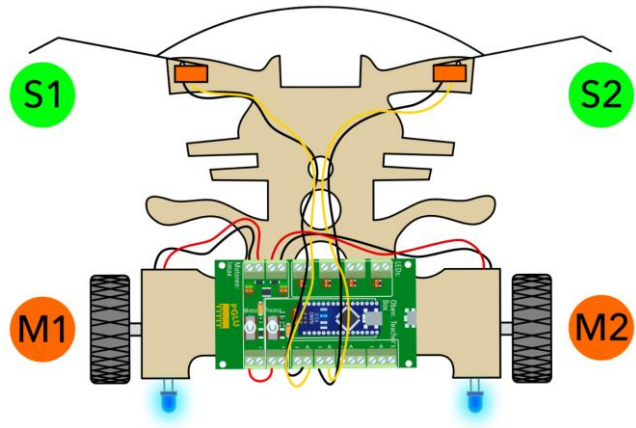


1

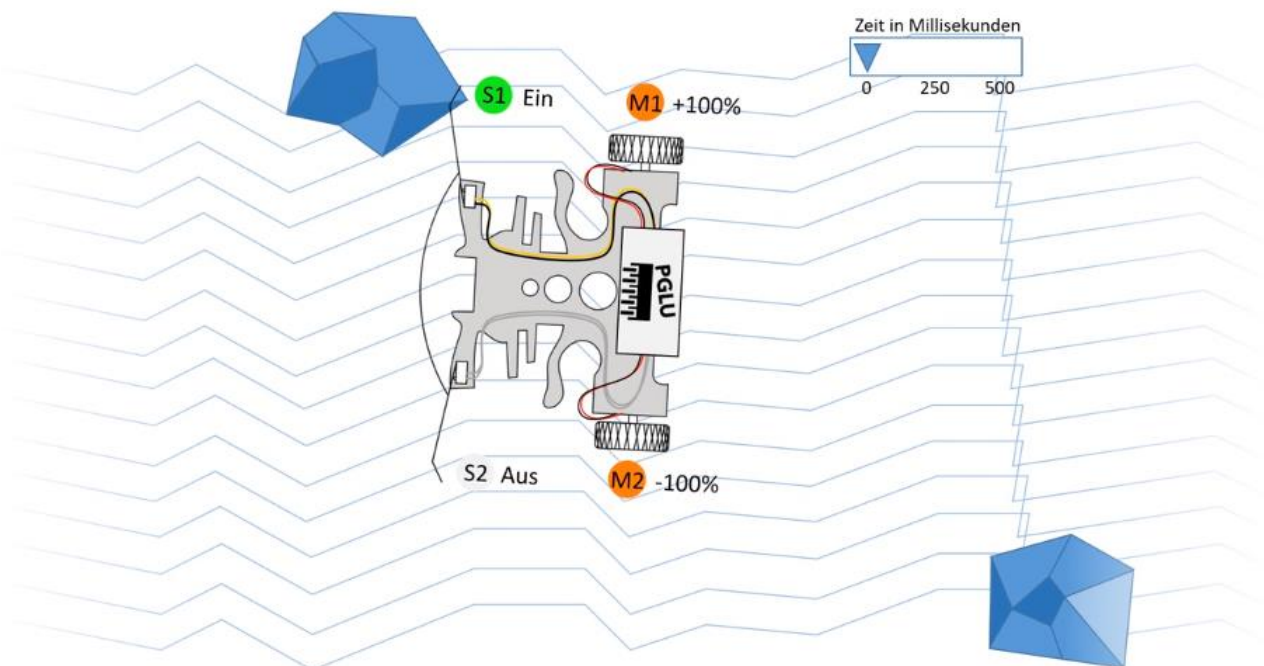


Aufgabe Beschreibe das Auto und das was es tut!

