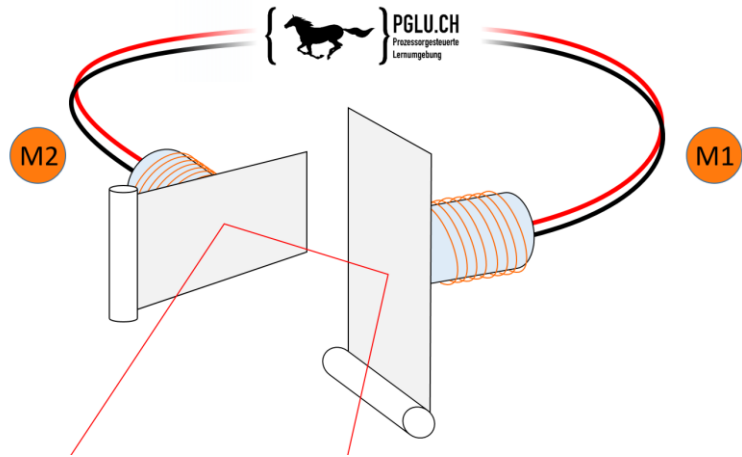


# 1



**Lösung** Beschreibe die Lasershow und das was sie tut!

Lese und verstehe diesen Text, damit es dir leichter fällt, die Lasershow und ihre Funktionen zu programmieren!

Weiterführende Google Keywords:

- Lissajous
- Lissajous figures
- Lissajous pattern

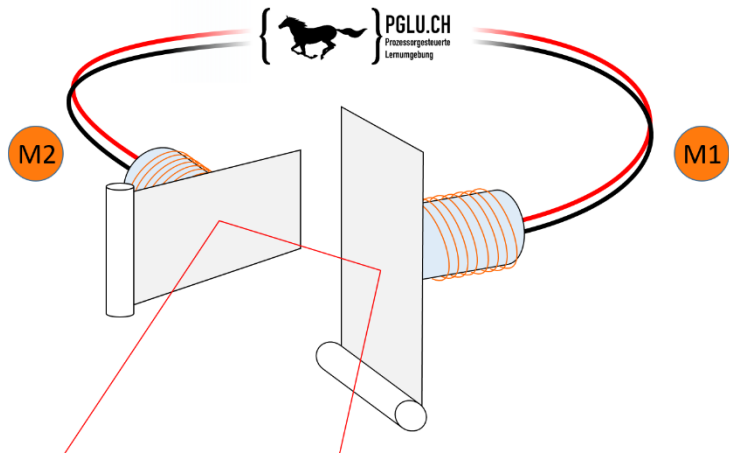
**Komponenten:** Die Lasershow besteht aus zwei Spiegeln, welche an den Magnetspulen zweier Lautsprecher und an Klebbandscharnieren befestigt sind. Ein Laserpointer erzeugt einen Strahl, der über beide Spiegel gelenkt wird.

**Elektronik und Mechanik:** die Spule umgibt einen Magnetkern und erzeugt dadurch eine Bewegung, wenn sie mit Strom durchflossen wird.

**Anordnung und Bewegung:** Durch die richtige Anordnung kann der eine Spiegel horizontal- und der andere vertikal nach vorne oder nach hinten schwenken.

**Steuerung:** Wenn im Programm ein Motor auf +100% gestellt wird, kippt der entsprechende Spiegel nach vorne. Bei -100% geschieht das Gegenteil. Natürlich können auch Zwischenstufen eingestellt werden, wie zum Beispiel +23%.

2



## Lösung Tic Tac

Teste dein Programm im Simulator und beobachte Die Ausgänge M1 und L1.

