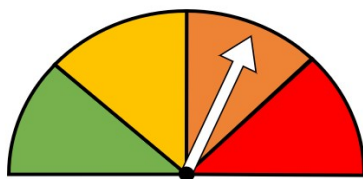


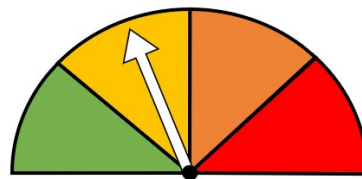


MicroPython

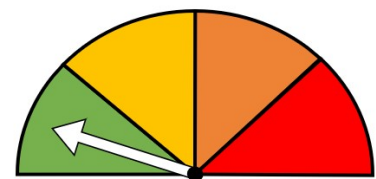
Station 4 | Farbthermometer



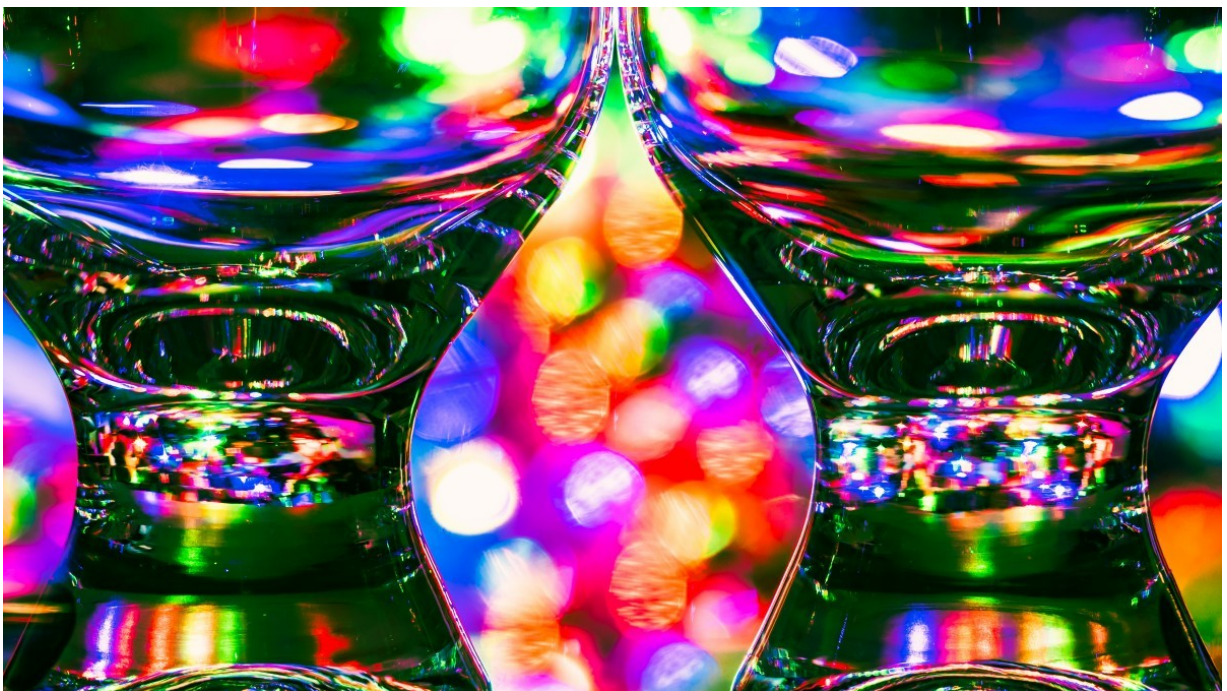
algorithmisches Denken



Programmieraufwand



Komplexität der Schaltung






Aotaro, „The Beauty of Bokeh“, pxhere.com , CC-BY 2.0

UM WAS WIRD ES IN DIESER STATION GEHEN?

Wie viele Menschen haben sich schon die Finger verbrannt, weil die Herdplatte noch heiß war? Oder haben sich am heißen Essen den Mund verbrannt? Das alles passiert nur weil der Mensch Temperatur nicht sehen kann. Das könnt ihr jetzt ändern! Eure Aufgabe heute ist es, ein Thermometer zu bauen, das mit verschiedenen Farben die Temperatur anzeigt.