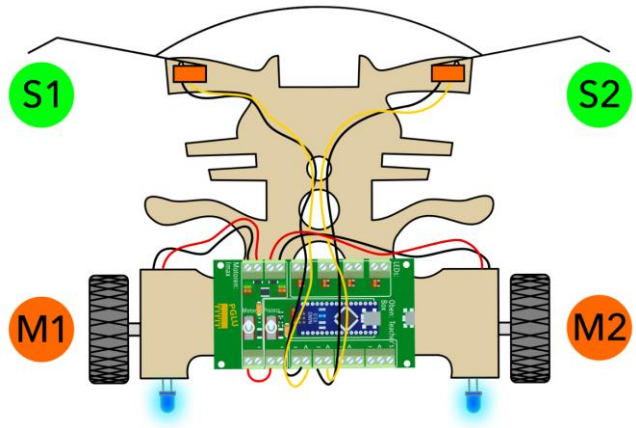
Schaue die Animation und beobachte was geschieht. Stelle dir vor, du müsstest das selbstfahrende Auto einer Person am Telefon erklären, um es ihr zu verkaufen.

Schreibe einen Text dazu!

- Aus welchen Komponenten besteht das Selbstfahrende Auto?
- Welche Einzelteile sind elektronisch, welche mechanisch?
- Wie werden die elektronischen Komponenten gesteuert, was ist der Auslöser für eine Steuerung?
- Wie bewegt sich das Auto, wenn es auf ein Hindernis trifft?
- Was macht der Timer?



2



Aufgabe Ein einzelner Motor (M1) vorwärts und bei Sensorberührung rückwärts.

Hinweis: der andere Motor (M2) ist noch nicht programmiert und läuft deshalb rückwärts. Beachte ihn einfach nicht.

Setze diese Blöcke richtig in den Hauptloop ein:

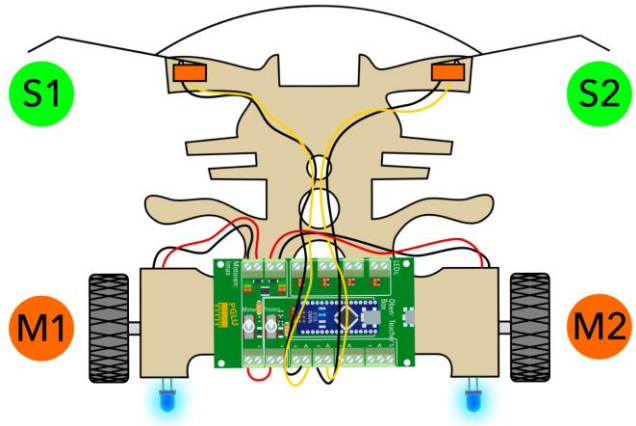
- Wenn Sensor 2 ein Hindernis berührt (S2=Ein), dann lasse Motor 1 Rückwärts laufen (M1=-100%)
- Sonst (wenn Sensor 2 nichts berührt und S2=Aus), lasse Motor 1 vorwärts laufen (M1=100%)

The screenshot shows the PGLU programming environment with the 'Code' tab selected. The main loop is set to 'Hauptloop: 100'000 mal pro Sekunde' with 'Blinkcode: kurz 1' and 'lang 1'. The code block contains the following logic:

```
if (Sensor 2 EIN) {
  Motor 1 auf -100 %
} else {
  Motor 1 auf 100 %
}
```

The 'wenn' (if) block is connected to the 'Sensor 2' block, and the 'sonst' (else) block is connected to the 'Motor 1' block. The 'Motor 1' block is set to 'auf -100 %' in the 'wenn' branch and 'auf 100 %' in the 'sonst' branch.