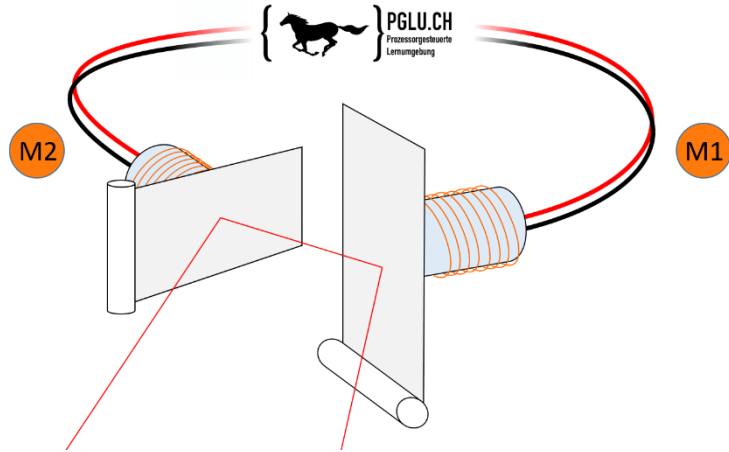
Du kannst den Wert des Blockes «pausiere» während laufender Simulation verändern!

The screenshot shows the PGLU.CH simulator interface. The 'Simulator' tab is selected. The interface includes a sidebar with categories like 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', and 'Ablauf II'. The main area displays a program with the following blocks:

- Vor Hauptloop: 1x**
  - setze LED 4 auf EIN
- Hauptloop: 100'000 mal pro Sekunde**
  - setze Motor 1 auf 100 %
  - pausiere 1000 ms
  - setze Motor 1 auf -100 %
  - pausiere 1000 ms

The status bar at the bottom shows the following values: M1 100, M2 0, L1 0, L2 0, L3 0, L4 100. The 'M1 100' and 'L4 100' values are circled in yellow.

3



### Lösung Tici Taci

Die Summe aller Pausen ergibt 2000ms. Dies ist gleichviel wie in Aufgabe 2.

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 block-based code for a robot. The code is organized into two main sections: "Vor Hauptloop: 1x" and "Hauptloop: 100'000 mal pro Sekunde".

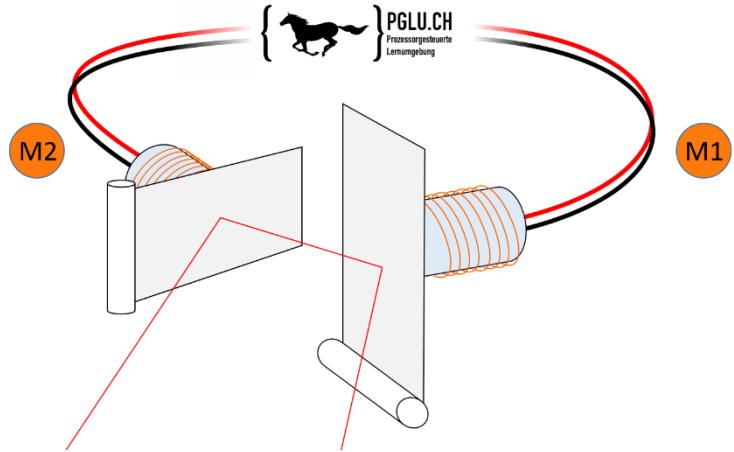
**Vor Hauptloop: 1x**

- setze LED 4 auf EIN

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

- setze Motor 1 auf 100 %
- pausiere 500 ms
- setze Motor 2 auf 100 %
- pausiere 500 ms
- setze Motor 1 auf -100 %
- pausiere 500 ms
- setze Motor 2 auf -100 %
- pausiere 500 ms