BENÖTIGTE BAUTEILE

Zusätzlich zum ESP32-Board und dem Steckbrett brauchst du folgende Bauteile:

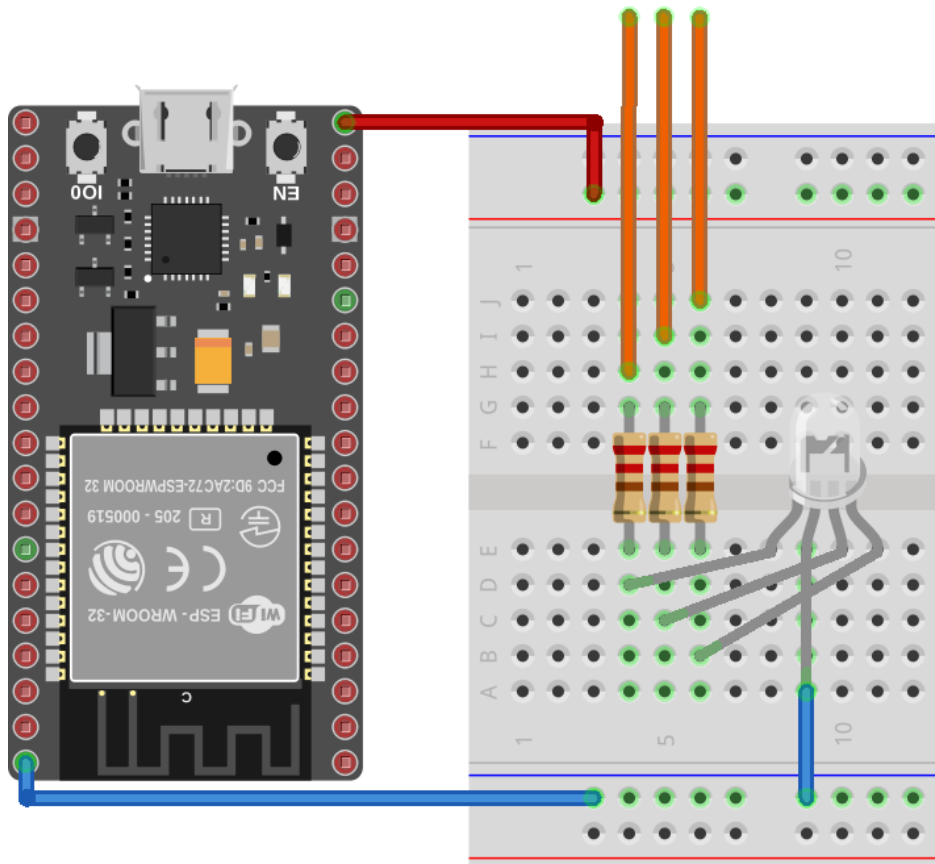
3 Widerstände	Widerstände schützen Bauteile vor Überhitzung durch zu viel Strom. Beachte die Farbe des Widerstandes, er sollte rot-braun-gold sein.	
Temperatur-Sensor	Der Temperatursensor misst die aktuelle Temperatur.	
RGB-LED	Die RGB-LED besteht aus einer roten, einer grünen und einer blauen LED. Sie kann in unterschiedlichen Farben leuchten.	



AUFGABE 1 – FARBEN MISCHEN

Als erstes bringst du die RGB-LED zum Leuchten und mischst verschiedene Farben.

1. RGB-LED anschließen



Versorge das Steckbrett wieder mit Strom, indem du die äußeren Leisten mit dem + Pol und – Pol verbindest. Verbinde das längste Beinchen der RGB-LED mit dem – Pol. Verbinde die restlichen Beinchen über einen Widerstand mit einem Kabel. Lasse die Enden der Kabel noch an einem Ende frei hängen.

i WAS IST EINE RGB-LED?

