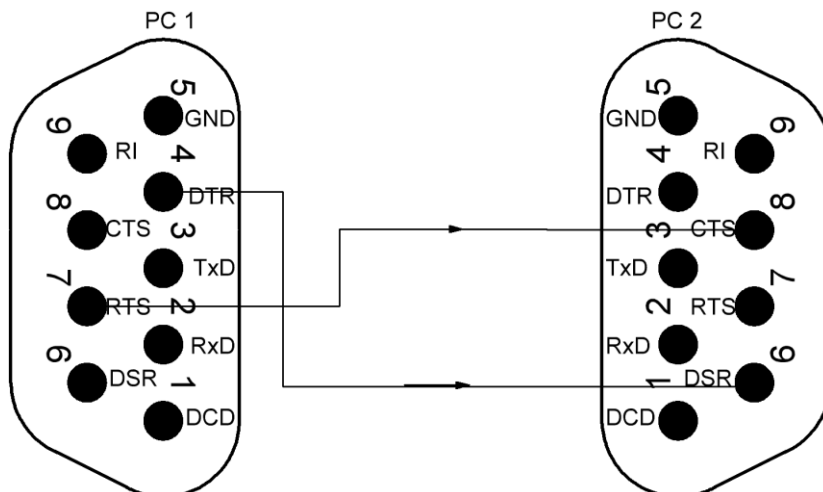


Beim **Hardware-Handshake** signalisieren sich die beiden Geräte (Sender und Empfänger) ihren jeweiligen Status über die bei der seriellen Schnittstelle (RS-232) vorhandenen zusätzlichen gekreuzten **Steuerleitungen RTS, CTS, DTR und DSR**.

In der Zeichnung sind nur die Verbindungen im Nullmodemkabel vom PC 1 zum PC 2 dargestellt (In die Gegenrichtung sind diese beiden Verbindungen natürlich ebenfalls enthalten.)



Zustände der Leitungen bei der Programmierung der seriellen Schnittstelle (kurz SSt) mit der **hessischen Landesabiturklasse Serial**:

Für die Nutzung der SSt muss zunächst ein Objekt der Klasse `Serial` erzeugt werden. Dabei sind alle Konfigurationsparameter mit anzugeben. Das muss bei beiden Kommunikationspartnern der SSt absolut identisch eingestellt sein.

Nach der Objekterzeugung ist die SSt dann technisch für die Nutzung zu öffnen. Das erfolgt mit dem Objekt dann durch einen Methodenaufruf mit `open()`.

Nach dem Öffnen haben die beiden Leitungen RTS und DTR am PC 1 (und damit CTS und DSR am PC 2) **den elektrischen / logischen Zustand 1 (bzw. HIGH bzw. true)**.

Der nachfolgende Codeausschnitt ergibt auf der Konsole die gezeigte Ausgabe:

```
...
com = new Serial("COM" + this.port, 9600, 8, 1, 1);
System.out.println("Status RTS: " + com.getRTS());
System.out.println("Status DTR: " + com.getDTR());

System.out.println("Port wird jetzt geoeffnet...");

this.com.open();

System.out.println("Status RTS: " + com.getRTS());
System.out.println("Status DTR: " + com.getDTR());
```

Konsequenz / Anwendung:

Man könnte damit am PC 2, dessen SSt dafür natürlich geöffnet sein muss, durch Abfrage der Leitungen CTS und DSR auf je gleichzeitig HIGH-Zustand (bzw. auf true) feststellen, ob die SSt des Kommunikationspartners PC 1 geöffnet ist bzw. noch nicht.

```
Status RTS: false
Status DTR: false
Port wird jetzt geoeffnet...
Status RTS: true
Status DTR: true
```