4

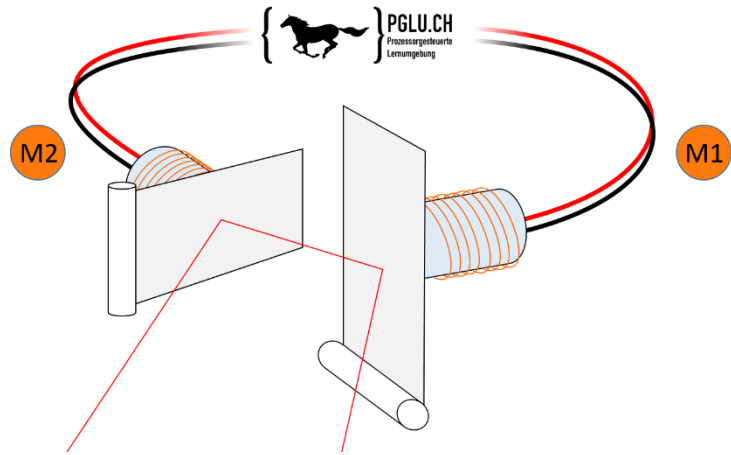


### Lösung Tici Taci mit Sensor

Teste diese Lösung im Simulator und schiebe den Regler S3 mit der Maus nach oben. Beobachte dabei die Ausgänge M1 und M2.

The screenshot shows the simulation software interface. The 'Simulator' tab is selected. The interface includes a top navigation bar with 'Programieren', 'Grafik', 'Code', 'Simulator', 'Organisieren', 'Programme', and 'Hilfe'. On the left, there is a sidebar with various tool categories like 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', 'Erweitern', and 'Ablauf II'. The main workspace displays a control panel with four sensors (Sensor 1-4) and their respective sliders and buttons. A 'Vor Hauptloop: 1x' block contains a 'setze LED 4 auf EIN' block. The main loop contains alternating 'setze Motor' and 'pausiere' blocks. The 'pausiere' blocks are linked to 'lese Wert von Sensor 3' and include 'Spannweite Regler' and 'Spannweite Pausenlänge' settings. At the bottom, motor status indicators show M1 and M2 at 100% and L1-L4 at 0%.

