

## Station 4 - Farbthermometer

### Temperatur zum Leuchten bringen

Wie viele Menschen haben sich schon die Finger verbrannt, weil die Herdplatte noch heiß war? Oder haben angewidert das Gesicht verzogen, weil der Tee schon kalt war? Das alles passiert nur, weil der Mensch Temperatur nicht sehen kann. Das könnt ihr jetzt ändern!



Eure Aufgabe ist es, ein Thermometer zu bauen, das mittels verschiedener Farben anzeigt, welche Temperatur gerade herrscht.



Abb. 1: Einige LEDs






Abb. 2: Ein Brunnen mit LEDs

### Farbmischung

#### Aufbau der Schaltung

Eure Schaltung beinhaltet folgende Elemente:

- **3x Widerstand (rot-rot-braun-gold)** 
- **1x Temperaturfühler** 
- **1x RGB-LED** 
- **4 gelbe lange, je 1 blaues und rotes kurzes und langes Kabel**

#### RGB-LEDs

RGB-LEDs sehen aus wie normale LEDs, haben aber mehr Beinchen. Eine RGB-LED verbindet nämlich eine **ROTE**, eine **GRÜNE** und eine **BLAUE** LED. Diese haben alle einen eigenen Pluspol, teilen sich aber einen gemeinsamen Minus-Pol. **(Achtung: Hier ist der Minus-Pol das längste Beinchen!)**

Diese drei Farben reichen, um alle sichtbaren Farben darzustellen. Doch dazu später mehr.

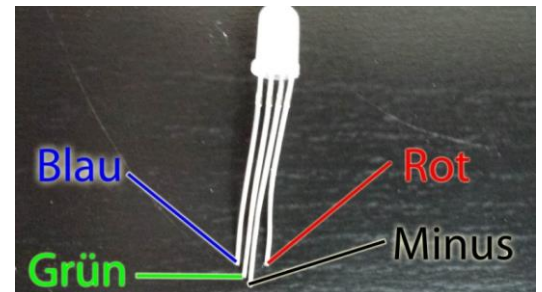


Abb. 3: RGB-LED



1. Verbindet die Plus-Leiste des Steckbretts mit 5V und die Minus-Leiste des Steckbretts mit GND auf dem Arduino.
2. Steckt die RGB-LED jetzt so in das Steckbrett, dass der Minus-Pin (das lange Beinchen) in der Minus-Leiste des Steckbretts steckt. Die anderen Pins kommen jeweils in eine eigene Reihe des Steckbretts.
3. Ergänzt anschließend die Widerstände und gelben Kabel wie in der Abbildung. Die oberen Enden der gelben Kabel hängen aber erstmal in der Luft.
4. Verbindet die gelben Steckkabel jeweils einzeln mit der Plus-Leiste und merkt euch, welche Farbe zu welchem Beinchen gehört! Prüft, welche Farben ihr erhaltet, wenn ihr die Beinchen jeweils paarweise und schließlich zu dritt anschließt!

## Station 4 - Farbthermometer

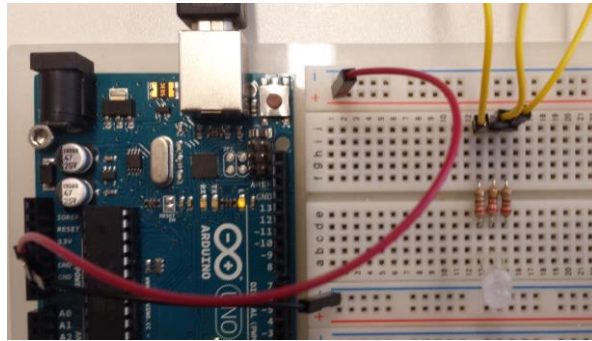


Abb. 4: Komplette Schaltung

### Additive Farbmischung

Im Gegensatz zur subtraktiven Farbmischung, die ihr aus dem Kunstunterricht kennt, nehmt ihr bei RGB-LEDs kein Licht weg, sondern mischt neues dazu. Mischt man alle Anteile, so erhält man hier weißes Licht. Die Grundfarben hier sind **Rot**, **Grün** und **Blau**.

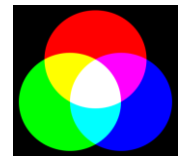


Abb. 5: Additive Farbmischung



Natürlich ist es sinnvoll, die RGB-LED mit dem Arduino zu steuern, statt immer Kabel umstecken zu müssen. Verbindet also die Kabel jetzt nicht mit der Plus-Leiste, sondern mit den digitalen Pins 9 (Blau), 10 (Grün) und 11 (Rot) am Arduino.

