

# Workshop #1

## Grundlagen, Motorsteuerung, Schleifen

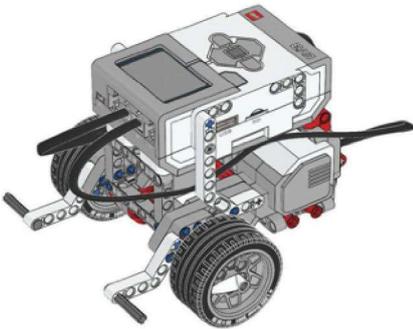
---

### Inhalte des Workshops

- Bauen und Kennenlernen des EV3 System
- Einstieg in die grafische Programmieroberfläche
  - Aufbau der grafischen Oberfläche
  - Wie kommt das Programm auf den Roboter?
  - Ansteuerung von Motoren
  - Sequentielle Programmierung
  - Programmierung von Schleifen
  - Ausgabe über Text, Bild, Ton und Statusleuchten

## Aufgabe 1: Aufbau des Standard Roboters

Baue den Standard Roboter (Bauanleitung bis Seite 41, Online PDF bis Seite 48)  
Download Bauanleitung: <http://goo.gl/Oc1GcH>



Der Standard Roboter stellt die Basis für die 4 Workshops dar.

## Aufgabe 2: Motoren – Standardsteuerung

- Der Roboter soll für 2 Sekunden gerade aus fahren und anschließend stehen bleiben.
- Probiere aus, wie sich die unterschiedlichen Einstellungen auswirken (Umdrehungen, Zeit, Grad).

**Tipp:** Mit Hilfe der Standardsteuerung können die beiden großen Motoren einfach angesteuert werden.



Standardsteuerung

## Aufgabe 3: Motoren – Alternative Steuerungen

- Der Roboter soll im Kreis fahren.
- Zusatz:** Finde heraus, wie sich die verschiedenen Einstellungen auf den Radius des Kreises auswirken.

**Tipp:** Verwenden den Großen Motor oder die Hebelsteuerung, um die beiden großen Motoren einzeln ansteuern zu können.

Großer Motor



Hebelsteuerung



## Aufgabe 4: Parcours – Eins nach dem Anderen

- Der Roboter soll mehrere Schritte hintereinander ausführen:
  1. 10 Umdrehungen nach vorne fahren
  2. 3 Sekunden warten
  3. 10 Umdrehungen rückwärts fahren
- Zusatz:** Der Roboter soll sich um 180° drehen und dann zurückfahren.
- Zusatz:** Bau auf dem Boden einen Parcours mit verschiedenen Gegenständen (Federpenal, LegoBausteine, Schultaschen, ...). Programme den Roboter nun so, dass er sich durch den Parcours bewegt, ohne die Gegenstände zu berühren.

## Aufgabe 5: Parcours – Vereinfachung mit Schleifen

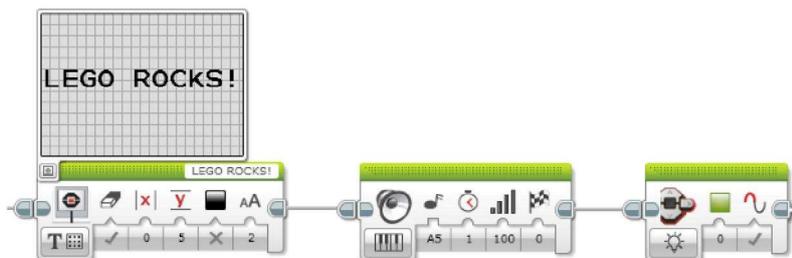
- Bau auf dem Boden einen quadratischen Parcours. Programme den Roboter so, dass er einmal komplett den Parcours abfährt und wieder an der Startposition zu stehen kommt.
- Erweitere dein Programm so, dass der Roboter 2 x den Parcours abfährt.
- Das Programm wird so ziemlich schnell unübersichtlich. Verwende den Baustein Schleife um den Roboter 4 x hintereinander den Parcours absolvieren zu lassen.



Mit eine Schleife können Anweisungen wiederholt werden.

## Aufgabe 6: Ausgabe

- Gib einen beliebigen Text über den Baustein Anzeige auf dem Display des Bricks aus
- Verwende den Baustein Klang, um einen beliebigen Ton für 3 Sekunden abzuspielen
- Lass die Statusleuchten des Bricks in den Farben rot, gelb und grün blinken. Verwende dazu den Baustein Stein-Statusleuchte.
- Zusatz:** Programme mit dem Baustein Klang die Melodie „Alle meine Entchen“ und spiele die Melodie mit Hilfe einer Schleife 3 x hintereinander ab.
- Zusatz:** Programme mit dem Baustein Anzeige eine Laufschrift, die von rechts nach links durch den Bildschirm läuft (x- und y-Werte verwenden). Verwenden eine Schleife, um die Laufschrift 5 x durch den Bildschirm laufen zu lassen.



