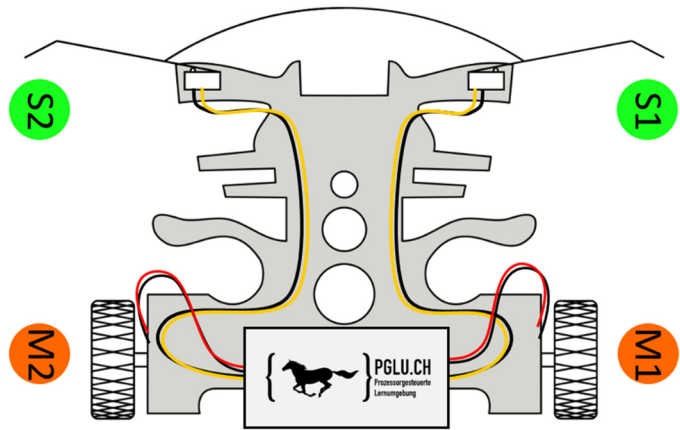


1

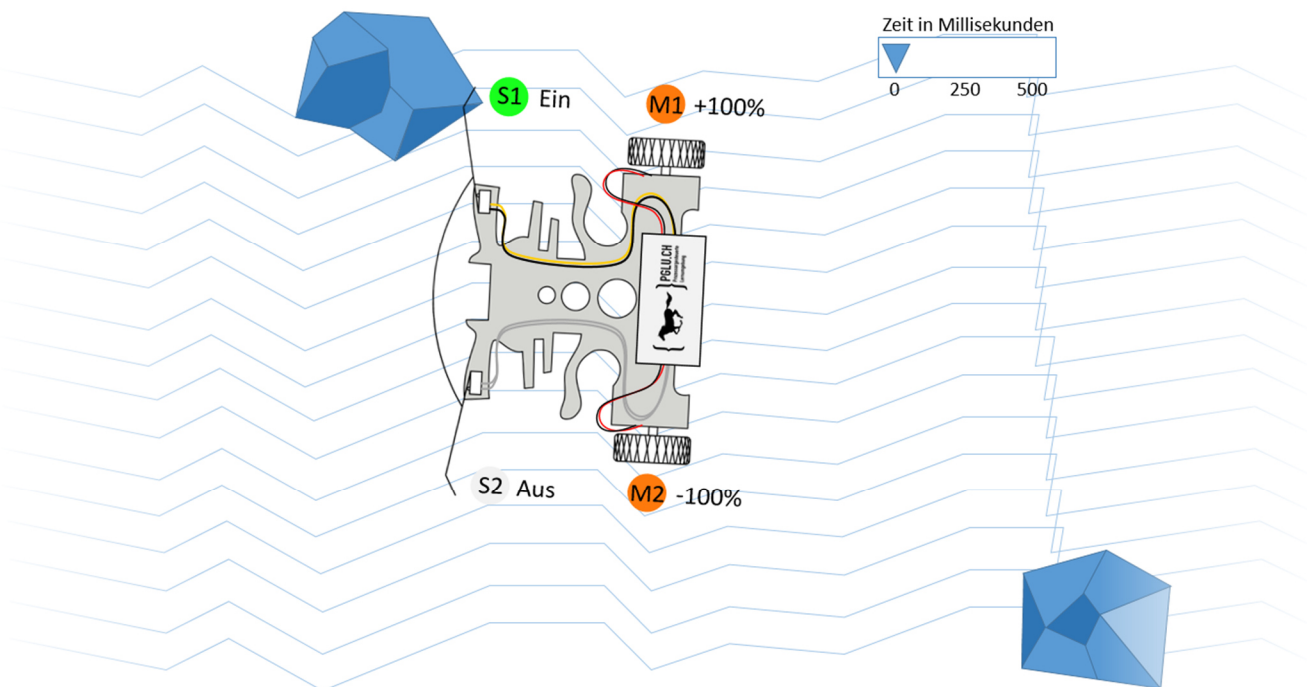


Aufgabe Beschreibe das Auto und das was es tut!

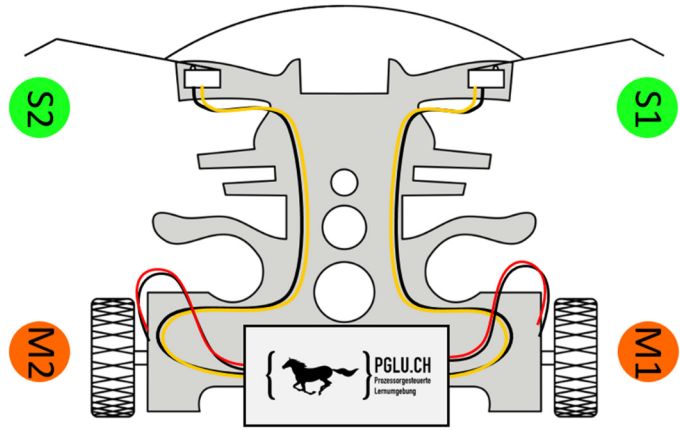
Schaue die Animation und beobachte was geschieht. Stelle dir vor, du müsstest das selbstfahrende Auto einer Person am Telefon erklären, um es ihr zu verkaufen.

