

1. Anmerkungen zu direkte Port-Programmierung

Die Kommunikation zwischen einem Programm und der Hardware werden häufig über so genannte Ports abgewickelt. Unter einem Port hat man sich einen 8 oder 16 Bit breiten Dateneingang bzw. -ausgang vorzustellen, der durch seine Adresse zwischen 0 und 65536 identifiziert wird..

Zur Kommunikation mit den Ports wird eine speziell hierfür entwickelte Klasse (IOPortAccess) bereitgestellt werden, die es ermöglicht, über einen Treiber Daten an einen Port zu schicken und von einem Port einzulesen.

Beispiel 1: Tonerzeugung über den Timer

Programmierung der Sound-Hardware

Die Aufgabe eines Programms, das Töne erzeugen soll, ist es, je nach der Frequenz des Tones, den Lautsprecher (Port-Adresse 61h) dazu zu veranlassen, sich in einer bestimmten Zeit so und so oft vor- und zurückzubewegen. Ein Programm zur Tonerzeugung müsste dem Lautsprecher über den Port mitteilen, ob die Membran sich vor- oder zurückbewegen soll.

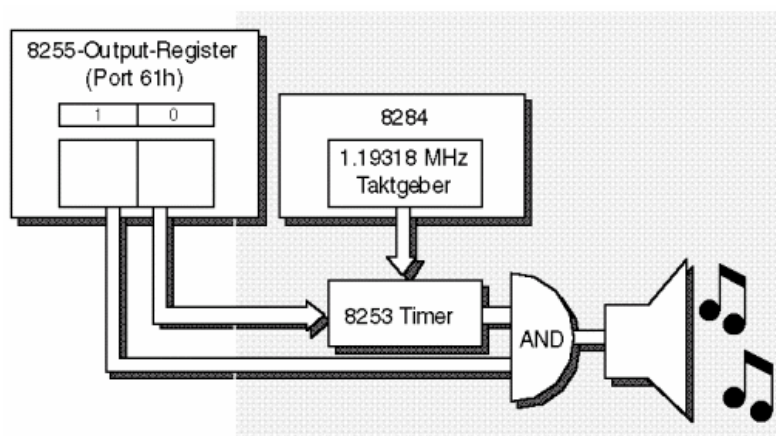
Der programmierbare Zeitgeber 8253

Zur Sounderzeugung bedient man sich zur Tonerzeugung des programmierbaren Zeitgeber mit der Bezeichnung 8253, der im PC eigentlich für die Steuerung der internen Uhr verantwortlich ist. Dass man mit seiner Hilfe auch Töne erzeugen kann, ist dabei eigentlich nur ein nützlicher Nebeneffekt.

Seinen Takt erhält der Zeitgeber 8253 über einen *Oszillator* vom Typ 8284, der 1.193.180 Impulse in der Sekunde erzeugt. Man kann dem 8253 nun mitteilen, nach wie vielen dieser Impulse eine bestimmte Aktion erfolgen soll. Im Fall der Tonerzeugung besteht diese Aktion darin, einen bestimmten Impuls an den Lautsprecher zu senden.

Da der 8253 mit der Angabe der Frequenz mit der Einheit Hertz (Schwingungen pro Sekunde) jedoch nichts anfangen kann, muss die Frequenz zunächst in die Anzahl der Schwingungen des Oszillators umgerechnet werden. Dies geschieht mit Hilfe folgender Formel:

$$\text{Zähler} = 1.193.180 / \text{Frequenz}$$



Erzeugung von Tönen

Programmierung des Zeitgebers 8253

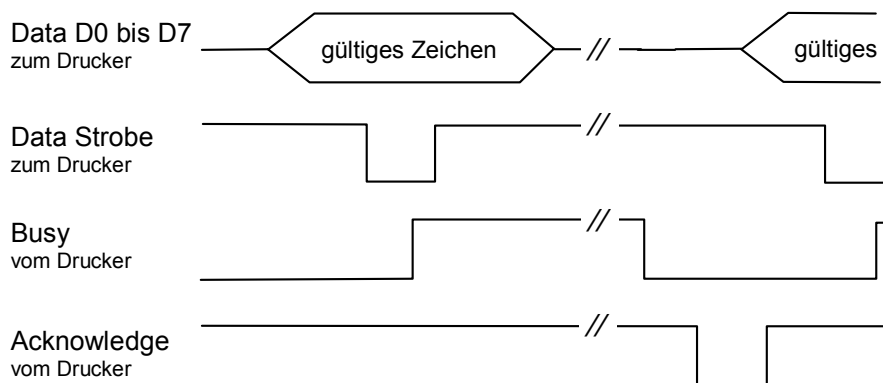
Die Kommunikation zwischen der CPU und dem 8253 erfolgt wiederum über Ports. Zunächst wird dabei der Wert 182 an den Port 43h gesandt. Dadurch wird der 8253 darauf aufmerksam

