

# Verwendung des COM-Ports zur Ansteuerung externer Hardware

## Mit QBasic-Programmbeispielen

### Version 1.0, 29.7.2003

Von "Skill" ( [skillfinger@gmx.de](mailto:skillfinger@gmx.de) )  
Lektorat und PDF-Konvertierung von Thomas Antoni ( [www.qbasic.de](http://www.qbasic.de) )

---

## 1. Vorwort

(Aus einer Mail von Skill an Thomas Antoni vom 4.7.2003)

Ich weiss nicht so recht, ob meine Anleitung was auf Deiner Q-Basic-Webseite zu suchen hat. Zum Einen ist mein Hobby eigentlich das Elektronik-Basteln. Der Rechner und das Programmieren sind mehr Mittel zum Zweck um z.B. Schaltungen anzusteuern. Meine QBasic-Kenntnisse sind auch gerade noch ausreichend, um mir ein paar Messgeräte zu programmieren; zum Glück sind die PCs ja heute so schnell, dass auch noch ein schlampig in Basic geschriebenes Oszi bis 20 kHz und mehr funktioniert ;-). Die Anleitung wäre also eher was für eine "Elektronik-Spielereien am COM-Port-FAQ".

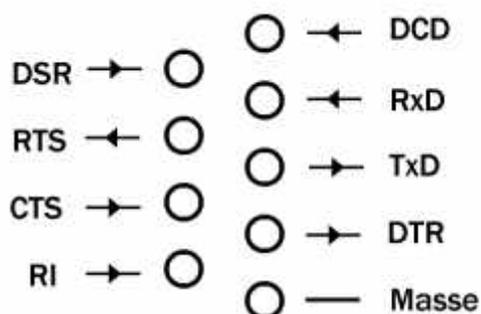
Ausserdem wird sie eigentlich erst durch die Bilder und Schaltpläne interessant. Wenn Du jedoch wie ich der Meinung bist, dass eine solche einfache Verbindung von Programmen mit der Außenwelt viele Anwendungsmöglichkeiten bietet und es mindestens genauso viel Spass bringt, ein Lämpchen statt eines Pixels blinken zu lassen, dann kannst Du vielleicht das Eine oder Andere übernehmen.

Ich bin erst seit wenigen Monaten im Internet, aber [www.qbasic.de](http://www.qbasic.de) hat von Anfang an seinen Platz in meiner "Favoritenliste". Ich finde es schön, dass es doch noch Leute wie euch gibt, die einfach aus Freude an einer Sache soviel Zeit und Mühe investieren.

Ich kann mich jedenfalls nur für alles, was ich dort bisher gelernt habe und bestimmt noch lernen werde, bedanken. In diesem Sinne würde es mich natürlich freuen, wenn ich ein bißchen davon an andere zurückgeben könnte, indem etwas von meiner Anleitung in Deine Monster-FAQ oder Tutorial-Rubrik eingehen würde.

## 2. Der COM-Port und die RS232-Norm

Die Steckbuchse des COM-Ports hat 9 Pins. Drei Ausgänge, fünf Eingänge und Masse. Der RxD-Anschluss (Recieve Data) wird als Empfangs-Datenleitung verwendet und hier nicht weiter beschrieben.



Die Masse ist sowas wie der Minuspol des COM-Ports, jedoch arbeitet dieser mit positiven und negativen Spannungen gegenüber der (eben deswegen so und nicht Minus genannten) Masse. Das ist Teil der sogenannten RS232-Norm.

Es gibt zwei Wege, die seriellen Schnittstelle anzusteuern. Zum Einen die zeichenorientierten Kommunikation mit RS232-tauglichen externen Geräten über den OPEN COM Befehl. Dabei werden zwar

Anfangs solche Dinge wie Datentransferrate, Nutzung eines Stoppbits etc. festgelegt, die eigentliche Datenübertragung wird dann aber vom Rechner intern gesteuert. Der COM-Befehl wird in dem hier vorliegenden Tutorial **nicht** behandelt .

Es gibt aber auch die Möglichkeit, über die entsprechenden Steuerregister auf alle Leitungen - außer den TxD-Eingang - direkt zuzugreifen. Diese Möglichkeit werden wir uns im Folgenden genauer anschauen.

Insgesamt besitzt der COM-Port neun Pins. Da neben der Masse insgesamt noch drei Ausgangs- und fünf Eingangsleitungen vorhanden sind, kann bei der hier beschriebenen Art des Gebrauchs auf TxD also leicht verzichtet werden.

