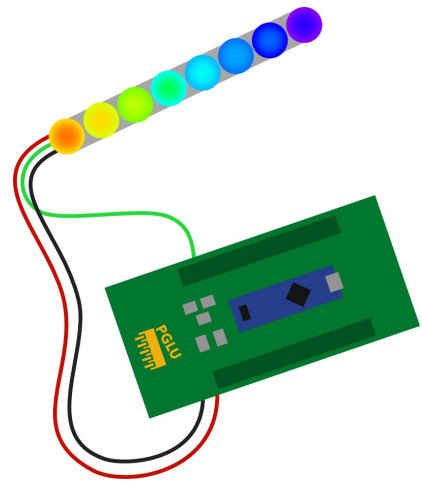


# 7



## Blubb Blubb - Video 7

Dieser Lichteffect ist perfekt zum Experimentieren, denn die Funktion INOISE8 steuert eine sogenannte PERLIN NOISE Funktion. Das ist eine Art weicher Zufallsgenerator, der wolkige oder blasenartige Lichteffecte schafft. Spiele mit den Zahlen!

**Blubb Blubb**

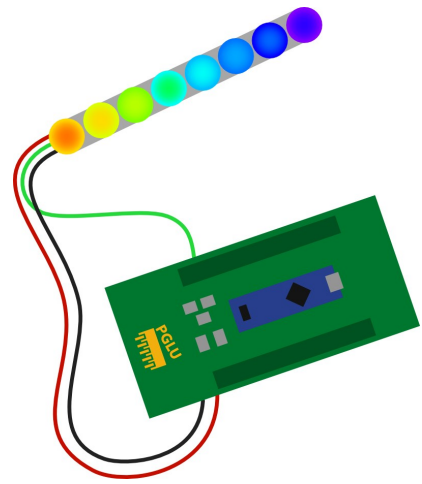
definiere LED-Strip Anzahl Pixel  Helligkeit total  % Ausgang

Hauptloop: 100'000 mal pro Sekunde Blinkcode: kurz 1 lang 1

- schreibe Weite =
- schreibe Tempo =
- zähle Pixel von  bis  alle
- schreibe Effekt =
- schreibe Farbe =  von Min  Max  zu Min  Max
- setze Zwischenspeicher ab Pixel  Anzahl  auf Farbe in  Sättigung in %  Helligkeit in %
- sende Pixel aus Zwischenspeicher an LED-Strip

ZWISCHENSPEICHER für die Darstellung mehrerer Farben gleichzeitig

# 8



## Feuer Simulator - Video 8

Ein Cheminée mit einem simulierten Feuer? Mit diesem Effekt leuchten deine Pixel wie eine digitale Kerze. Dafür verwendet dieses Programm wiederum den weichen Zufallsgenerator, gleich wie im Sketch Blubb Blubb.

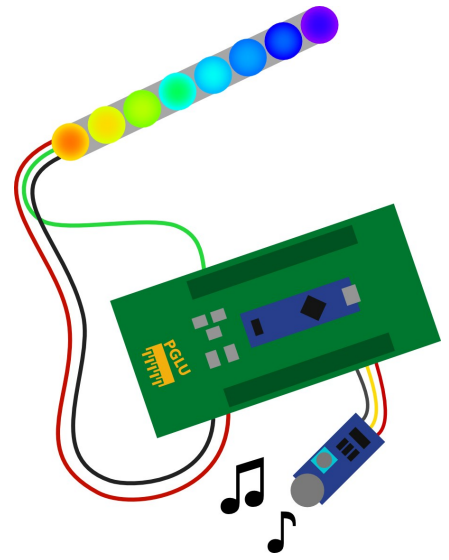
The screenshot shows the PGLU IDE interface with the 'FEUER SIMULATOR' code block. The code is written in a block-based language and includes the following elements:

- definiere LED-Strip:** Anzahl Pixel: 8, Helligkeit total: 100, % Ausgang: LED 1.
- Hauptloop:** 100'000 mal pro Sekunde, Blinkcode: kurz 1, lang 1.
- schreibe Weite = 60**
- schreibe Tempo = Zeit in ms \* 0.16**
- zähle Pixel von 1 bis 8 alle 0.05**
- schreibe Effekt = inoise8(Pixel\*Weite, Tempo)**
- schreibe Farbe = ändere Effekt von Min 50 Max 190 zu Min 18 Max 42**
- schreibe Helligkeit = ändere Effekt von Min 50 Max 100 zu Min 30 Max 80**
- setze Zwischenspeicher ab Pixel Pixel Anzahl 1 auf Farbe in Farbe Sättigung in 100 Helligkeit in Helligkeit**
- sende Pixel aus Zwischenspeicher an LED-Strip**

Annotations in the image include:

- Yellow arrows pointing to the 'zähle Pixel' and 'schreibe Effekt' blocks.
- A yellow box around the 'setze Zwischenspeicher' block with the text: "ZWISCHENSPEICHER für die Darstellung mehrerer Farben gleichzeitig".