RGB-LEDs sehen zwar aus wie normale LEDs (die du ja aus dem Einstiegsprojekt kennst), aber in Wirklichkeit verstecken sich darin drei verschiedene LEDs. Nämlich eine rote, eine grüne und eine blaue LED, daher auch der Name RGB-LED. Das längste Beinchen der RGB-LED sind die Minuspole der einzelnen LEDs, die schon im Gehäuse miteinander verbunden sind. Die anderen drei Beinchen sind die + Pole der einzelnen LEDs. Wenn nicht nur eine LED leuchtet, sondern zum Beispiel die rote und grüne LED gleichzeitig, entsteht eine neue Farbe. Das kannst du dir vorstellen, wie den Farbkasten in der Schule. Welche Farben du mischen kannst, sollst du gleich herausfinden.

2. Farben mischen

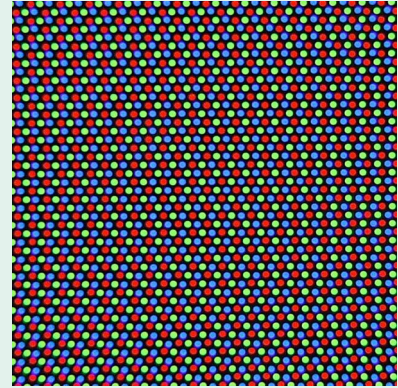
Stecke die frei hängenden Kabel nacheinander in die äußere Leiste, die mit dem + Pol verbunden ist. Welche Farbe nimmt die RGB-LED an? Du kannst die Farben auch mischen, indem du mehrere Kabel mit dem + Pol verbindest. Schreibe deine Ergebnisse auf.



Kabel	Farbe
links	
mitte	
rechts	
links + mitte	
links + rechts	
rechts + mitte	
links + rechts + mitte	

i FARBMISCHUNG DER RGB-LED

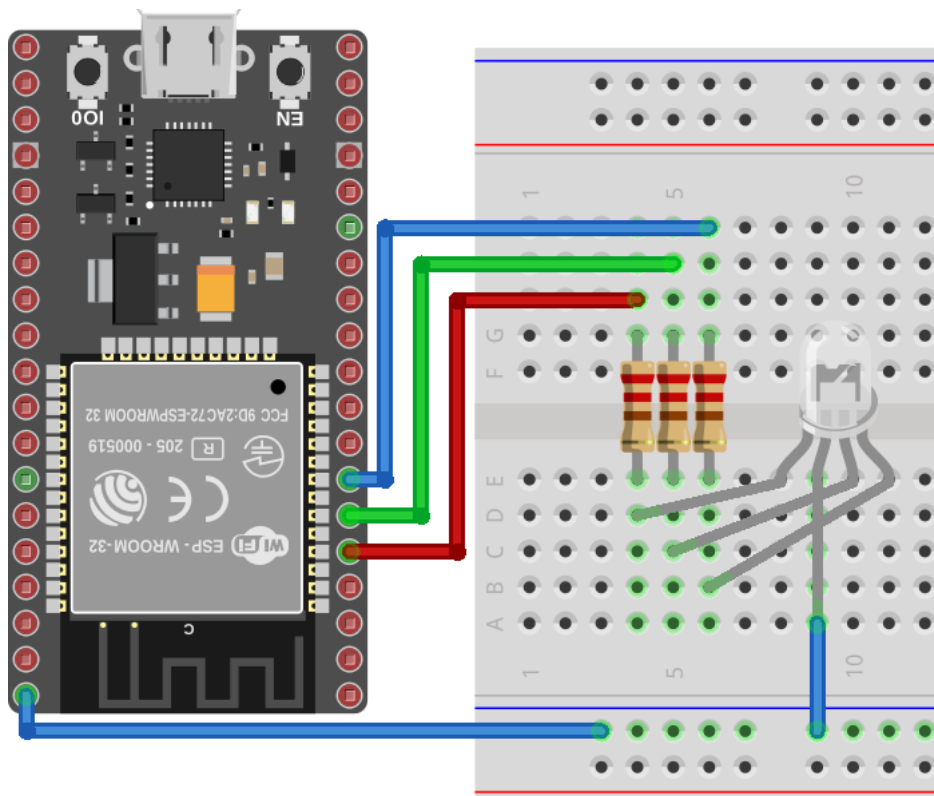
Das ist aber komisch! Hast du schon einmal gesehen, dass die Mischung von mehreren Farben weiß ergibt? Tatsächlich haben LEDs eine andere Farbmischung, als du sie vielleicht von einem Malkasten kennst. Sie heißt **additive Farbmischung**. Diese nutzen auch Fernseher und andere Bildschirme. Wenn du einen Bildschirm aus der Nähe betrachtest, kannst du ganz viele rote, grüne und blaue Bildpunkte sehen. Alleine aus diesen Farben können alle weiteren Farben gemischt werden.