In der Regel liegt an einem Ausgabepin bei AUS bzw. NULL eine negative Spannung von -12 Volt gegenüber der Masse an, bei AN bzw. EIN eine positive von +12 Volt. Die Werte können aber bei manchen Rechnern - vor allem Laptops - abweichen und z.B. nur +/- 3 Volt betragen. Die Eingänge wiederum lesen Spannungen von über + 1 Volt als EIN oder EINS, und solche unter + 1,25 Volt als AUS bzw. NULL. Die Differenz zwischen Ein- und Ausgangsspannung soll eine sichere Übertragung garantieren, ist für unsere Bedürfnisse aber eher hinderlich.

Dazu später mehr. Grundsätzlich will ich mich aber mit elektrotechnischen Details zurückhalten und sie nur soweit behandeln, dass man's nachbauen kann. Wer tiefer einsteigen will, kann sich ja mal im Internet über die Funktionen von Transistoren, Logik-ICs, Zenerdioden usw. informieren oder bei mir nachfragen.

### 3. Zugriff in QBasic

Die einzelnen Bits, die den logischen Wert 1 oder 0 entsprechend EIN oder AUS der einzelnen Pins des COM-Ports enthalten und über die diese gesetzt bzw. ausgelesen werden können, liegen in festen Speicherzellen, den so genannten Steuerregistern.

Gelegentlich wird von Problemen bei der Ansteuerung der seriellen Schnittstelle durch DOS-Programme unter Windows NT/2000/XP berichtet. Ich habe derartige Probleme jedoch noch nicht feststellen können. Im Zweifelsfalle kann man seinen Rechner für COM-Anwendungen mit der Startdiskette unter DOS starten, was zusätzlich den Effekt hat, dass Programme viel schneller laufen.

Die Adresse der Steuerregister setzt sich aus Basisadresse (BA) und Offset zusammen. Die Basisadresse von COM 1 ist 1016 und für COM 2 gleich 760. Wenn's nicht so sein sollte, vielleicht mal die Einstellungen in der Systemsteuerung überprüfen.

Die Register enthalten je acht Bit. Die Werte für die Steuerleitungen DTR und RTS liegen auf dem ersten und zweiten Bit des Registers mit dem Offset 4.

Je nach COM 1 oder 2 errechnet sich so die Adresse entweder als

$$1016 + 4 = 1020$$

oder

$$760 + 4 = 764$$

Im Folgenden benutze ich für Beispiele immer 1016 als BA (Basisadresse). Da das erste Bit ja für die Einer und das Zweite für die Zweier steht, schaltet der Befehl

```
OUT (1020) , 1
```

nur den Ausgang DTR von COM 1 auf EIN und

```
OUT (1020) , 2
```

nur den Ausgang RTS.

OUT (1020) , 3

schaltet beide auf EIN (Bit 1 = 1 und Bit 2 = 1)

OUT (1020) , 0

schaltet DTR und RTS aus.

Der Ausgang TxD wird über das siebte Bit des Registers mit dem Offset 3 gesteuert. Das siebte Bit steht für die 64er.

OUT (1019) , 64

schaltet also TxD ein und

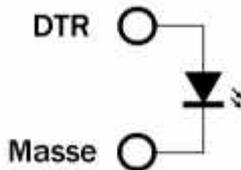
OUT (1019) , 0

wieder aus. Der Zustand von DTR und RTS ändert sich dabei nicht, da diese ja in einem anderen Steuerregister liegen.

#### 4. Erste Schaltungen

Da die Ausgangsströme auf 10 mA begrenzt sind lässt sich so schon direkt eine Leuchtdiode ansteuern. Natürlich auch eine IR-Diode um eine Fernsteuerung zu bauen oder ein Optokoppler als Treiberstufe.

Da an DTR im AUS-Zustand ja eine negative Spannung anliegt, leuchtet die Leuchtdiode auch wenn sie "verkehrt" herum eingebaut wird. Nur ist sie dann eben bei AUS ein- und bei EIN ausgeschaltet. Für die erste Funktionsprüfung ist das also ideal.