Ausgabe über das Display, den Statusleuchten und dem Lautsprecher

# Workshop #2

## Sensoren, Bedingungen

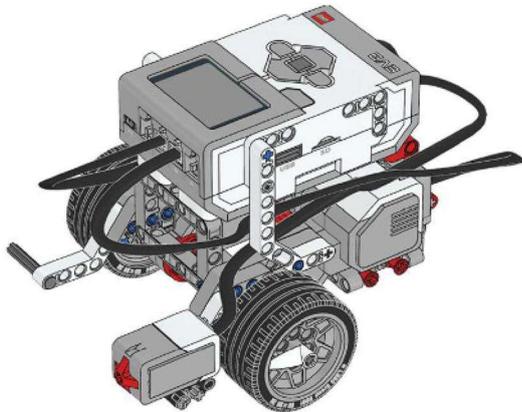
---

### Inhalte des Workshops

- Verbauen und Kennenlernen der **Sensoren** des EV3 Systems
- Programmierkenntnisse
  - Auslesen von **Sensor-Werten**
  - **Bedingungen** – Reagieren auf Sensor-werte
  - Wiederholung **Schleifen**

## Aufgabe 1: Berührungssensor

Baue den Berührungssensor-Aufsatz (Bauanleitung Seite 73–76, Online PDF bis Seite 63–67)  
Download Bauanleitung: <http://goo.gl/Oc1GcH>

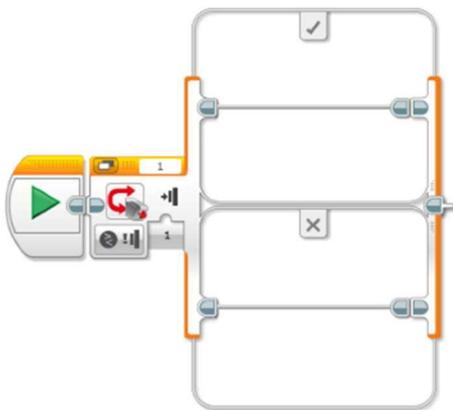


Der Berührungssensor reagiert wie ein Schalter.

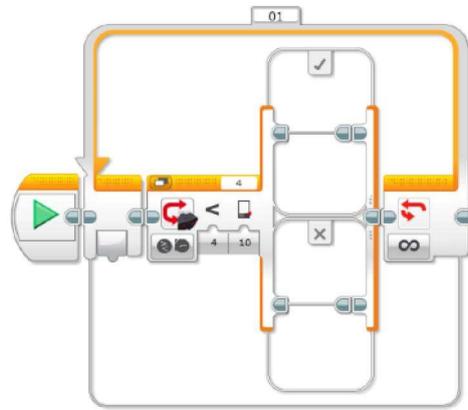
### Aufgabe 1.a: Schalter – Reagieren auf eine Startbedingung

Ist der Berührungssensor beim Start des Programms gedrückt, so soll der Roboter 3 Sekunden vorwärts fahren, sonst rückwärts. Anschließend soll der Roboter stehen bleiben.

**Tipp:** Mit einem Schalter kannst du auf verschiedene Sensor-Werte reagieren.



Schalter



Oft ist es hilfreich, Schalter mit Schleifen zu kombinieren.

### Aufgabe 1.b: Schalter – Kontinuierliche Überprüfung von Sensoren

Immer, wenn der Berührungssensor gedrückt ist, so soll die Statusleuchte rot aufleuchten. Ist der Berührungssensor nicht gedrückt, so soll die Statusleuchte nicht leuchten.

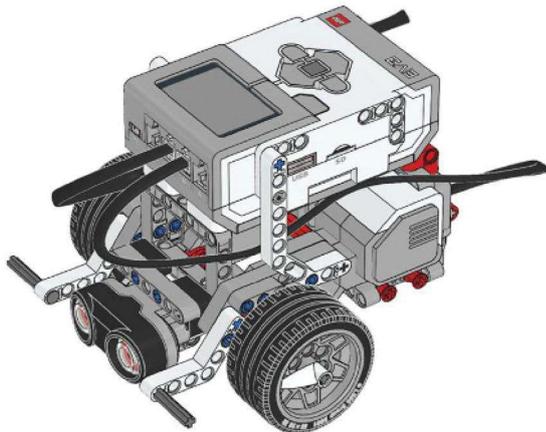
**Tipp:** Verwende eine Schleife, damit der Roboter eine Aufgabe laufend erfüllen kann.

## Aufgabe 2: Ultraschallsensor

**Achtung:** Sollte der Ultraschallsensor-Baustein nicht in der Programmieroberfläche vorhanden sein, so muss der Treiber dafür installiert werden:

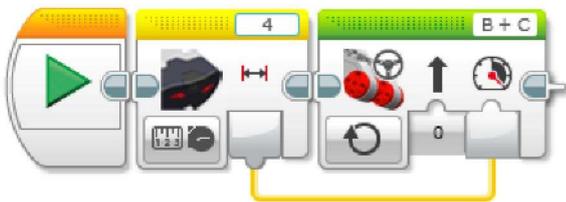
Der Treiber dafür ist unter <https://www.lego.com/de-de/mindstorms/downloads> im Bereich **EV3SOFTWAREBLÖCKE** zu finden. Einfach herunterladen und in der Programmieroberfläche mit **Werkzeuge** ⇒ **Blöcke importieren** auswählen und **Importieren**.