3

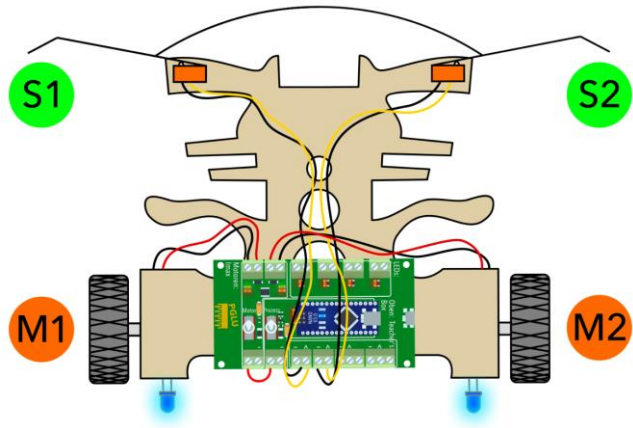


Aufgabe Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde rückwärts

- Wenn Sensor 2 ein Hindernis berührt (S2=Ein), dann lasse Motor 1 rückwärts drehen (M1=-100%). **Tue dies während einer halben Sekunde, auch wenn die Berührung nur sehr kurz war!**
- Wenn Sensor 2 nichts berührt (S2=Aus), dann lasse Motor 1 vorwärts laufen (M1=100%)

The screenshot shows the PGLU programming environment with the 'Code' tab selected. The main workspace contains a 'Hauptloop: 100'000 mal pro Sekunde' block with a 'Blinkcode: kurz 1 lang 1' dropdown. Below it is a logic block: 'Sensor 2 = EIN prüfe standard'. This is followed by a 'wenn/sonst' conditional block. The 'wenn' branch contains 'setze Motor 1 auf -100 %', and the 'sonst' branch contains 'setze Motor 1 auf 100 %'. A 'pausiere 500 ms' block is connected to the end of the 'wenn/sonst' block. The left sidebar shows various tool categories: Aktion, Logik, Zahl, Variable, Mathe, Struktur, Funktion, and Spezial. The 'Sensor' category is highlighted, and yellow lines connect the 'Sensor 2' block in the code to the 'Sensor' category in the sidebar.

4



Erweiterungsaufgabe Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde Rückwärts, mit LED-Blinker

- Wenn Sensor 2 nichts berührt (S2=Aus), soll M1 mit voller Geschwindigkeit vorwärts drehen (M1=100%).
- Wenn Sensor 2 auf ein Hindernis trifft (S2=Ein), soll M1 mit vollem Tempo während einer halben Sekunde rückwärts drehen (M1=-100%). Während dies geschieht soll LED1 3x aufblinken.

The screenshot shows the PGLU programming environment with the following components:

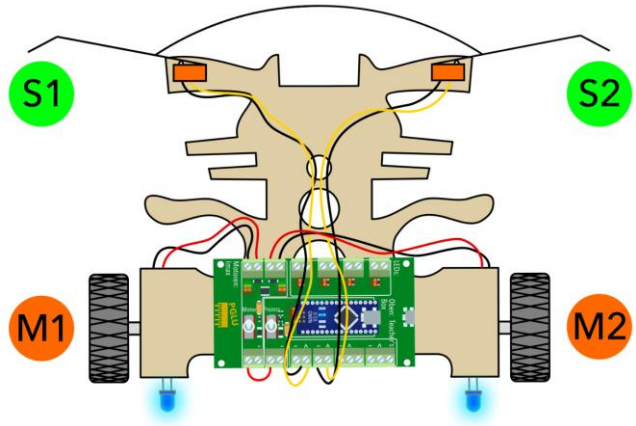
- Header:** PGLU, Grafik, Code, Simulator, Programme.
- Loop:** Hauptloop: 100'000 mal pro Sekunde, Blinkcode: kurz 1, lang 1.
- Sensor:** Sensor 2, EIN, prüfe standard.
- Motor:** setze Motor 1 auf -100 %, setze Motor 1 auf 100 %.
- LED:** setze LED 1 auf EIN, setze LED 1 auf AUS, setze LED 1 auf EIN, setze LED 1 auf AUS, setze LED 1 auf EIN, setze LED 1 auf AUS.
- Logik:** wenn, sonst.
- Flow:** pausiere 100 ms.

