



Mit MATRIX "Roboter Sumo" lernen

Beispiel mit MATRIX Essential Set(MR0001)

Erstellt von Water Xu & MATRIX Robotics am 24.09.2025

Benötigte Materialien



Panzertape (optional)

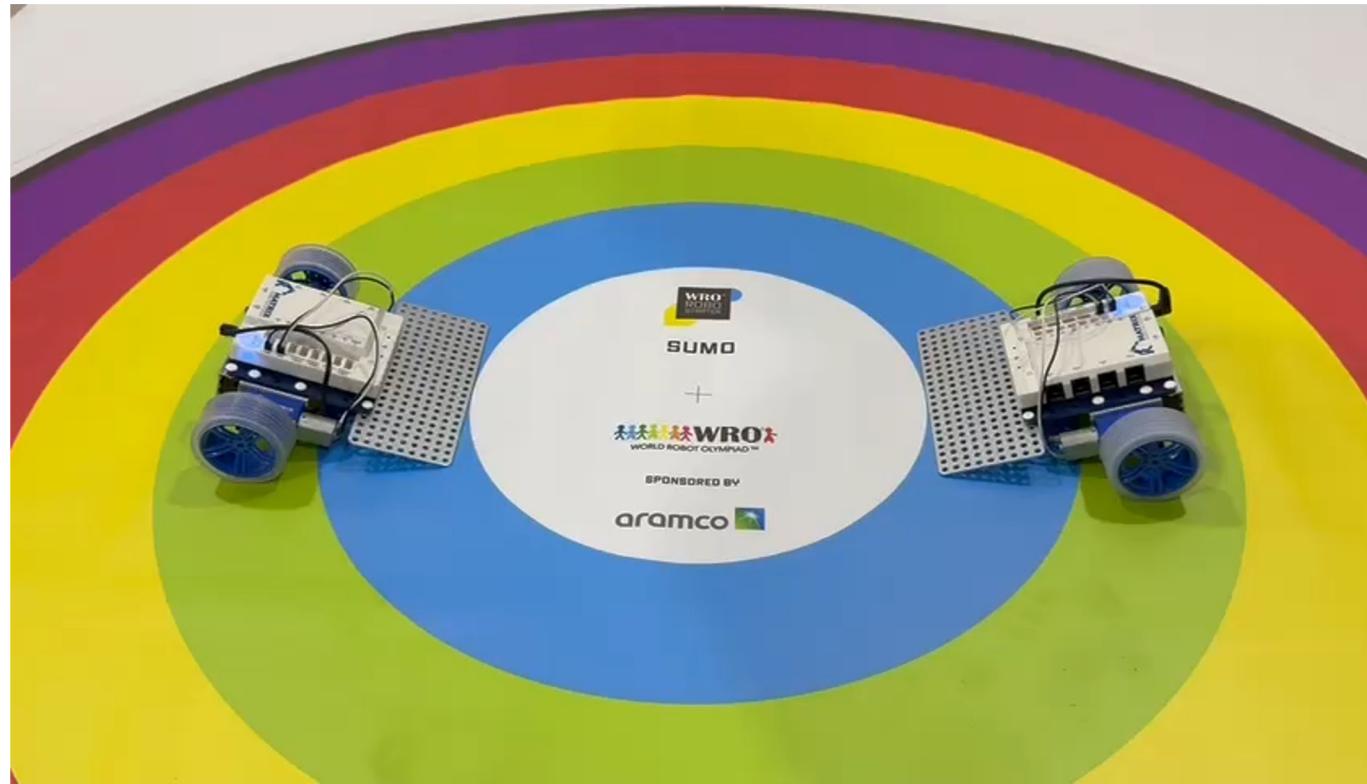


MATRIX Mini Essential Set V2

Was ist ein Sumo-Roboter?



Was ist ein Sumo-Roboter?



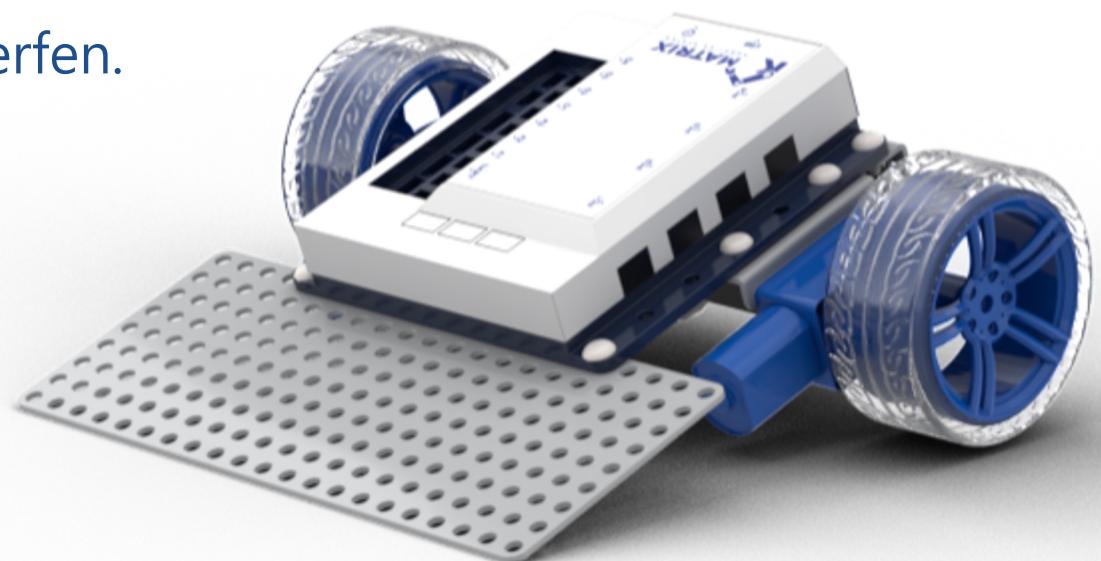
Was sollte ein starker Sumo-Roboter alles haben?

Was sollte ein starker Sumo-Roboter alles haben?

| Design Fokus | Strukturelle Eigenschaften | Vorteile | Nachteile |
|-------------------|---|----------------------|--|
| Stabilität | breites Chassis, niedrigen Schwerpunkt | schwierig umzuwerfen | weniger Vortrieb |
| Offensive | spitz zulaufender oder abgewinkelter Rumpf | hohe Aufschlagskraft | verliert schneller die Kontrolle, wird leichter umgeworfen |

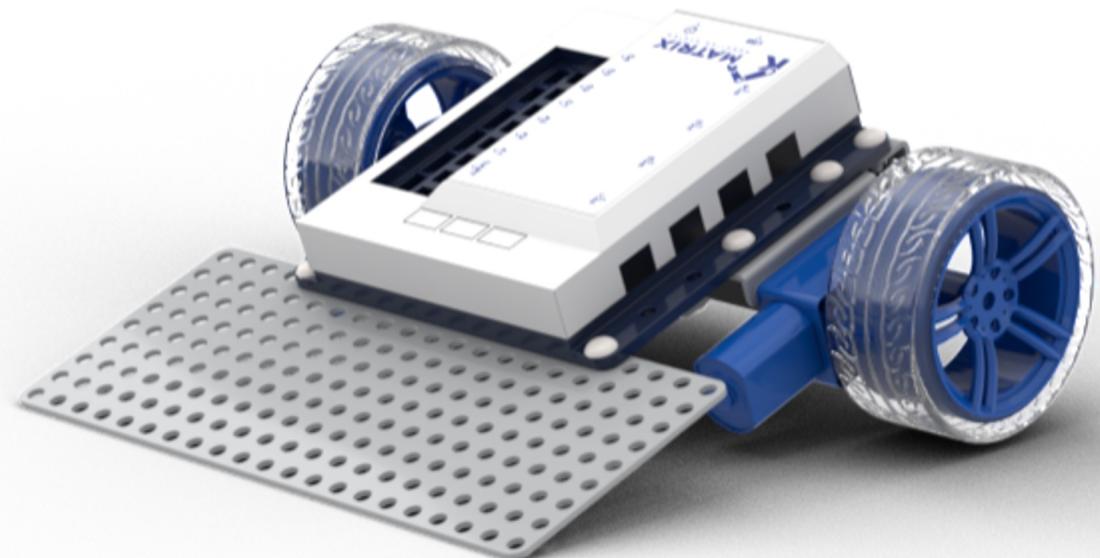
Lernziele:

1. Roboter-Sumo verstehen.
2. Anpassen von Parametern und Veränderungen verstehen.
3. Spielstrategien für Roboter Sumo entwerfen.