Jetzt geht es daran, die drei LEDs ein- und auszuschalten! Dies macht ihr wieder mit ArduBlock! Legt dazu ein neues Programm an und speichert alles unter einem sinnvollen Namen!



- Ihr müsst für 3 LEDs die entsprechenden Blöcke herausuchen. Die LEDs sind an Ausgängen angeschlossen, also müsst ihr unter **Output** suchen.
- Schaltet die LEDs nacheinander an und aus:
- Benutzt **Warte Millisekunden** zwischen den Schaltvorgängen, um die Unterschiede zu sehen.
- Speichern und ausprobieren!



### Mehr als nur drei Farben

Wie ihr ja bereits gesehen habt, können RGB-LEDs noch mehr als nur drei Farben darstellen. Dazu müsst ihr einfach 2 oder auch 3 der eingebauten LEDs gemeinsam anschalten.



- Kombiniert verschiedene LEDs, um unterschiedliche Farben zu erzeugen: **gelb**, **pink**, **türkis** und **weiß**
- Testet euer Programm und vergesst nicht das Speichern!

All diese Farben könnt ihr nutzen, um verschiedene Temperaturen darzustellen. So sieht selbst euer kleines Geschwisterchen auf den ersten Blick den Unterschied zwischen warm und kalt!



## Station 4 - Farbthermometer



- Um den Wert dieser Variable zu berechnen, muss erst der Sensorwert (0 bis 1023) in eine Spannung (0 bis 5.0 V) und anschließend die Spannung in die Temperatur (0,01 V pro 1°C) umgerechnet werden:



Die Blöcke für die Formel findet ihr unter **Math. Operatoren**, die Blöcke für die Zahlen könnt ihr unter **Variablen/Konstanten** finden. Hier ist wichtig, dass bei der 5.0 auch wirklich „5 Punkt 0“ steht.

- Am Ende sollte das Ganze so aussehen:



- Um die Temperatur zu überprüfen, kann man den **Seriellmonitor** nutzen:



Die Blöcke findet ihr unter **Kommunikation**.

- Euer Programm sollte jetzt so aussehen:



Zum Test klickt ihr zunächst auf **Hochladen auf den Arduino** und anschließend auf **Seriellmonitor**. Wenn die ausgegebenen Werte ungefähr mit der Raumtemperatur übereinstimmen ist alles in Ordnung.

# Station 4 - Farbthermometer

## Das Farbthermometer

Nun zu den Farben! Es bietet sich an, im Bereich zwischen Raumtemperatur (ca. 20°C) und Körpertemperatur (ca. 36°C) zu arbeiten.

Ihr legt zunächst drei Farbbereiche fest: Unter 24°C soll die LED grün leuchten, zwischen 24 und 27°C blau, über 27°C rot. Also müssen drei Fälle abgedeckt werden.



1. Baut für jeden Temperaturbereich einen *falls*-Block aus **Steuerung** ein:

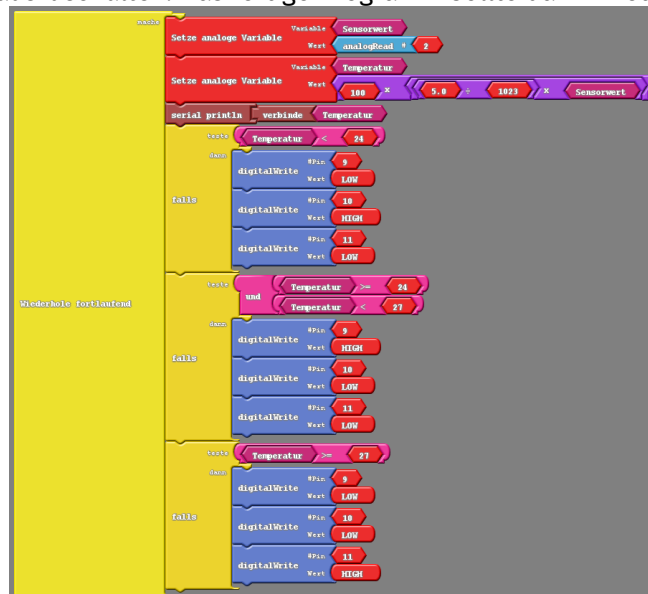


Zum Vergleichen braucht ihr  $<$ ,  $>$  und  $\leq$  bzw.  $\geq$  sowie den *und*-Block aus **Log. Operatoren**.

