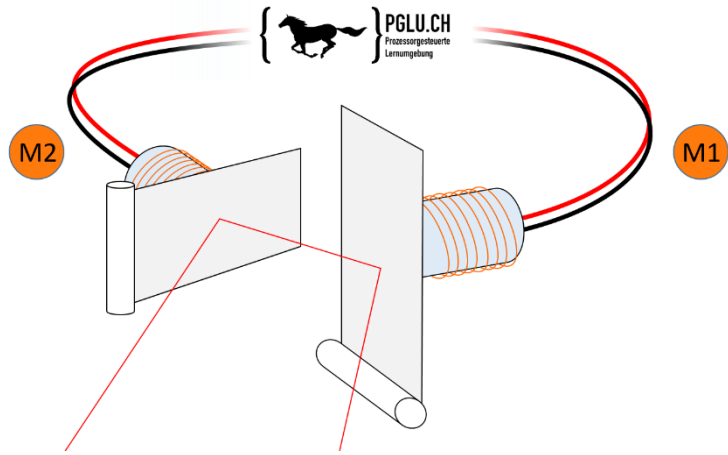


# 1



## Variante Quadrat gefahren, nicht gesprungen

In Aufgabe 3 (Tici Taci) hast du bereits ein Quadrat gezeichnet. Der Strahl ist dabei direkt von einer Ecke zur nächsten gesprungen.

Lerne hier die Zähle-von-bis-Schleife kennen, welche dir ermöglicht, den Laser kontrolliert von der einen Ecke in die andere fahren zu lassen!

Steuere das Tempo dieser Bewegung mit Sensor 3.

Ein wichtiges Detail: «Schritt» darf nie den Wert 0 annehmen können, sonst bleibt die Schleife ewig hängen ☹️

Programmieren Grafik Code Testen Simulator Organisieren Hilfe

Agieren

Sensor

Motor

LED

Steuern

Frage

Ablauf

Dimensionieren

Variable

Mathe

Vereinfachen

Funktion

Erweitern

Ablauf II

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

zähle Position von -100 bis 100 alle Schritt

setze Motor 1 auf Position %

zähle Position von -100 bis 100 alle Schritt

setze Motor 2 auf Position %

zähle Position von 100 bis -100 alle Schritt

setze Motor 1 auf Position %

zähle Position von 100 bis -100 alle Schritt

setze Motor 2 auf Position %

Vor Hauptloop: 1x

schreibe Schritt = 0.05

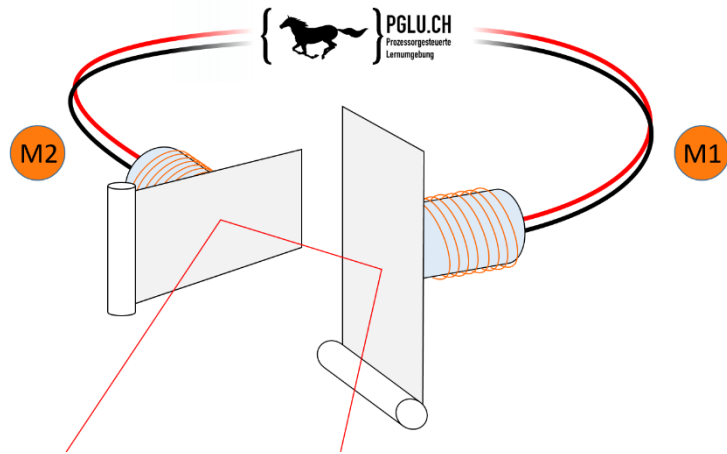
setze LED 4 auf EIN

Parallel zu Hauptloop: alle 1ms

schreibe Schritt = ändere lese Wert von Sensor 3 in % von Min 0 Max 100 zu Min 0.01 Max 2

Garantiert, dass «Schritt» nie den Wert 0 annehmen kann!

# 2



## Variante Lasershow Blubber

Die Erhöhung des Schrittes für die sin- und cos-Berechnung, welche du in Aufgabe 6 kennengelernt hast, wird hier mit der inneren zähle-von-bis-Schleife ausgeführt.

In Aufgabe 6 liessst du die Variable x von 0 bis Unendlich hochzählen. Hier stoppst du die Berechnungen von sin und cos nach jeder Kreisumdrehung (sobald x gleich 6.282 ist). Dann erhöhst/verringerst du den Durchmesser des Kreises mit der äusseren zähle-von-bis-Schleife um 0.1 und startest deine innere Rechnung wieder bei 0!