Praktisches Beispiel

1. Sumo-Roboter Aufbau (15 Minuten)
2. Sumo-Roboter Programmierung, Parameter optimieren (15 Minuten)

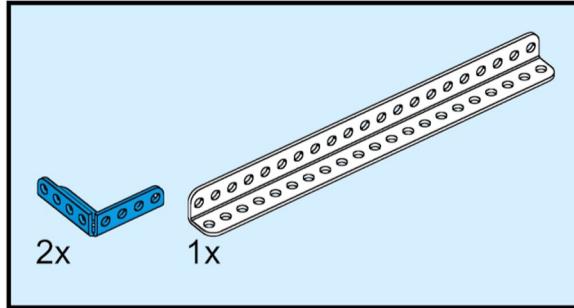




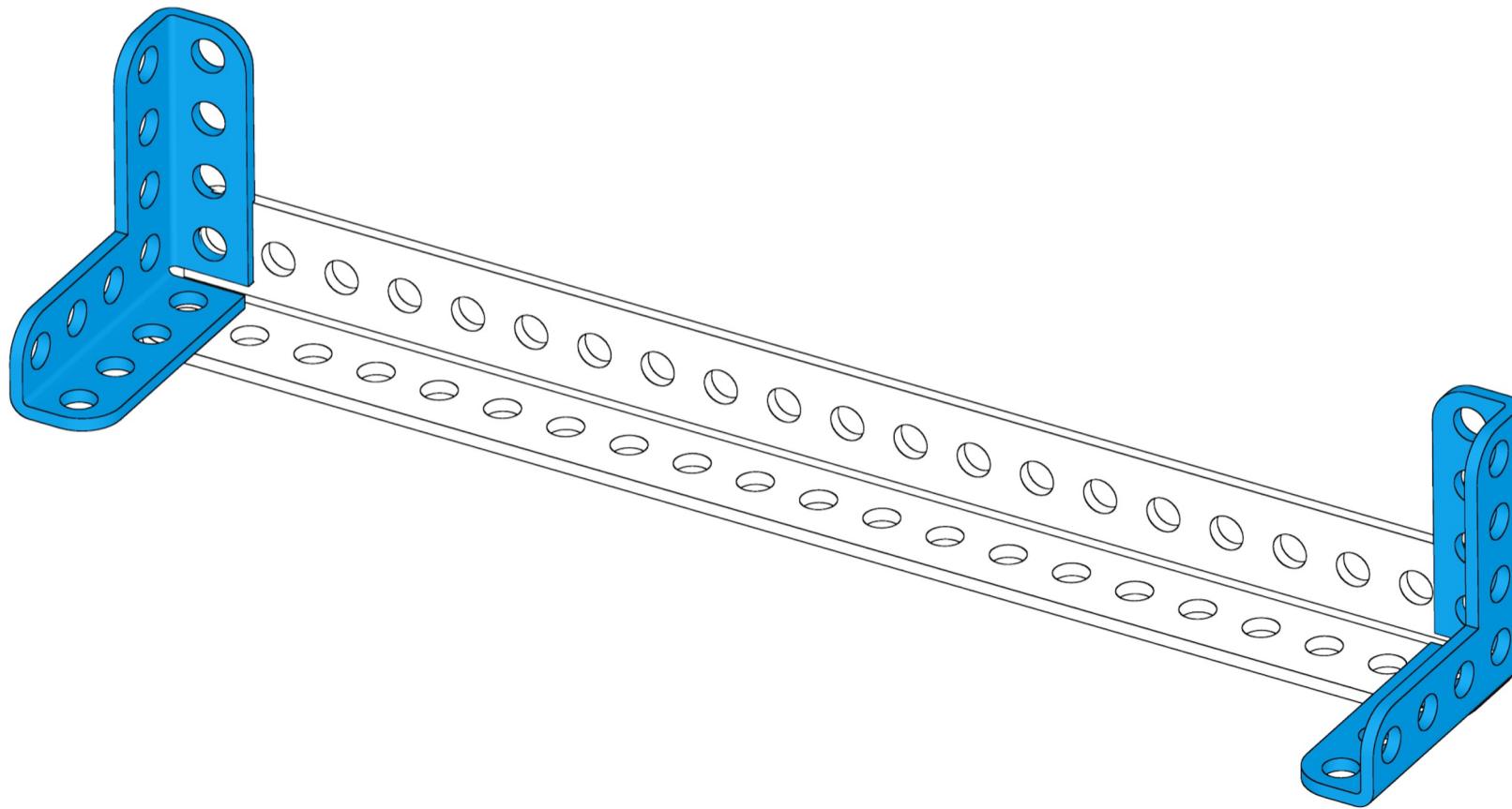
Sumo-Roboter Aufbau

STEAM EDUCATION, FUTURE TECHNOLOGY.