Baue den Ultraschallsensor-Aufsatz (Bauanleitung Seite 42–47, Online PDF bis Seite 68–76)



Mit dem Ultraschallsensor kann man Abstände messen.

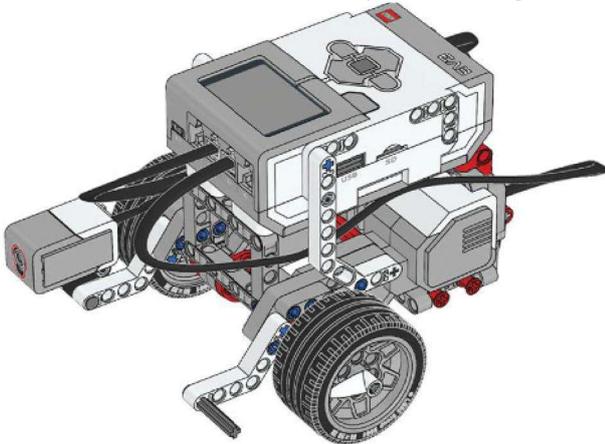
- Der Roboter fährt solange vorwärts, bis er ein Hindernis in weniger als 10cm Entfernung erkennt und bleibt dann stehen. Sobald das Hindernis weg ist, soll der Roboter weiterfahren, bis wieder ein Hindernis erkannt wird.
- Zusatz:** Der Roboter fährt bis auf 10 cm an ein Hindernis heran, fährt im Anschluss um 30 cm zurück, dreht um 45 Grad und beginnt dann von neuem.
- Zusatz:** Verschachtelte Bedingung – Der Roboter fährt bis auf 50 cm an ein Hindernis heran und wird dann mit abnehmender Distanz langsamer. Sobald der Berührungssensor aktiv wird, soll der Roboter sofort stehen bleiben.



Der gemessene Abstand kann zum Beispiel auch als Wert für die Geschwindigkeit verwendet werden.

## Aufgabe 3: Farbsensor

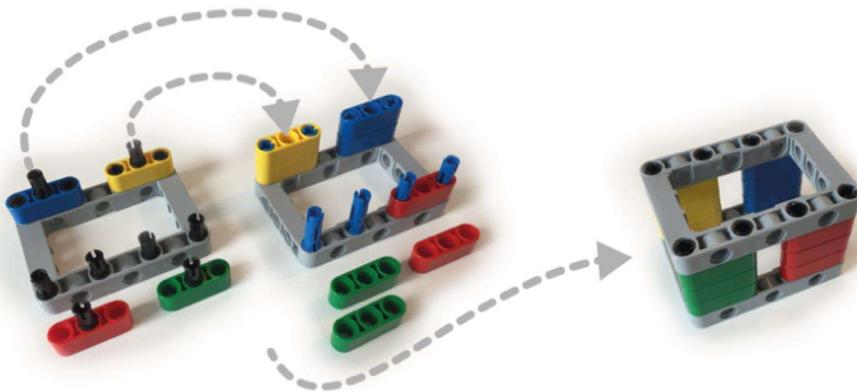
Baue den Farbsensor-Aufsatz (Bauanleitung Seite 69–76, Online PDF bis Seite 60–66)



Der Farbsensor hat drei verschiedene Modi – Farbe, Stärke des reflektierten Lichts und Stärke des Umgebungslichts.

### Aufgabe 3.a: Farbe erkennen

Baue den Farbwürfel (Bauanleitung Seite 4–6)



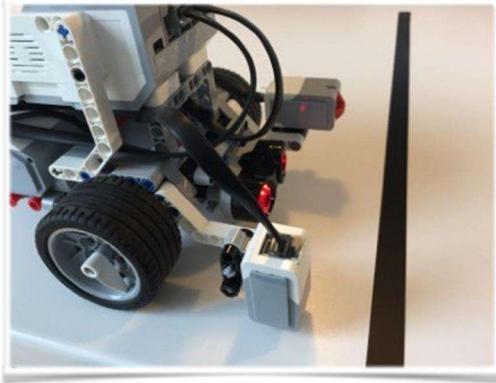
Halte den Würfel vor den Farbsensor und gib die Farbe als Text am Bildschirm aus.

**Tip:** Du kannst innerhalb eines Schalters einen weiteren Schalter verwenden.

**Zusatz:** Wird eine rote, eine gelbe oder eine grüne Farbe erkannt, so soll zusätzlich die entsprechende Statusleuchte leuchten (rot-rot, gelb-gelb, grün-grün). Wird keine Farbe erkannt, so soll die Statusleuchte nicht leuchten.

