These slides serve as a visual aid for the lecture, not as a comprehensive document or script.

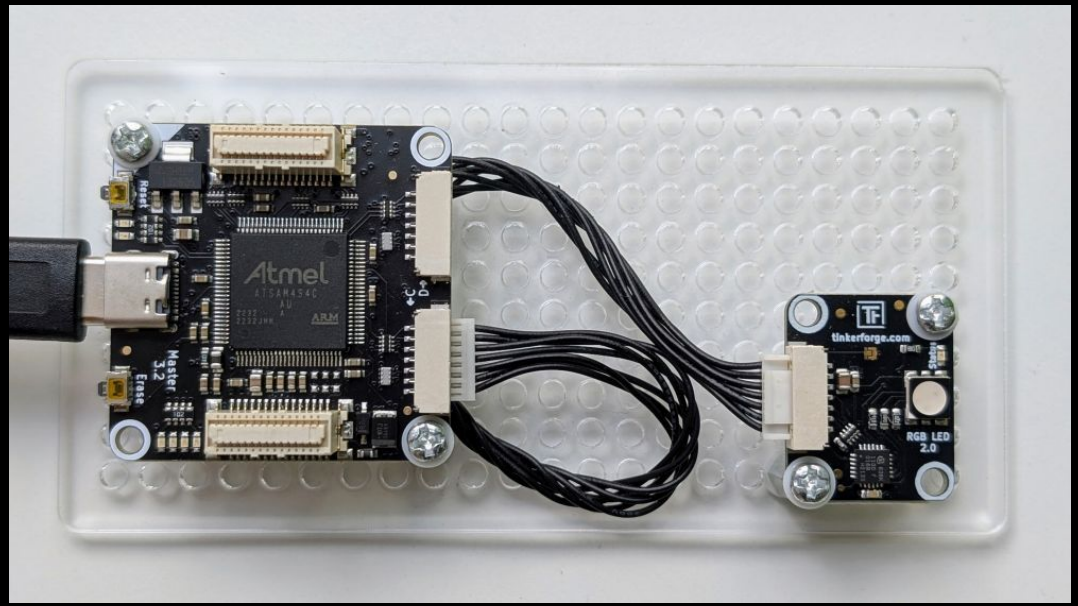
Please refrain from printing these slides to help protect the environment.

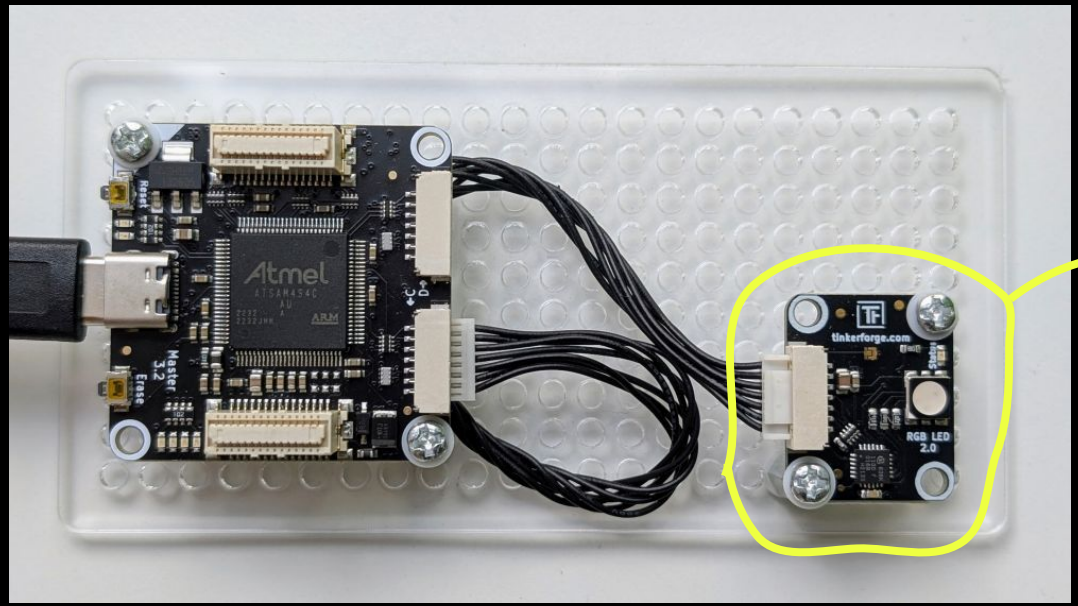
For any comments or feedback, please contact n.meseth@hs-osnabrueck.de.