Die vier Eingänge CTS, DSR, RI und DCD werden in dieser Reihenfolge über Bit fünf bis acht des Registers mit dem Offset 6 mit dem INP-Befehl abgefragt. Um diese Bits einzeln abfragen zu können, muss mit einer AND-Verknüpfung noch eine entsprechende Maskierung über den Wert von INP gelegt werden.

ein = INP (1022) AND 16

liest so also CTS aus

ein = INP (1022) AND 32

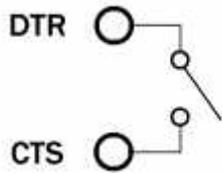
folglich den DSR-Eingang usw. Kleiner Tip: Wenn ihr nur 0 und 1 Werte haben wollt dann macht es z.B. so:

ein = INP ((1022) AND 128) / 128

Dabei muss die AND-Verknüpfung in Klammern gesetzt werden, da BASIC sonst zuerst 128 / 128 rechnet und dann INP (1022) AND 1 .

Da die Ausgänge kurzschlußfest sind, lässt sich eine einfache Schalterabfrage z.B eines Endschalters

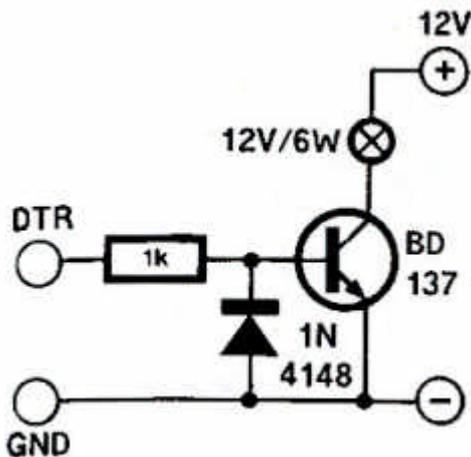
oder mit allen Eingängen (CTS-DCD) die vier Richtungen eines Joypads ohne weitere Beschaltung realisieren. Bei der gezeigten Schaltung muss DTR natürlich eingeschaltet sein.



Kombiniert man diese ersten beiden Schaltungen, kann man mit einer Lochscheibe wie in einer mechanischen Maus schon Drehzahlen messen.

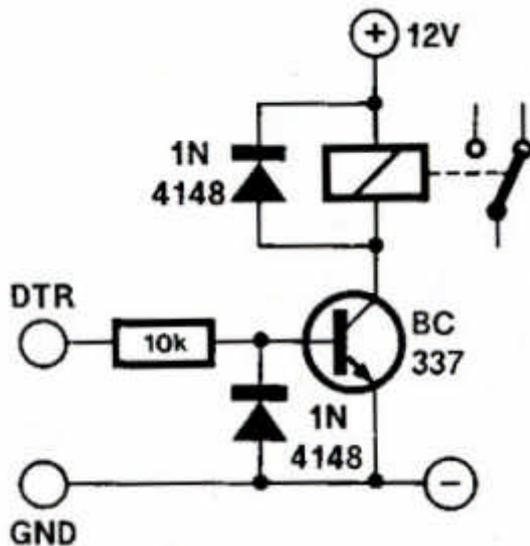
Um grössere Lasten wie z.B. kleine Glühlämpchen zu schalten, braucht man eine externe Stromversorgung. Im Beispiel sind 12 V / 6 W angegeben. Diese Angaben beziehen sich auf den Transistor BD 137. Werden andere NPN-Transistoren benutzt, können natürlich auch andere Lasten geschaltet werden. Auf den 1k-Widerstand am Eingang kann verzichtet werden. Für diejenigen unter euch, die Schaltpläne nicht so gut lesen können: In der Anlage "Bilder" werden die Bauteile und ihre Anschlüsse genauer erklärt.

Die Diode 1N4148 leitet hier die negative -12 V Spannung im AUS-Zustand direkt an die Masse ab.



Um galvanisch getrennte Stromkreise wie z.B. den Märklin Modellbahn-Wechselstrom oder Netzspannung ( VORSICHT ! ) zu schalten, kommt man um die Verwendung eines Relais nicht herum. Das Relais kann über eine Transistorschaltung wie oben gesteuert werden. Zu beachten ist dann aber, dass parallel zur Spule des Relais eine so genannte Freilaufdiode geschaltet wird, da das Relais beim Ausschalten nochmal einen entgegengesetzten, evtl. sehr hohen Spannungsstoß abgibt, der über diese zweite Diode abgeleitet wird. Aber Vorsicht ! Die Freilaufdiode muß unbedingt richtig herum eingebaut werden, da sie sonst die Relais-Stromversorgung kurzgeschlossen wird. Man hat ja dann schon drei Stromkreise:

- den Steuerstrom zwischen DTR und Masse
- den Strom, der die Spulen der Relais versorgt (z.B. 5 oder 12 Volt) und dessen Masse mit derjenigen des COM-Ports verbunden ist und schließlich
- den ganz unabhängigen Stromkreis , der über das Relais geschaltet wird.