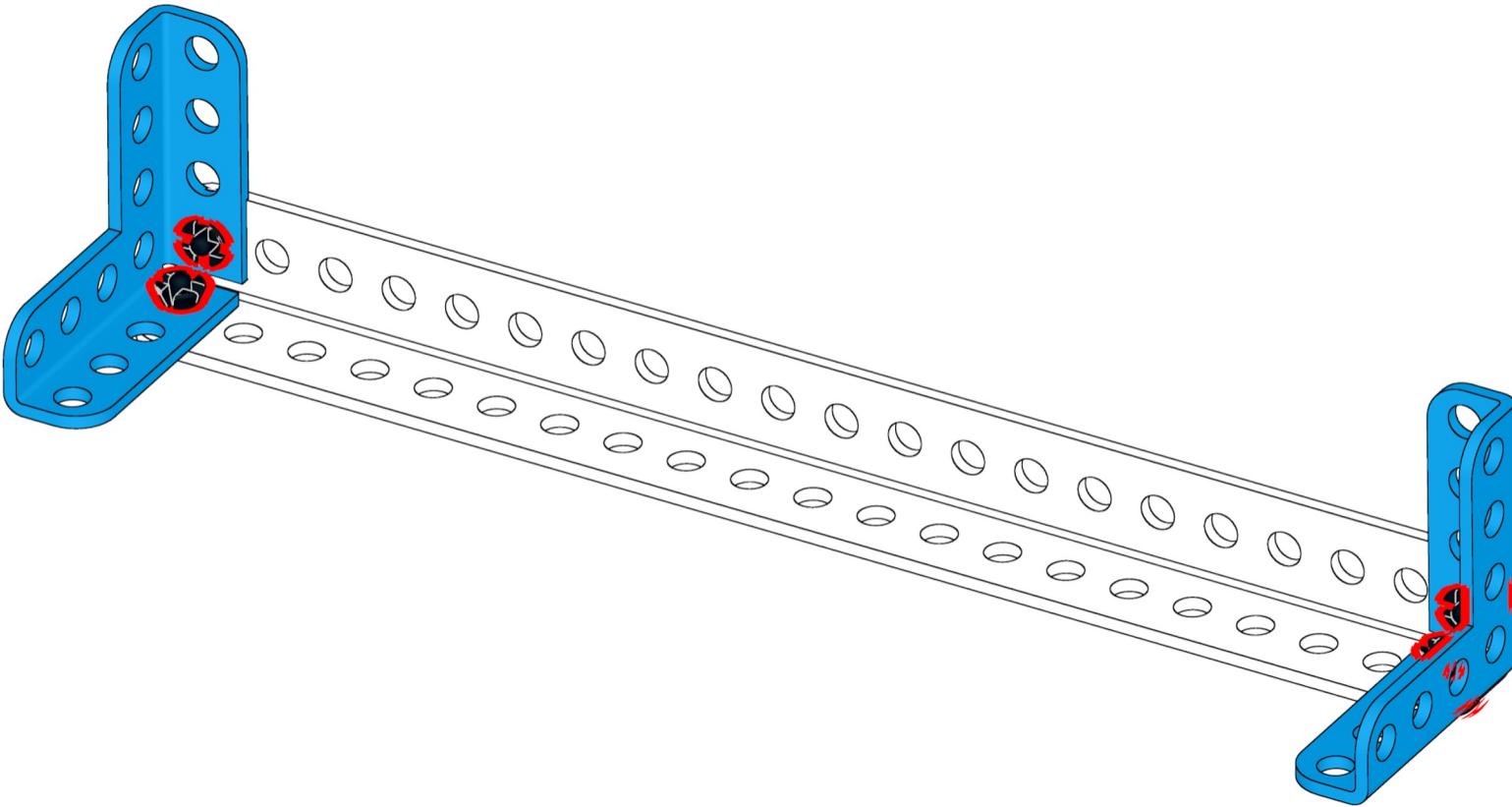
1



Baue das Angriffsmodul

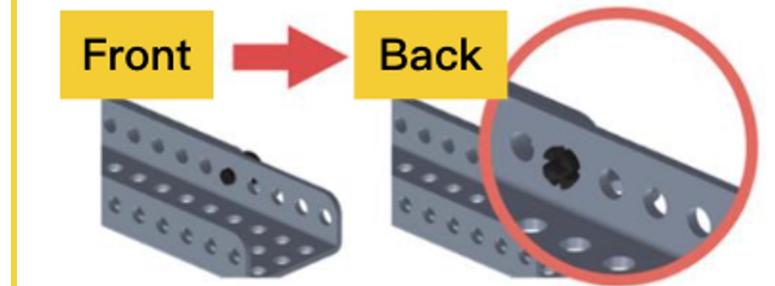


2



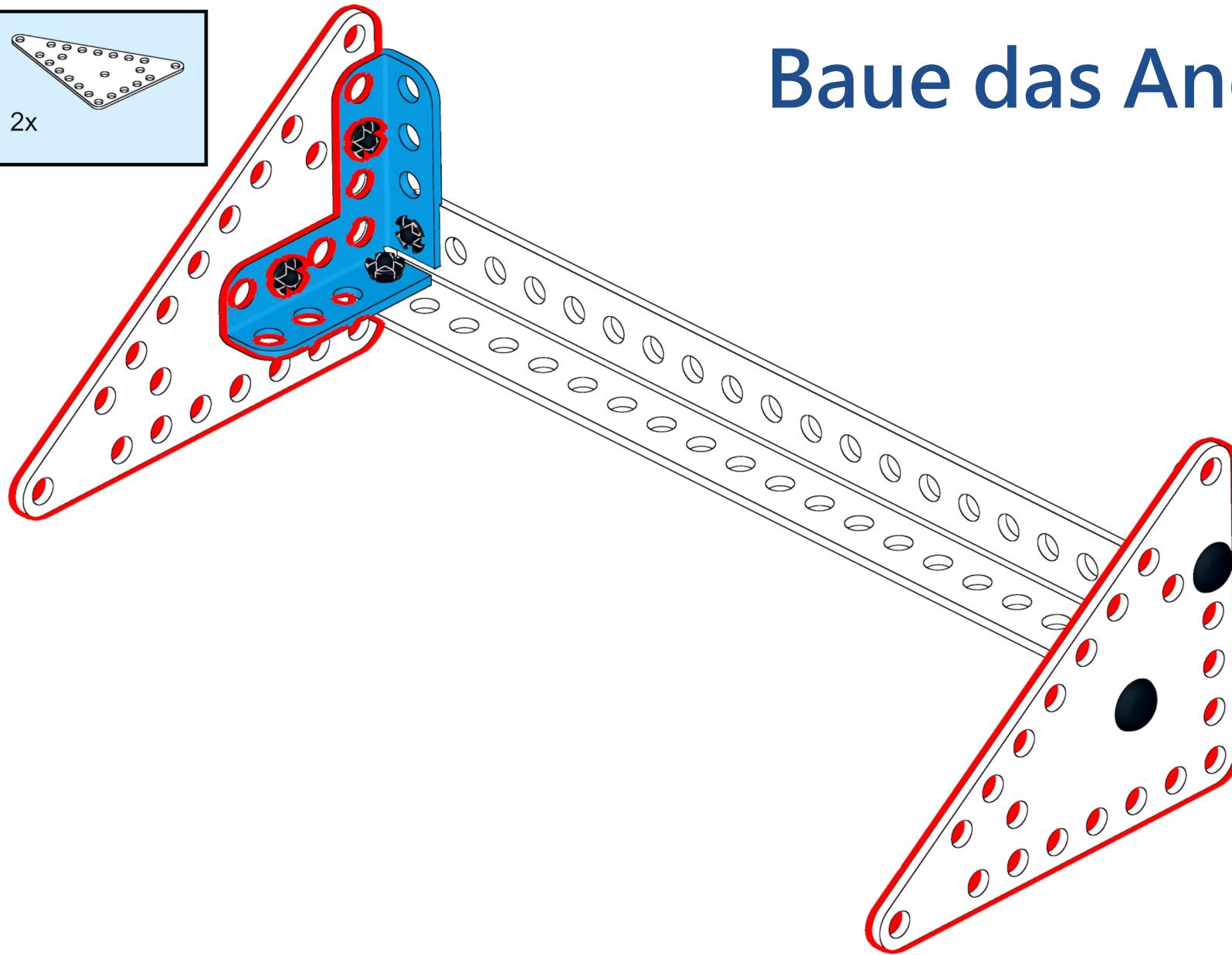
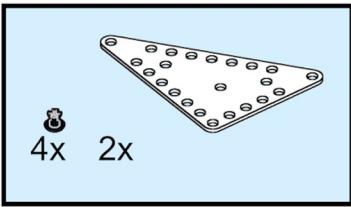
Baue das Angriffsmodul

Tips für den Quick Connector



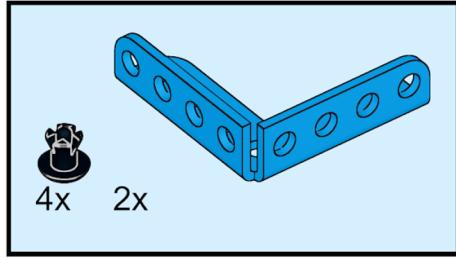
When the connector is inserted,
the claws will spread out.