gemacht, dass er mit der zyklischen Generierung eines Signals beginnen soll, sobald - und das ist der nächste Schritt - ihm die Pause zwischen den einzelnen Signalen mitgeteilt worden ist. Dabei handelt es sich um den mit Hilfe der obigen Formel ausgerechneten Wert. Da der 8253 diesen Wert intern in Form einer 16-Bit-Zahl (ein Wert zwischen 0 und 65535) speichert, schränkt das den Bereich der erzeugbaren Töne auf Frequenzen zwischen 18 und 1.193.180 Hertz ein. Diese Zahl muss an den Port 42h übertragen werden. Da es sich jedoch um einen 8-Bit-Port handelt, können die 16 Bit dieser Zahl nicht auf einmal übertragen werden. Vielmehr müssen zunächst die niederen und dann die höheren 8 Bit übertragen werden.

Jetzt erfolgt der zweite Schritt, indem das Signal des 8253 an den Lautsprecher weitergeleitet wird. Der Zugriff auf den Lautsprecher erfolgt dabei über den Port 61h, der mit einem programmierbaren Peripheriebaustein verbunden ist. Um das Signal des 8253 an den Lautsprecher weiterzuleiten, müssen die beiden untersten Bits dieses Ports auf 1 gesetzt werden. Die übrigen 6 Bits werden für andere Zwecke genutzt und dürfen daher nicht verändert werden. Sobald diese Bits entsprechend gesetzt sind ertönt ein Ton, dessen Erzeugung erst durch das Zurücksetzen der genannten Bits auf 0 beendet wird.

Beispiel 2: Drucken über die Druckerschnittstelle

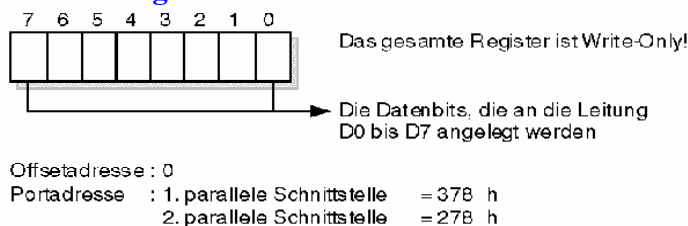
Um ein Zeichen an den Drucker zu übertragen ist best. ein *Übertragungsprotokoll* einzuhalten. Hierzu wird zunächst das auszugebende Byte in das Datenregister der parallelen Schnittstelle geschrieben und dadurch auf die Datenleitungen D0 bis D7 gelegt. Zwar kommen diese Signale sofort beim Drucker an, doch benötigt der Drucker ein weiteres Signal, um sie zu verarbeiten.



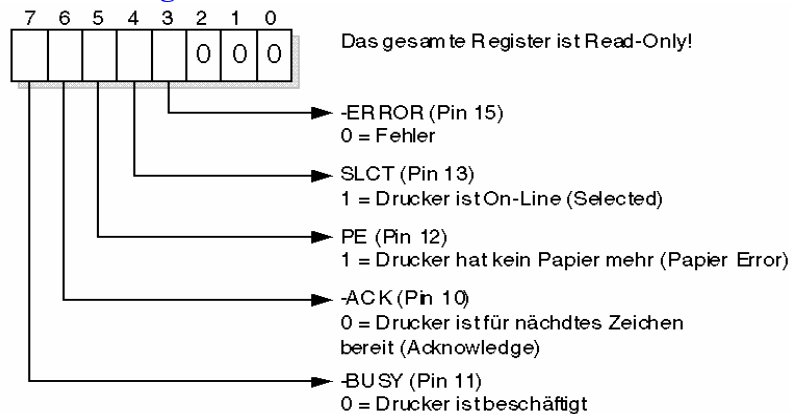
Zusammenspiel von Data Strobe, Busy und Acknowledge bei der Datenübertragung vom Computer zum Drucker