## Aufgabe 3.b: Farbe als Stopp-Linie

- Zusatz:** Suche einen schwarzen flachen Gegenstand, ein schwarzes Papier oder ein Klebestreifen, welchen/welches du als Stopp-Linie verwenden willst. Drehe den Farbsensor-Aufsatz so um, dass der Sensor nach unten misst und etwa 10mm vom Boden entfernt ist. Der Roboter soll jetzt solange fahren, bis die Stopp-Linie erreicht ist und anschließend stehen bleiben.
- Zusatz:** Verwende statt dem Farbvergleich den Vergleich des reflektierenden Lichts.



Der Farbsensor wird so angebracht, dass er Richtung Untergrund misst.

---

## Aufgabe 4: Kombinationen

- Zusatz:** Der Roboter soll jetzt auf alle drei Sensoren gleichzeitig entsprechend folgendem  Pseudo-Quelltext reagieren:

```
Schleife ( Unbegrenzt )
```

```
  Schalter ( Abstand > 50 cm ) ?
```

```
  JA:
```

```
    Roboter fährt schnell
```

```
    Aktuelle Farbe am Bildschirm ausgeben
```

```
  NEIN:
```

```
    Roboter fährt langsam Schalter (
```

```
    Berührungssensor aktiv ) ?
```

```
  JA:
```

```
    Roboter bleibt stehen
```

```
    Aktuelle Farbe am Bildschirm ausgeben
```

```
Ende Schleife
```

- Zusatz:** Denke dir weitere Aufgaben aus, die der Roboter erfüllen soll und setze diese um.

# Workshop #3

## Variablen

### Inhalte des Workshops

- Programmierkenntnisse: Einführung in Variablen
  - **Schreibender** Zugriff auf Variablen
  - **Lesender** Zugriff auf Variablen
  - **Verarbeiten** von Variablen
  - **Sensorwerte** in Variablen **speichern**
  - **Logische Operatoren**

### Einführung in Variablen

#### Variablen – Ich packe in meinen Koffer ...

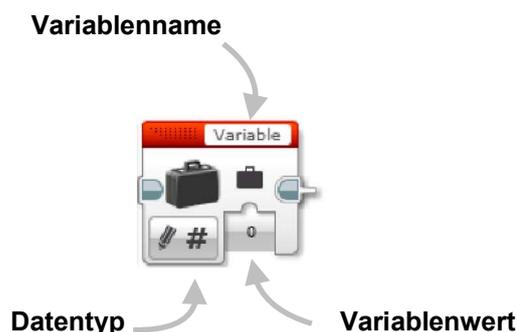
Beim Programmieren ist es oft so, dass ich der Computer (Roboter) verschiedene Informationen verarbeiten muss. Damit er jederzeit auf diese Informationen zugreifen kann, muss er sich zuerst diese Informationen merken und „in einen Koffer packen“. So ein Koffer wird in der Informatik **Variable** genannt.

Damit der Computer später den richtigen Koffer wieder findet, beschriftet er den Koffer. Bei Programmieren wird diese Beschriftung **Variablenname** genannt. Über diesen Variablenamen kann der Computer jetzt jederzeit den richtigen Koffer öffnen und auf den Wert (**Variablenwert**) darin zugreifen.

Da in einen Koffer unterschiedliche Dinge gepackt werden können, gibt es unterschiedliche Arten von Koffern, wie zum Beispiel eine Laptop-Tasche oder ein Reisekoffer. Genauso verhält es sich beim Programmieren – hier unterscheiden sich die Variablen durch ihren **Datentyp**. Es können 3 unterschiedliche Informationen in einer Variable gespeichert werden:

- **Text** ( z.B.: „Hallo Lego“)  T Text
- **Numerischer Wert** ( z.B.: 142 )  # Numerischer Wert
- **Logischer Wert** ( Ja oder Nein )  ✘ Logischer Wert

Eine **Variable** besteht also immer aus dem **Variablennamen** und dem zugehörigen **Variablenwert** und hat eine bestimmten **Datentyp**.

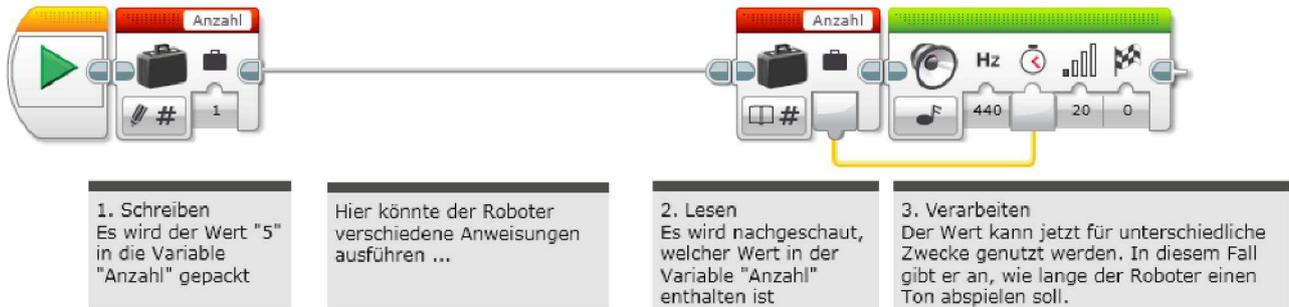


## Lesen und Schreiben von Variablen

Beim Programmieren gibt es zwei unterschiedliche Operationen, die du mit Variablen durchführen kannst.

