

Gegenstand	Klasse	Datum	Serielle Schnittstelle	Seite
LAB4			Grundlagen	1 von 3

SERIELLE SCHNITTSTELLE

ALLGEMEINES

Bei der seriellen Schnittstelle werden die Datenbytes "hintereinander" digital über **eine** Leitung übertragen.

Anders als beim PC mit seinen genormten Steckern (9 pol. bzw. 25 pol.) gibt es in der Mikrocontrollertechnik keine Norm für die Anschlußbelegung. Außerdem wird bei den Mikrocontrollern meist nur das Softwarehandshake unterstützt, das bedeutet für die Datenübertragung werden nur 3 Leitungen (TX: Transmitt, Sendeleitung; RX: Receive, Empfangsleitung; GND: Ground, Masseverbindung) verwendet.

TECHNISCHE DATEN

Tabelle für die Standard Serielle Schnittstelle:

Bezeichnung:	COM1: bis COM4: oder RS 232C (RS = recommended standard)
Datenübertragung:	seriell (Bit-seriell, Byte-seriell)
Datenrichtung:	bidirektional (auch unidirektional möglich)
Spannungen:	+3 V...+20 V / -3 V...-20 V (bei PC's auch +5 V/0 V)
Logik:	Steuerleitung: pos. Logik ("1" = +12 V, "0" = -12 V)
Datenleitung:	neg. Logik ("1" = -12 V, "0" = +12 V), Ruhezustand "1", Startbit "0", Stopbit "1"
Anzahl Leitungen:	min. 2, max. 9
Leitungslängen:	bis 30 m, je nach Versorgungsspannung und Übertragungsrate auch mehr
Anschluß:	9- oder 25-poliger Sub-D-Stecker (am Rechner Stecker = männlich)
Übertragungsgeschwindigkeit = Baudrate:	75, 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 115200 Baud
Anzahl Datenbits:	1 Startbit ("0" = -12 V), 5 bis 8 Datenbits (Wertigkeit: 1-2-4-8-16-32-64-128), 0 oder 1 Kontrollbit (Paritybit), 1, 1,5 oder 2 Stoppbits ("1" = -12 V)
Vorteil:	einfach, bidirektional, wenig Leitungen, störsicher, lange Leitungen, viele Varianten möglich
Nachteil:	langsam, Übertragungsprotokoll muß festgelegt werden (und stimmen!)
Verwendung:	Verbindung Terminal mit Modem, Terminal mit Computer; beim PC Anschluß von Maus, Plotter, Meßgeräte usw.
Besonderheit:	Vor Beginn der Datenübertragung müssen erst die Übertragungsparameter festgelegt werden !

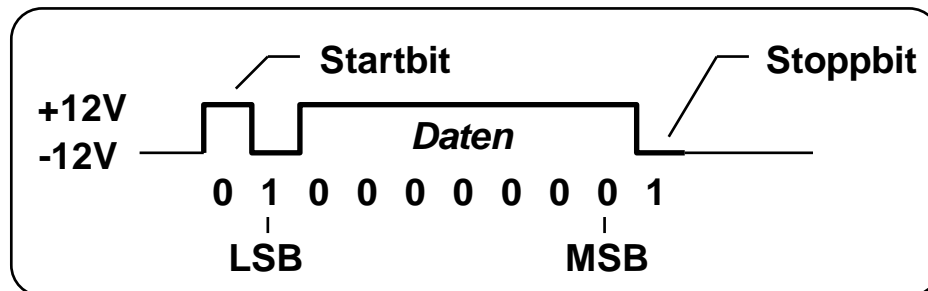
Bei den Mikrocontollern sind die Spannungen 0V und 5V definiert, weiters existieren die Leitungen für die Steuerlogik meist nicht und die Anschlüsse am Mikrocontroller sind auch nicht definiert.

Für die Umsetzung der Spannungen von 0/5V auf +12/-12V ist ein sogenannter serieller Leitungstreiber (z.B.: MAX232) notwendig.

Gegenstand	Klasse	Datum	Serielle Schnittstelle	Seite
LAB4			Grundlagen	2 von 3

ÜBERTRAGUNG EINES BYTES

Die RS-232 ist eine asynchrone, serielle Schnittstelle. Aus diesem Grund müssen der Beginn einer Übertragung, die Übertragungsgeschwindigkeit und das Ende einer Übertragung festgelegt werden.