3

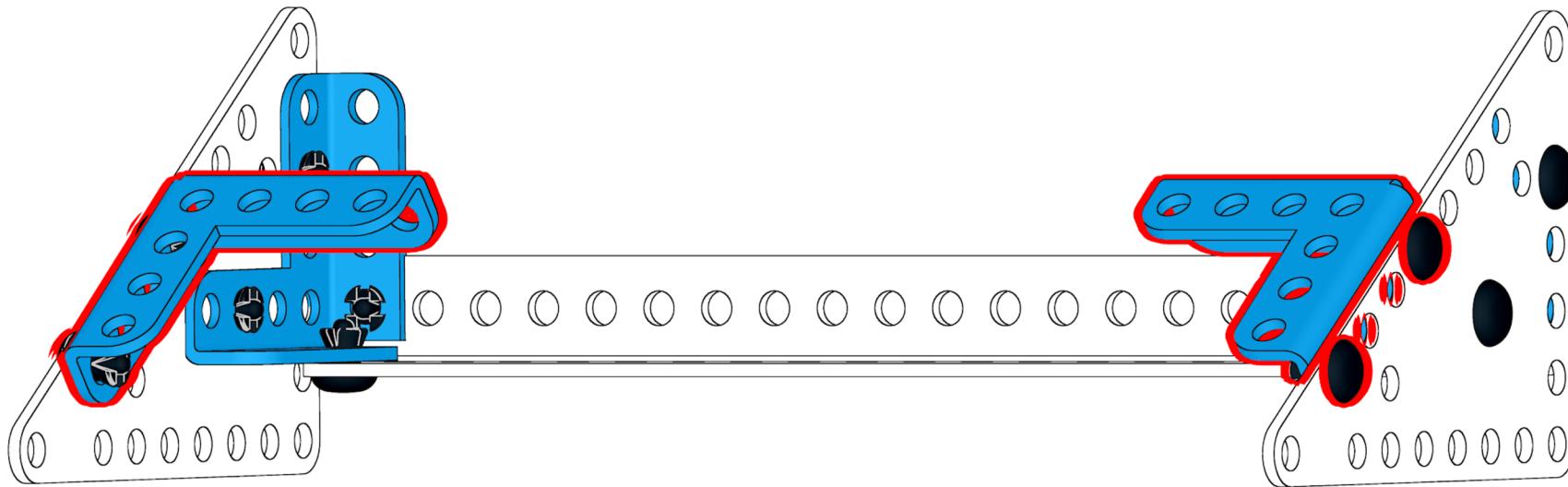


Baue das Angriffsmodul

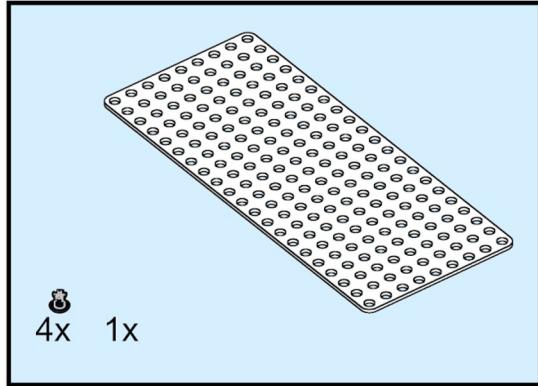
4



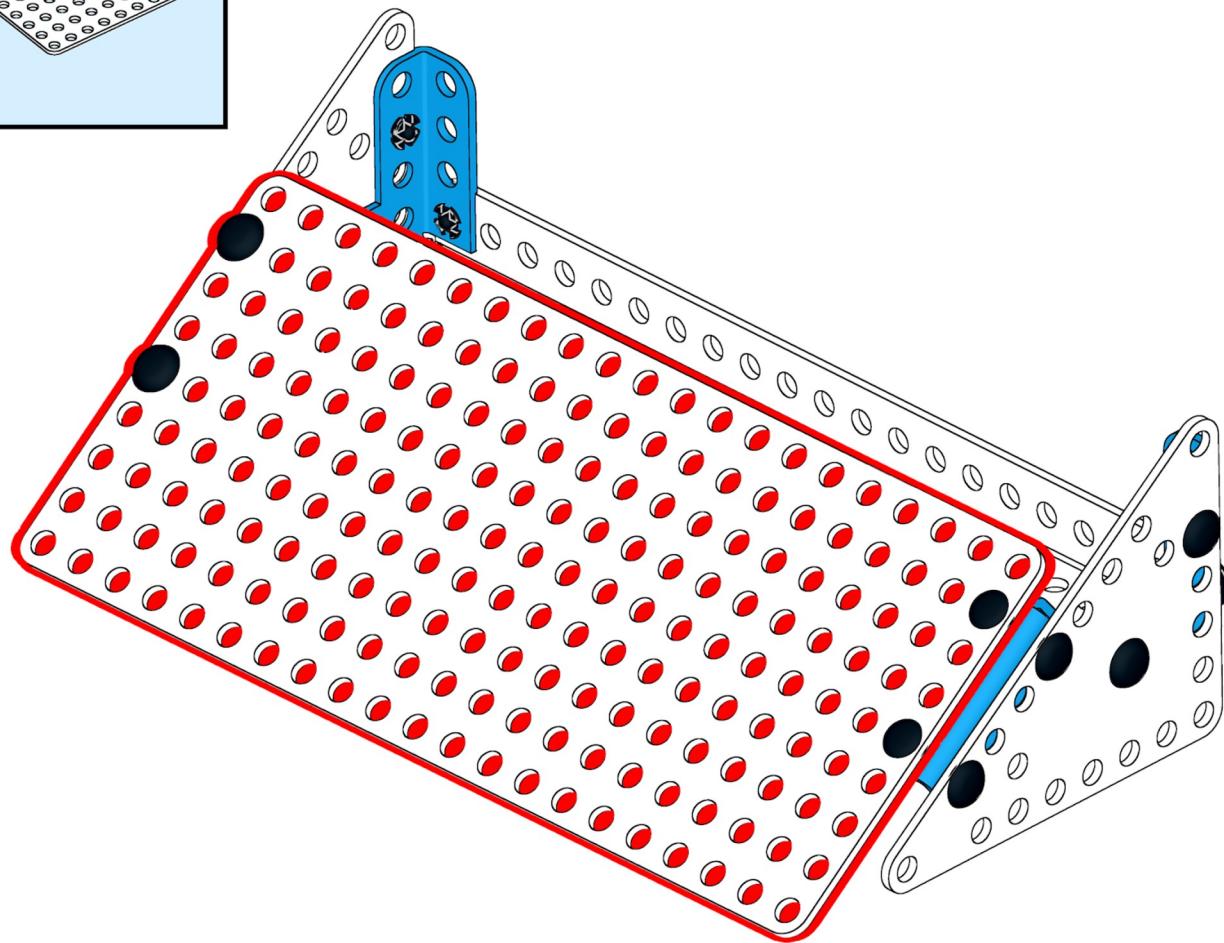
Baue das Angriffsmodul



5