Einfacher geht's mit dem Treiber (=Verstärker)-IC ULN 2803 (Conrad BestNr. 171824). Dieser hat 8 Treiberstufen, die jeweils bis 12 V / 500 mA belastbar sind und schon die Freilaufdioden eingebaut haben. Die Anschlußbelegung mit einem Schaltbeispiel für zwei Glühlämpchen (es könnten auch 2 Relais sein) findet ihr im Anhang unter "Bilder".

Im Prinzip enthält der IC acht komplette Transistorstufen. Zu beachten ist dabei nur, dass hier die Verbraucher (Lämpchen, Relais etc.) auf einem gemeinsamen Pluspol liegen und sozusagen die Verbindung zur Masse geschaltet wird. Folglich muß am Ausgang des ICs ein AUS-Signal ( 0 ) anliegen, um den Verbraucher einzuschalten. Aber die Ausgangswerte werden durch die im IC enthaltene Schaltung auch negiert ( 1 = 0 und 0 = 1 ). Dadurch wird doch wieder durch eine 1 an DTR oder RTS die Lampe oder was auch immer eingeschaltet.

Kurzer Tip zum Löten von ICs: Da diese relativ hitzeempfindlich sind, solltet ihr erst einen Stecksockel (gibt's bei Conrad für alle Größen) einlöten und dann den IC einstecken. Das hat auch den Vorteil, dass man die ICs, die teilweise mehr kosten als der Rest der Schaltung, für mehrere Anwendungen nutzen kann.

## 5. seriell / parallel

Der einzige Nachteil des COM-Ports ist seine kleine Anzahl von Steuerleitungen. Durch Benutzung eines seriell/parallel-Wandlers für die Erweiterung der Aus- bzw. eines parallel/seriell-Wandlers für die der Eingänge, lässt sich das aber ausgleichen. Außerdem sind diese Elemente beliebig auf 16, 24 oder mehr Ein- und Ausgänge erweiterbar.

Zur Verwendung von Logik-IC's (TTL / CMOS) ist nochmal etwas E-Technik nötig. TTL-IC's (Transistor-Transistor-Logik) arbeiten mit einer Versorgungsspannung von 5 Volt, und auch die Ein- und Ausgangssignale sind 5 V bei EIN ( 1 ) bzw. 0 V (Masse) bei AUS. Dieser TTL-Pegel ist mit praktisch allen anderen ICs kompatibel und soll auch bei den folgenden Beispielen die Grundlage sein. Die Eingänge des COM-Ports können den TTL-Pegel direkt lesen.

Die Ausgangsspannung muß jedoch von 12 auf fünf Volt begrenzt werden. Das geschieht ganz einfach, indem man anstatt der Diode 1N4148 eine so genannte Zenerdiode benutzt. Diese hat die Eigenschaft, auch in Sperrichtung ab einer bestimmten Spannung Strom durchzulassen. Die Zenerdiode ZPD 4,7 V lässt, wie der Name schon sagt, ab einer Spannung von 4,7 V den Strom an die Masse abfließen. 4,7 Volt reichen aus um TTL-Eingänge zu schalten. Seht euch dazu vielleicht die entsprechenden Datenblätter mal genauer an.

Der unter "Bilder" beschriebene serien/parallel-Wandler 744094 (Best.Nr. 151653; ich gebe die Bestell Nr.

auch an, weil unter der auch das Datenblatt des Bauteils bei Conrad zu finden ist) besteht zum Einen aus einem 8-Bit Schieberegister. Dieses schiebt bei einem Signal am Takteingang (Clock) - also DTR an und wieder aus – die acht alten Bits um eine Bitposition nach hinten. Das letzte geht entweder verloren oder kann per Übergabe an ein zweites Schieberegister, das vom selben Takt gesteuert wird, übertragen werden.

Als erstes Bit wird der Wert ( 1 oder 0 ) übernommen, der am Dateneingang (Data) anliegt. Sind so alle acht Bit neu "aufgefüllt" worden, werden sie durch das Übernahmesignal (Strobe) an ein zweites Register übergeben, das die Ausgänge steuert. Die Ausgänge ändern sich also immer alle gleichzeitig und nur bei einem Strobe-Signal.

