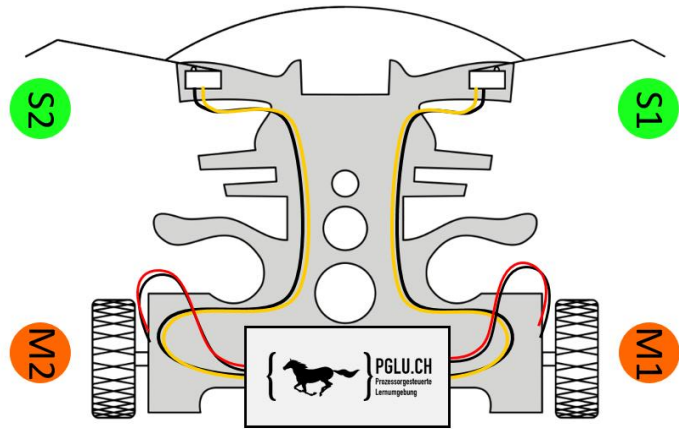


# 1



**Lösung** Beschreibe das Auto und das was es tut!

Lese und verstehe diesen Text, damit es dir leichter fällt, das Auto und seine Funktionen zu programmieren!

Weiterführende Google Keywords:

- Robotik
- Robot Competition/Contest
- Line follower

**Komponenten:** Das Selbstfahrende Auto besitzt 2 Motoren und 2 Sensoren.

**Mechanik:** Die Sensoren bestehen aus einem federnden Bügel an der Front sowie zwei seitlichen Antennen, welche ein Hindernis erkennen können.

**Elektronik:** Der federnde Bügel ist an 2 Schaltern befestigt, welche jede Berührung von vorne oder den Seiten an die Platine melden.

**Steuerung:** Ein Sensor steuert jeweils seinen gegenüberliegenden Motor. Trifft z.B. S1 auf ein Hindernis, dreht M2 während einer halben Sekunde rückwärts.

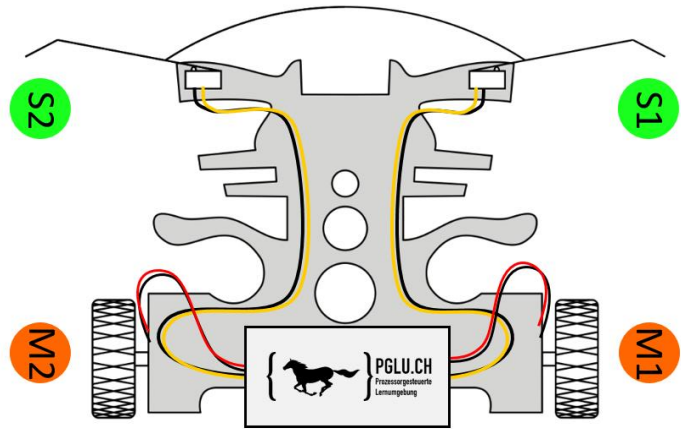
**Bewegung:** Berührt das Fahrzeug ein Hindernis, dreht es an Ort wieder davon weg.

**Timer:** Um dieses Abdrehen zu verlängern, wird nach jeder Berührung das Programm für eine kurze Zeit pausiert. Das heisst, dass die Anweisung «fahre wieder geradeaus» um eine halbe Sekunde hinausgezögert wird!

Mit der Länge dieser Zeit kann der Winkel des Abdrehens eingestellt werden.

In Erweiterungsaufgabe 5 lernst du, dass diese einfache Art, die Drehung zu verlängern auch Nachteile mit sich bringt!

# 2



**Zwei Lösungen** Ein einzelner Motor vorwärts und bei Sensorberührung rückwärts

Die Version 2 ist die elegantere von beiden. Bei Programmieraufgaben gibt es immer viele mögliche Lösungen! Achte darauf, immer die einfachste zu suchen. Das hilft dir, komplizierte Programme besser zu verstehen oder zu erweitern!