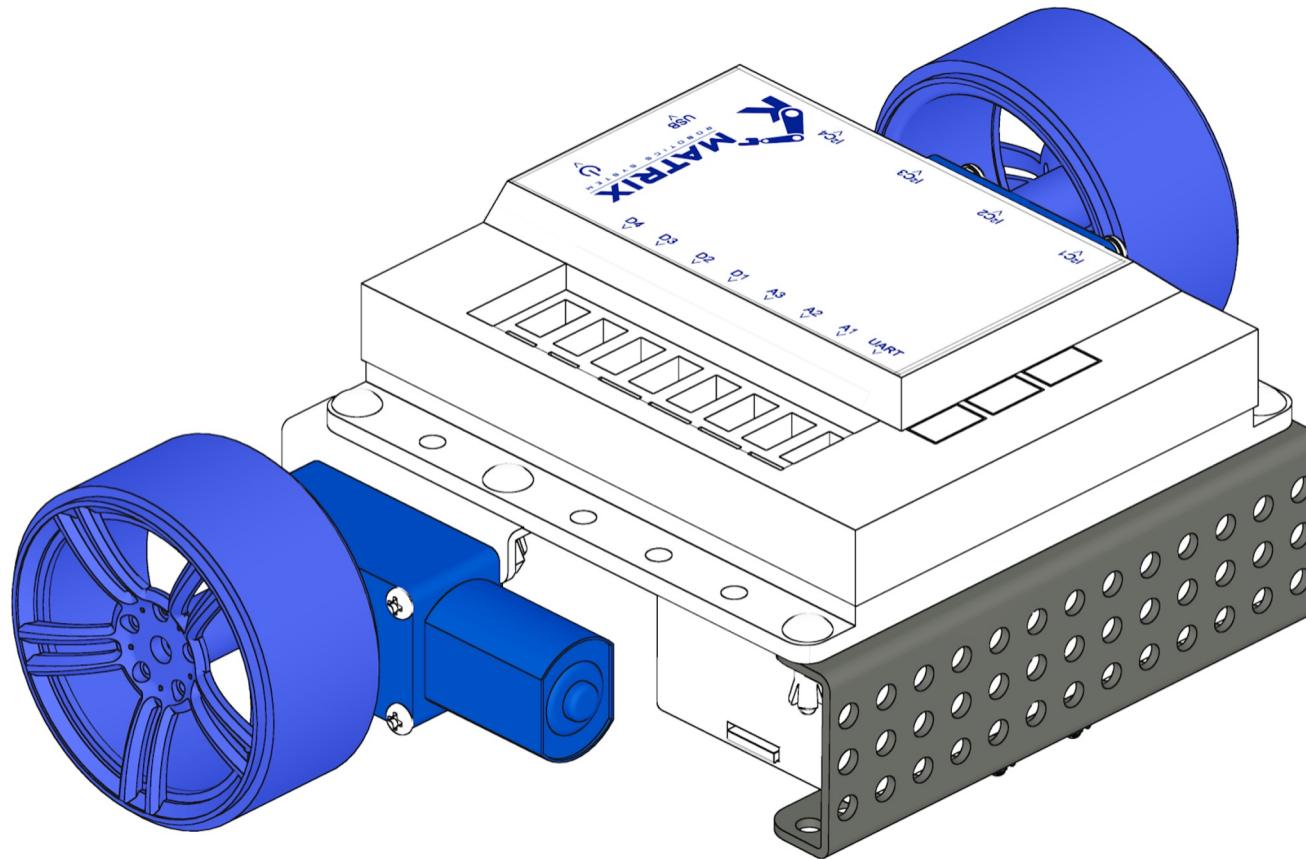


Baue das Angriffsmodul

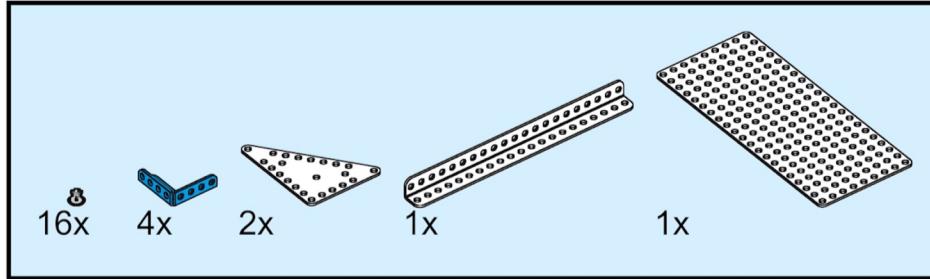


1

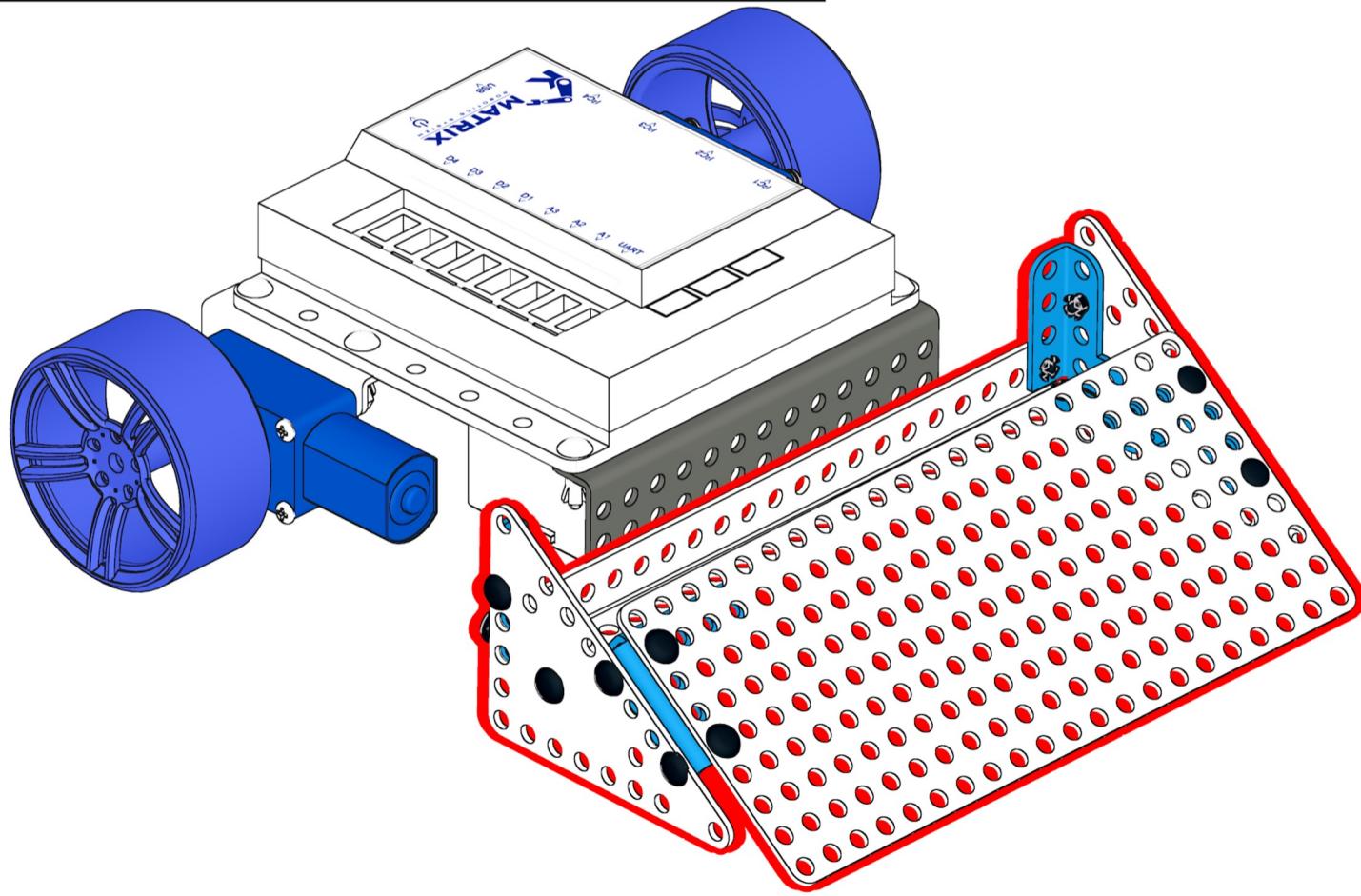
Verbinde dein Fahrzeug aus Lektion 2
mit dem Angriffsmodul