**Vor Hauptloop: 1x**

- schreibe Anfang = 0
- schreibe Ende = 100
- setze LED 4 auf EIN

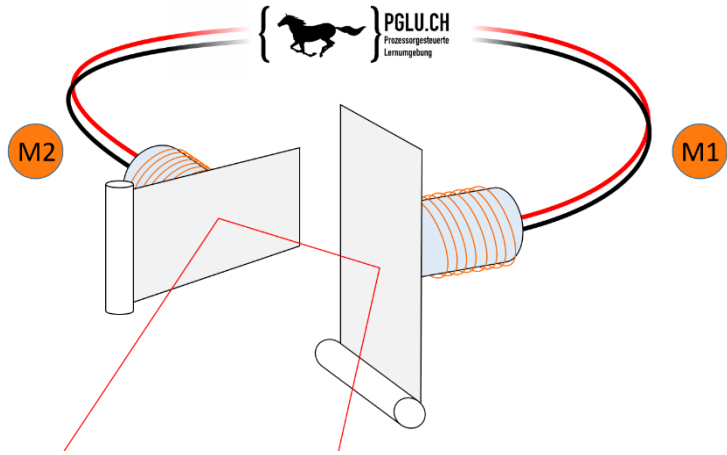
Sorgt dafür, dass die Schleife «Zähle Kreisgrösse von Anfang bis Ende» abwechslungsweise hoch- und hinuntergezählt wird!  
«Absolutwert» bedeutet, dass negative Zahlen automatisch positiv gemacht werden.

**Hauptloop: 100'000 mal pro Sekunde** (Blinkcode: kurz 1 lang 2)

- schreibe Anfang =  $|\text{Anfang} - 100|$
- schreibe Ende =  $|\text{Ende} - 100|$
- zähle Kreisgrösse von Anfang bis Ende alle 10
- zähle x von 0 bis 6.282 alle 0.1
- setze Motor 1 auf  $\sin(x) \cdot \text{Kreisgrösse} \%$
- setze Motor 2 auf  $\cos(x) \cdot \text{Kreisgrösse} \%$

Die innere Schleife wird 62 x ausgeführt. Dann wird die äussere Schleife 1 x ausgeführt und «Kreisgrösse» um 10 erhöht. Dann kommt wieder die innere Schleife 62 x zum Einsatz... usw

# 3



## Variante Lasershow pulsierend (Siehe Video 5+6: Projekte>Lasershow)

Dies ist eine Variante des Programms aus Aufgabe 8. Zusätzlich wird jede Millisekunde gefragt, ob der Laser ein- oder ausgeschaltet sein soll!

Die Variable «Intervall» bestimmt die Zeit in Millisekunden, in der der Laser ein- und wieder ausgeschaltet werden soll. In der ersten Hälfte des Intervalls leuchtet der Laser auf und in der zweiten Hälfte löscht er ab.

**Parallel zu Hauptloop: alle 1ms**

```

schreibe Zeit = [Zeit] + [1]
wenn [Zeit] < [Intervall / 2]
  setze LED 4 auf EIN
sonst
  setze LED 4 auf AUS
wenn [Zeit] = [Intervall]
  schreibe Zeit = 0
  
```

**Vor Hauptloop: 1x**

```

schreibe Intervall = 16
  
```

Spieler mit «Intervall» mit Werten von 4 bis 32.

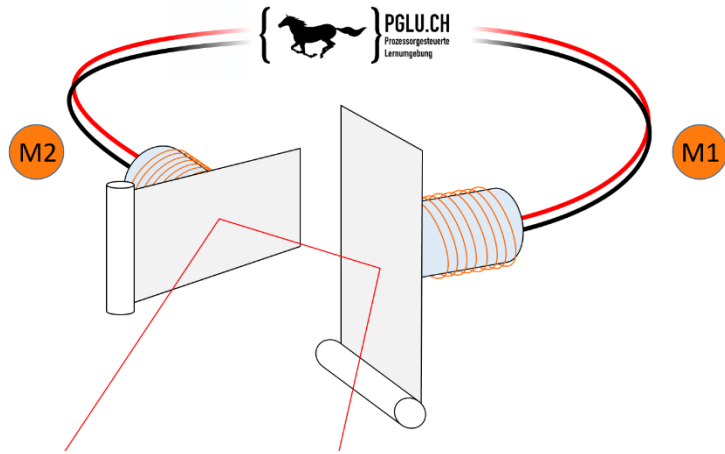
Fragt, ob sich der Ablauf in der ersten- oder zweiten Hälfte des vorgegebenen Intervalls befindet!