**Version 1 mit 2x Fragen**

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

- wenn Sensor 2 = EIN prüfe standard
  - setze Motor 1 auf -100 %
- wenn Sensor 2 = AUS prüfe standard
  - setze Motor 1 auf 100 %

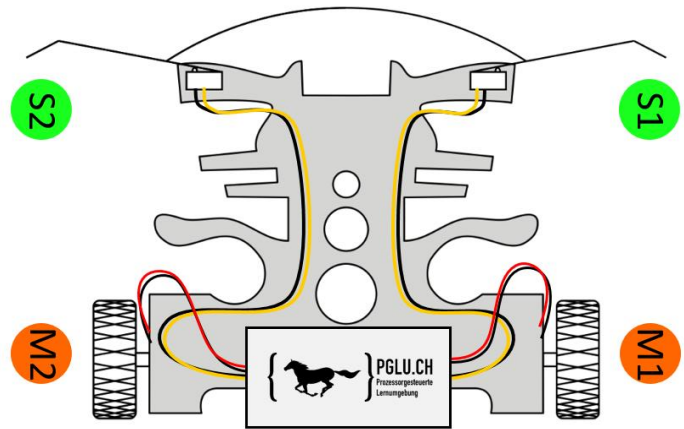
**Version 2 mit 1 x Fragen und sonst-Anweisung**

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

- wenn Sensor 2 = EIN prüfe standard
  - setze Motor 1 auf -100 %
- sonst
  - setze Motor 1 auf 100 %

Für Sonst-Anweisung «wenn» aus dem Menu nehmen, auf schwarzen Punkt klicken und «sonst» unten anfügen

3

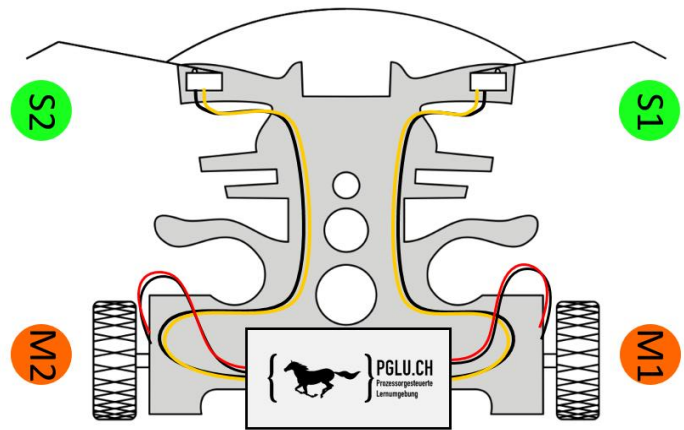


**Lösung** Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde rückwärts

Öffne den Simulator und teste dein Programm: drücke auf den Taster von Sensor 1 in der oberen Leiste und beobachte den Ausgang M1 in der unteren Leiste.

Auf diese Weise kannst du das Programm jederzeit testen. Du kannst es während laufender Simulation auch verändern!

# 4



**Drei Lösungen Erweiterungsaufgabe** Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde rückwärts, mit LED Blinker

Mit diesen drei Lösungen lernst du ein Programm zu vereinfachen und zu kürzen.

Programm 1 ist die Grundversion. In der Version 2 wird das Blinken in eine «Funktion» ausgelagert und in Version 3 kommt zusätzliche eine «Wiederhole-Schleife» zum Einsatz.

## Grundversion

The screenshot shows a block-based programming environment. The main loop is titled 'Hauptloop: 100'000 mal pro Sekunde'. The program starts with a 'wenn' (if) block that checks if 'Sensor 2' is 'EIN' (ON) using a 'prüfe standard' (check standard) block. If true, it sets 'Motor 1' to '-100%' and 'LED 1' to 'EIN'. It then pauses for 100 ms, sets 'LED 1' to 'AUS', pauses for 100 ms, sets 'LED 1' to 'EIN', pauses for 100 ms, sets 'LED 1' to 'AUS', pauses for 100 ms, sets 'LED 1' to 'EIN', and finally pauses for 100 ms. If the sensor is not ON, it sets 'Motor 1' to '100%'.