COLORS

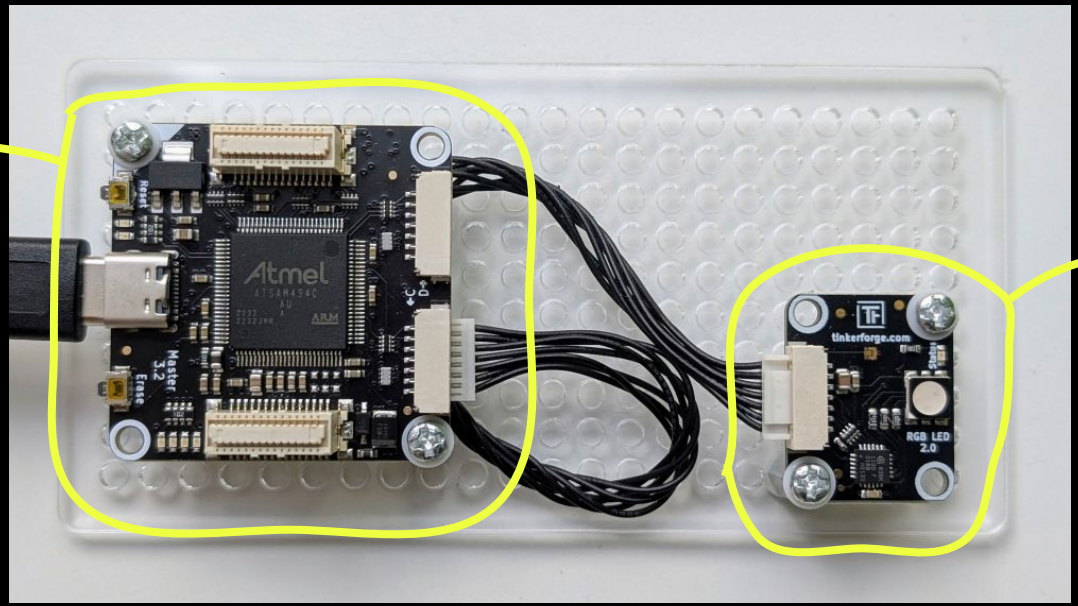
Supporting slides for chapter 1 of the book
Hands-On Computer Science