# 9



## Musik 🎵 - Video 9

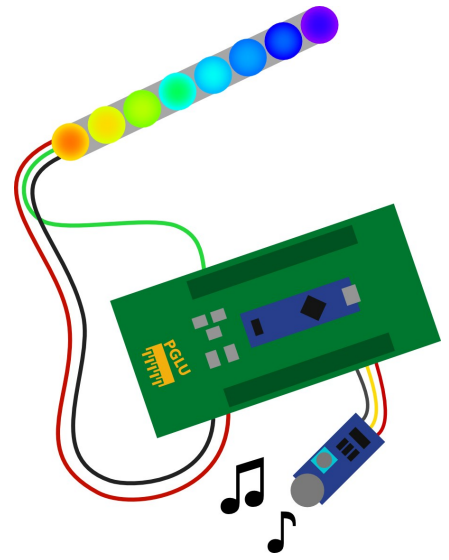
Steuere deine Pixel mit der Musik, die dein Soundsensor hört. Denn damit Gestaltest du eine interaktive Partylampe, eine Möbelbeleuchtung oder irgend etwas anderes. Hauptsache, die Pixel reagieren auf deinen Sound!

**WICHTIG:** Stelle die Empfindlichkeit des Soundsensors am kleinen Drehregler auf dem blauen Bauteil des Sensors sorgfältig ein.

The screenshot shows the PGLU IDE interface with the following code blocks:

- Aktion:** A 'MUSIK!' block.
- LED:** 'definiere LED-Strip' block with 'Anzahl Pixel' set to 8, 'Helligkeit total' set to 100, and '% Ausgang' set to LED 1.
- Logik:** A 'Hauptloop: 100'000 mal pro Sekunde' block containing:
  - A 'wenn' (if) block: 'Sensor 3 = EIN prüfe standard'.
    - 'setze LED-Strip' block: 'ab Pixel' 1, 'Anzahl' 8, 'auf Farbe in °' Farbe, 'Sättigung in %' 100, 'Helligkeit in %' 100.
    - 'pauriere' block: 20 ms.
  - 'sonst' (else) block: 'setze LED-Strip' block: 'ab Pixel' 1, 'Anzahl' 8, 'auf Farbe in °' Farbe, 'Sättigung in %' 100, 'Helligkeit in %' 20.
- Mathematik:** 'schreibe' block: 'Farbe = Farbe + 0.01'.

# 10



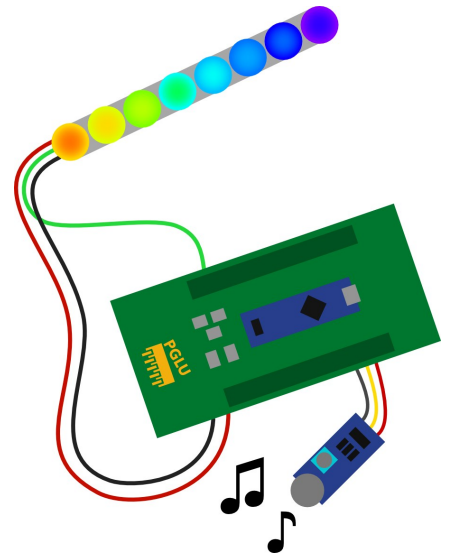
## Beat Fader 🎵 - Video 10

Der Beat-Fader ist ein besonders wirkungsvoller Lichteffekt, denn alle 8 Pixel dimmen nach jedem Rhythmus-Schlag wieder langsam gegen 0%. Spiele mit der Dimm-Zeit und den Farben, die hier mit einem Zufallsgenerator gesteuert werden.

The screenshot shows the PGLU IDE interface with the following code blocks:

- Aktion:** BEAT FADER
- definiere LED-Strip:** Anzahl Pixel: 8, Helligkeit total: 100, % Ausgang: LED 1
- Hauptloop:** 100'000 mal pro Sekunde, Blinkcode: kurz 1, lang 1
- schreibe Helligkeit =:** Helligkeit - 0.8
- schreibe Helligkeit =:** begrenze Helligkeit auf Min 20 und Max 100
- wenn:** Sensor 3 = EIN prüfe standard
- schreibe Farbe =:** ganzzahliger Zufallswert zwischen 20 bis 225
- schreibe Helligkeit =:** 100
- setze LED-Strip ab Pixel:** 1, Anzahl: 8, auf Farbe in: Farbe, Sättigung in %: 100, Helligkeit in %: Helligkeit
- pausiere:** 5 ms

# 11



## Sound Wiper 🎵 - Video 11

Wie ein Scheibenwischer pendelt dieser Pixel hin und her, und das im Takt zu deinem Sound. Experimentiere mit dem Tempo dieses interaktiven Laufflichts und erreiche, dass der Pixel in beide Richtungen läuft!

Spiele auch mit der Hintergrundfarbe!

definiere LED-Strip Anzahl Pixel 8 Helligkeit total 100 % Ausgang LED 1

Hauptloop: 100'000 mal pro Sekunde Blinkcode: kurz 1 lang 1

wenn Sensor 4 = EIN prüfe standard

zähle Pixel von 1 bis 8 alle 1

setze Zwischenspeicher ab Pixel 1 Anzahl 8 auf Farbe in ° Farbe Sättigung in % 100 Helligkeit in % 7

setze Zwischenspeicher ab Pixel Pixel Anzahl 1 auf Farbe in ° Farbe Sättigung in % 100 Helligkeit in % 100

sende Pixel aus Zwischenspeicher an LED-Strip

pausiere 20 ms

schreibe Farbe = Farbe + 5

sonst setze LED-Strip ab Pixel 1 Anzahl 8 auf Farbe in ° Farbe Sättigung in % 100 Helligkeit in % 20

ZWISCHENSPEICHER für die Darstellung mehrerer Farben gleichzeitig