Willst du Sachen in den Koffer packen, kannst du das mit der Operation **Schreiben** machen.

Mit der Operation **Lesen** kannst du nachsehen, was sich in einem Koffer befindet. Bevor du nachsehen kannst, was sich in einem Koffer befindet, musst du natürlich erst einmal Sachen in den Koffer packen:



Nachdem ein Wert in die Variable Anzahl geschrieben wurden kann jederzeit darauf zugegriffen werden. Die Ein- und Ausgänge der Bausteine werden mit Leitungen verbunden.

## Sensoren-Werte merken

Um einen Sensor-Wert in eine Variable zu Schreiben, müssen eine Leitung vom Sensor zur Variable gezogen werden. Der Wert, der jetzt in der Variable gespeichert ist, kann im späteren Programmverlauf zum Beispiel als Wert für die Leistung eines Motors verwendet werden.



Der gemessene Abstand wird in der Variable **Abstand** gespeichert. Der Inhalt der Variable **Abstand** wird als Wert für Variable **Abstand** eine Einstellung bei einem Motor verwendet.

## Aufgabe 1: Logische Werte

Verwende für diese Aufgabe den Berührungssensor-Aufsatz.

- Zu Beginn der Programms soll überprüft werden, ob der Berührungssensor gedrückt ist oder nicht. 



Im Anschluss wartet der Roboter 2 Sekunden, fährt dann für 3 Sekunden vorwärts und wartet wieder 2 Sekunden, egal wie der Zustand des Berührungssensors war. 



War der Berührungssensor zu Beginn aktiv, so soll der Roboter sich um 180 Grad drehen und wieder für 3 Sekunden vorwärts fahren. War der Berührungssensor nicht gedrückt, so soll der Roboter 3 Sekunden rückwärts fahren, ohne sich zu drehen.

```
Variable speichern ( Zustand Berührungssensor ) > Gedrueckt
```

```
2 Sekunden warten
3 Sekunden vorwärts fahren
  2 Sekunden warten
```

```
Schalter ( Variable Gedrueckt ) ?
```

**JA:**

```
  Umdrehen
  3 Sekunden vorwärts fahren
```

**NEIN:**

```
  3 Sekunden rückwärts fahren
```

**Tipp:** In einer Variable können auch logische Werte gespeichert werden. Ein logischer Wert hat nur zwei mögliche Zustände: Ja / Nein. Die Variable kann sich so also gut merken, ob der Berührungssensor gedrückt ist oder nicht.

## Aufgabe 2: Schalter – Kontinuierliche Überprüfung von Sensoren

Hier soll der Roboter 2 Aufgaben ausführen!

- 1. Aufgabe:** Es soll der Zustand des Berührungssensor laufend in die Variable mit dem Namen **Gedrueckt** geschrieben werden.
- 2. Aufgabe:** Der Zustand der Variable soll laufend ausgelesen werden –  Ist der Wert in der Variable „**JA**“, so soll die Statusleuchte des Bricks **grün** leuchten, ist der Wert „Nein“, so soll die Statusleuchte des Bricks **rot** leuchten.
- Zusatz:** Der Roboter soll die beiden Aufgaben **gleichzeitig** ausführen.

**Tipp:** Um Aufgaben laufend durchzuführen, kannst du wieder eine **Schleife** verwenden.

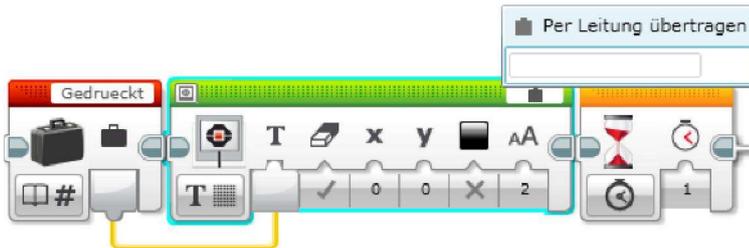
**Tipp:** Damit der Roboter mehrere Aufgaben zur gleichen Zeit durchführen kann, kannst du mehrere **StartBausteine** verwenden. Beim **Herunterladen und Ausführen** beginnen dann alle **Start-Bausteine** gleichzeitig zu arbeiten.

### Aufgabe 3: Schalter – Zählen mit Mathe

- Es soll mitgezählt werden, wie oft der Berührungssensor gedrückt wurde. Die Anzahl der Berührungen soll auf dem Display ausgegeben werden.
- Zusatz:** Führe die 2 Aufgaben **gleichzeitig** durch (Zählen, Am Display anzeigen).

**Tipp:** Mit dem Baustein **Mathe** kannst du verschiedene Rechnungen durchführen, wie zum Beispiel zwei Zahlen addieren.