EXPERIMENT SETUP

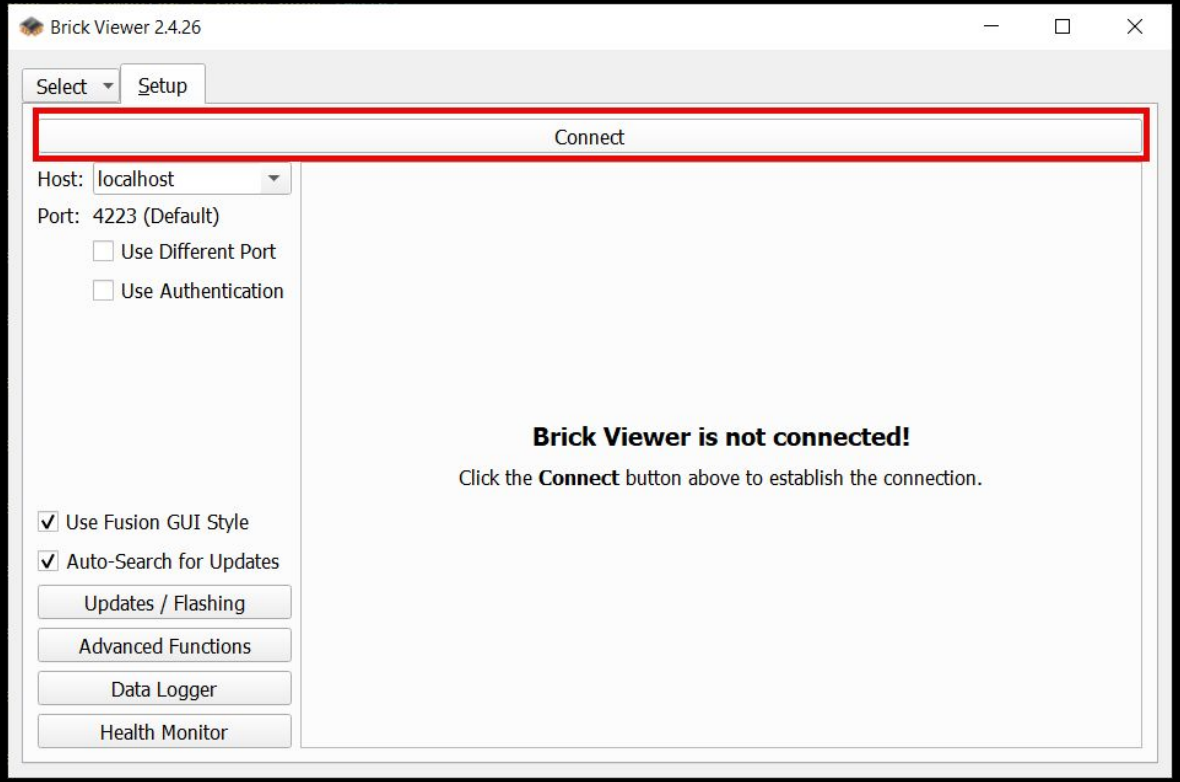

Atmel
ATSAM4S4C
AU
ARM
Reset
Master
3.2
Erase
tinkerforge.com
RGB LED
2.0

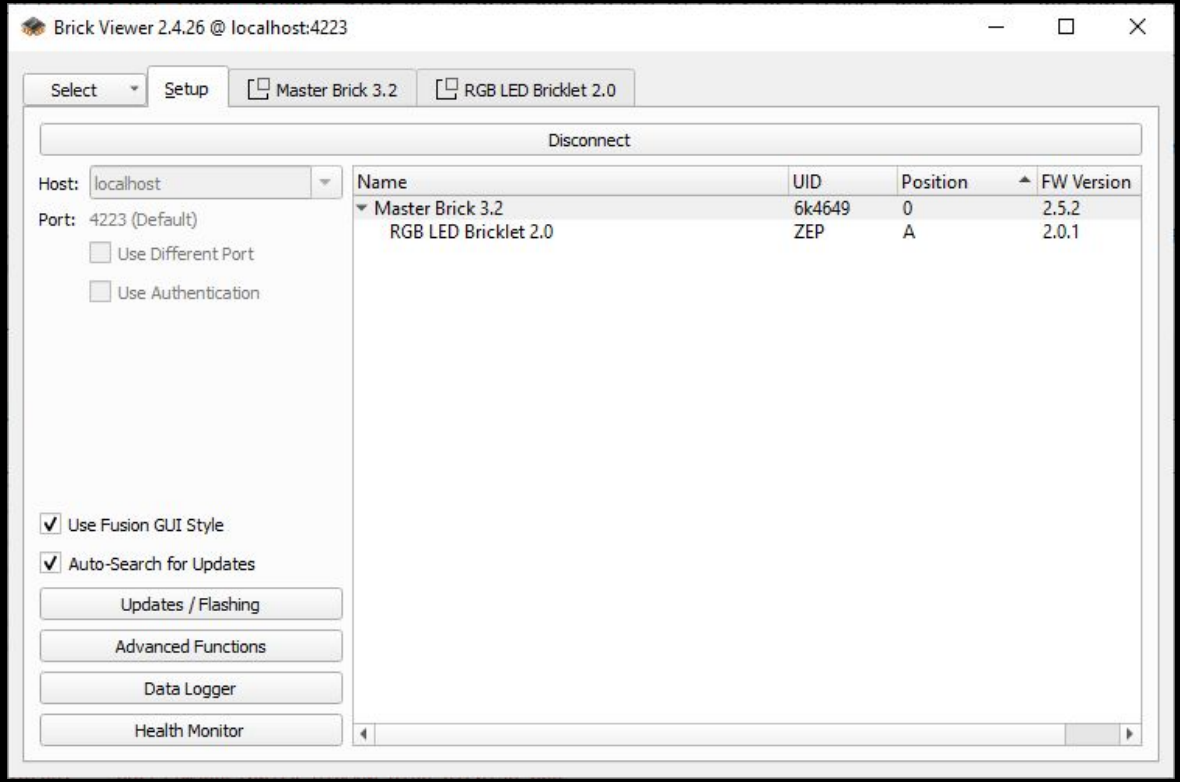

RGB LED
Atmel
tinkerforge.com
RGB LED 2.0

Master Brick
(Microcontroller)



RGB LED


Brick Viewer 2.4.26
Select
Setup
Connect
Host: localhost
Port: 4223 (Default)
Use Different Port
Use Authentication
Brick Viewer is not connected!
Click the Connect button above to establish the connection.
Use Fusion GUI Style
Auto-Search for Updates
Updates / Flashing
Advanced Functions
Data Logger
Health Monitor


Brick Viewer 2.4.26 @ localhost:4223
Select
Setup
Master Brick 3.2
RGB LED Bricklet 2.0
Disconnect
Host:
localhost
Port:
4223 (Default)
Use Different Port
Use Authentication
Name
UID
Position
FW Version
Master Brick 3.2
6k4649
0
2.5.2
RGB LED Bricklet 2.0
ZEP
A
2.0.1
Use Fusion GUI Style
Auto-Search for Updates
Updates / Flashing
Advanced Functions
Data Logger
Health Monitor