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



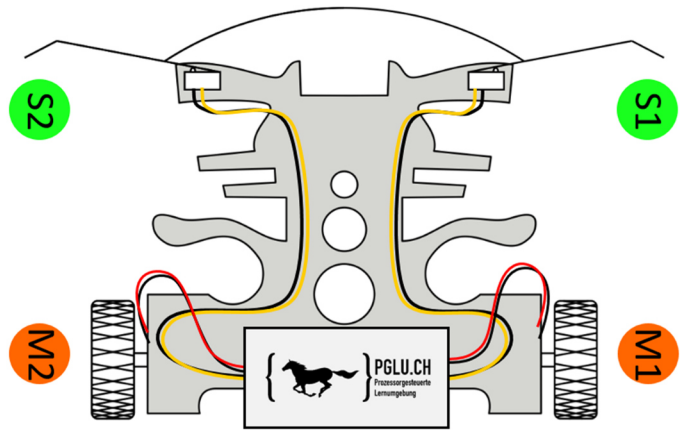
Aufgabe Ein einzelner Motor vorwärts und bei Sensorberührung rückwärts. Beachte: der andere Motor läuft standardmässig rückwärts.

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 a block-based programming environment. The top navigation bar includes 'Programmieren', 'Testen', 'Organisieren', and 'Hilfe'. The main workspace contains a 'Hauptloop: 100'000 mal pro Sekunde' block. Inside the loop, there is a 'Sensor 2' block set to 'EIN' and 'prüfe standard'. This is followed by a 'wenn' block. The 'wenn' block has two branches: one for 'Motor 1' set to 'auf -100 %' and another for 'Motor 1' set to 'auf 100 %'. A 'sonst' block is also present. The left sidebar shows various categories like 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', 'Erweitern', and 'Ablauf II'. Yellow lines connect the 'Sensor 2' block to the 'wenn' block and the two 'setze' blocks.

3

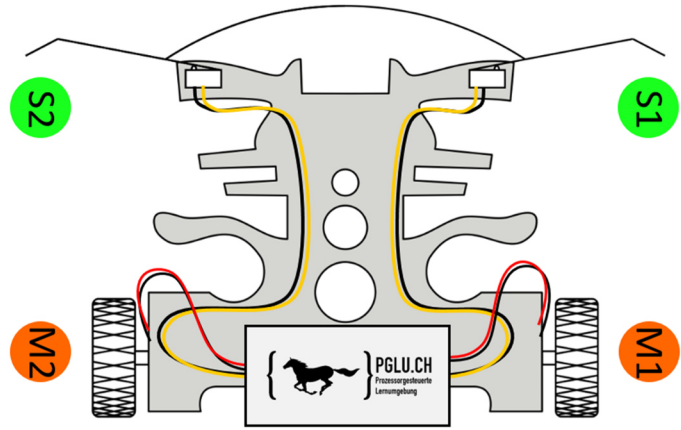


Aufgabe Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde rückwärts

- 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 a block-based programming environment. The top navigation bar includes 'Programmieren', 'Testen', 'Organisieren', and 'Hilfe'. Below this, there are tabs for 'Grafik', 'Code', 'Simulator', and 'Programme'. The main workspace contains a 'Hauptloop: 100'000 mal pro Sekunde' block with 'Blinkcode: kurz 1 lang 1' settings. The 'Sensor' block is set to 'Sensor 2' with a value of 'EIN' and 'prüfe standard'. The 'Motor' block is set to 'Motor 1' with a value of '-100 %'. The 'LED' block is set to 'Motor 1' with a value of '100 %'. The 'Frage' block is set to 'wenn' and 'sonst'. The 'Ablauf' block is set to 'pausiere' with a value of '500 ms'. The 'Dimensionieren' block is set to 'Variable'.

4



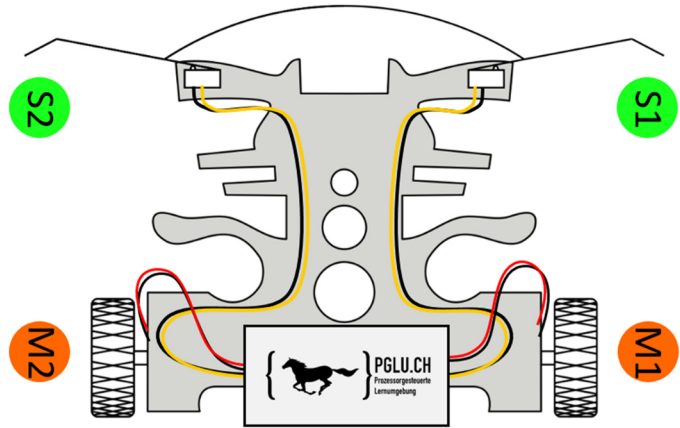
Erweiterungsaufgabe Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde Rückwärts, mit LED-Blinker