Die Register der Schnittstelle

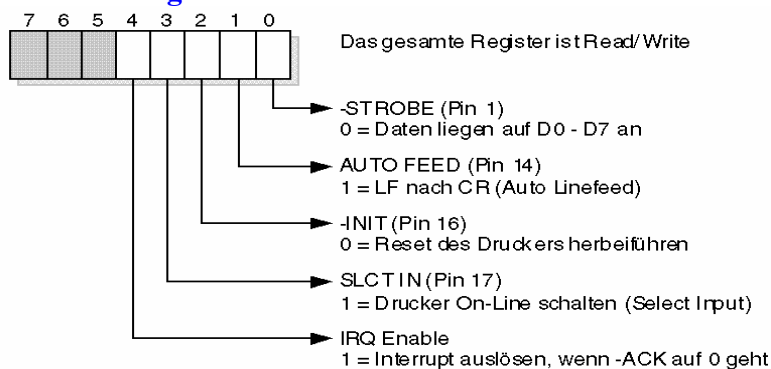
⇒ Datenregister



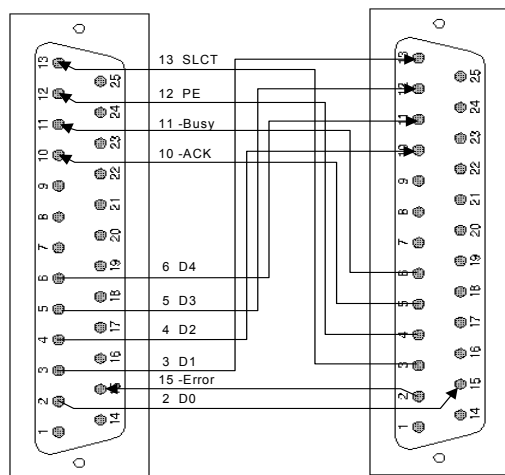
⇒ Statusregister



⇒ Steuerregister



Beispiel 3: Datenübertragung über die parallele Schnittstelle



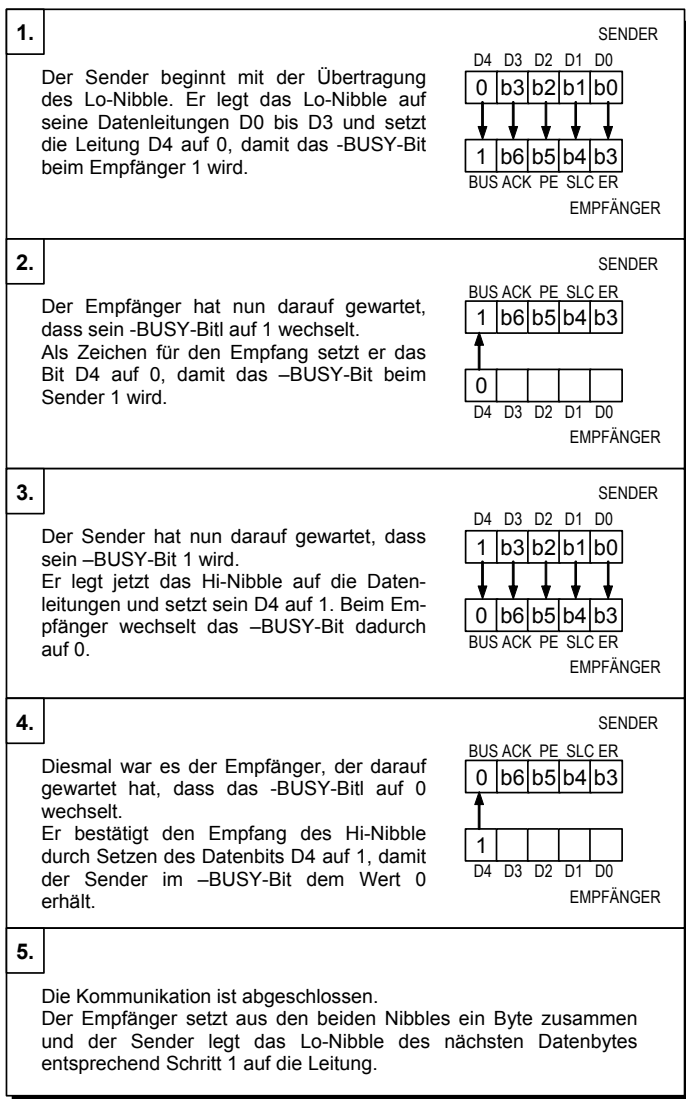
Das parallele Null-Modem-Kabel

Mit einem parallelen Null-Modem-Kabel bietet sich die Möglichkeit, zwei PCs miteinander zu verbinden, um Daten zwischen ihnen auszutauschen.

Das *Übertragungsprotokoll* arbeitet auf zwei Ebenen, die separat behandelt werden:

- die *Ebene der Byte* und
- die *Ebene der Datenblöcke*, die auf der Byte-Ebene sitzt.

Die Byte-Ebene arbeitet hardwareorientiert, während es sich bei der Block-Ebene um ein reines Software-Protokoll handelt.



Das Kommunikationsprotokoll auf der Byte-Ebene