Ein einfaches Steuerprogramm für die unter "Schaltungen" gezeigte Schaltung könnte so aussehen. Mit den Tasten 1-8 werden die acht Ausgänge des 744094 separat geschaltet. Die Basisadresse (BA) ist wieder 1016.

```
schleife:
taste$ = INKEY$
IF taste$ = "" THEN GOTO schleife
IF taste$ = CHR$(32) THEN END           ' Abbruch mit ENTER
wert = ASC (taste$) - 48
IF wert < 1 OR wert > 8 THEN GOTO schleife
IF bit(wert) = 0 THEN bit(wert) = 64 ELSE bit(wert) = 0
FOR takt = 1 TO 8
OUT (1019) , bit(takt)
OUT (1020) , 1                          ' DTR schalten
OUT (1020) , 0
NEXT
OUT (1020) , 2                          ' RTS schalten
OUT (1020) , 0
OUT (1019) , 0                          ' TxD aus falls bit(8) = 1 war
GOTO schleife
```

Der parallel/seriell-Wandler 4021 (Best.Nr. 172766) arbeitet nach dem selben Prinzip für die Erweiterung auf acht Eingänge oder mehr.

Bei einem Strobe-Signal werden die an den acht Eingängen anliegenden Werte in ein Register übertragen.

Danach liegt am Data-Ausgang des 4021 (hier werden ja jetzt Werte zum Rechner hin übertragen) das erste Bit an, nach dem Clock-Signal das zweite usw. Auch diese Schaltung lässt sich beliebig erweitern und durch Programmierung eines entsprechenden "Übertragungsprotokolls" auch gleichzeitig beliebig viele Ein- und Ausgänge steuern und abfragen.

Ein entsprechendes Programm für den 4021, das den CTS-Eingang benutzt, funktioniert prinzipiell so

```
schleife:
OUT (1020) , 2
OUT (1020) , 0
FOR takt = 1 TO 8
bit(takt) = ( INP (1022) AND 16 ) / 16
OUT (1020) , 1
OUT (1020) , 0
NEXT
LOCATE 1,1
FOR wert = 1 TO 8
PRINT "Eingang Nr." ; wert ; ": " ; bit(wert)
NEXT
IF INKEY$ = "" THEN GOTO schleife ELSE END
```

Zu beachten ist noch, dass bei superschnellen Rechnern vielleicht noch eine Zeitschleife zwischen Ein- und Ausschalten der Steuerleitungen eingebaut werden muss. Das Datenblatt macht auch Angaben über

die Mindestdauer von Impulsen.

Da die ICs aber bis in den MHz-Bereich arbeiten, ist das kaum nötig, außer vielleicht beim Beschreiben von EPROMS oder dem Auslesen von seriellen A/D-Wandlern etc.

Für diejenigen, die sich mit Schaltplänen nicht so gut auskennen, habe ich mal einige Beispiele unter "Schaltungen" angefügt, die auch unerfahrene Elektronikbastler nachbauen können. Ich hoffe jedenfalls, sie sind verständlich - hab' mir ziemlich Mühe gegeben (-:

## 6. Nachwort

So, das war's erst mal. Mit den Beispielschaltungen läßt sich z.B. eine 24-Kanal-Relaiskarte realisieren. Eine Anregung dazu unter "Schaltungen".

Überhaupt ist es für Experimentierzwecke sehr hilfreich, einzelne Elemente wie die hier gezeigten Schaltstufen und seriell/parallel- bzw. parallel/seriell-Wandler als einzelne Module aufzubauen. Ich verbinde diese dann ja nach Bedarf mit kleinen Steckkabeln.

Auf jeden Fall gibt es Anwendungen ohne Ende. Auch uralte Rechner können da noch zu Ehren kommen - wer will eine Modellbahnweiche schon mit 50 kHz hin- und herschalten...

Eurer Phantasie sind also keine Grenzen gesetzt. Ich hoffe, ich konnte ein paar Anregungen geben. Wie gesagt, könnt ihr auch gerne nachfragen, auch wenn's manchmal mit der Antwort auch etwas dauert.

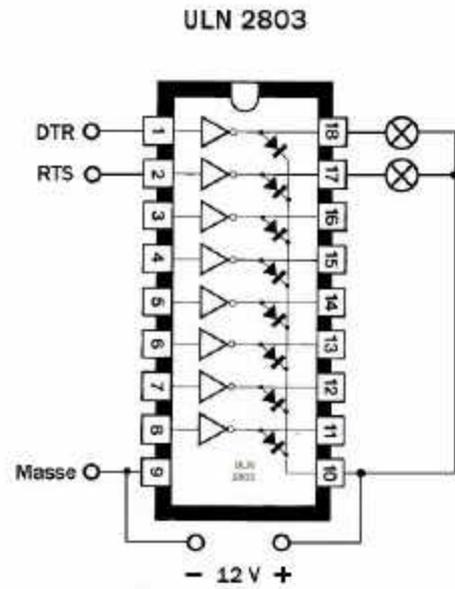
# Anlage "Bilder"



Theorie...



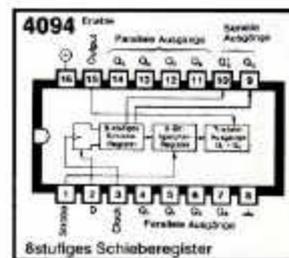
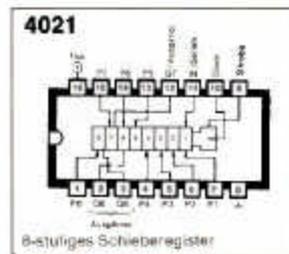
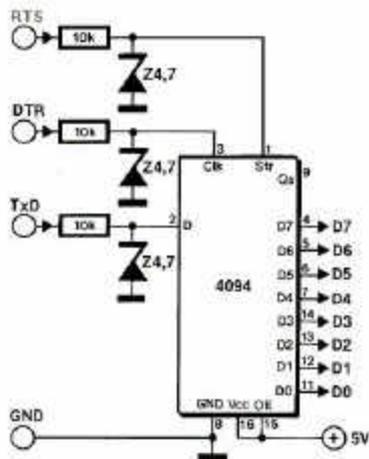
...und Praxis



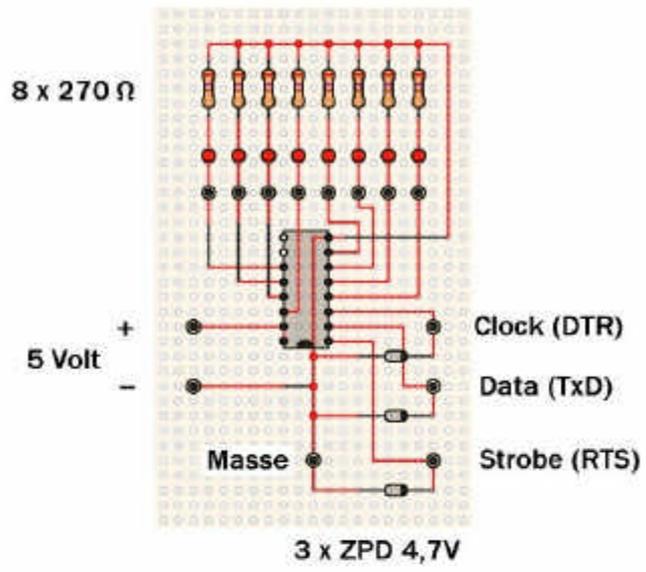
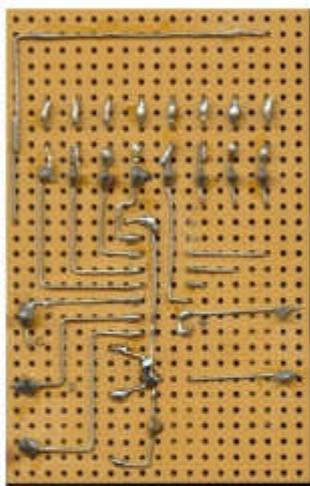
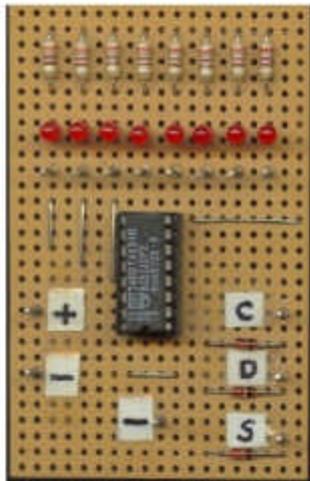
## Dioden



Der Ring zeigt an, wo welche Seite der Diode liegt. Hier ist sie so wie auf dem jeweiligen Symbol links.



# Anlage "Schaltungen"



- Leitung, die über der Platine verläuft
- Leitung, die unter der Platine verläuft
- Zwei Leitungen kreuzen sich ohne Berührung
- Verbindung zwischen zwei Leitungen
- Anschlusspunkt
- Widerstand
- Diode ZPD 4,7 V
- Leuchtdiode