**Tipp:** Um den Inhalte von Variablen auf dem Display anzuzeigen, muss die Option "Per Leitung übertragen" ausgewählt werden:

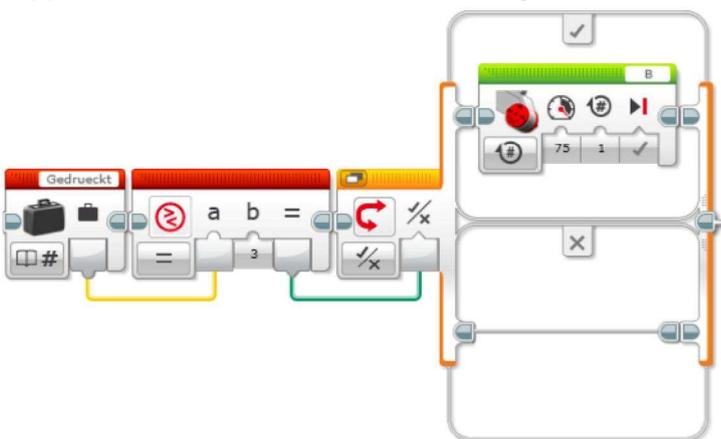


Anzeigen von Variablenwerten auf dem Display des Bricks.

### Aufgabe 4: Schalter – Zählen und Vergleichen

- Wurde der Berührungssensor **weniger als 5 mal** gedrückt, so soll die Statusleuchte **grün** leuchten, für alle anderen Werte soll die Statusleuchte **rot** leuchten.
- Zusatz:** Wurde der Berührungssensor **weniger als 5 mal** gedrückt, so soll die Statusleuchte **grün** leuchten. Wurde er **mehr als 10 mal** gedrückt, so soll die Statusleuchte **rot** leuchten. Für alle Werte dazwischen soll die Statusleuchte **gelb** leuchten.
- Zusatz:** Die Anzahl der Berührungen sollen zusätzlich am Display des Bricks angezeigt werden.
- Zusatz:** Führe die 3 Aufgaben **gleichzeitig** durch (Zählen, Am Display anzeigen, Statusleuchte).

**Tipp:** Um den Inhalte von Variablen zu Vergleichen, kannst du den Baustein **Vergleichen** verwenden:



Beispiel: Mit dem Baustein **Vergleichen** wird der Wert der Variable **Gedruickt** mit dem Wert **3** verglichen. Trifft das zu, so wird ein Motor angesteuert, sonst nicht.

**Tipp:** Verwende mehrere Schalter.

## Aufgabe 5: Zählen und mehr ...

**Zusatz:** Roboter soll in einer Schleife 2 Sekunden nach vor und dann zurück fahren. Im Display soll angezeigt werden, wie oft er die Strecke schon gefahren ist.

**Zusatz:** Roboter soll in einer Schleife 2 Sekunden vorwärts fahren, umdrehen, 2 Sekunden zurückfahren und wieder umdrehen. [L] [SEP]

Nach jedem Durchlauf soll der Roboter so oft piepsen, wie er Strecken gefahren ist.

**Schleife** (Unbegrenzt)

```
Variable speichern ( 0 ) > Anzahl
```

```
2 Sekunden vorwärts fahren
```

```
Umdrehen
```

```
2 Sekunden vorwärts fahren
```

```
Umdrehen
```

```
Variable speichern ( +1 ) > Anzahl
```

```
Schleife ( Anzahl )
```

```
    Ton abspielen
```

```
    1 Sekunde warten
```

```
Ende Schleife
```

```
Ende Schleife
```

---

## Aufgabe 6: Farben lesen (Für die ganz Schnellen)

**Zusatz:** Zeige dem Roboter 10 mal nacheinander mit 5 Sekunden Pause dazwischen verschiedene Farben. Die aktuelle Farbe soll immer am Display angezeigt werden. [L] [SEP]

[L] [SEP]

Nach den 10 Durchläufen soll im Display angezeigt werden, wie oft die Farbe rot dabei war.

**Zusatz:** Lege eine farbige Fläche auf den Boden. [L] [SEP]

Der Roboter soll bis zur Fläche vorwärts fahren und dort die Farbe erkennen. Anschließend soll der Roboter umdrehen und wieder zurückfahren. Am Ausgangspunkte angekommen, soll er die gelesene Farbe am Display anzeigen.

