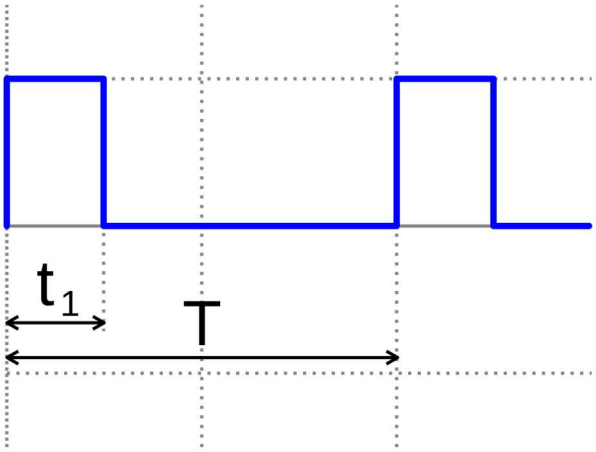
Pulsweitenmodulation (PWM)

Im „realen“ Leben gibt es im Gegensatz zur Digitaltechnik mehr als zwei Zustände („High“ oder „Low“, „Wahr“ oder „Falsch“, „1“ oder „0“, „Ja“ oder „Nein“). Zum Beispiel kann man mittels eines Dimmers die Helligkeit einer Glühlampe einstellen, während ein Schalter nur die Lampe an- oder ausschalten kann. Da ein Mikrocontroller, so auch der ARDUINO, eigentlich nur zwei Zustände an seinem Port- Pins schalten kann, bedient man sich hier der PWM, um ein analoges Signal zu „simulieren“.
„Die **Pulsweitenmodulation** (kurz *PWM*, auch **Pulsdauermodulation** *(PDM)*, **Pulslängenmodulation** (*PLM*), **Unterschwingungsverfahren** oder **Pulsbreitenmodulation** *(PBM)* (auf englisch: *pulse-width modulation* (PWM)) ist eine Modulationsart, bei der eine technische Größe (z. B. elektrische Spannung) zwischen zwei Werten wechselt. Dabei wird bei konstanter Frequenz der Tastgrad eines Rechteckpulses moduliert, also die Breite der ihn bildenden Impulse.“ (Quelle: Wikipedia)
Die PWM wird dadurch erzeugt, in dem ein Port- Pin kurzzeitig auf „High“ gesetzt wird aber sonst den Wert „Low“ annimmt. Nun kann man die Zeit „t_1“, die der Pin auf „High“ eingestellt ist von 0 auf t_1=T (Periodendauer) variieren (s. Bild).



Square wave.svg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=3683151

Beim ARDUINO wird die PWM mittels dem Befehl „`analogWrite()`“ realisiert. Diese Funktion funktioniert, obwohl sie ein analoges Signal simuliert mit den Digitalpins 3, 5, 6, 9,10 und 11 (wobei die Pin 5 und 6 technisch bediungt mit einer kürzeren Periodendauer betrieben werden). Der Funktion „`analogWrite()`“ werden zwei Parameter übergeben, nämlich der Pin und die Pulsweite (0 bis 255). Danach wird die PWM an dem Pin solange ausgeführt bis der Pin mit „`digitalWrite()`“ auf einen festen Wert gesetzt wird.