

### Digitale Eingabe und Ausgabe

Unter „digital“ versteht man Signale, die entweder an oder aus (Spannung vorhanden oder Spannung auf Null) sein können, aber keine Werte dazwischen annehmen. Die digitalen Pins sind die mit Beschriftung von 0 bis 13.

Befehl	Beschreibung	Beispiel
<code>pinMode(Pin, INPUT);</code>	Legt den Pin als Dateneingang fest, d.h. der Arduino kann Signale empfangen. Typischerweise benutzt Du diesen Befehl innerhalb von <code>setup()</code> . Englisch: pin = Nadel / Stecknadel / Anschluss Englisch: mode = Art / Betriebsweise Englisch: in = herein / innen	<code>pinMode(3, INPUT);</code>
<code>pinMode(Pin, OUTPUT);</code>	Legt den Pin als Datenausgang fest, d.h. der Arduino kann Signale verschicken. Typischerweise benutzt Du diesen Befehl innerhalb von <code>setup()</code> . Englisch: out = heraus / außen	<code>pinMode(4, OUTPUT);</code>
<code>int name = digitalRead(Pin);</code>	Liest ein Signal und entscheidet, ob es an oder aus ist. Dazu muss der Pin als <code>INPUT</code> definiert sein. Englisch: read = lesen / eingeben	<code>pinMode(3, INPUT);</code> <code>int name = digitalRead(3);</code>
<code>digitalWrite(Pin, HIGH);</code>	Schält das Signal an. Dazu muss der Pin als <code>OUTPUT</code> definiert sein. Englisch: write = schreiben	<code>pinMode(4, OUTPUT);</code> <code>digitalWrite(4, HIGH);</code>
<code>digitalWrite(Pin, LOW);</code>	Schält das Signal aus. Dazu muss der Pin als <code>OUTPUT</code> definiert sein.	<code>pinMode(4, OUTPUT);</code> <code>digitalWrite(4, LOW);</code>
<code>analogWrite(Pin, Wert);</code>	Simuliert ein analoges Signal durch unterschiedlich langes Wechseln des digitalen Ausgangs <sup>*)</sup> . Werte: von 0 (immer aus) bis 255 (immer an)	<code>pinMode(4, OUTPUT);</code> <code>analogWrite(4, 30);</code>

<sup>\*)</sup> dieses Verfahren nennt sich PWM (Puls-Weiten-Modulation). Du kannst mit dem Oszilloskop untersuchen, wie es sich verhält. Hier ein paar Beispiele:

