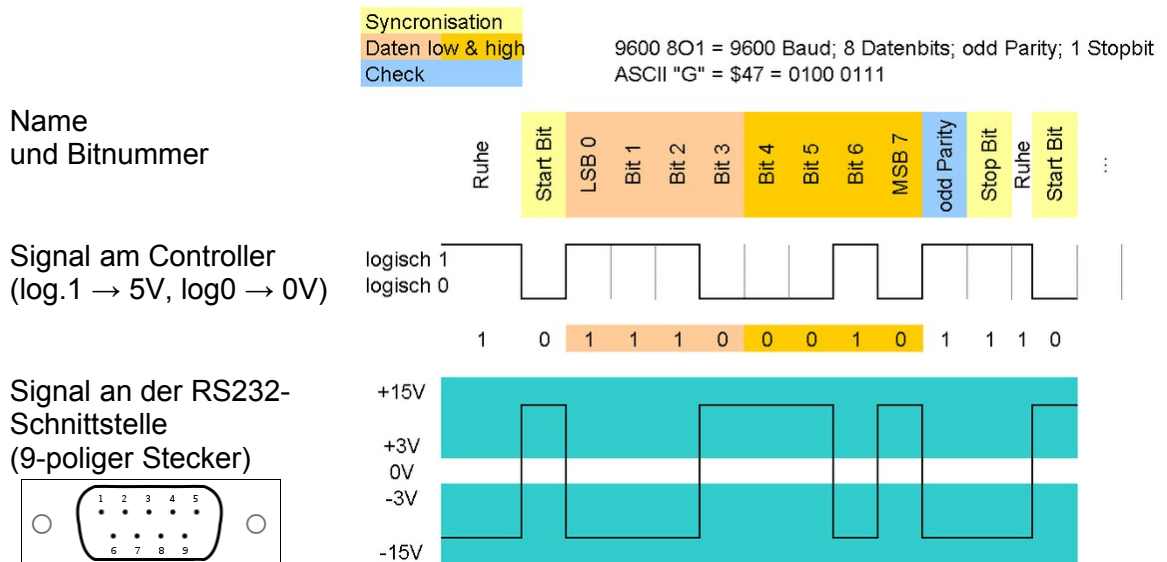


5 Datenaustausch über die serielle Schnittstelle (COM) mit der Mikrocontrollerplatine

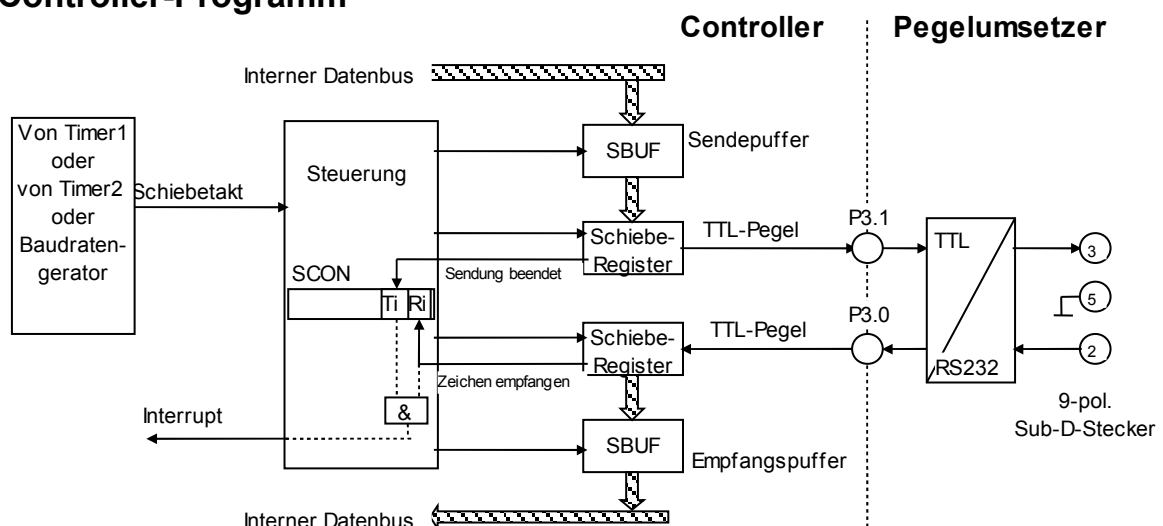
Mikrocontrollerplatine und PC werden über die serielle Schnittstelle miteinander verbunden. Es kann auf PC-Seite ein Seriell-USB-Umsetzer verwendet werden. Dieser kann dann im PC wie eine normale serielle Schnittstelle angesprochen werden.