Brick Viewer 2.4.26 @ localhost:4223
Select
Setup
Master Brick 3.2
RGB LED Bricklet 2.0
UID: ZEP
FW Version: 2.0.1
Timeouts: 0
Status LED:
Show Status
Reset
More
R:
0
G:
255
B:
0
H:
120
S:
255
L:
128
Black
White
Red
Yellow
Green
Cyan
Blue
Magenta

FIRST PROGRAM

boilerplate code

```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```

libraries or modules

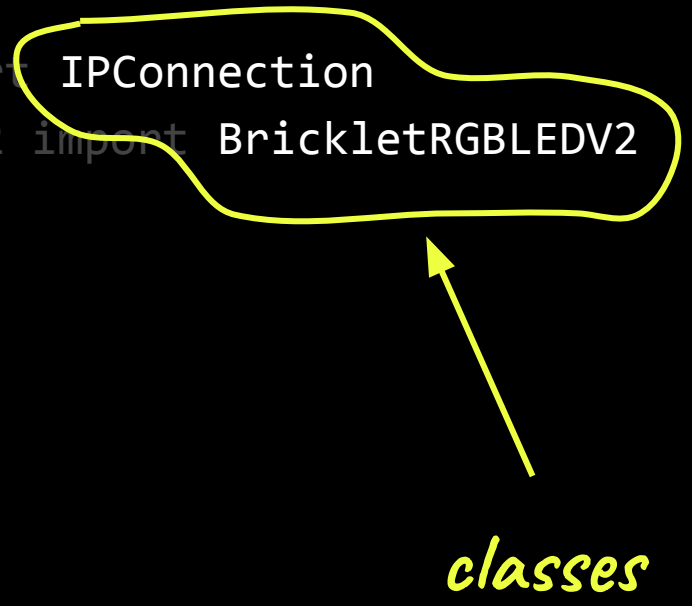
```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```

classes and objects

```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```

classes and objects

```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```



classes and objects

```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```

instantiate objects from classes

classes

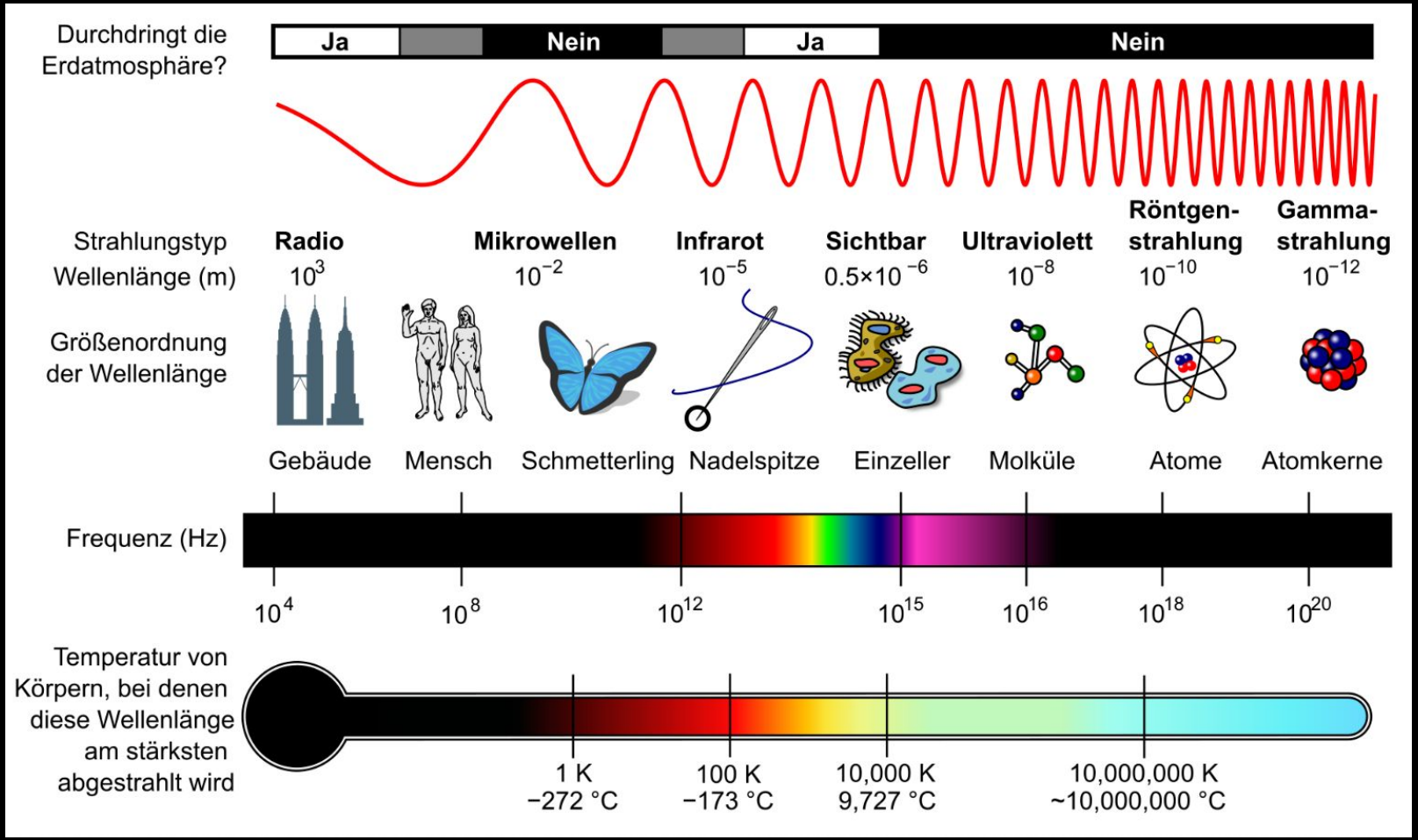
reserved keywords

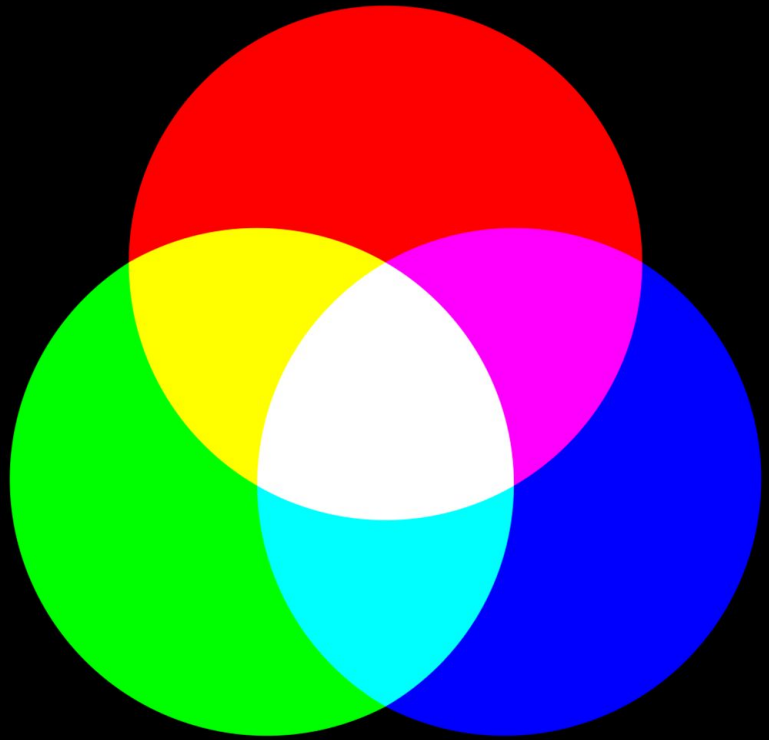
```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```