The code logic is as follows:

```

Hauptloop: 100'000 mal pro Sekunde
  Sensor 2 (EIN) prüfe standard
  wenn (Sensor 2 = EIN)
    setze Motor 1 auf -100 %
    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
  sonst
    setze Motor 1 auf 100 %
  
```

5



Erweiterungsaufgabe Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde rückwärts – Version «nicht blockierend»

Dieses Programm hat genau den gleichen Effekt, wie dasjenige in Aufgabe 3. Der Unterschied ist, dass nie ein Programmblock «pausiere-500-ms» den Programmablauf stoppt. Das ist gut, wenn du eine Lichtanimation parallel zur Fahrfunktion programmieren möchtest.

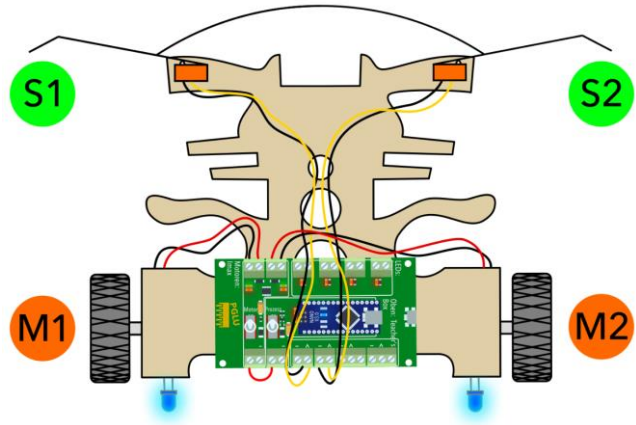
Tip 1: Zeit-in-ms gibt die Anzahl Millisekunden zurück, die seit dem Einschalten des Mikrocontrollers verstrichen sind.

Tip 2: Verstehe workshop.pglu.ch > Sketch > Zeit-in-ms

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

- Vor Hauptloop: 1x**
 - schreibe Zeitstempel Sensor 2 = -501
- Hauptloop: 100'000 mal pro Sekunde**
 - setze Motor 1 auf 100 %
 - wenn Sensor 2 = EIN prüfe standard
 - schreibe Zeitstempel Sensor 2 = Zeit in ms
 - setze Motor 1 auf -100 %
 - wenn Zeit in ms < Zeitstempel Sensor 2 + 500
 - sonst

6



Aufgabe Das Hauptprogramm «Selbstfahrendes Auto»

Programmiere das Selbstfahrende Auto, so dass es sich gleich verhält, wie du es in der Animation beobachtet hast! Verwende deine Erkenntnisse aus den Übungen 1-3. Füge noch blinkende LEDs aus Übung 4 ein!

Falls du die Erweiterungsaufgabe 5 erfolgreich gelöst hast, kannst du die erweiterte Version mit zwei Variablen «Timer 1» und «Timer 2» ausprobieren. Die Lösung dazu findest du in Variante 3.

A screenshot of the PGLU programming environment. The interface is divided into several tabs: Grafik, Code, Simulator, and Programme. The main workspace shows a block-based program. At the top, there is a 'Hauptloop: 100'000 mal pro Sekunde' block with 'Blinkcode: kurz 1' and 'lang 1' options. Below this, there are two 'pausiere 500 ms' blocks. The program uses a 'wenn...sonst' (if...else) logic structure. The 'wenn' blocks are connected to 'Sensor 1' and 'Sensor 2' blocks. The 'Sensor 1' block is set to 'EIN' and 'prüfe standard'. The 'Sensor 2' block is also set to 'EIN' and 'prüfe standard'. The 'sonst' blocks are connected to 'setze Motor 2 auf -100 %' and 'setze Motor 2 auf 100 %' blocks. The 'setze Motor 1 auf -100 %' and 'setze Motor 1 auf 100 %' blocks are also present. The left sidebar shows various categories: Aktion, Sensor, Motor, LED, Pixel, Logik, Frage, Ablauf, Zahl, Variable, Mathe, Struktur, Funktion, Loop, and Spezial.