# 5

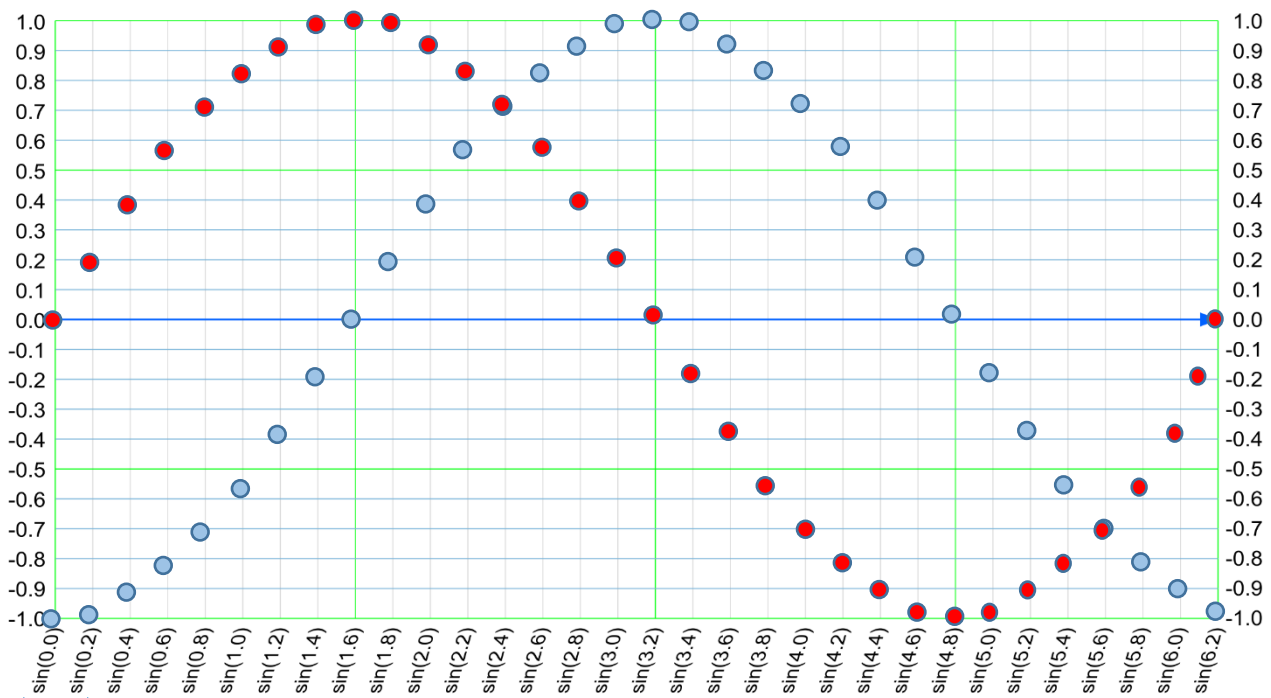


## Lösung Grundlagen Sinuskurve kennenlernen

Die roten Punkte zeigen eine Sinuskurve. Diese Wellenförmige Bewegung kennst du aus Natur und Technik: Die Tageslänge im Verlauf der Jahreszeiten oder die Bewegung eines Uhrenpendels.

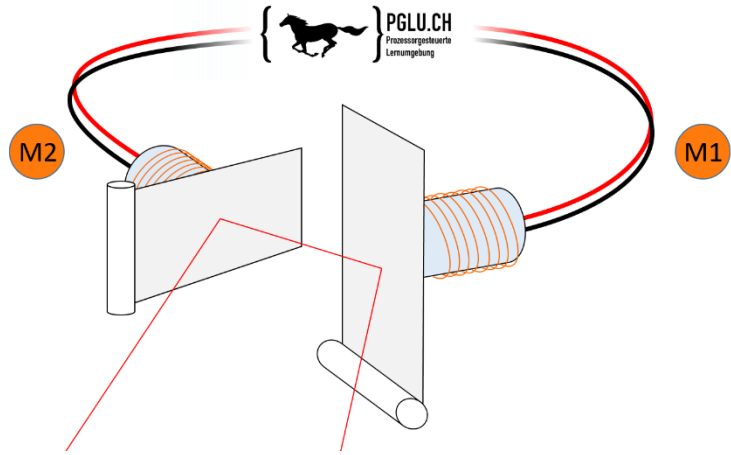
In Aufgabe 7 wirst du auch noch die Cosinus-Kurve kennenlernen. Sie ist hier mit hellblauen Punkten dargestellt und wird mit der Formel  $\cos(0.0)$ ,  $\cos(0.2)$ ,  $\cos(0.4)$ ... etc. berechnet!

Frage deine Mathe Lehrperson nach weiteren Verwendungen von Sinus und Cosinus.



Schrittweite zwischen 2 Rechnungen: Je kleiner, desto genauer die Kurve aber aufwändiger und langsamer die Berechnung!

# 6



## Lösung Sinus-Berechnung automatisieren

Gratuliere! Du hast viel Fleissarbeit an deinen Prozessor abgegeben und kannst dich nun voll und ganz auf die kreative Weiterentwicklung deiner Laserbilder konzentrieren.

Der Spiegel an Ausgang M1 führt nun eine harmonische Schaukelbewegung aus. Das Tempo dieser Bewegung hängt von der Schrittweite (hier 0.005) zwischen zwei Sinus-Rechnungen ab!

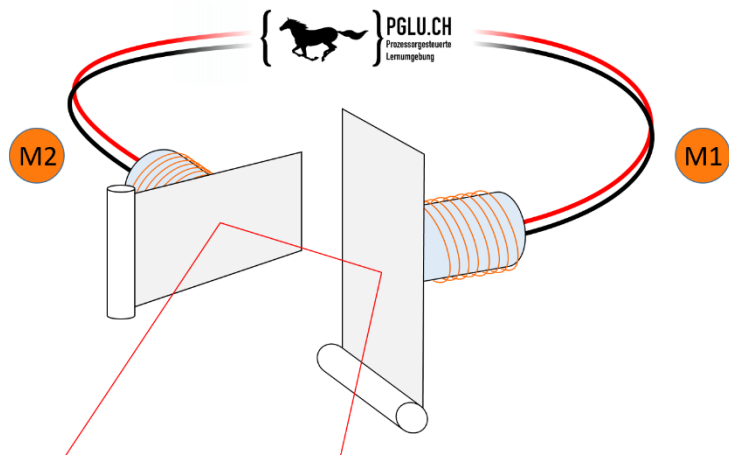
Faktor 100 bestimmt die Kreisgröße!