methods

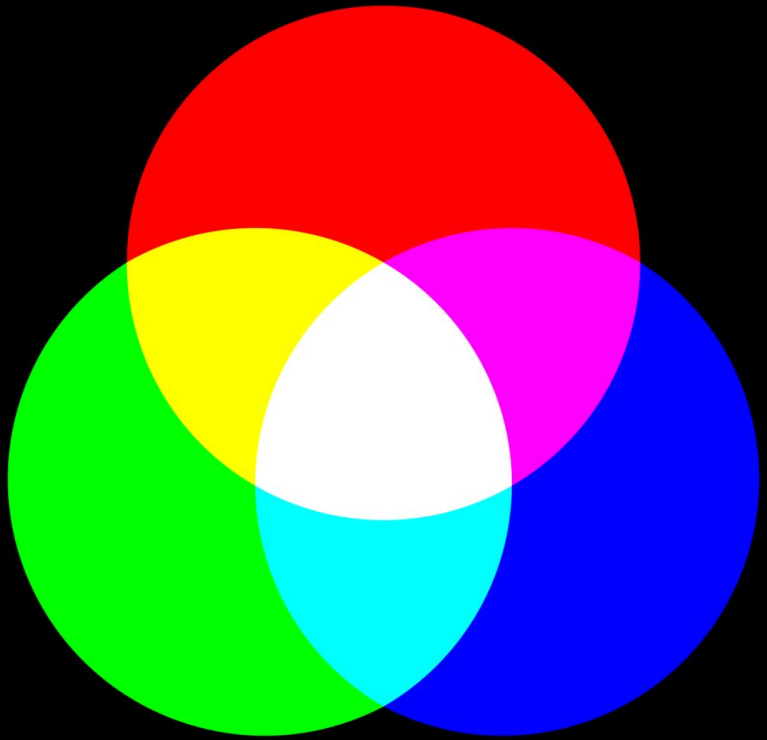
```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rgb_led_v2 import BrickletRGBLEDV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
led = BrickletRGBLEDV2("ZEP", ipcon)
led.set_rgb_value(0, 255, 0)
```