Marcin Floryan, „A close-up on a CRT screen“, wikipedia.org, gemeinfrei

3. RGB-LED an analoge Pins schließen

Wenn du für die Temperaturen unterschiedliche Farben verwenden willst, ist es unpraktisch immer die Kabel umstecken zu müssen. Verbinde die drei Kabel der LEDs deswegen mit den digitalen Pins 32, 34 & 35, um sie mit dem Board anzusteuern.





AUFGABE 2 - DISKOLICHT

Nachdem du die Farben schon durch Umstecken der Kabel gemischt hast, sollst du die Farben jetzt durch das Board steuern. Dazu sollst du ein kleines Diskolicht programmieren, das die Farben automatisch ändert.

1. Module importieren

Um auf die Pins am ESP32-Board zugreifen zu können, benötigt MicroPython die Bibliothek `machine`. Um steuern zu können, wie schnell das Programm abläuft, solltest du zusätzlich die Funktion `sleep` aus der Bibliothek `time` verwenden. Weißt du noch, wie du Module importieren kannst? Falls nicht, schaue einfach nochmal in der 2. Aufgabe der Einführungsstation nach.

2. Pin-Variable definieren und digitalen Pin verbinden


Wie in der Einführungsstation musst du auch hier den ESP32 wissen lassen, an welchen Pins die LEDs angeschlossen sind. Definiere dafür drei Variablen. Am besten ist es, den Variablen den Namen der Farbe zu geben, die an dem Pin angeschlossen ist. Beachte aber, dass MicroPython keine Umlaute Ä, Ö, Ü versteht!

Außerdem muss das Board wissen, dass an dem Pin etwas angeschlossen ist, das angeschaltet werden soll. Lege deswegen den Pinmodus auf `Pin.OUT` fest.

Falls du dafür Hilfe brauchst, schaue dir noch einmal in der Einstiegsstation Aufgabe 3 Punkt 2 an.

3. LEDs anschalten

Die Befehle zum An- und Ausschalten einer LED kennst du bereits. Wie lauten sie?

 Hier muss stehen, welche LED gemeint ist.

Einschalten: _____ .value(_____)

Ausschalten: _____ .value(_____)

Hier soll der Befehl zum Ein- oder Ausschalten stehen, also 1 oder 0.

Schalte mithilfe der Befehle erst eine LED an, dann aus. Als nächstes soll eine andere LED leuchten. Bedenke wieder, dass das Board sehr schnell arbeitet. Er muss also eine Pause machen, nachdem die LED angeschaltet

wurde, damit man sie auch sehen kann. Wie lautet noch einmal der Befehl für die Pause?



4. Farben mischen

Wenn das Einschalten der LEDs funktioniert hat, dann kannst du noch mehr Farben für das Diskolicht programmieren. Erstelle die Lichtabfolge rot – grün – blau – gelb – pink – türkis – weiß. Schalte für die gemischten Farben mehrere LEDs gleichzeitig an und dann wieder aus. Wie die Farben gemischt werden, kannst du in deiner Tabelle von vorhin nachsehen.