## Version 2 mit Funktion

The screenshot shows the LEGO Mindstorms software interface. The main workspace contains a program with a 'Hauptloop: 100'000 mal pro Sekunde' block. Inside the loop, there is a 'wenn' block with the condition 'Sensor 2 = EIN prüfe standard'. The 'wenn' block has two branches: 'wenn' and 'sonst'. The 'wenn' branch contains a 'setze Motor 1 auf -100 %' block and a 'Blinke mit LED 500ms' function block. The 'sonst' branch contains a 'setze Motor 1 auf 100 %' block. The function block is named 'Blinke mit LED 500ms'. The function's internal code is shown in a separate window, containing a sequence of blocks: 'setze LED 1 auf EIN', 'pausiere 100 ms', 'setze LED 1 auf AUS', 'pausiere 100 ms', 'setze LED 1 auf EIN', 'pausiere 100 ms', 'setze LED 1 auf AUS', 'pausiere 100 ms', 'setze LED 1 auf EIN', 'pausiere 100 ms', and 'setze LED 1 auf AUS'. The left sidebar shows the 'Funktion' menu item highlighted. Below the screenshot, there are three yellow arrows pointing to the function block in the main workspace and the function definition window, with corresponding text instructions.

Programmieren Grafik Code Simulator Programme ?

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

wenn Sensor 2 = EIN prüfe standard

setze Motor 1 auf -100 %

Blinke mit LED 500ms

sonst setze Motor 1 auf 100 %

Name Blinke mit LED 500ms

setze LED 1 auf EIN

pausiere 100 ms

setze LED 1 auf AUS

pausiere 100 ms

setze LED 1 auf EIN

pausiere 100 ms

setze LED 1 auf AUS

pausiere 100 ms

setze LED 1 auf EIN

pausiere 100 ms

setze LED 1 auf AUS

>Zuerst Funktionsklammer in Menu holen

>Funktion einen Namen geben

>Funktionsblock holen und in Hauptloop einsetzen

## Version 3 mit Wiederhole-Schleife

The screenshot shows the LEGO Mindstorms software interface. The main workspace contains a program with a 'Hauptloop: 100'000 mal pro Sekunde' block. Inside the loop, there is a 'wenn' block with the condition 'Sensor 2 = EIN prüfe standard'. The 'wenn' block has two branches: 'wenn' and 'sonst'. The 'wenn' branch contains a 'setze Motor 1 auf -100 %' block and a 'Blinke LED 500ms' function block. The 'sonst' branch contains a 'setze Motor 1 auf 100 %' block. The function block is named 'Blinke LED 500ms'. The function's internal code is shown in a separate window, containing a 'wiederhole 50 mal' block. Inside the loop, there are four blocks: 'setze LED 1 auf EIN', 'pausiere 5 ms', 'setze LED 1 auf AUS', and 'pausiere 5 ms'. The left sidebar shows the 'Funktion' menu item highlighted. Below the screenshot, there are three yellow arrows pointing to the function block in the main workspace and the function definition window, with corresponding text instructions.

Programmieren Grafik Code Simulator Programme ?

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

wenn Sensor 2 = EIN prüfe standard

setze Motor 1 auf -100 %

Blinke LED 500ms

sonst setze Motor 1 auf 100 %

Name Blinke LED 500ms

wiederhole 50 mal

setze LED 1 auf EIN

pausiere 5 ms

setze LED 1 auf AUS

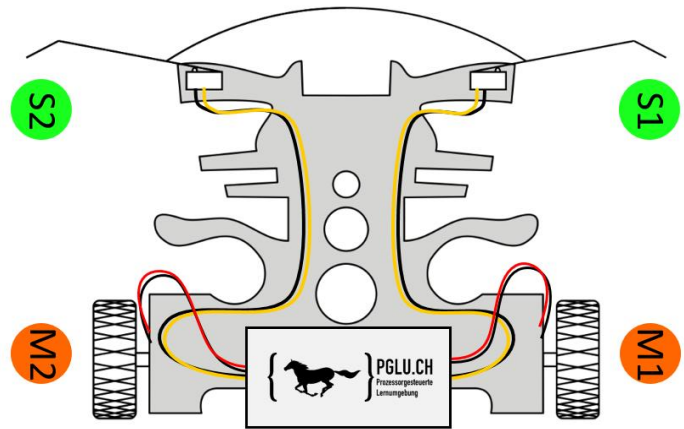
pausiere 5 ms

>Zuerst Funktionsklammer in Menu holen

>Funktion einen Namen geben

>Funktionsblock holen und in Hauptloop einsetzen

# 5



**Lösung Erweiterungsaufgabe** Ein einzelner Motor vorwärts und bei Sensorberührung eine Sekunde rückwärts ohne Programmpause: so würde ein Profi programmieren!