5.1 Datenformat und Übertragungsraten



- alle Bits werden nacheinander (seriell) übertragen
- Grundzustand der Leitung ist logisch 1
- Setzt der Sender nach der Ruhe die Leitung auf log.0 (Startbit), so erkennt der Empfänger den Beginn der Übertragung.
- Danach werden 8 Datenbit gesendet, im Beispiel der Buchstabe G.
- Anschließend kann zur Fehlerüberprüfung noch ein Paritätsbit gesendet werden.
- Dann wird die Leitung durch mindestens 1 Stopbit wieder in den Grundzustand gebracht.
- Die Übertragungsrate (Baudrate) legt fest, wie viele Bits (inkl. Start-, Stopbit usw.) pro Sekunde gesendet werden.
- Wir arbeiten mit: 9600 Bit/s, 8 Datenbits, kein Paritätsbit, 1 Stopbit

5.2 Controller-Programm



- Zunächst müssen der Baudratengenerator und die serielle Schnittstelle initialisiert werden.
- Dann schreibt man per Befehl ein Byte in den Sendepuffer SBUF.
- Der Controller schickt nun selbständig das Byte inkl. Start- und Stopbit zur Sendeleitung P3.1
- Sobald das Byte übertragen ist, wird das Bit TI gesetzt.
(Durch Abfrage dieses Bits erfährt man, ob der Sendevorgang schon beendet ist)
- Wird seriell über die Empfangsleitung P3.0 ein Byte empfangen, so setzt der Controller das Bit RI und schreibt das empfangene Byte in den Empfangspuffer, der leider auch SBUF heißt.
- Wenn man per Befehl das Byte aus SBUF abgeholt hat, setzt man RI wieder zurück um auch beim nächsten Byte RI wieder sinnvoll abfragen zu können.

```
/*-----
File:      seriell_labview1.c
Autor:     Bubbers
Klasse:    TGJ1
Controller: AT89C5131 (Atmel)
Datum:     16.11.10
Version:    1.0
-----
Beschreibung:
Controller schickt bei Tastendruck P3_3 (blau) das mit den 8 Schaltern
an P1 eingestellte Byte mit 9600 Bit/s über die serielle Schnittstelle
Dieses Zeichen wird auch an P0 mit 8 LEDs dargestellt.
Ein empfangenes Byte wird an die 8 LEDs von P2 ausgegeben
-----*/

#include <at89c5131.h>          // Registeradressen fuer Atmel AT89C5131

sfr at P2  AnzeigeEmpfang; // 8 LEDs
sfr at P0  AnzeigeSenden;
sfr at P1  Sendebyte;
sbit at P3_3 Sendetaster;    // Blauer Taster    = Byte an 8 Schaltern senden

#define reloadBRL 217          // Reloadwert des Baudratengenerators 217 für 9600 Bit/s
                                // BRL = 256- ( 2^SMOD1 * Fper ) / ( 6^(1-SPD) * 32 * Baudrate)
                                // Fper = Fquarz/2 = 6MHz wenn kein x2-Mode (bei X2-Mode Fper = 12 MHz)

void entprellung (void) //----- Zeitverzögerung zur Entprellung -----
{ unsigned char c;
  for (c=255;c!=0;c--);
}

void main (void) //----- Hauptprogramm -----
{
  AnzeigeEmpfang = 0;          // Anzeigen aus
  AnzeigeSenden = 0;           //
                                // Initialisierung serielle Schnittstelle
  PCON = PCON | 0b10000000;    // SMOD=1 bei 9600 Baud
  BRL = reloadBRL;              // Reloadwert für int. Baudratengenerator
  BDRCON = 0b00011110;         // int. Baudraten-Generator für Senden und Empfangen, SPD=1
  SCON = 0b01010000;           // 8-Bit-UART, Empfang zulassen
  ES = 0;                      // kein serieller Interrupt
  RI = 0;                      // Empfangsbit löschen: kein Zeichen empfangen
  TI = 1;                      // Sendebit setzen: letztes Byte wurde gesendet

  while (1)                    // Endlosschleife Hauptprogramm
  {
    if (RI)                    // wenn Zeichen empfangen
    { AnzeigeEmpfang = SBUF;    // abholen und anzeigen
      RI = 0;                  // und für neuen Empfang vorbereiten (Empfangsflag löschen)
    }
    else if (Sendetaster)      // wenn Sendertaster gedrückt
    { while (!TI);              // warten falls altes Byte noch nicht fertig gesendet
      AnzeigeSenden=Sendebyte; // Sendebyte dual anzeigen
      SBUF = Sendebyte;        // und senden
      TI = 0;                  // Sendeflag löschen
      entprellung();           //
      while (Sendetaster);     // warten bis Taster losgelassen
      entprellung();           //
    } // Ende Senden
  } // Ende Hauptprogrammschleife
} // Ende Hauptprogramm
```

5.3 Labview: 1 Byte senden und empfangen

Die VI's zur Kommunikation über die serielle Schnittstelle finden sich bei rechteMaus → Instrumenten-I/O → Seriell.

