

Station 4 – Color Thermometer – Bonus 2

New Challenges

You want new challenges? You can get it! Somewhere, we have hidden the biggest challenge of the InfoSphere.

So far, you always check, if the temperature is in a certain range, and then you give the RGB LED a previously programmed color. This works perfectly, but it also leads to quite hard transitions between the colors. But there is another way. Knowing the temperature, you can calculate a suitable color. How do you calculate the color? That is easy, because of the fact, that you can control the red, green and blue LED separately. While the red part increases more and more, the blue part decreases with increasing temperature.

- Are you good at math? Then you can start calculating, and figure out, how to approach the problem.
- You are good at math, but like real programmers you are too lazy to calculate yourself? Below, you will find two functions. But beware: They can do the calculation for you, but you still have to think on your own. 😊



The following examples are based on the German grading system. In this system, a 1 means a very good performance. A 6 means a very poor performance.

The Map Function

Basically, the `map()` function is a mathematical function, which converts a value from one range of values to another. Example: You write a class assignment at school. In total, there are 40 points to get. But your teacher has an evaluation scheme in percent. Therefore, she uses the following function to convert the `number-of-points` (0-40) in percent (0-100):

```
map(number-of-points, 0, 40, 0, 100);
```

For example, if you enter 20 as the number of points, the function will return the value 50. This also works the other way around:

```
map(number-of-points, 0, 40, 6, 1);
```

Here, you will get a 6 for 0 points, and a 1 for 40 points.

How does that help you?

If the red LED should still be off at 20 °C (i.e. has the value 0), but should shine as bright as possible at 25 °C (i.e. 255), you can use the following command:

```
map(temperature, 20, 25, 0, 255);
```

The Constrain Function

`map()` is a really great function. However, it has one disadvantage: If you have studied well for your class assignment, do everything right, and on top of that get 10 bonus points for beautiful writing (so you have 50 points) `map(number-of-points, 0, 40, 6, 1);` might output a 0 or even a -1. This is where `constrain()` comes in. This function makes sure, that a number stays within certain limits. This way, the result of

```
constrain(value, a, b);
```

always is $\geq a$ and $\leq b$.

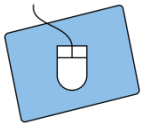
Your teacher can still correct the calculated grade.

```
real-grade = constrain(calculated-grade, 1, 6);
```

Station 4 – Color Thermometer – Bonus 2

So, the 0 becomes a 1 again.

For you, this means the following: You have learned, that the values for `analogWrite` must be between 0, and 255. `constrain` can do this for you.

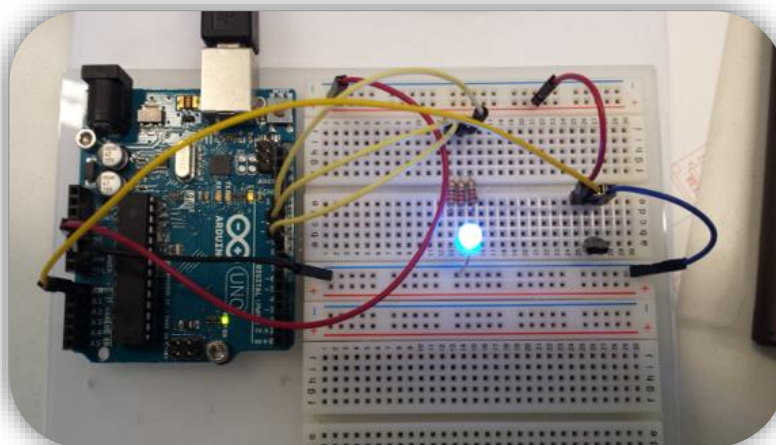


Use the `map`, and the `constrain` function for your color thermometer.

Once, you have mastered this challenge, too, you can move on to your own challenges. Here, you can find a few ideas on how you could continue:

- other temperature ranges: Is there a hair dryer or fan heater nearby? Try to adjust your circuit accordingly.
- variable temperature range: In your InfoSphere Kit, you can find a button in the material box of station 0. Can you use it to switch the temperature range at the push of a button, so that temperatures in the range of 20 to 25 °C, and temperatures above 30 °C are converted into colors?

Have fun while tinkering!



[1]



List of references:

Fig. 1, , ,  – Source: InfoSphere