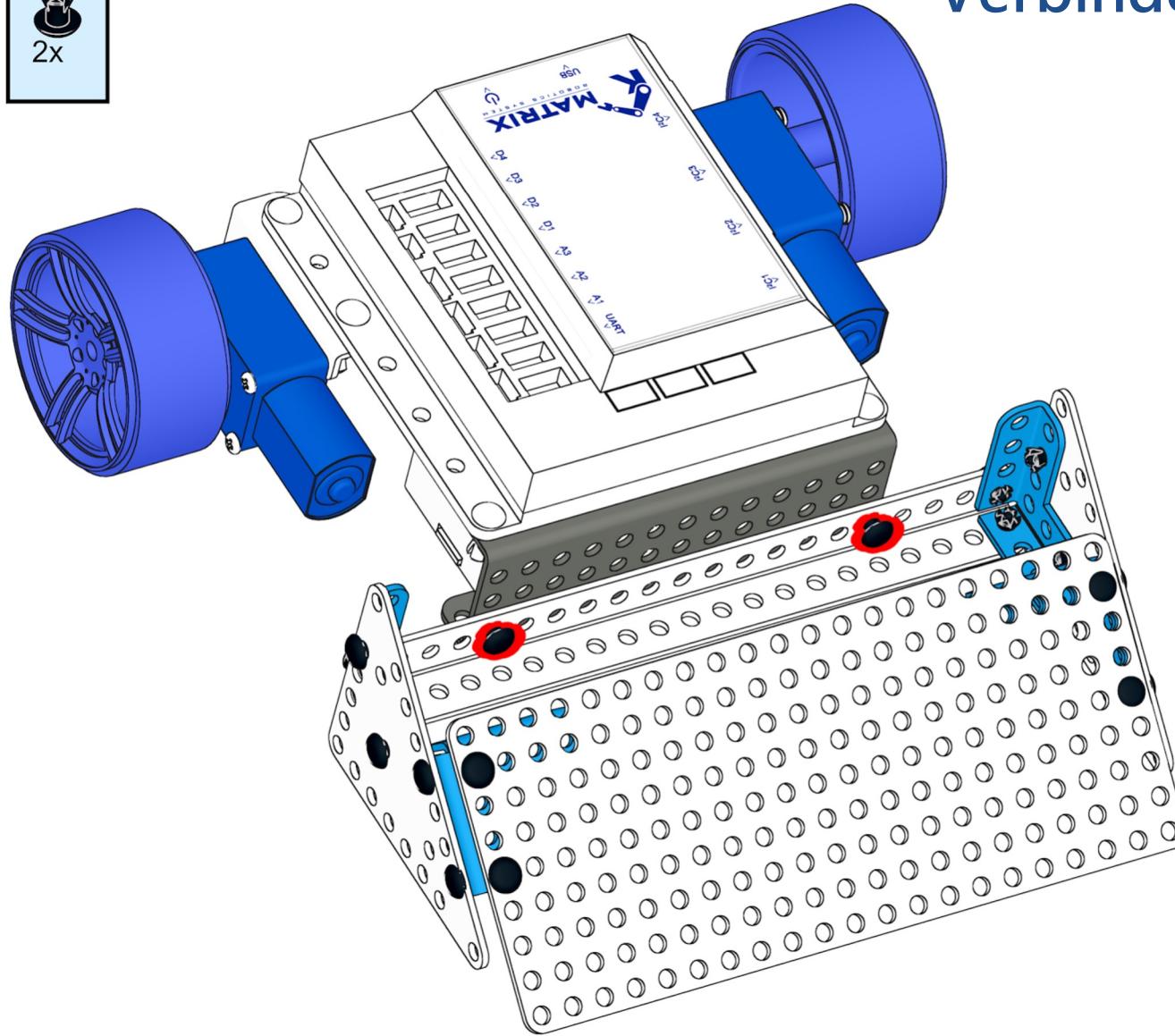
2



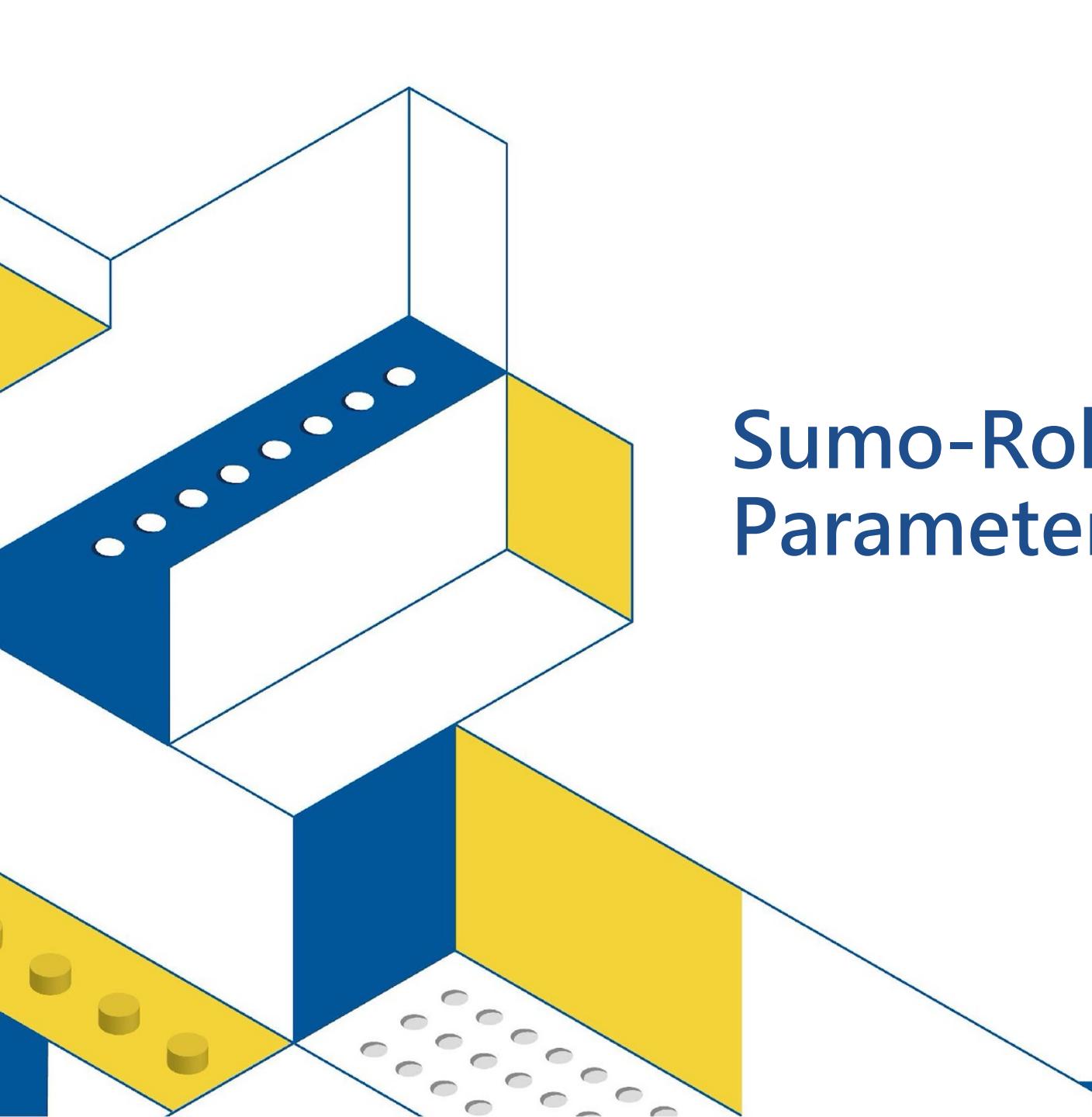
Verbinde dein Fahrzeug aus Lektion 2 mit dem Angriffsmodul



3

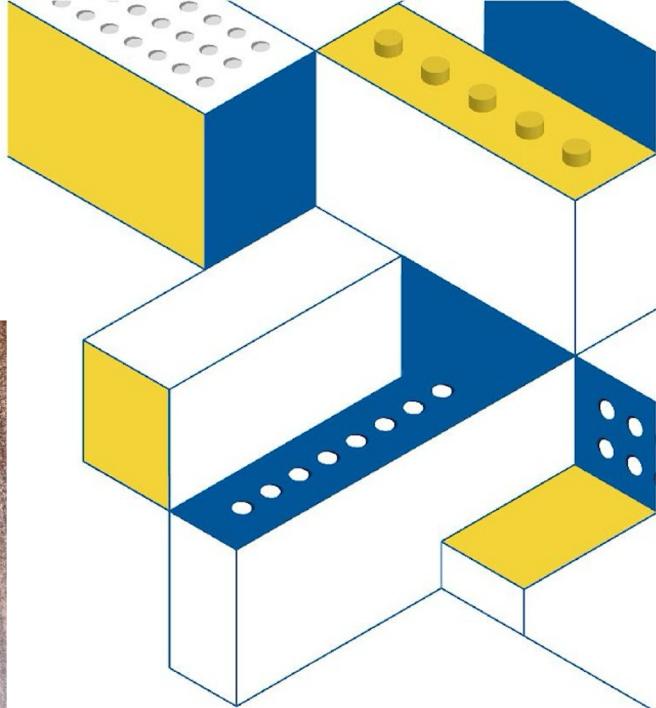
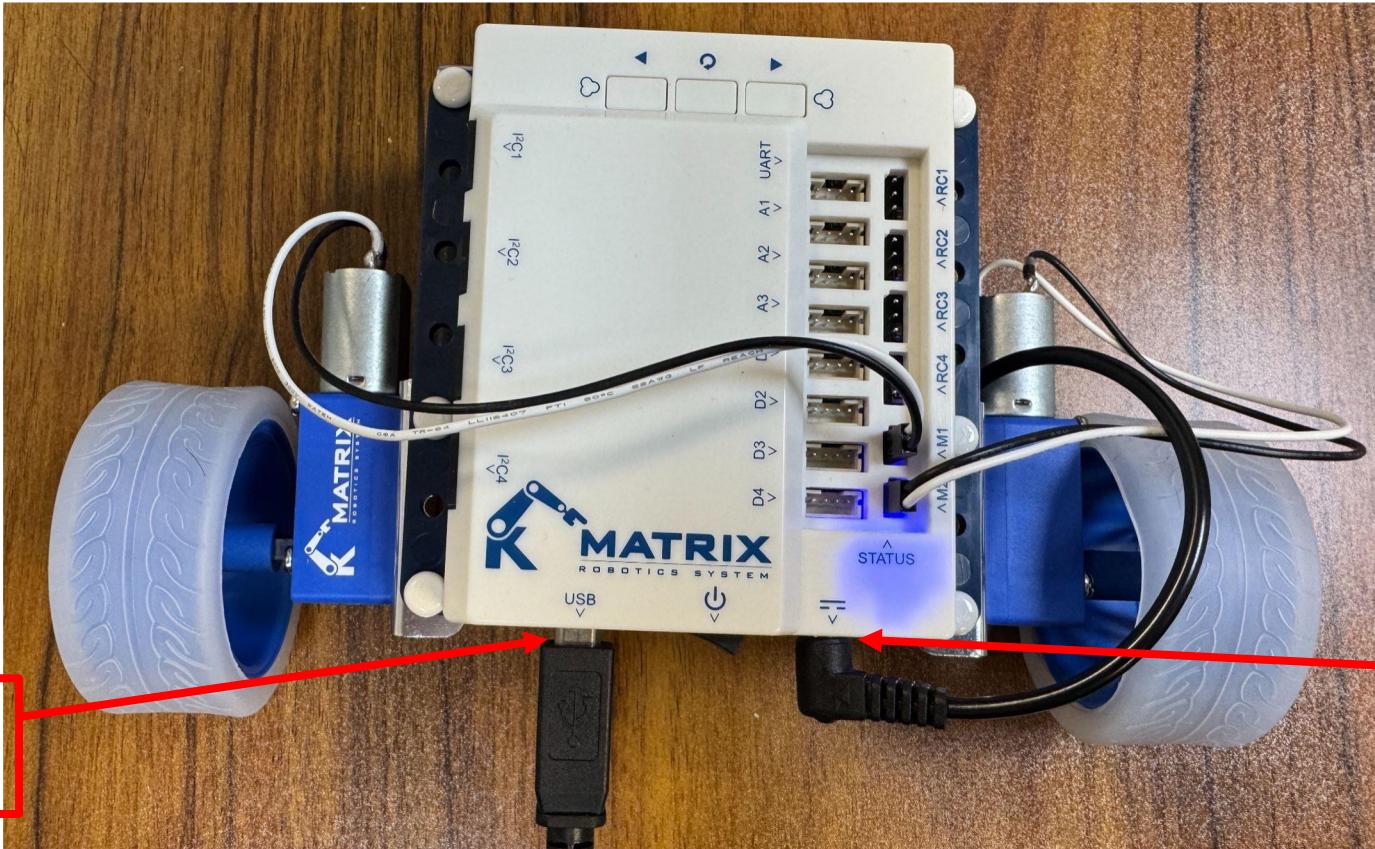