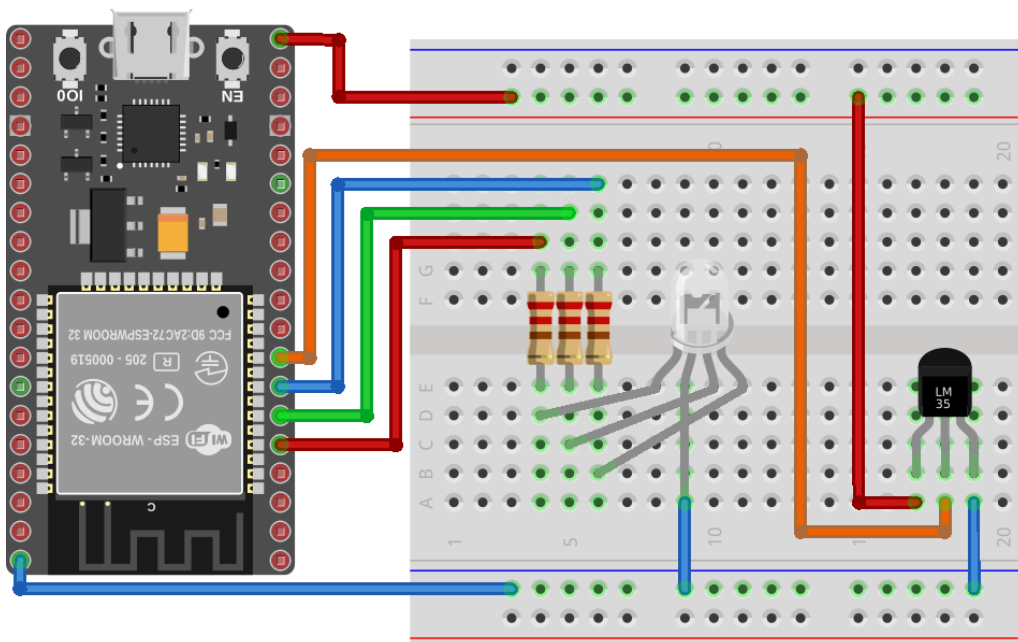
AUFGABE 3 – TEMPERATUR MESSEN

Super! Jetzt kannst du schon verschiedene Farben programmieren. Fehlt nur noch die Temperatur, der du eine Farbe zuweisen kannst. Als nächstes sollst du deswegen die Temperatur messen.

1. Neues Programm öffnen

Speicher dein Diskokugel-Programm unter dem Namen Diskokugel ab. Du wirst es später noch einmal brauchen. Jetzt sollst du aber erst einmal ein neues Programm anfangen. Klicke dafür in der Programmierumgebung auf `Datei` und dann auf `New`.

2. Temperatursensor anschließen



Wenn die flache Seite des Temperatursensors zu dir zeigt, verbinde das linke Beinchen mit dem + Pol und das rechte Beinchen mit dem – Pol. Das mittlere Beinchen verbindest du mit einem Pin am Board.

3. Pin-Variable definieren

Auch für den Temperatursensor musst du eine Variable definieren, die speichert, an welchem Pin der Sensor angeschlossen ist. Nenne sie am besten `messwert` und weise ihr den Pin zu, den du mit dem Mess-Beinchen des Temperatursensors verbunden hast.

4. Messwerte umwandeln

Der Temperatursensor gibt analoge Messwerte an den ESP32 weiter. Damit er sie richtig interpretieren kann, musst du sie in digitale Werte umwandeln. Nutze hierfür den `ADC()`-Befehl.

Die Variable, die du zum Messen der Temperatur verwendest

_____ = ADC (Pin (_____))

5. Messeinstellungen

Um den vollständigen Messbereich (0 – 3,3 Volt) des ESP32 nutzen zu können, musst du folgenden Befehl ergänzen:

Die Variable, die du zum Messen der Temperatur verwendest

Erweiterung des Spannungsmessbereichs auf 3,3V



_____ .atten(ADC.ATTN_11DB)

6. Daten erfassen

Nutze den `read()`-Befehl, um die Temperaturmessung durchzuführen:

Die Variable, die du zum Messen der Temperatur verwendest



_____ .read()

7. Daten ausgeben lassen und Programm verlangsamen

Um die gemessenen Werte auf der Konsole ausgeben zu lassen, kannst du einfach den `print()`-Befehl verwenden. Überlege, welche Variable deines Programms in die Klammern gehört. Integriere den Befehl in eine `while True` – Schleife, um die Messung wiederholt auszuführen.