Eine Variable «Timer» zählt im Millisekundentakt von 0 bis unendlich hoch. Ist «Timer» kleiner als 500 dreht M1 rückwärts (M1=-100%).

Sobald Sensor 2 ein Hindernis spürt, wird «Timer» auf 0 gesetzt und ist somit eine halbe Sekunde lang kleiner als 500.

The screenshot shows the PGLU.CH programming interface. The top navigation bar includes 'Programmieren', 'Grafik', 'Code', 'Testen', 'Simulator', 'Organisieren', and 'Hilfe'. The left sidebar lists various components: Agieren, Sensor, Motor, LED, Steuern, Frage, Ablauf, Dimensionieren, Variable, Mathe, Vereinfachen, Funktion, Erweitern, and Ablauf II.

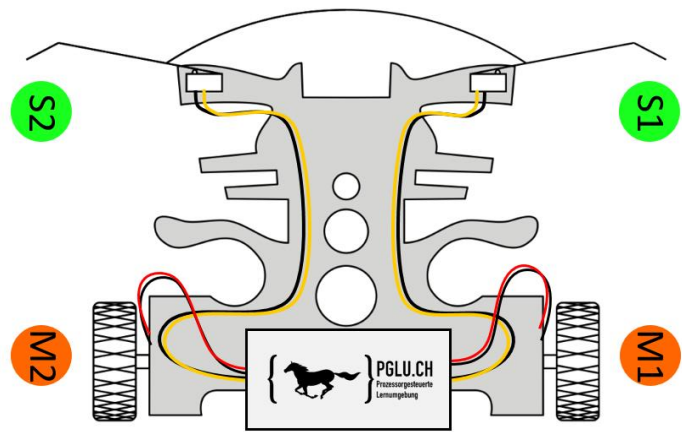
The code editor contains the following blocks:

- Vor Hauptloop: 1x**
  - schreibe x = 500
- Parallel zu Hauptloop: alle 1ms**
  - schreibe Timer = Timer + 1
  - wenn Sensor 2 = EIN prüfe standard
    - schreibe Timer = 0
- Hauptloop: 100'000 mal pro Sekunde**
  - wenn Timer < 500
    - setze Motor 1 auf -100 %
  - sonst
    - setze Motor 1 auf 100 %

Additional text in the interface:

- Blinkcode:** kurz 1, lang 1
- LED 1** auf Timer %
- Text:** Für einen Test im Simulator, setze hier 20 statt 500 ein, da der Simulator langsamer arbeitet, als die PGLU-Platine
- Text:** Setze diesen Block ein, wenn du das Programm im Simulator testen willst. Du kannst dann an Ausgang L1 den Wert von «Timer» beobachten solange er unter 100 ist.

# 6



## Lösung Das Hauptprogramm «Selbstfahrendes Auto»

Gratuliere! Dies ist das vollständige Programm, das dein Auto zum Fahren bringt. Es basiert auf den Programmübungen 1-3, die du gemacht hast.

Spendiere deinem Auto noch zusätzliche Funktionen, indem du die Übungen 4-5 einbaust und LEDs blinken lässt oder eine Version ohne Pausieren-Block entwickelst.

Teste auch die Varianten 1-3 im separaten Dokument.

```
Hauptloop: 100'000 mal pro Sekunde Blinkcode: kurz 1 lang 1
wenn Sensor 2 = EIN prüfe standard
  setze Motor 1 auf -100 %
  pausiere 500 ms
sonst
  setze Motor 1 auf 100 %
wenn Sensor 1 = EIN prüfe standard
  setze Motor 2 auf -100 %
  pausiere 500 ms
sonst
  setze Motor 2 auf 100 %
```