Verbinde dein Fahrzeug aus Lektion 2
mit dem Angriffsmodul



Sumo-Roboter Programmierung, Parameter optimieren

Verkabelung





File ▾

*Example6.mbn2

✖ ✅ ⌂ No Device

Mini Core

Mini Begin 18650x2 ▾ UART: On ▾ Baud: 9600 ▾

Setup

Loop

Mini RGB LED LED1 ▾ R: 25 G: 0 B: 0

Mini DC Motor M1 ▾ Power: 50

Mini Servo Motor RC1 ▾ Angle: 50

Mini Button Button1 ▾ is Pressed?

Mini Ultrasonic D1 ▾ Distance(cm)

Mini GPIO

Mini D1 ▾ Digital Signal

Mini D1 ▾ Set to HIGH ▾

Mini A1 ▾ Analog Signal

Serial

Serial Print Hello

Serial Print World with New Line

Serial Write (ASCII) 65

Serial Chart DataSet 32 64 128 , Interval 500 ms

Is Serial Available?

Ziehe den Hauptprogramm Block heraus

#include "MatrixMini.h"

void setup()

{

 Mini.begin(LI_2, 0, 9600);

 Serial.begin(9600);

}

8

void loop()

{

11

}

12



Send



STEAM EDUCATION, FUTURE TECHNOLOGY.



File ▾

*Example6.mbn2

✖ ✅ ⌂ No Device

Mini Core

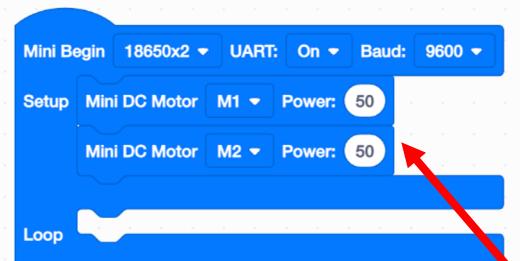
- Mini Core
- Serial
- Sensing
- Extension Boxes
- 3rd Party Sensors
- Control
- Operators
- Variables
- My Blocks

Mini GPIO

- Mini D1 Digital Signal
- Mini D1 Set to HIGH
- Mini A1 Analog Signal

Serial

- Serial Print Hello
- Serial Print World with New Line
- Serial Write (ASCII) 65
- Serial Chart DataSet 32 64
- Is Serial Available?



1. Ziehe zwei Motorkontrolle Blocks heraus.

2. Setze M1, M2 power auf 50.

```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.M1.set(50);
8     Mini.M2.set(50);
9 }
10
11 void loop()
12 {
13 }
```



File ▾

*Example6.mbn2

No Device

- Control
- Mini Core
- Serial
- Sensing
- Extension Boxes
- 3rd Party Sensors
- Control
- repeat (10)
- Operators
- Variables
- My Blocks

wait 1000 millisecond

2. Ziehe einen Warteblock heraus

Mini Begin 18650x2 ▾ UART: On ▾ Baud: 9600 ▾

Setup Mini DC Motor M1 ▾ Power: 50

Mini DC Motor M2 ▾ Power: 50

wait 10000 millisecond

Loop

1. Select Control

2. Setze wait auf 10000 Millisekunden
(fahre 10 Sekunden lang geradeaus)

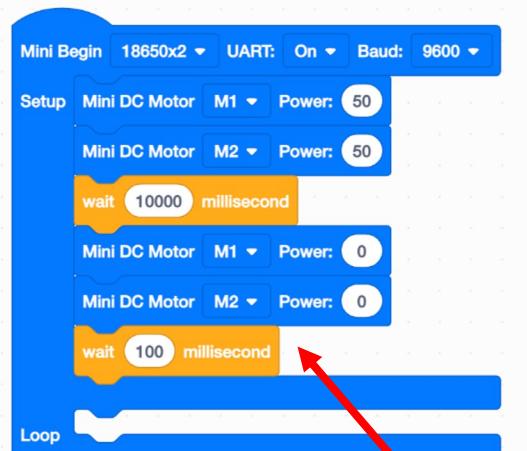
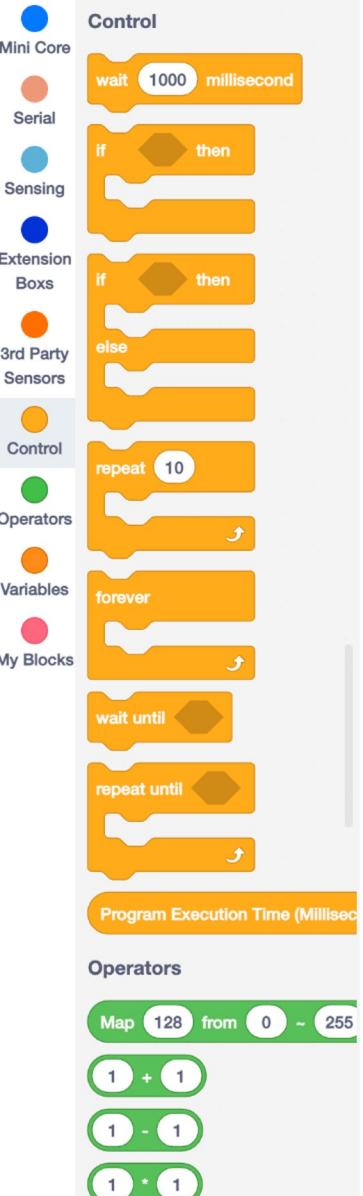
```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.M1.set(50);
8     Mini.M2.set(50);
9     delay(10000);
10 }
11
12 void loop()
13 {
14 }
```