**Zusatz:** Denke dir weitere Aufgaben aus, die der Roboter erfüllen soll und setze diese um.

# Workshop #4

## Anwenden, Line-Follower

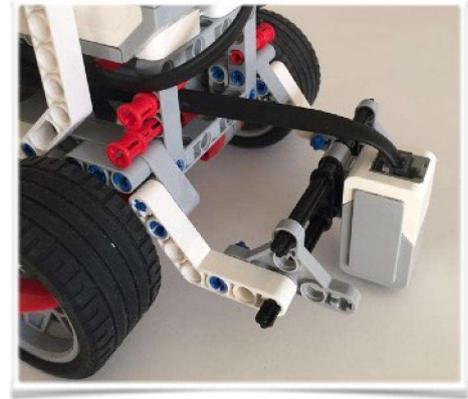
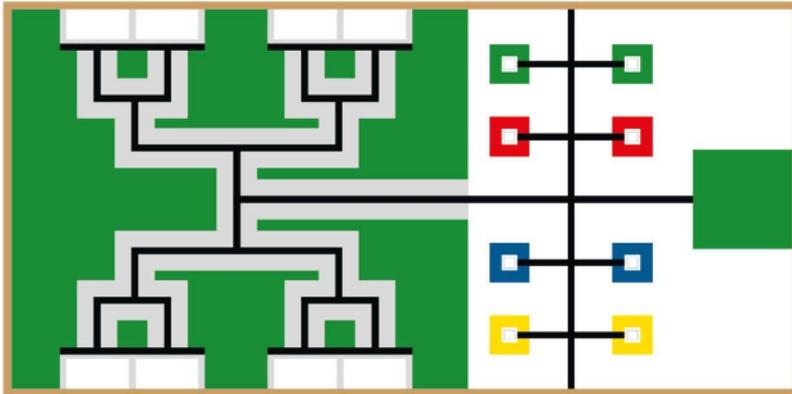
---

### Inhalte des Workshops

- Programmierkenntnisse
  - Vertiefung **Variablen**
  - Verwenden von **Text-Bausteinen**
  - Durchführen von **Berechnungen**
- Anwenden der bereits erworbenen Kenntnisse

## Aufgabe 1: Entlang der schwarzen Linie

Verwende für diese Aufgabe die Wettbewerbsmatte von 2016 und baue den Farbsensor-Aufsatz so um, dass er nach unten zeigt und **mittig** vor dem Roboter angebracht ist. Der Sensor soll sich etwa 5-10 mm über dem Boden befinden.



Spielfeldmatte „Rap the scrap“

Der Farbsensor-Aufsatz wird mittig angebracht.

Um einer Linie zu folgen, gibt es verschiedene Methoden. Einige dieser Methoden benötigen zwei Farbsensoren. Die folgende Methode kommt mit einem Farbsensor aus:



Folgende Schritte sind notwendig, damit der Roboter auch bei unterschiedlichen Lichtverhältnissen gut klar kommt:

1. **Kalibrieren des Farbsensors**
2. **Finden der schwarzen Linie**
3. **Folgen der schwarzen Linie**

## Aufgabe 1.a: Kalibrieren des Farbsensors

Die Lichtverhältnisse haben einen starken Einfluß auf die Werte der Farbsensoren. Darum ist es wichtig, diese Einflüsse auszugleichen und die Farbsensoren zu „**kalibrieren**“.

Wenn zwischen einer schwarzen Fläche und einer nicht schwarzen Fläche unterschieden werden soll, dann bietet es sich an den Modus **Messen – Stärke des reflektierenden Lichts** an.

- Zeige die aktuelle Stärke des reflektierenden Lichts laufend am Display an.
- Der Roboter soll die Anweisungen entsprechend folgendem Pseudo-Quelltext umsetzen:

```

"Weiss messen" am Bildschirm ausgeben

Schleife ( 5 Sekunden )
    Messen Stärke des reflektierenden Lichts
    Aktuelle Stärke des reflektierenden Lichts am Bildschirm ausgeben
Ende Schleife

Variable speichern ( Stärke des reflektierenden Lichts ) > Weiss

"Schwarz messen" am Bildschirm ausgeben

Schleife ( 5 Sekunden )
    Messen Stärke des reflektierenden Lichts
    Aktuelle Stärke des reflektierenden Lichts am Bildschirm ausgeben
Ende Schleife

Variable speichern ( Stärke des reflektierenden Lichts ) > Schwarz

Inhalt der Variablen Weiss und Schwarz am Bildschirm ausgeben
10 Sekunden warten
    
```



Abfolge der Anzeige zum Lesen und Speichern der Stärke des reflektierenden Lichts auf dem Brick.

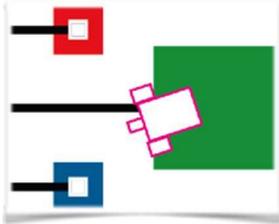
**Tip:** Um Texte miteinander zu verbinden kannst du den Baustein **Text** verwenden:



## Aufgabe 1.b: Start – Finde die schwarze Linie

### Ausgangsstellung

- Der Roboter soll so positioniert werden, dass er sich mit einem der Räder auf der grünen Startfläche befindet und der Farbsensor darüber hinaus steht. Der Sensor soll sich dabei links von der schwarzen Linie befinden.



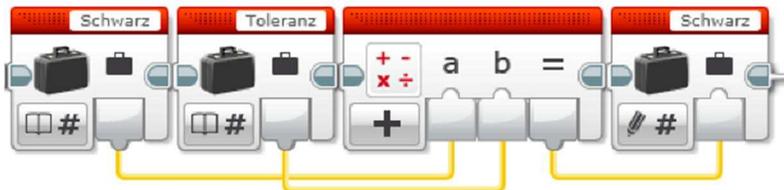
- Schreibe in die Variablen **Schwarz** und **Weiss** die Werte, die in **Aufgabe 1.a** ermittelt wurden.

