt werden und logisch Eins um es zu deaktivieren.

Die Eingänge \overline{RD} und \overline{WR} werden an den Steuerbus angeschlossen. Über sie kann bestimmt werden, ob die Register beschrieben oder ausgelesen werden sollen.

Alle weiteren PC-seitigen Anschlüsse werden entweder nicht verwendet oder sind für das Verständnis der Funktionsweise des 8250A nicht relevant.

5.5.3 Adressierung des 8250A in Turbo Pascal

Das Speichern und Auslesen von Daten in resp. aus Registern aller Art ist in Turbo Pascal sehr einfach. Mit folgendem Befehl wird zum Beispiel das Zeichen 'a' (hexadezimal: 61) in das Register mit der Adresse 03F8 gespeichert.

```
port[$03F8] := $61;
```

Um das Register mit der Adresse \$03FE auszulesen gibt man folgenden Befehl ein:

```
VARIABLE := port[$03FE];
```

Aufgaben zum Anschluss und Adressierung des 8250A:

1. Wandle alle Hexadezimal-Adressen der COM1-Schnittstelle in binäre Adressen um.
2. Gib die logischen Zustände der Eingänge D0 bis D7, A0 bis A2, \overline{CS} , \overline{WR} und \overline{RD} nach der Ausführung des folgenden Befehls an.

```
port[$03F8] := $61;
```

3. Die Adresse \$3E8 bewirkt genau wie die Adresse \$3F8 dass drei Nullen an den Eingängen A0 bis A2 anliegen. Trotzdem darf der 8250A-Baustein der COM1-Schnittstelle nicht auf diese Adresse reagieren. Wie verhindert der Computer, dass der 8250A der COM1-Schnittstelle nach folgendem Befehl das Zeichen 'a' in sein erstes Register schreibt?

```
port[$03E8] := $61;
```

5.5.4 Chip-Select-Schaltung

In der vorherigen Aufgabe wurde festgestellt, dass der 8250A der COM1 Schnittstelle über den Chip-Select-Eingang \overline{CS} deaktiviert werden muss, wenn die Adresse auf dem Adressbus außerhalb des Bereichs \$03F8 bis \$03FF liegt. Die Schaltung die aufgrund der Zustände der Adressleitungen A_3 bis A_{15} entscheidet ob der Chip aktiviert wird oder nicht nennt man Chip-Select-Schaltung.

Aufgabe:

Entwirf fachgerecht die Chip-Select-Schaltung durch Schaltungssynthese unter Verwendung handelsüblicher Gatter.

5.5.5 Programmierung des 8250A

(Quelle: Mikrorechnerpraktikum 1, Institut für Anwendungen der Informatik, Universität Karlsruhe)

Nachdem der Anschluss des Bausteins an das Rechnersystem erläutert wurde, folgt nun die Beschreibung der Register und eine Einführung in die Programmierung und Nutzung des 8250A.

Der Baustein belegt acht Adressen, über die 11 Register ausgewählt werden können. Dies ist möglich, da sich zum einen das Sende- und das Empfängerhaltereregister eine Adresse teilen - die Unterscheidung erfolgt anhand des Zugriffs (lesend oder schreibend). Zum anderen bestimmt das Bit 7 im Leitungskontrollregister welche Register an den beiden niedrigsten Adressen (A1 und A2 Low-Pegel) angesprochen werden. Ist das Bit 0, erreicht man die Halteregister und das Interrupt-Freigabe-Register, ist das Bit 1, erreicht man die Teilerfaktor-Register.

Hier nun die Beschreibung der einzelnen Register in aufsteigender Reihenfolge. Die Basisadresse der COM1-Schnittstelle ist normalerweise \$03F8.

Adresse: Basisadresse + 0

Abhängig vom Bit 7 im Leitungskontrollregister (Basisadresse + 3), befindet sich hier entweder die niederwertige Hälfte des schreib- und lesbaren Teilerfaktorregisters bzw. bei Schreibzugriffen das Sendehaltereregister, oder bei Lesezugriffen das Empfangshaltereregister. Die Halteregister speichern das zu sendende oder zuletzt empfangene Zeichen. Zu beachten ist, dass ein Datenbyte im Empfangshaltereregister überschrieben wird, sobald ein neues Zeichen komplett empfangen wurde.