- 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 a block-based programming environment. The top navigation bar includes 'Programmieren', 'Testen', 'Organisieren', and 'Hilfe'. Below this are tabs for 'Grafik', 'Code', 'Simulator', and 'Programme'. The main workspace contains a 'Hauptloop: 100'000 mal pro Sekunde' block with a 'Blinkcode: kurz 1 lang 1' dropdown. The logic is as follows:

- Sensor 2** is checked for 'EIN' (ON) using a 'Sensor 2 = EIN prüfe standard' block.
- If Sensor 2 is ON:**
 - Motor 1 is set to -100% ('setze Motor 1 auf -100 %').
 - Motor 1 is set to 100% ('setze Motor 1 auf 100 %').
 - A 'wenn' (if) block is used to trigger a sequence of LED 1 actions:
 - 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
- If Sensor 2 is NOT ON (sonst):** Motor 1 is set to 100% ('setze Motor 1 auf 100 %').

5



Erweiterungsaufgabe Ein einzelner Motor vorwärts und bei Sensorberührung eine halbe Sekunde rückwärts - ohne Programmpause

Programmiere den Motor gleich wie in Aufgabe 3. vermeide aber den Block «pausiere 500ms». Dieser führt dazu, dass das Programm pausiert und in dieser Zeit keine Sensorberührungen registriert werden können!

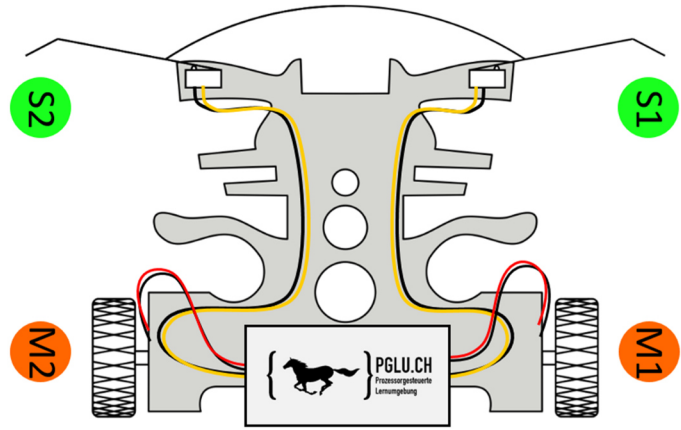
Tipp: Benenne eine Variable mit «Timer». Sie soll im Millisekundentakt von 0 bis unendlich hochzählen. Setze «Timer» durch jede Sensorberührung wieder auf 0! Lasse M1 rückwärts laufen, solange «Timer» kleiner als 500 ist.

The screenshot shows a programming environment with a dark background and a sidebar on the left. The sidebar has categories: Agieren, Sensor, Motor, LED, Steuern, Frage, Ablauf, Dimensionieren, Variable, Mathe, Vereinfachen, Funktion, Erweitern, and Ablauf II. The main workspace contains the following code blocks:

- Vor Hauptloop: 1x** (red box): `schreibe Timer = 500`
- Parallel zu Hauptloop: alle 1ms** (red box): `schreibe Timer = Timer + 1`
- Hauptloop: 100'000 mal pro Sekunde** (blue box): `Blinkcode: kurz 1 lang 1`
- if-else block** (blue box):
 - wenn** (blue box): `Timer < 500`
 - sonst** (blue box):
 - `setze Motor 1 auf -100 %`
 - `setze Motor 1 auf 100 %`
- if block** (blue box): `wenn` `Sensor 2 = EIN prüfe standard`
- schreibe Timer = 0** (yellow box)

Yellow lines connect the 'Timer' variable in the code blocks to the 'Variable' category in the sidebar.

6



Aufgabe Das Hauptprogramm «Selbstfahrendes Auto»

Programmiere das Selbstfahrende Auto, so dass es sich gleich verhält, wie du es 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.

The screenshot shows a programming interface with a dark background. At the top, there are navigation tabs: 'Programmieren' (with sub-tabs 'Grafik', 'Code'), 'Testen' (with sub-tab 'Simulator'), 'Organisieren' (with sub-tab 'Programme'), and 'Hilfe'. A horse logo is in the top left corner.

The main workspace contains a script area with the following blocks:

- Agieren:** A 'Hauptloop: 100'000 mal pro Sekunde' block with 'Blinkcode: kurz 1' and 'lang 1' dropdowns.
- Steuern:** Two 'wenn' blocks with 'sonst' blocks. The first 'wenn' block has a 'pausiere 500 ms' block attached to its 'wenn' part. The second 'wenn' block has a 'pausiere 500 ms' block attached to its 'sonst' part.
- Dimensionieren:** Two 'Sensor' blocks. The first is 'Sensor 1' with a dropdown set to 'EIN' and 'prüfe standard'. The second is 'Sensor 2' with a dropdown set to 'EIN' and 'prüfe standard'.
- Vereinfachen:** Four 'setze' blocks. Two are for 'Motor 2' (set to -100% and 100%) and two are for 'Motor 1' (set to -100% and 100%).

On the left side, there is a vertical menu with categories: 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', 'Erweitern', and 'Ablauf II'. Yellow lines connect the 'Sensor' and 'Motor' categories to the corresponding blocks in the script area.