Der *und*-Block ist für den mittleren Bereich notwendig. Die Blöcke gehören in den teste-Bereich:



2. Lasst je nach Bereich die richtigen LEDs leuchten. *Vergesst dabei nicht, die jeweils anderen LEDs wieder auszuschalten.* Das fertige Programm sollte dann in etwa so aussehen:



Wenn alles geklappt hat, sollte es bei euch etwa wie in den beiden Bildern auf der nächsten Seite funktionieren! Ihr könnt natürlich auch mehr Temperaturbereiche anlegen und Farben mischen.

## Station 4 - Farbthermometer

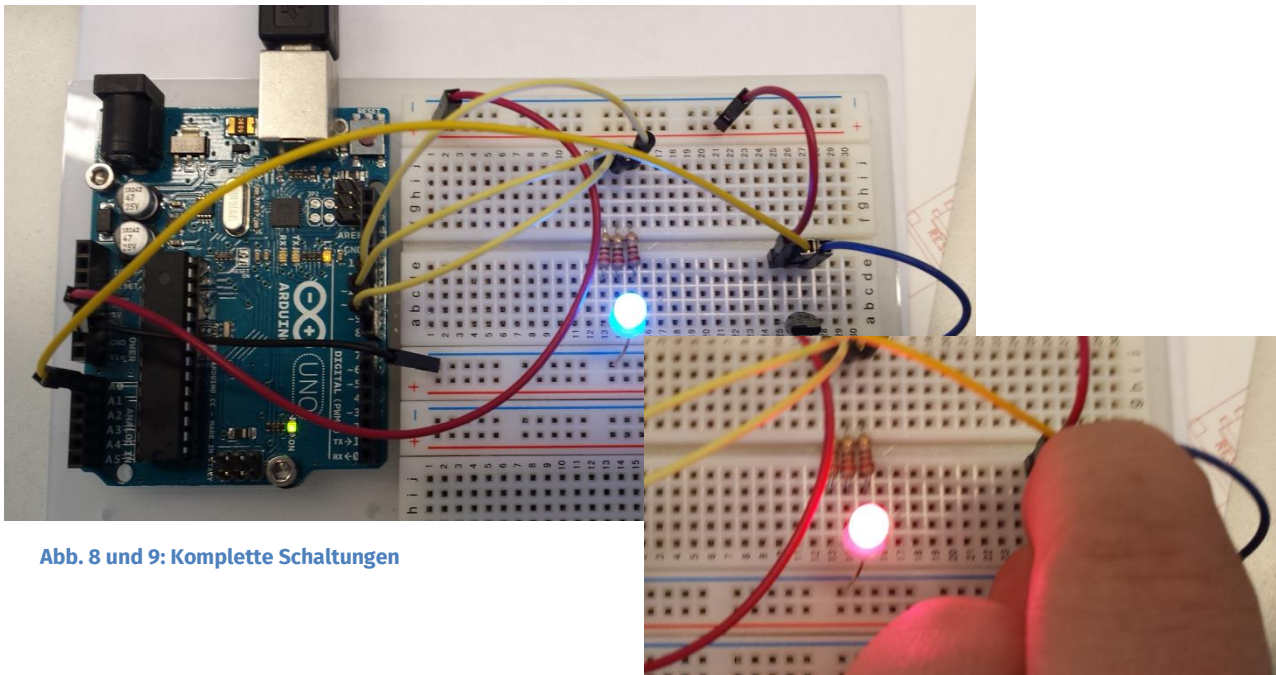


Abb. 8 und 9: Komplette Schaltungen

Alles klappt? Herzlichen Glückwunsch!



### Quellenverzeichnis:

**Abb. 1** – Quelle: [wikipedia.org](http://wikipedia.org), Autor: Akimbomidget (CC BY-SA 2.5)

**Abb. 2** – Quelle: [wikipedia.org](http://wikipedia.org), Autor: Diliff (CC BY-SA 3.0)

**Abb. 7**, , – Quelle: Screenshots der Fritzing-Software (<http://fritzing.org>)

**Abb. 3, 4, 6, 8, 9, 10** – Quelle: InfoSphere

**Abb. 5** – Quelle: [wikipedia.org](http://wikipedia.org), Autor: Quark67 (CC SA 3.0)

**Screenshots der Programmelemente aus ArduBlock**

Alle weiteren Grafiken/Icons – Quelle: InfoSphere