Adresse: Basisadresse + 1

Auch hier entscheidet Bit 7 im Leitungskontrollregister über das ansprechbare Register. Ist das Bit eins, erreicht man die höherwertige Hälfte des schreib-/lesbaren Teilerfaktorregisters. Ansonsten das ebenfalls schreib-/lesbare Interrupt-Freigabe-Register. Die einzelnen Bits in diesem Register haben folgende Bedeutung:

(Ein gesetztes Bit bewirkt die genannte Funktion)

- Bit 0: Interrupt, sobald Zeichen empfangen wurde (ID 1).
- Bit 1: Interrupt, sobald Sendehalteregister frei, d.h. Zeichen gesendet (ID 2).
- Bit 2: Interrupt bei fehlerhaftem Empfang (ID 0).
- Bit 3: Interrupt, wenn sich der Pegel auf eine der Steuerleitungen ändert (ID 3).
- Bit 4-7: nicht benutzt, beim Lesen Null.

Die in Klammern angegebene Nummer ist die Identifizierungsnummer, welche nach einem Interrupt im Interrupt-ID-Register angegeben wird.

Adresse: Basisadresse + 2

Das nur lesbare Interrupt-Identifizierungs-Register.

Die einzelnen Bits haben folgende Bedeutung:

- Bit 0: Dieses Bit ist null, wenn ein Interrupt ausgelöst wurde. Durch Auslesen dieses Register wird das Bit wieder gesetzt.
- Bit 1 und 2: Hier befindet sich die binär codierte Identifizierungsnummer des Interrupts.
- Bit 3 bis 7: Unbenutzt (null).

Adresse: Basisadresse + 3

Das schreib-/und lesbare Leitungs-Kontroll-Register.

Bit 0 und 1: Anzahl der Datenbits (siehe Kapitel 5.2.2). Es gilt folgende Zuordnung:

| Bit 1 | Bit 0 | Datenbits |
|-------|-------|-----------|
| 0 | 0 | 5 Bit |
| 0 | 1 | 6 Bit |
| 1 | 0 | 7 Bit |
| 1 | 1 | 8 Bit |

Bit 2: Anzahl der Stoppbits. Gelöschtes Bit bedeutet ein Stoppbit, gesetztes Bit bedeutet 2 Stoppbits.

Bit 3: Ein Setzen dieses Bits schaltet die Paritätserzeugung und Prüfung ein.

Bit 4: Ist Bit 3 gesetzt, bestimmt dieses Bit die Art der Parität. Gesetztes Bit bedeutet gerade Parität, gelöschtes Bit ungerade Parität.

Bit 5 und 6: Diese Bits müssen immer gelöscht sein.

Bit 7 (DLAB): Dieses Bit entscheidet, welche Register an den Adressen Basisadresse+0 und Basisadresse+1 angesprochen werden. Bei gesetztem Bit erreicht man die Teilerfaktorregister.

Adresse: Basisadresse + 4

Das schreib-/lesbare Modem-Kontroll-Register. Dieses Register bestimmt den Pegel der Ausgänge OUT1 und OUT2 sowie der Steuerleitungen DTR und RTS. Es gilt folgende Zuordnung:

Bit 0: DTR

Bit 1: RTS

Bit 2: OUT1

Bit 3: OUT2

Bit 4-7: Unbenutzt, müssen immer gelöscht sein.

Adresse: Basisadresse + 5

Das nur lesbare Leitungs-Status-Register. In diesem Register wird der aktuelle Status des 8250A gespeichert.

Bit 0: Data Ready

Ein gesetztes Bit zeigt an, dass ein Datenwort im Empfangshalteregister steht.

Bit 1: Overrun Error

Ein gesetztes Bit, zeigt an, dass ein nicht abgeholtes Datenwort im Empfangshalteregister durch ein Neues überschrieben wurde.

Bit 2: Parity Error

Ist dieses Bit gesetzt, wurde beim Empfang des letzten Datenworts eine falsche Parität festgestellt. Dieser Fehler kann nur auftreten, wenn die Paritätsprüfung eingeschaltet ist.

Bit 3: Framing Error

Ein gesetztes Bit zeigt an, dass ein Datenwort mit falscher Wortlänge empfangen wurde.

Bit 4: Break Interrupt

Dieses Bit ist gesetzt, wenn das BREAK-Signal empfangen wurde. Dies ist der Fall, wenn die RxD-Leitung länger als für die Übertragung eines Datenwortes nötig, aktiv ist (logisch null).

Bit 5: Transmitter Holding Register Empty

Ein gesetztes Bit zeigt an, dass das Sendehalteregister leer ist, das zuletzt geschriebene Datenwort also gesendet wird.

Bit 6: Transmitter Empty

Diese Bit zeigt zusätzlich zu Bit 5 an, dass sowohl das Sendehalteregister als auch das Sende-Schieberegister leer sind, ein Datenwort also komplett gesendet wurde.