Du wirst schnell merken, dass das Board in kurzer Zeit sehr viele Messwerte aufnimmt. Es ist daher sinnvoll, zwischen den Messungen kurze Pausen einzufügen. Erinnerung dich – wie lautet der Befehl für *Pause*?

Falls es Probleme gibt, schaue einfach nochmal in der 4. Aufgabe der Einführungsstation nach.



AUFGABE 5 - ERGEBNISSE UMRECHNEN

Hmm... Irgendwie wirken die Messergebnisse komisch. Tatsächlich misst der Temperatursensor nicht in °C, sondern in Spannung. Mit der Zahl können wir aber nichts anfangen. Deswegen ist deine nächste Aufgabe, das Messergebnis in °C umzurechnen.

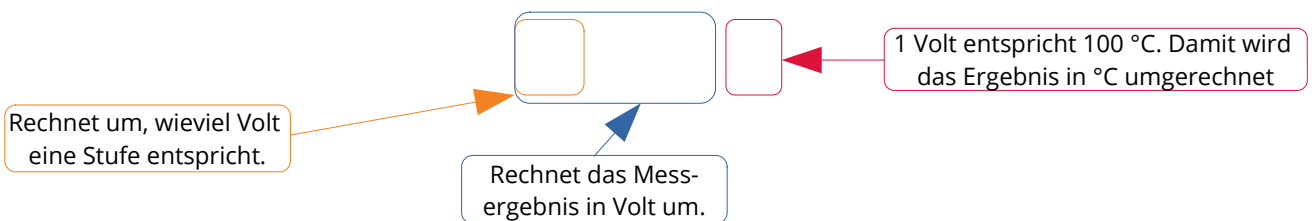


WIE FUNKTIONIERT EIN TEMPERATURSENSOR?

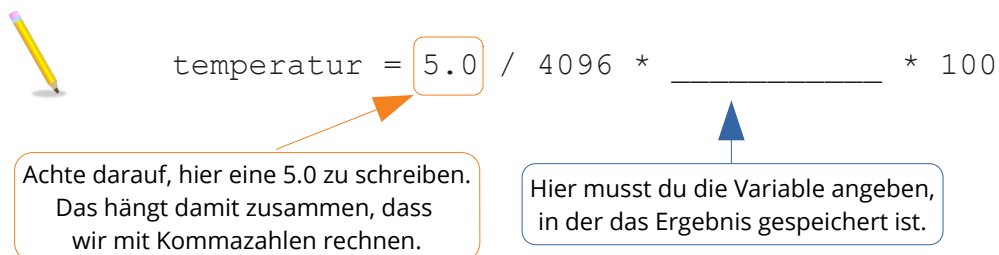
Der Temperatursensor ist mit dem ESP32 über einen Schaltkreis verbunden. Die Kabel sind die einzige Verbindung über die sie Informationen austauschen können. Der Sensor schickt das Ergebnis seiner Messung als elektrisches Signal. Je höher die Temperatur ist, desto größer ist die Spannung auf dem Kabel. Die Spannung ist zwischen 0 und 5 Volt groß. Das Board misst die Spannung am Pin. Es muss die Information in eine Form bringen, die für den Computer geeignet ist. Deswegen speichert er das Ergebnis in einer von 4096 Stufen. Wenn du eine 0 auf der Konsole als Ergebnis bekommst, heißt das, es ist 0 °C. Eine 4095 wäre 500°C.