LIGHT AND COLORS

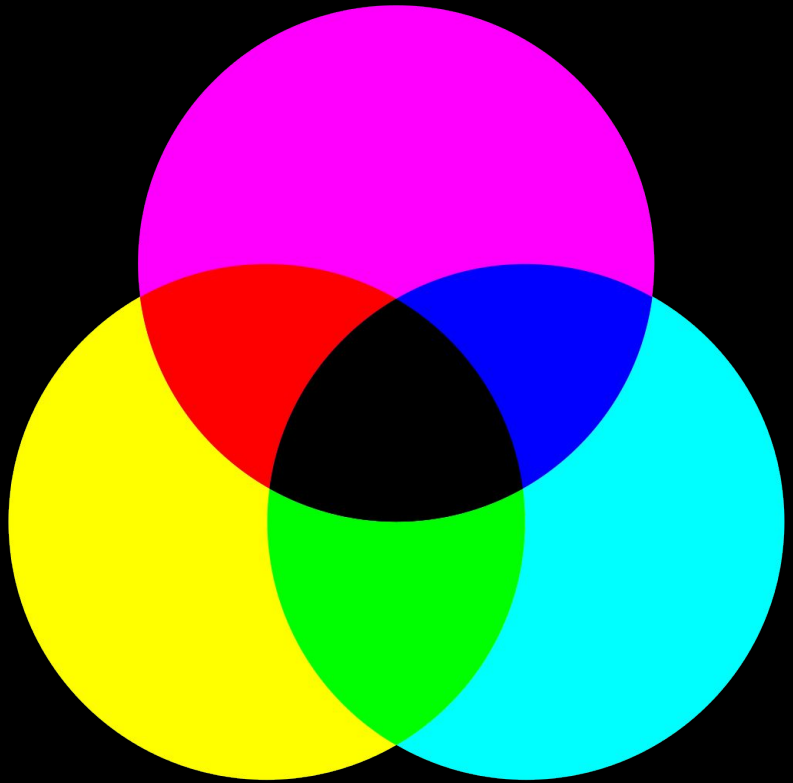

Durchdringt die Erdatmosphäre?
Ja
Nein
Ja
Nein
Strahlungstyp
Wellenlänge (m)
Größenordnung der Wellenlänge
Radio
10^{3}
Mikrowellen
10^{-2}
Infrarot
10^{-5}
Sichtbar
0.5×10^{-6}
Ultraviolett
10^{-8}
Röntgen-strahlung
10^{-10}
Gamma-strahlung
10^{-12}
Gebäude
Mensch
Schmetterling
Nadelspitze
Einzeller
Molküle
Atome
Atomkerne
Frequenz (Hz)
10^{4}
10^{8}
10^{12}
10^{15}
10^{16}
10^{18}
10^{20}
Temperatur von Körpern, bei denen diese Wellenlänge am stärksten abgestrahlt wird
1 K
−272 °C
100 K
−173 °C
10,000 K
9,727 °C
10,000,000 K
~10,000,000 °C



additive color mixing



additive color mixing



subtractive color mixing


KY-TK5150BK
KY-TK5150C
KY-TK5150M
KY-TK5150Y

PULSATING LED

FOR-LOOP

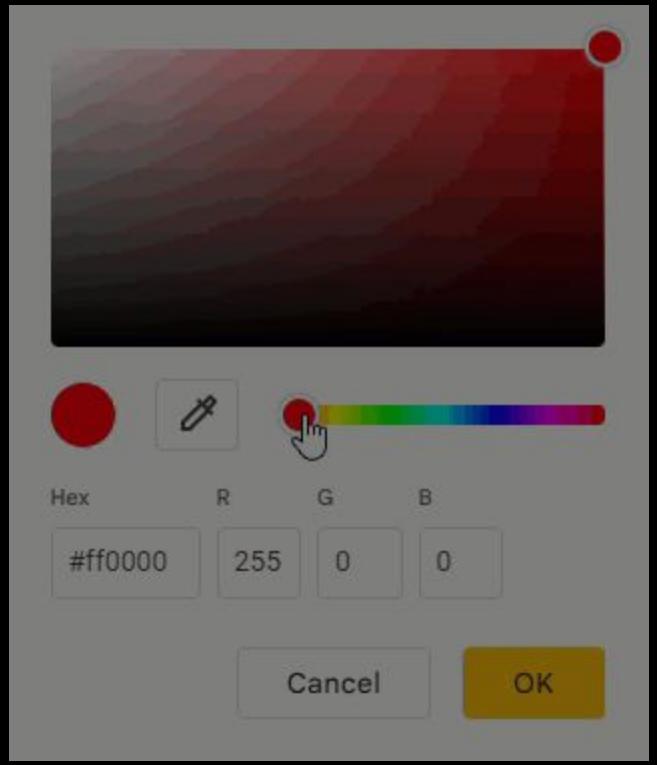
```
for r in range(256):
    led.set_rgb_value(r, 0, 0)
```

