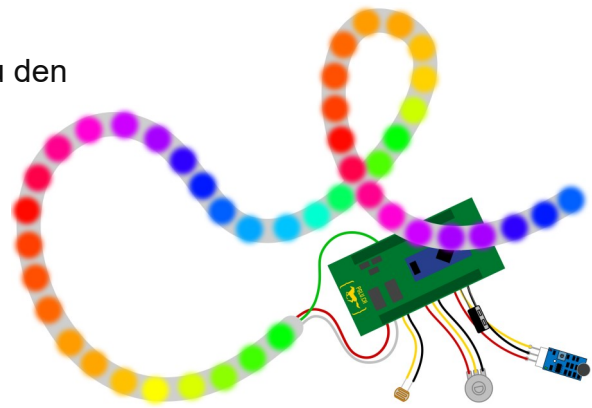


## Challenges zu den Schnippseln „Strip“

Diese Aufgaben sind mögliche Erweiterungen zu den Schnippseln und Videos 1-13



1

### Ein Pixel leuchtet direkt

- > Lasse deinen Pixel in anderen Farben leuchten. Teste auch Farben mit weniger Sättigung. Beachte: sobald du die Sättigung der Farbe heruntersetzt, steigt dein Strombedarf.
- > Wieviel Ampères „zieht“ dein Strip, wenn alle Pixel weiss leuchten? Beachte Arbeitsblatt „Speedypixel Basics“ Seite 2 sowie die Leistungsfähigkeit (maximaler Strom) deiner Stromquelle!

2

### Sensor steuert die Lage und Farbe eines Pixels

- > Verändere mit der Position der Pixel auch ihre Farbe. Tipp: In Schnippsel 4 wird ebenfalls die Farbe abhängig von der Lage der Pixel verändert. Verstehst du, weshalb hier der Sensorwert mit 3.6 multipliziert wird? Es geht darum, aus dem Sensorwert 100% 360° zu machen.

3

### Sensor steuert Anzahl Pixel

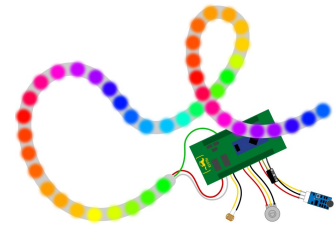
- > Spiele mit den Farben: je mehr Pixel leuchten, desto mehr verändert sich die Farbe entlang dem Verlauf des Farbkreises.
- > Ein Spezialfall: Wenn der erste Pixel alleine leuchtet soll dieser die Farbe grün 120° haben. Leuchten alle 72 Pixel, soll der Strip rot 0° leuchten. Programmiere dies so, dass die Farben dazwischen dem Verlauf des Farbkreises entlang *rückwärts* folgen. Von *grün* über *orange* direkt zu *rot*. Rechne so: Pixelfarbe = 120 - (Sensorwert x 1.2)

4

### Sensor steuert Anzahl und Farbe der Pixel

- > Ändere die Farben der Pixel so, dass ein einzelner Pixel türkis 160° leuchtet. Wenn alle 72 Pixel leuchten, sollen diese die Farbe purpur 270° haben!

## Challenges zu den Schnippseln „Strip“



5

### Sensor steuert Lage und Anzahl der Pixel

- > Jetzt geht's ins Detail: In Video 5 siehst du, dass ein Pixel ständig leuchtet, auch wenn das Mikrofon keine Musik „hört“. Dies ist Pixel Nummer 36, welcher auf dem Strip in der Mitte zu liegen scheint. Tatsächlich kann Pixel 36 jedoch nicht der mittlere Pixel sein, da der Strip in zwei Hälften aufgeteilt ist: Pixel 1-36 und 37-72. Kurz: es gibt keinen mittleren Pixel, wenn der Strip in zwei Hälften aufgeteilt wird und deshalb ist die Animation in Video 5 auch nicht ganz symmetrisch.
- > Ändere das Programm so, dass auf dem Strip eine echte Symmetrie entsteht und bei Stille gar keine Pixel leuchten. Hört das Mikrofon Musik, sollen auf beiden Seiten des Pegels jeweils gleichviele Pixel aufleuchten. Tipp: Arbeite mit einer Wenn - Sonst Frage und setze bei Sensorwert  $< 5\%$  den ganzen Strip auf dunkel. Schau auch Schnippsel 9 an und arbeite mit zwei Ebenen, je eine für jede Seite der Symmetrie.

6

### Breath

- > Es ist gar nicht so wichtig, dieses Schnippsel vollständig zu verstehen. Dennoch lohnt es sich, mit den Farben gezielt spielen zu können: Ändere das Programm so, dass der „Atem“ zwischen aqua  $180^\circ$  und blau  $225^\circ$  variiert.
- > Schaffst du es, je näher der Atemzug der Farbe blau  $225^\circ$  kommt, die Sättigung um  $20\%$  zu senken um sie bei  $180^\circ$  wieder auf  $100\%$  zu erhöhen?