Im Ruhezustand liegt die Datenleitung (TxD) auf -12 V (= logisch 1!). Die Übertragung wird durch ein

- „**Startbit**“ (= logisch 0) eingeleitet, danach folgen die
- **Datenbits**, beginnend mit dem niederwertigsten Bit (LSB), schließlich beendet ein
- „**Stopbit**“ die Übertragung. (= logisch 1, dh die Leitung kehrt in den Ruhezustand zurück)

FESTLEGEN DER ÜBERTRAGUNGSPARAMETER

Bevor eine Datenübertragung über die RS-232 möglich ist, müssen fünf Parameter festgelegt werden:

1. Nummer der Schnittstelle
2. Zeit pro Bit (= „Übertragungsgeschwindigkeit“)
3. Prüfbit (Parity-Bit)
4. Zeichenlänge
5. Anzahl der Stopbits

Je nach Betriebssystem erfolgt die Eingabe dieser Werte auf unterschiedliche Weise (MS-DOS: MODE-Befehl, Windows: Systemsteuerung,)

Details zu den Übertragungsparametern

1. Nummer der Schnittstelle: COM1: bis COM4:
2. Möglichkeiten für die Übertragungsgeschwindigkeit: 115200, 57600, 38400, 28800, 19200, 9600, 4800, 2400, 1200, 600, 300, ...
3. Prüfzeichen = Parity-Bit: **N** (No = keine), **E** (even = gerade), **O** (odd = ungerade)
M (Mark = immer 1) und S (Space = immer 0)
Beispiel: E = Ergänzung auf gerades Paritybit:
Summe aller "1" incl. Prüfbit ist gerade z.B. 0011 1001 ⇒ Prüfbit 0
4. Zeichenlänge: 7 oder 8 Bits (manchmal auch 5 oder 6)
5. Stopbits: Anzahl 1 oder 1,5 oder 2

Gegenstand	Klasse	Datum	Serielle Schnittstelle	Seite
LAB4			Grundlagen	3 von 3

BERECHNUNG DER ÜBERTRAGUNGSGESCHWINDIGKEIT

Je nach eingestelltem „Overhead“ (Prüfbit ja/nein, Anzahl Datenbits 5/6/7/8, Stopbits 1/1,5/2) ergibt sich auch bei gleicher Baudrate ein unterschiedlicher „Netto-Durchsatz“. Die Anzahl der Bytes, die pro Sekunde übertragen werden können, ergibt sich aus folgender Rechnung:

Zeit pro Bit: $t_{\text{bit}} = 1/b$
 Zeit pro Byte: $t_{\text{Byte}} = (z + p + s + 1) \cdot t_{\text{bit}}$
 Anzahl Byte/Sek: $1/t_{\text{Byte}}$

b.....Baudrate (in Baud, zB 9600 Bd)

z.....Anzahl Bits pro Zeichen (meist 7 Bit oder 8 Bit)

p.....Paritybit (0, gerade, ungerade. Gerade bedeutet, daß das Paritybit gleich 1 ist, wenn die Summe ALLER übertragenen Bits gerade ist)

s.....Anzahl Stopbits (1 oder 2)

Beispiel: Berechnung der Übertragungsgeschwindigkeit

Wieviele Bytes pro Sekunde können mit den Übertragungsparametern 9600,e,7,1 übertragen werden?

Lösung:

Zeit pro Bit: $t_{\text{bit}} = 1/b = 1/9600 = 104,166 \mu\text{s}$

Zeit pro Byte: $t_{\text{Byte}} = (z + p + s + 1) \cdot t_{\text{bit}} = (7 + 1 + 1 + 1) \cdot 104,166 \mu\text{s} = 1,0417 \text{ ms}$

Anzahl Byte/Sek: $1/t_{\text{Byte}} = 1/1,0417 \text{ m} = 960 \text{ Byte/sec}$

HANDSHAKE

Handshake: Sender und Empfänger zeigen an, daß sie zum Empfang von Daten bereit sind. Dabei unterscheidet man zwischen Hardware- und Software-Handshake.

Hardware-Handshake

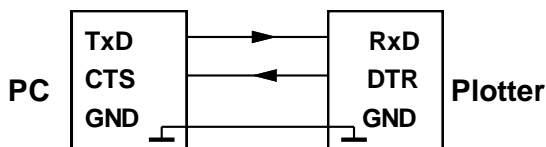


Bild 1: Hardware-Handshake

Der Handshake wird auf dafür vorgesehenen Steuerleitungen durchgeführt. Leider ist die Verwendung der Leitungen nicht einheitlich.

Andere Kombination von Handshake-Leitungen:

RTS - CTS oder DTR - DTS oder ...

Beispiel: Datentransfer vom PC zum Plotter.

Software-Handshake

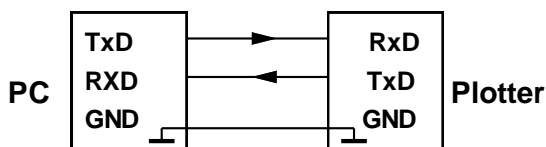


Bild 2: Software-Handshake

Der Empfänger schickt dem Datensender über die Datenleitung bestimmte Steuerzeichen, die angeben, ob er zum Empfang bereit ist:

- X-ON = 11h: es dürfen Zeichen gesendet werden,

- X-OFF = 13h: interner Zeichenpuffer ist voll.

Vorteil: kein Problem mit den vielen Varianten an Handshake-Leitungskombinationen.

Nachteil: Es muß ständig geprüft werden, ob ein Steuerzeichen gesendet wurde.

Bei den Mikrocontrollern wird meist das Softwarehandshake verwendet!!!