1. Umrechnen

Die Formel zum Umrechnen ist folgende:



Erstelle eine Variable für die Temperatur berechne ihren Wert mithilfe der Temperaturformel:



2. Temperatur drucken

Bisher steht noch im Code, dass das Messergebnis (in Stufen) gedruckt werden soll. Ändere den Code so ab, dass stattdessen die Temperatur in °C gedruckt wird.

$$Temperatur = \frac{5}{4096} \cdot messwert \cdot 100$$



AUFGABE 6 - FARBEN FÜR TEMPERATUREN AUSWÄHLEN

Wenn die Temperatur in der Konsole erscheint, kannst du mit der letzten Aufgabe beginnen. Jetzt sollst du die Farben für die Temperaturen auswählen.

1. Zuordnung der Farben

Mache dir zuerst einen Plan, bei welcher Temperatur welche Farbe leuchten soll. Es wird schwierig sein, eine genaue Temperatur zu messen. Gebe deswegen in der Tabelle an, bis wieviel Grad die Farbe gelten soll. Fange in der Tabelle links mit der kältesten Temperatur an. Lasse sie Schritt für Schritt wärmer werden. Für Farbe 5 musst du keine Maximaltemperatur angeben. Warum, siehst du im nächsten Schritt. Zur Erinnerung, dir stehen die Farben rot - grün - blau - gelb - pink - türkis - weiß zur Auswahl.



	Farbe 1	Farbe 2	Farbe 3	Farbe 4	Farbe 5
gilt für Temperatur bis ___ °C					X
Farbe					

2. Wenn - Ausdrücke

Um deine Liste zu programmieren brauchst du **Wenn** und **Sonst, wenn** - Ausdrücke. Auf gut Deutsch:

Wenn Temperatur kleiner ist, als ___ Grad, soll Farbe _____ leuchten.
Sonst, wenn die Temperatur kleiner ist, als ___ Grad, soll Farbe _____ leuchten. **Sonst** soll Farbe _____ leuchten.

Füge die Ausdrücke in die `while True` - Schleife ein, nachdem die `temperatur` berechnet wurde.

```

if ( temperatur < _____ ) :
    Farbe 1 leuchtet
elif ( temperatur < _____ ) :
    Farbe 2 leuchtet
else:
    Farbe 5 leuchtet

```

Füge hier die kleinste Temperatur deiner Tabelle ein

Und hier die Temperatur der Farbe 2

Für die anderen Farben kannst du an dieser Stelle weitere **Sonst, wenn** - Ausdrücke hinzufügen. Die Farbe im letzten Sonst-Ausdruck leuchtet, wenn die Temperatur größer ist, als alle anderen Zahlen. Deswegen brauchst du hier keine Maximaltemperatur angeben.

3. Farb-Befehle

In dem Code zum Diskolicht, hast du schon das Leuchten der Farben programmiert. Kopiere aus dem Diskolicht-Code die Variablen-Definitionen und die Befehle zum Pinmodus. Füge sie an den richtigen Stellen im Programm des Temperaturmessens ein.

Kopiere auch den Befehl, der Farbe 1 aus der Tabelle leuchten lässt (zusammen mit der Pause und dem Ausschalten). Setze ihn anstelle des Texts „Farbe 1 leuchtet“ im **Wenn**-Ausdruck ein. Wiederhole das für die anderen Farben und **Sonst, wenn** - Ausdrücke.

4. Blinken stoppen

Wenn du jetzt das Programm installierst, wird das Licht blinken. Das liegt an der Pause, die vorher dafür gesorgt hat, dass nicht zu viele Messergebnisse auf einmal in der Konsole erscheinen. Durch das Anschalten des Lichts, hast du eine zweite Pause dem Code hinzugefügt, sodass die alte Pause nicht mehr nötig ist. Lösche sie deswegen aus dem Code.

Fotos: RWTH Aachen, InfoSphere

Screenshots: fritzing electronics made by easy und Arduino IDE 1.8.12 (windows)

Alle weiteren Grafiken: Patrick Binkert, EduInf@TUD

5. Programm installieren und testen

Installiere das Programm auf dem ESP32. Teste es, indem du die Temperatur des Sensors veränderst. Die Farbe des Lichts sollte sich verändern. Falls nicht, kannst du versuchen die Maximaltemperaturen der Farben zu verändern.

Super! Mithilfe deines Programms kann man jetzt Temperaturen sehen.