Die Rechnung  $\sin(x)$  ergibt Werte zwischen -1 und 1. Damit für den Ausgang M1 verständliche %-Werte entstehen, muss das Ergebnis noch mit 100 multipliziert werden.

```
Vor Hauptloop: 1x
setze LED 4 auf EIN

Hauptloop: 100'000 mal pro Sekunde
setze Motor 1 auf sin(x) * 100 %
schreibe x = x + 0.005
```

7



## Lösung Kreis mit Sinus und Cosinus

Spiele mit den Werten 0.005 und setze für x und y auch unterschiedliche Zahlen ein. Z.B 0.5 und 0.25

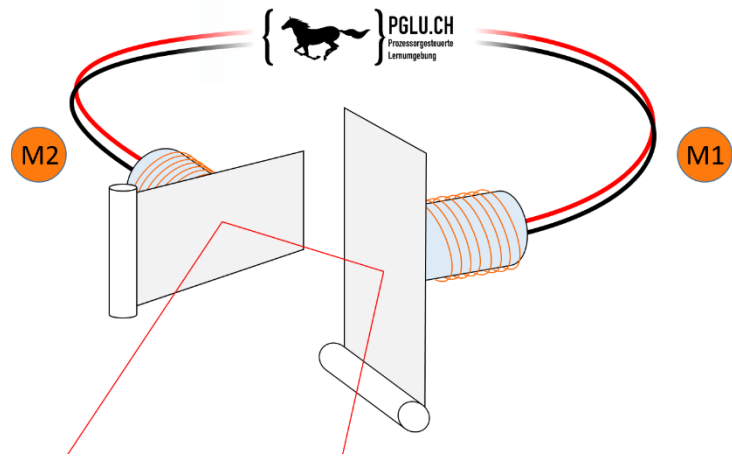
Programmieren Testen Organisieren Hilfe  
Grafik Code Simulator Programme ?

Agieren  
Sensor  
Motor  
LED  
Steuern  
Frage  
Ablauf  
Dimensionieren  
Variable  
Mathe  
Vereinfachen  
Funktion  
Erweitern  
Ablauf II

Vor Hauptloop: 1x  
setze LED 4 auf EIN

Hauptloop: 100'000 mal pro Sekunde Blinkcode: kurz 1 lang 1  
schreibe x = x + 0.005  
schreibe y = y + 0.005  
setze Motor 1 auf sin x \* 100 %  
setze Motor 2 auf cos y \* 100 %

Spiele mit diesen Werten!  
Möglich sind: 0.005 bis 0.5



## Lösung Hauptprogramm Laserbilder live verändern

Teste dein Programm im Simulator und verschiebe die Regler S3 und S4.

Beachte: Der Simulator arbeitet langsamer als der Prozessor auf deiner PGLU-Platine!

The screenshot shows the PGLU simulator interface. The top navigation bar includes 'Programmieren', 'Grafik', 'Code', 'Simulator' (highlighted with a yellow circle), 'Organisieren', and 'Programme'. The left sidebar lists various components: Agieren, Sensor, Motor, LED, Steuern, Frage, Ablauf, Dimensionieren, Variable, Mathe, Vereinfachen, Funktion, and Ablauf II.

The main control panel features four sensors (Sensor 1 to Sensor 4) with sliders and buttons. Sensor 3's slider is set to 77 and is circled in yellow. A yellow callout points to this slider with the text 'Spannweite Regler' and 'Spannweite Schrittweite (Siehe Lösung 5)'. A 'Simulationsgeschwindigkeit' slider is also visible.

The program logic is as follows:

- Vor Hauptloop: 1x**
  - schreibe x = 0
  - schreibe y = 0
  - setze LED 4 auf EIN
- Hauptloop: 100'000 mal pro Sekunde**
  - schreibe x = x + (lese Wert von Sensor 3 in % von Min 0 Max 100 zu Min 0 Max 0.5)
  - schreibe y = y + (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) %

At the bottom, the status bar shows motor positions: M1 -59, M2 77, L1 0, L2 0, L3 0, L4 100.