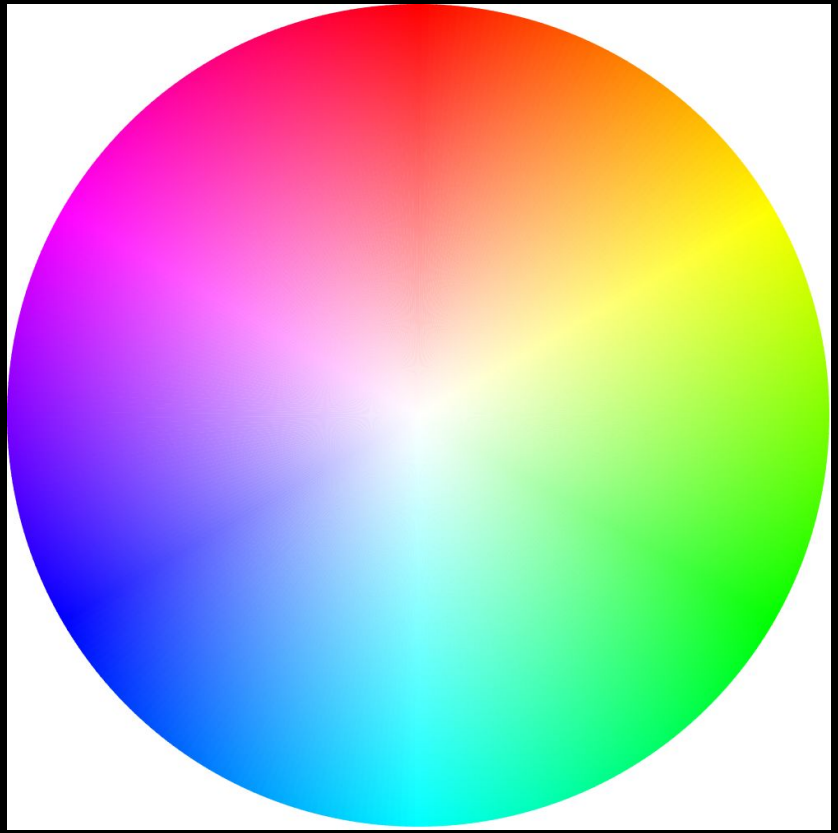
```
time.sleep(0.001)
```

WHILE-LOOP

```
while True:
    print("I will loop forever")
    time.sleep(1)
```

COLOR CIRCLES


Hex
R
G
B
#ff0000
255
0
0
Cancel
OK



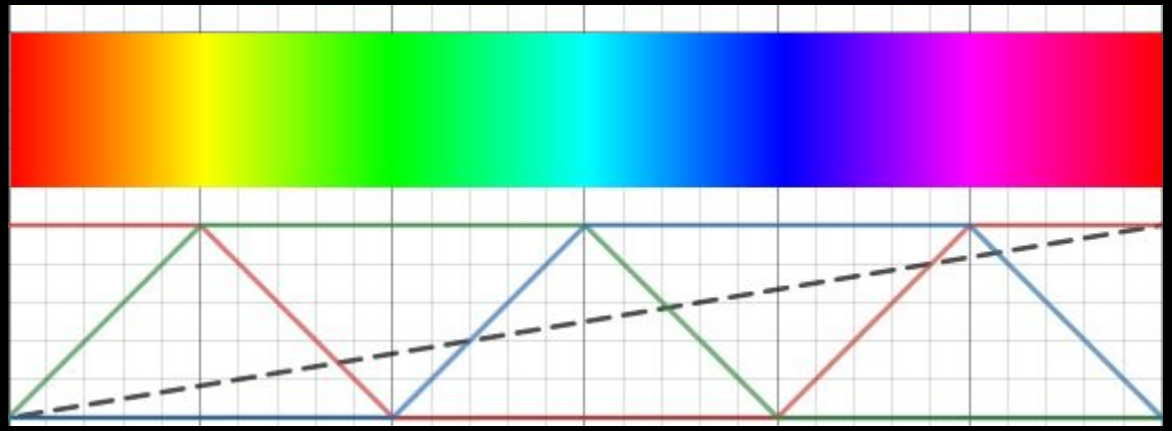
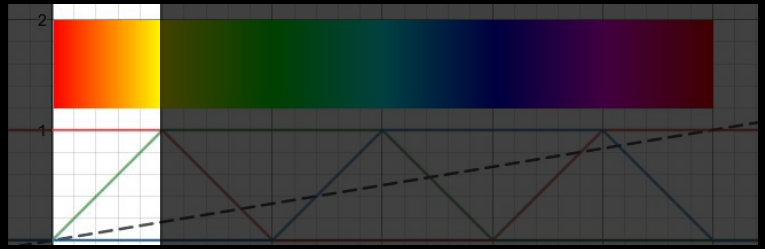
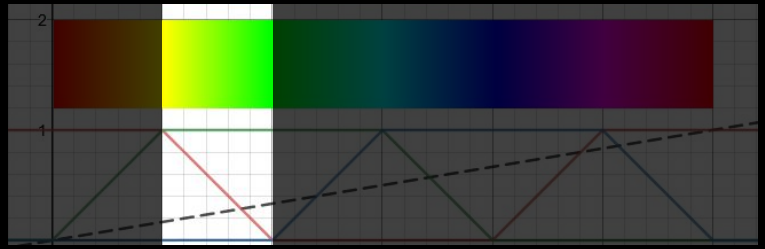