7

### Rainbow still

- > Steuere mit einem Potentiometer oder Mikrofon die *Weite* der Farbabstände im Regenbogen: von ganz *eng* bis zu sehr *weit*. Welche fixe Zahl muss durch einen Sensor ersetzt werden und welchen Faktor benötigst du, um sinnvolle Effekte zu erhalten?

8

### Rainbow flow

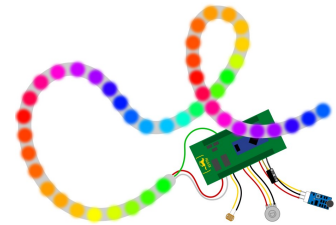
- > Steuere mit einem Potentiometer oder Mikrofon das Tempo des Farbflusses im Regenbogen: von *langsam* zu *schnell*. Welche fixe Zahl muss durch einen Sensor ersetzt werden und welchen Faktor benötigst du, um sinnvolle Effekte zu erhalten?

9

### Musikpegel mit Hintergrund

- > Lege einen „Regenbogen still“ in den Hintergrund anstelle der fixen Farbe grün.

## Challenges zu den Schnippseeln „Strip“



10

### Musikpegel mit Peak

- > Steuere die Zeit des „Herunterfallens“ mit einem Potentiometer. Welche fixe Zahl muss durch einen Sensor ersetzt werden und welchen Faktor benötigst du, um sinnvolle Effekte zu erhalten?

11

### Spin

- > Spiele mit der Zeit des Ausblendens der Pixel. Welchen Wert musst du dabei ändern?

12

### Knight Rider

- > Steure die Länge des Schweifs mit einem anderen Wert. Kannst die auch die Farben in Abhängigkeit zur Pixelposition variieren?

13

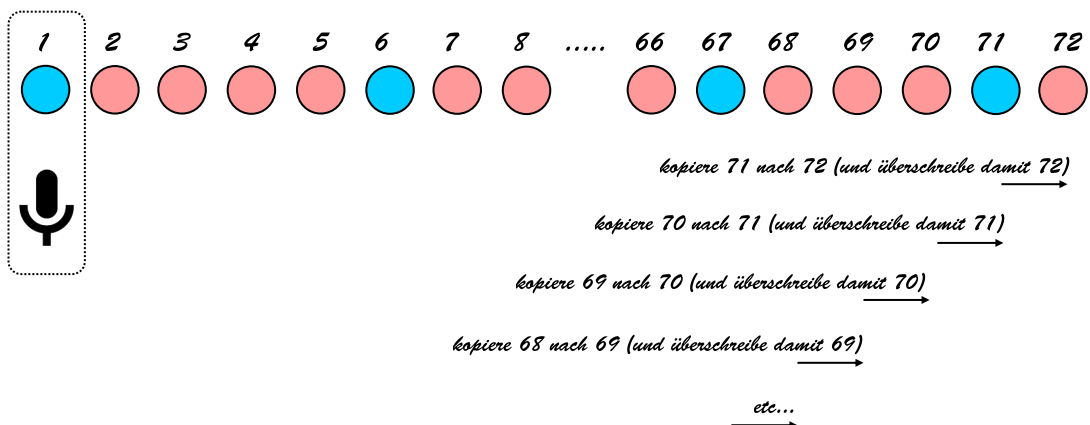
### Sparkle

- > Dieses Programm spielt mit Zufallswerten für die Position, Farbe und Sättigung eines Pixels. Steuere mit einem Sensor das Tempo und/oder die Länge des Dimmeffektes.

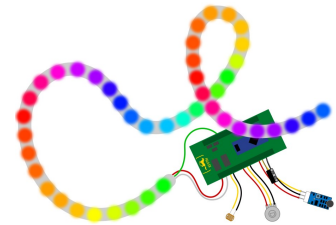
14a

### Pixelshooter

- > Pixel 1 reagiert auf das Mikrofon an Sensor 4 mit Farbe und Helligkeit
- > Farbe und Helligkeit von Pixel 1 werden Pixel für Pixel nach oben kopiert
- > Um den Kopiervorgang zu verstehen, skizziere ihn auf einem Blatt Papier
- > Verwende die Blöcke auf der folgenden Seite. Die Reihenfolge kann eventuell anders sein, als auf dem Bild.



## Challenges zu den Schnippseeln „Strip“



definiere LED-Strip Anzahl Pixel  Helligkeit total  %

zähle Pixel von bis alle

sende Pixel aus Zwischenspeicher an LED-Strip

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

kopiere im Zwischenspeicher Pixel  nach Pixel

lese Wert von Sensor  in %  \*

lese Wert von Sensor  in %

14b

### Pixelshooter symmetrisch

Hier sollen die beiden mittleren Pixel des Strips auf die Musik reagieren und in beide Richtungen nach aussen kopiert werden. Tipp: skizziere den Vorgang auf einem Blatt Papier und versuche zu verstehen, wo du mit dem Kopieren beginnen kannst.

E

### Extra

Recherchiere im Web nach weiteren Effekten mit diesen Keywords:

- > Neopixel VU meter
- > Neopixel music pattern
- > Neopixel music visualizer
- > Neopixel Audio / Sound / Spectrum