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

```

schreibe x = [x] + [ändere lese Wert von Sensor 3 in % von Min 0 Max 100 zu Min 0 Max 0.5]
schreibe y = [y] + [ändere lese Wert von Sensor 4 in % von Min 0 Max 100 zu Min 0 Max 0.5]
setze Motor 1 auf [sin x * 100] %
setze Motor 2 auf [cos y * 100] %
  
```

# 4



## Variante Blubber Musikgesteuert (Siehe Video 7: Projekte>Lasershow)

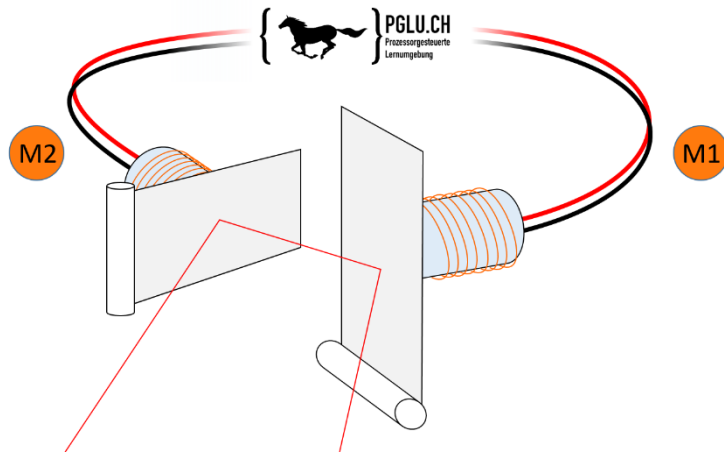
In dieser Variante des Blubbers soll die Kreisgröße mit dem Pegel von Musik gesteuert werden. Hierzu könnte im Programm aus Aufgabe 7 der Faktor 100 durch eine Sensorabfrage an einem Mikrophon ersetzt werden. Dies würde jedoch nur zu chaotischen Bildern führen.

Deshalb wird der Musikpegel am Mikrophon nur alle 10 Kreisumdrehungen gemessen und daraus die Kreisgröße abgeleitet. Zwischen diesen Messungen wird der Kreis nach jeder Umdrehung um 3% verkleinert.

The screenshot shows a programming environment with a dark theme. The top navigation bar includes 'Programmieren', 'Testen', 'Organisieren', and 'Hilfe'. The main workspace contains the following script:

- Vor Hauptloop: 1x**
  - setze LED 4 auf EIN
- Hauptloop: 100'000 mal pro Sekunde**
  - Blinkcode: kurz 1 lang 2
  - schreibe Musikpegel = lese Wert von Sensor 4 in %
  - schreibe Blubbergröße = Musikpegel
  - wenn Musikpegel > 20
    - wiederhole 10 mal
      - schreibe Blubbergröße = Blubbergröße - 3
      - zähle x von 0 bis 6.282 alle 0.1
      - setze Motor 1 auf sin x \* Blubbergröße %
      - setze Motor 2 auf cos x \* Blubbergröße %

# 5



## Variante Rotierende Linie, wie ein Propeller

Der Strahl fährt eine Linie vor und zurück. Nach jedem Durchgang, wird der Winkel der Linie um einen Schritt geändert.

Das Programm ist in 4 Abschnitte aufgeteilt, welche nacheinander jeweils 90° der Drehung ausführen.

Diese Figur kann sicher noch einfacher programmiert werden. Versuchs!

Just for fun!

The screenshot shows a programming environment with a dark background and a sidebar on the left. The sidebar contains various categories: Agieren, Sensor, Motor, LED, Steuern, Frage, Ablauf, Dimensionieren, Variable, Mathe, Vereinfachen, Funktion, Erweitern, and Ablauf II. The main area displays a code editor with a block-based program. The program is organized into several sections:

- Hauptloop: 100'000 mal pro Sekunde** (Blinkcode: kurz 3, lang 2)
- Funktion mit:** Weite Start (1), Weite Ende (100), Schritt Start (0), Schritt Ende (6.282), Startpunkt = (0)
- Funktion mit:** Weite Start (99), Weite Ende (0), Schritt Start (0), Schritt Ende (6.282), Startpunkt = (-3.141)
- Funktion mit:** Weite Start (1), Weite Ende (100), Schritt Start (3.141), Schritt Ende (9.423), Startpunkt = (0)
- Funktion mit:** Weite Start (99), Weite Ende (0), Schritt Start (3.141), Schritt Ende (9.423), Startpunkt = (-3.141)

The main function block is titled "Name: Funktion mit: Weite Start, Weite Ende, Schritt Start, Schritt Ende, Startpunkt". It contains the following logic:

- zähle Weite** von Weite Start bis Weite Ende alle Winkel
- zähle Schritt** von Schritt Start bis Schritt Ende alle Speed
- setze Motor 1** auf  $\sin(\text{Schritt} * \text{Weite} \% \text{Schritt Ende})$
- setze Motor 2** auf  $\sin(\text{Schritt} + \text{Startpunkt} * \text{Absolutwert}(\text{Weite} + -100) \% \text{Schritt Ende})$

There is a "Vor Hauptloop: 1x" block that sets LED 4 to EIN.

Below the main function, there is a "Parallel zu Hauptloop: alle 1ms" section with two blocks:

- schreibe Winkel =** ändere lese Wert von Sensor 3 in % von Min (0) Max (100) zu Min (5) Max (50)
- schreibe Speed =** ändere lese Wert von Sensor 4 in % von Min (0) Max (100) zu Min (0.01) Max (0.5)