

## Station 4 – Farbthermometer – Bonus 1

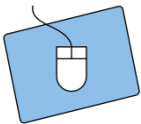
### Mehr als nur an und aus

Auf dem ersten Arbeitsblatt wurde schon angedeutet, dass man mit den drei Grundfarben alle sichtbaren Farben erzeugen kann. Allerdings reicht es dafür nicht aus, die LEDs einfach nur an- und auszuschalten. Zum Glück kann der Arduino mehr! Vielleicht ist euch schon aufgefallen, dass an einigen Pins des Arduinos kleine Schlangen (~) abgebildet sind. Diese Pins werden auch **PWM-Pins** genannt. Auch die Pins 9 bis 11, die ihr eben mit der RGB-LED verbunden habt, sind mit solchen ~ markiert. Diese Pins können 256 verschiedene Werte annehmen. Die passende **Anweisung** lautet:





```
analogWrite(pin-name, wert);
```

Der `pin-name` ist wie im Einstiegsprojekt der Name der Variablen, die die Nummer eures entsprechenden Pins enthält. `wert` ist eine Zahl von 0 bis 255 oder der Name einer Variablen, die eine solche Zahl enthält. Wie lautet die Anweisung, um die blaue LED komplett einzuschalten?

```
analogWrite(_____, _____);
```



1. Nehmt jetzt euren bisherigen Sketch und speichert ihn nochmal unter einem neuen Namen.
2. Ersetzt den Befehl `digitalWrite()` durch `analogWrite()` und mischt weitere Farben! Notiert in der Tabelle die Werte, für die eure LED den dargestellten Farben am nächsten kommt!

	Rot	Grün	Blau
	255		0
	146		80
		50	160
	255	255	

Jetzt wisst ihr, wie die Farbmischung mit einer RGB-LED funktioniert und könnt wirklich alle Farben erzeugen. Ihr könnt euer Farbthermometer beliebig anpassen. Viel Spaß dabei!


Temperatur	Farbe	R(ot)	G(rün)	B(lau)

**Hinweis:** Hier reicht es jetzt nicht mehr, nur einzutragen, ob die LED an oder aus ist, sondern ihr müsst die entsprechenden Werte eintragen.



## Station 4 – Farbthermometer – Bonus 1

### Quellenverzeichnis:

 – Quelle: InfoSphere, CC BY-SA 4.0 Attribution-ShareAlike 4.0 International  
(<https://creativecommons.org/licenses/by-sa/4.0/>)