RS232 mit GPIO, Timer, und Interrupt bauen

# Notizen

* Was ist ein RS232?
* Wie kann die Schnittstelle mittels GPIO, Timer und Interrupts realisiert werden
* Was kann man alles mit einer Seriellen Verbindung machen
* Was könnte man Vorzeigen
  + RS232 Schnittstellen auf dem Nucleo Board miteinander kommunizieren lassen
* Entertainment Faktor!

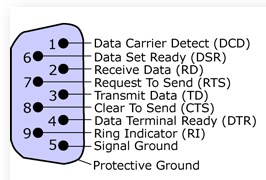
# Was ist zu tun?

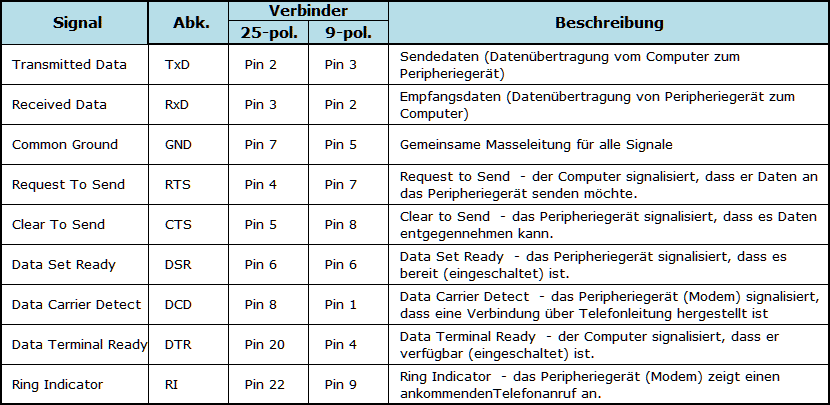
* Informationen Sammeln
* Nucleo Stetup

# Wie Funktioniert ein RS232

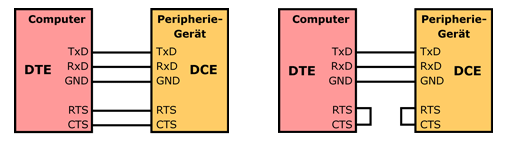
* Dienen zum Seriellen Austausch von Daten von einem PC zu ein Externen Gerät
* Wird auch COM genannt
* Einfach implementierbar und Zuverlässig
* RS = Recommended Standard
* Wortweise Datenübertragung
  + Kann aus 5–9 Bits bestehen; Üblich 7-8
  + Sind in Hexadezimal nach ASCII codiert
  + Verschickt 7-8 Bit lange ASCII kodierte Codewörter
* Datenübertragung Asynchron
  + Heißt der Empfänger muss aus dem Datenfluss das Taktsignal zurückgewinnen
* Flanken werden mittels „Samplen“ des empfangenen Signals mit einem Vielfachen der **Datenrate** erkannt
* Die Synchronisationen des Empfängers wird mittels den Start/Stop Bits im Datenstrom vorgenommen
* Vor dem Start der Übertragung auf der Datenleitung ist dieser im Ruhezustand auf logisch „1“
* Bei beginn geht dieser auf logisch „0“ (invertierter Pegel des Start Bits)
* Dann folgen die Datenbits der Nutzdaten
* Abgeschlossen wird das ganze mit dem Stop Bit welches eine logische „1“ ist
* Die Dauer eines einzelnen Bits ist abhängig von der eingestellten **Bitrate**
* Der Empfänger synchronisiert sich auf die Mitte der einzelnen Datenbits ab und Tastet diese jeweils mit seiner eigen **Bitrate** ab
* Wie maximal erreichbare Datenrate ist jedoch zum Großteil abhängig von Hardwarefaktoren

# Port Informationen

* Portanschluss ist immer als „male“ ausgelegt
* Es existieren Zwei Varianten der RS232 Ports
  + Die „klassische “e 25-Polige Sub-D-Verbinder (Zwei Kanäle)
  + Die 9-Polige Sub-D-Verbinder, In der EDV gängig da nur die 9 Pins benötigt werden (Ein Kanal)
* Pin Adressierung   
  

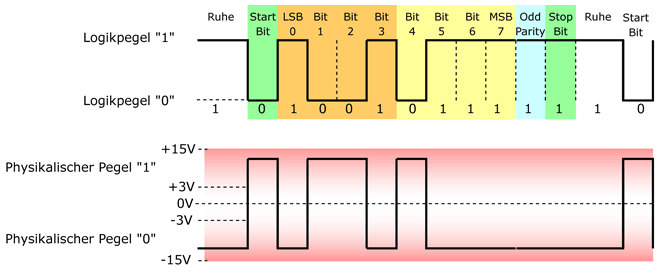


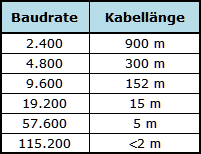
* Von diesen 9 Belegungen werden heutzutage nur mehr 5 benötigt da es keine Modem mehr zwischen den Schnittstellen mehr gibt.
  + TxD, RxD, GND, RTS, CTS
* Wenn auf die Hardware Flusssteuerung (Handshake DTE-DCE) verzichtet, benötigt man nur mehr 3 (TxD, RxD, GND), Dabei wird diese beim Softwarehandshake durchgeführt



# Signalpegel

* Die logischen „1“ und „0“ Pegel können in verschieden physischen Signalpegel werte haben
  + +/- 5V
  + +/- 10V
  + +/- 12V
  + +/- 15V
  + Die tatsächlich anliegenden Pegel sind abhängig von der Treiberschaltung
* Für TxD und RxD ist „0“ ein negativer Pegel und „1“ ein positiver
* Steuerleitungen RTS und CTS ist der Pegel umgekehrt



* Weil RS2323 Signalpegel höher sind Logischen Pegel von Logikbausteinen können Daten nicht direkt verarbeitet werden. Zur Anpassung benötigt man **Pegelwandlerbausteine** welche zwischengeschaltet werden
* Baudrate  
  

# Verbindung

* In unserem Projekt hätten wir eine DTE-DTE Verbindung (gleiches Gerät)
* Bei dem Kabel sollten die gesendeten Pins mit den Empfangenen Pins verbunden werden   
  