Bit 7: Nicht benutzt, Zustand undefiniert.

Adresse: Basisadresse + 6

Das nur lesbare Modem-Status-Register. In diesem Register wird der Zustand der Steuerleitung CTS, DSR, DCD und RI vermerkt. Außerdem wird auch noch jede Änderung dieser Signale seit dem letzten Auslesen des Registers vermerkt.

Bit 0: Delta Clear to Send

Dieses Bit ist gesetzt, wenn sich seit dem letzten Lesen der Pegel der Leitung CTS geändert hat.

Bit 1: Delta Data Set Ready

Wie Bit 0 nur für die Leitung DSR.

Bit 2: Trailing Edge Ring Indicator

Dieses Bit wird durch eine ansteigende Flanke auf der RI-Leitung gesetzt.

Bit 3: Delta Data Carrier Detect

Wie Bit 0 und 1, nur für die Leitung DCD.

Bit 4: Clear to Send

Zeigt den aktuellen Zustand der Leitung CTS an.

Bit 5: Data Set Ready

Wie Bit 4, nur Leitung DSR.

Bit 6: Ring Indicator

Wie Bit 5, nur Leitung RI.

Bit 7: Data Carrier Detect

Wie Bit 6, nur Leitung DCD.

Adresse: Basisadresse + 7

Dieses Register hat auf die Funktion des 8250A keine Auswirkung. Es ist ein frei verwendbares 8-Bit Schreib-/Leseregister.

Somit sind alle Register beschrieben worden. Nun noch einige Hinweise zu Programmierung und Nutzung des Bausteins.

Programmierung und Nutzung des Bausteins:

Für die Datenübertragung muss der Baustein zuerst richtig programmiert werden. Dazu trägt man in das Leitungs-Kontroll-Register das Datenübertragungsformat ein. Angenommen die Übertragung erfolgt mit 8-Bit Wörtern, ohne Paritätsprüfung und einem Stoppbit, so muss der Wert 00000011b, bzw. 93h eingetragen werden. Danach muss das Teilerfaktorregister in Abhängigkeit von der Übertragungsgeschwindigkeit (Baudrate) geladen werden. Der Faktor errechnet sich bei Verwendung eines 1,8432 MHz Quarzes zu:

$$\text{Faktor} = 115200 / \text{Baudrate}$$

Für eine Baudrate von 9600 Baud/s ergibt sich z.B. der Wert 12 bzw. 000Ch.

Werden die Steuerleitungen benutzt, muss man zuletzt noch für den richtigen Pegel auf den Ausgängen DTR und RTS sorgen. Danach kann mit der Datenübertragung begonnen werden.

Am einfachsten gestaltet sich der Vorgang der Datenübertragung, wenn auf die Steuerleitungen verzichtet werden kann. Um ein Datenwort auszugeben, wird zuerst anhand Bit 5 und 6 des Leitungs-Status-Register geprüft, ob der Sender frei ist und dann das Datenwort in das Sendehalterregister geschrieben. Ob das Datenwort beim Empfänger angekommen ist kann man allerdings nicht feststellen.

Sollen Zeichen empfangen werden, kann man entweder ständig das Bit 0 im Leitungs-Statusregister abfragen und bei gesetztem Bit das Datenwort aus dem Halterregister lesen, oder die Interruptfähigkeiten des Bausteins nutzen.

Zuletzt sei noch angemerkt, dass nach jeder Programmierung des 8250A-Bausteins das Empfangshalterregister sicherheitshalber gelesen werden sollte, um eventuell enthaltene alte Daten zu löschen.

Aufgaben zur Programmierung des 8250A:

1. Vervollständige folgende Tabelle mit allen Registern, deren Adressen und dem Wert des DLAB.

| Registernamen (deutsch) | Adresse | DLAB | lesen schreiben |
|--------------------------------|----------------|-------------|----------------------------|
|--------------------------------|----------------|-------------|----------------------------|

| | | | |
|--------------------------|------------------|---|-----------|
| Empfängerhaltereregister | Basisadresse + 0 | 0 | nur lesen |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

2. Gib den Turbo-Pascal-Befehl an mit dem man das DLAB-Bit gezielt auf 0 oder 1 setzen kann, ohne dabei die restlichen Bits zu verändern? Es gibt nur den port-Befehl zum Verändern von Registern.
3. Wie lauten in Turbo Pascal die Bedingungen, um zu überprüfen ob das Sendehaltereregister leer ist?