### Toleranzen einbauen

Bei der Erkennung der Farben ist es hilfreich eine gewisse Toleranz einzubauen – <sup>[1]</sup><sub>[SEP]</sub>  
 Wenn der Wert für Schwarz z.B. 10 beträgt, so soll der Roboter bereits ab einem Wert von 15 Schwarz erkennen. Die Toleranz ist die Differenz aus den beiden Werten – in diesem Fall also 5 <sup>[1]</sup><sub>[SEP]</sub>

- Erstelle eine Variable **Toleranz** und speichere den Wert **10** darin.
- Addiere zum Variablenwert **Schwarz** die **Toleranz** und speichere das Ergebnis wieder in die Variable **Schwarz**.
- Subtrahiere vom Variablenwert **Weiss** die **Toleranz** und speichere das Ergebnis wieder in die Variable **Weiss**.

**Tip:** Um mit Variablen rechnen zu können kannst du den Baustein **Mathe** verwenden:



Die Inhalte der beiden Variablen werden addiert und anschließend wieder gespeichert.

### Drehen bis Schwarz erkannt wird

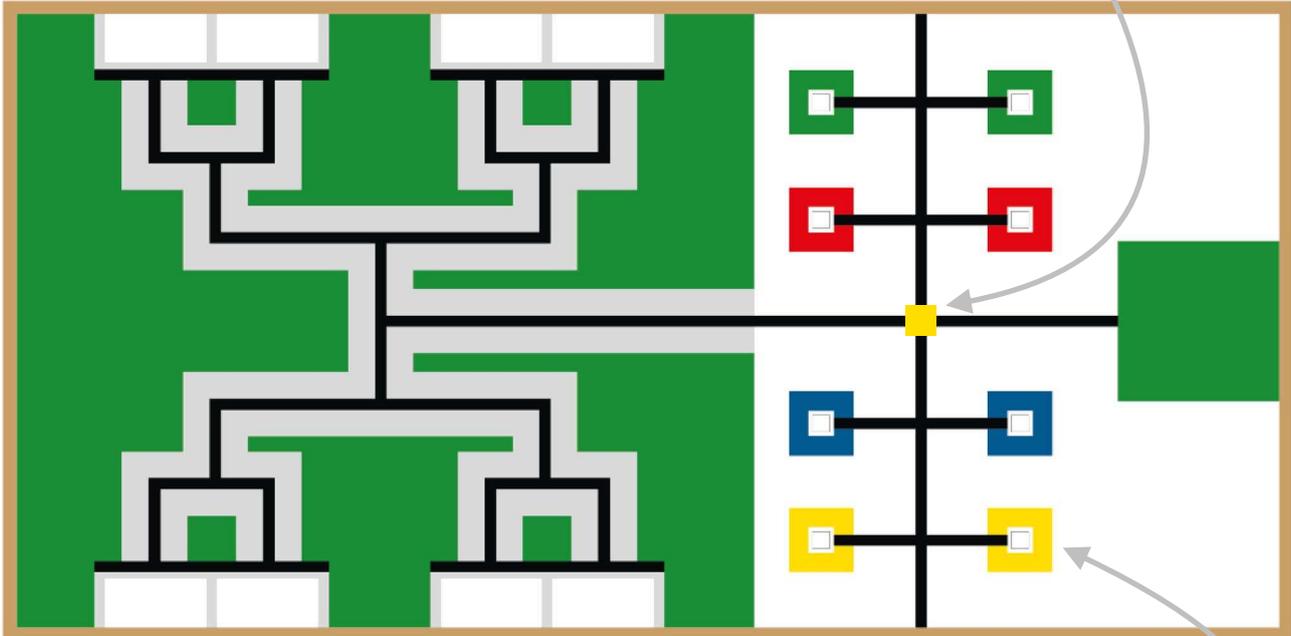
- Der Roboter dreht sich solange im Uhrzeigersinn, bis der Sensor eine schwarze Linie erfasst. Anschließend soll der Roboter aufhören sich zu drehen
- Zusatz:** Zeige laufend den aktuelle **Stärke des reflektierenden Lichts** sowie den Inhalt der Variable **Schwarz** an.



## Aufgabe 2: Explorer (Für die ganz Schnellen)

### Ausgangsstellung

- Der Roboter soll so positioniert werden, dass er sich mit einem der Räder auf der grünen Startfläche befindet und der Farbsensor darüber hinaus steht. Der Sensor soll sich dabei links von der schwarzen Linie befinden.
- Auf den ersten Kreuzungspunkt der schwarzen Linien wird ein farbiges Plättchen gelegt.



### Aufgabe

- Der Roboter fährt der schwarzen Linie entlang, bis er ein farbiges Plättchen erkennt. Anschließend merkt sich der Roboter die Farbe und fährt wieder in den Startbereich zurück, wo er die Farbe des Plättchens auf dem Display ausgibt.
- Zusatz:** Der Roboter befindet sich vollständig über dem grünen Startbereich.
- Zusatz:** Der Roboter fährt auf den Zielbereich mit der Farbe, die das Plättchen angibt.

**Tip:** Für die Ermittlung der Farben muss der Farbsensor auf einen anderen Modus umgeschaltet werden.