Image Source:https://www.ronja-tutorials.com/post/041-hsv-colorspace/

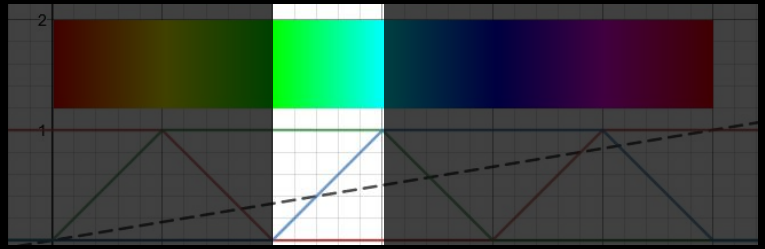
RAINBOW LED



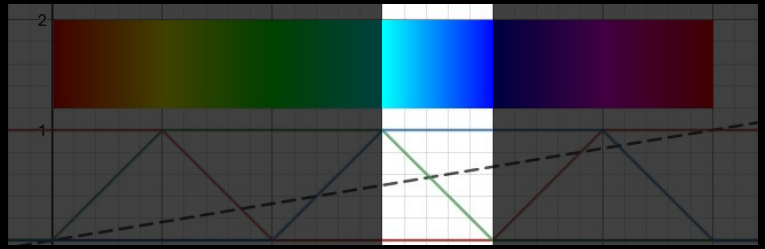
```
# phase 1: red = 255, blue = 0
for green in range(256):
    led.set_rgb_value(255, green, 0)
```



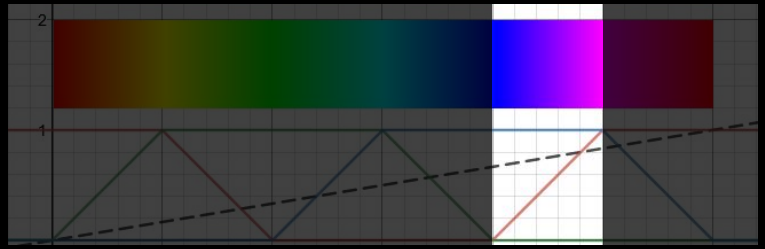

```
# phase 2: green = 255, blue = 0
for red in range(255, -1, -1):
    led.set_rgb_value(red, 255, 0)
```



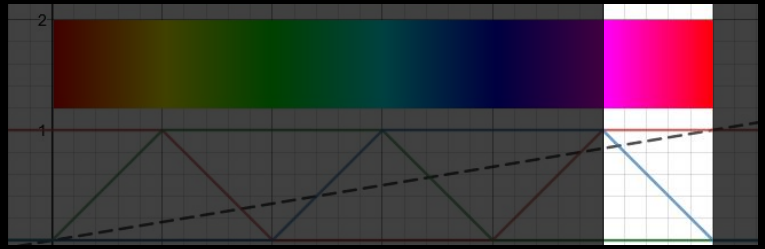

```
# phase 3: green = 255, red = 0
for blue in range(256):
    led.set_rgb_value(0, 255, blue)
```



```
# phase 4: blue = 255, red = 0
for green in range(255, -1, -1):
    led.set_rgb_value(0, green, 255)
```



```
# phase 5: blue = 255, green = 0
for red in range(256):
    led.set_rgb_value(red, 0, 255)
```




```
# phase 6: red = 255, green = 0
for blue in range(255, -1, -1):
    led.set_rgb_value(255, 0, blue)
```

CONTROLLING SPEED