File ▾

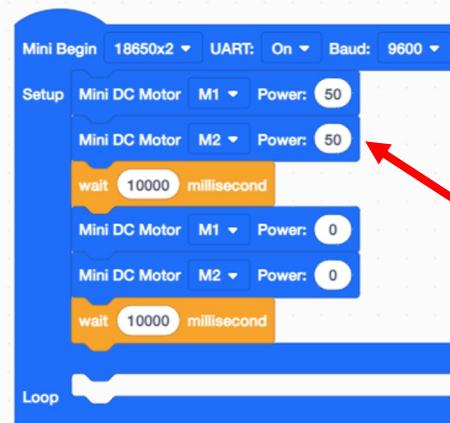
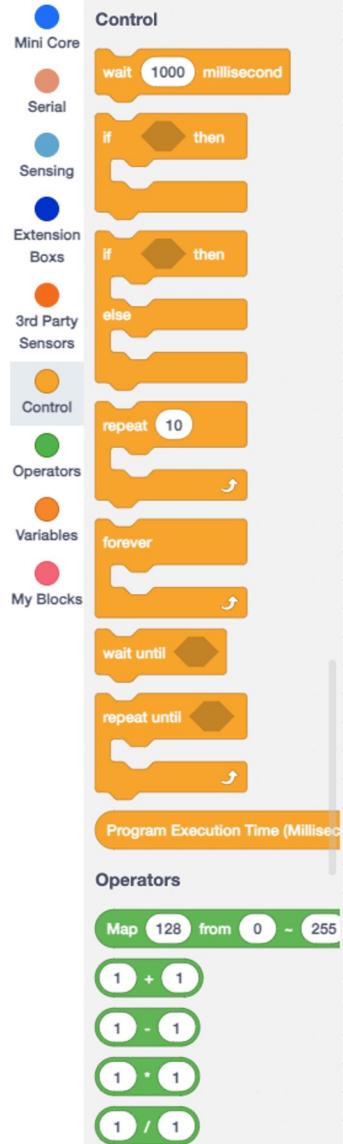
*Example6.mbn2

No Device



```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.M1.set(50);
8     Mini.M2.set(50);
9     delay(10000);
10    Mini.M1.set(0);
11    Mini.M2.set(0);
12    delay(100);
13 }
14
15 void loop()
16 {
17 }
```

Ziehe ein Paar von M1 und M2 blocks heraus,
setze power auf 0 und warte 100
Millisekunden.
(Dieses Programm stoppt die Bewegung).



```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.M1.set(50);
8     Mini.M2.set(50);
9     delay(10000);
10    Mini.M1.set(0);
11    Mini.M2.set(0);
12    delay(10000);
13 }
14
15 void loop()
16 {
```

Erklärung von Motor Leistung und Geschwindigkeit

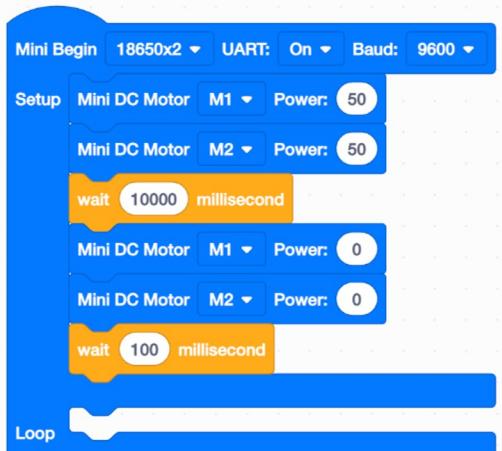
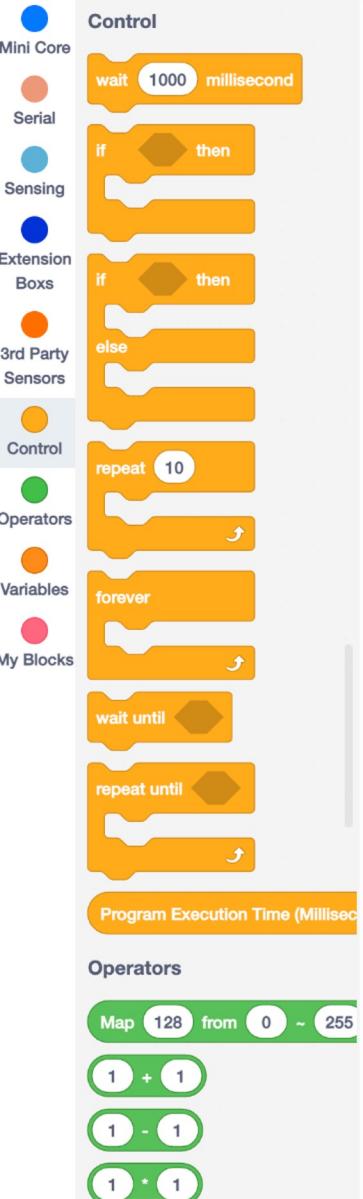
1. **Leistung (Power)** stellt die Spannung und Stromstärke im Motor ein.
2. Je größer die Leistung, desto schneller dreht sich der Motor; je kleiner die Leistung, desto langsamer dreht sich der Motor.
3. Ist der Wert für Leistung positiv, dreht sich der Motor nach vorne; ist der Wert negativ nach hinten.
4. Weil jeder Motor im Herstellungsprozess ein kleines bisschen anders ist als die anderen, kann es sein, dass zwei Motoren bei gleicher Leistung unterschiedlich schnell drehen.
5. Falls der Roboter nicht in einer geraden Linie fährt, ist einer der Motoren schneller als der andere.
6. In diesem Fall musst du die Motorleistung individuell anpassen, sodass beide gleich arbeiten.



File ▾

*Example6.mbn2

No Device



```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(U_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.M1.set(50);
8     Mini.M2.set(50);
9     delay(10000);
10    Mini.M1.set(0);
11    Mini.M2.set(0);
12    delay(100);
13 }
14
15 void loop()
```

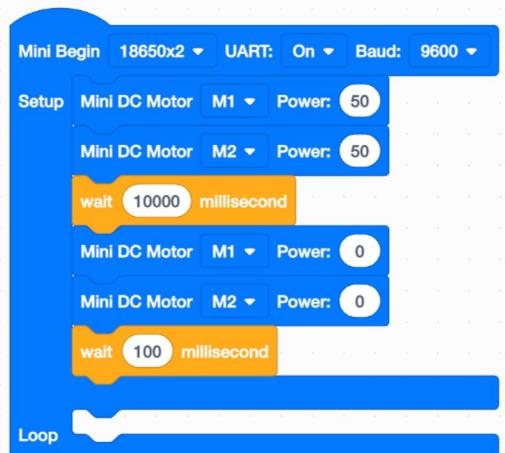
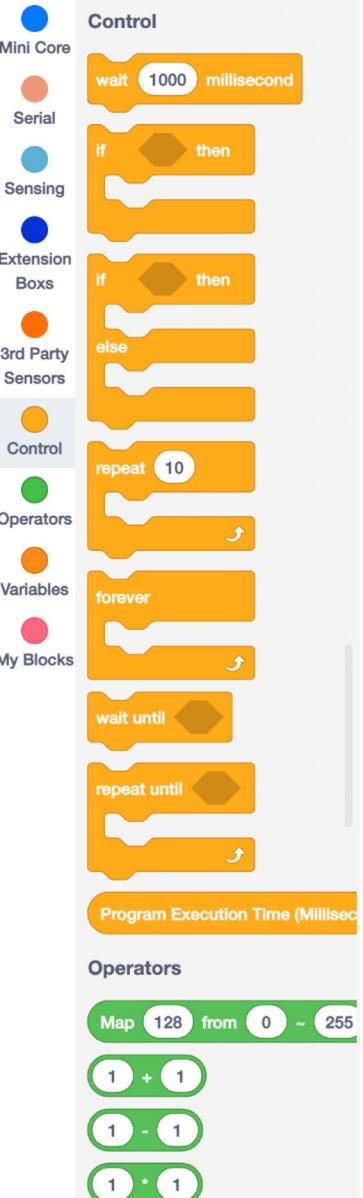
Wähle den kleinen
Roboter aus.



File ▾

*Example6.mbn2

X ✓ ⌂ No Device



```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.M1.set(50);
8     Mini.M2.set(50);
9     delay(10000);
10    Mini.M1.set(0);
11    Mini.M2.set(0);
12    delay(100);
13 }
14
15 void loop()
16 {
17 }
```

Hier klicken zum
Hochladen auf
den Roboter.

Send ⚙

Aufgabe für den Unterricht

Tragt die ersten Sumo Spiele mit anderen Teams aus!

Der Roboter, der zuerst das Spielfeld verlässt, verliert!

Falls noch kein Spielfeld vorhanden ist oder keins frei ist, kann man mit Panzertape auch eine 1m x 1m Fläche abkleben und benutzen!