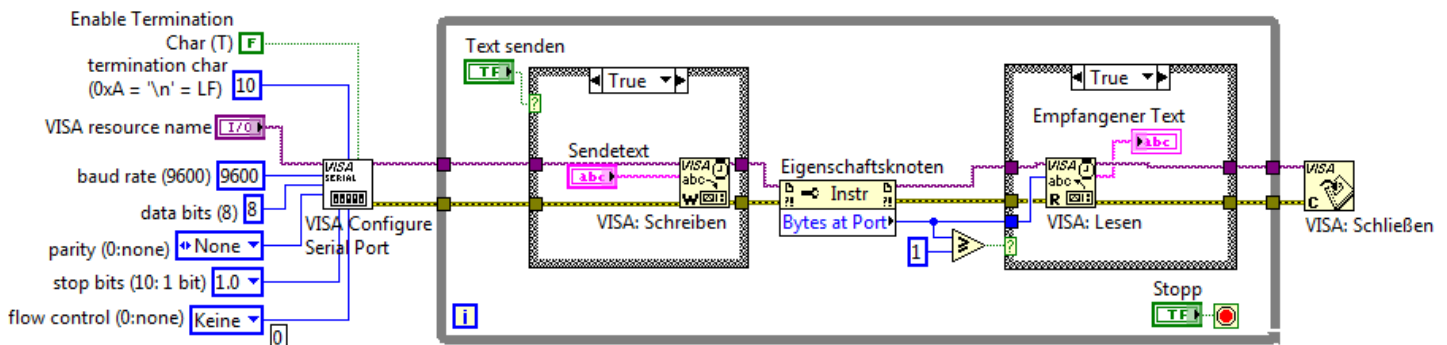
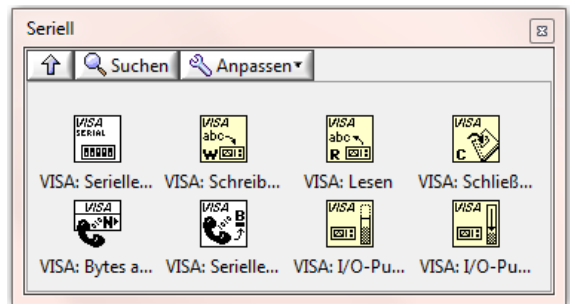
Sie benötigen die oberen vier Komponenten und die linke in Zeile 2.

Visa Konfiguration:

Alle Einstellungen außer resource name mit Konstanten versehen, da sie fest vorgegeben werden.

Mit resource name soll der Anwender die Schnittstelle (COM1, COM2, ...) wählen können.

Alle VISA-Komponenten werden mit einer Visa-Ressourcenname-Leitung und einer Fehler-Leitung verbunden.



Visa Schreiben:

Eingabefeld für Textfeld erstellen.

Senden nur bei Druck auf einen Taster (Erstellen → Bedienelement, dann Schaltverhalten ändern und bei Datenoperation den Standard (nicht gedrückt) auf False ändern)

Visa Bytes am Port erzeugt einen **Eigenschaftsknoten** mit dem man feststellen kann, ob die serielle Schnittstelle Zeichen (vom Controller) empfangen hat.

Visa Lesen:

Nur wenn mindestens ein Zeichen im Empfangspuffer der seriellen Schnittstelle vorhanden ist, sollen alle Zeichen gelesen und angezeigt werden.

Visa Schließen:

Nach Druck auf den Stopp-Button wird die serielle Schnittstelle wieder geschlossen.

5.4 Empfangenes Byte dezimal, hexadezimal und dual darstellen.

5.5 Zahl zwischen 0 und 255 senden

5.6 Empfangene Bytes in einem Array „sammeln“.

5.7 Empfangene Bytes in einem Array „sammeln“, grafisch darstellen und in einer Datei speichern.

5.8 Mehrere in einem Array gespeicherte Werte im Abstand von 1s in einer „Endlosschleife“ an den Controller schicken.