



# Problemstellungskarte Bereich: Muskeln

## Problemstellung

Die Angestellten der Gärtnerei *Spichtig* sind durchtrainierte Arbeiter. Trotzdem ist es ganz normal, dass eine einzelne Person nicht mehr als zwei Säcke Blumenerde auf einmal tragen kann. Und das liegt nicht einmal daran, dass die „Muskelpakete“ zu wenig Kraft hätten – nein – sie haben schlicht und einfach zu wenig Arme.

Nun ist bei der Geschäftsleitungssitzung Jennifer Spichtig, der Tochter der Geschäftsinhaber Martin und Isabelle Spichtig, die geniale Idee in den Sinn gekommen, die Angestellten etwas zu entlasten und ihnen einen *Schubkarrenroboter* zur Seite zu stellen, der sich mithilfe eines *Greifarms* automatisch beladen kann.

Dieser Roboter soll selbstständig das Transportieren von Material übernehmen. Dabei **belädt der Greifarm die Schubkarre**, welche wiederum das **Material von einem Ort zum anderen fährt**. Anschliessend soll sie **wieder** zum Ausgangspunkt **zurückfahren**.

*Modellierung: Da wir keine Säcke voll Blumenerde haben, müssen wir uns anderweitig zu helfen wissen. Ein Tennisball soll einen Blumenerde-Sack simulieren. Ausgangs- und Schlusspunkt des Transportes dürft ihr selbst bestimmen. Wichtig ist aber, dass diese Orte identisch sind!*

## Leitfragen

Durch die Beantwortung der folgenden Leitfragen wird die Inbetriebnahme dieses Schubkarrenroboters für euch einfacher werden:

- Wie kann erreicht werden, dass der Greifarm sich zum Tischtennisball hinbewegt?
- Wie kann erreicht werden, dass der Greifarm sich im richtigen Moment schliesst und so den Tischtennisball einfängt?
- Wie kann erreicht werden, dass der Schubkarrenroboter von einem Ort zum anderen fährt und nachher zur Ausgangsposition zurück?
- Wie kann erreicht werden, dass der Schubkarrenroboter mit Greifarm an der richtigen Stelle den Tischtennisball aufhebt, weiterfährt und an der gewünschten Position den Tischtennisball wieder runterlässt.



## Hypothesen

### 1. Ideen festhalten

Es ist nun an der Zeit, euren Ideen freien Lauf zu lassen. *Was denkt ihr? Wie könnte wohl eine mögliche Lösung für das vorangehende Problem aussehen?* Überlegt euch mögliche Szenarien, wie ihr das Problem lösen könntet und haltet eure Ideen im Forscherbuch fest.

### 2. Modellzeichnungen

Zeichnet euch skizzenhaft eine Modellzeichnung eines möglichen Lösungsprogramms und erläutert schriftlich die Bedeutung der einzelnen Bausteine. Hierzu stehen euch als Hilfestellung die untenstehenden Sprachbausteine zur Verfügung.

#### Sprachbausteine

Start, Mittlerer Motor (greifen), Bewegungslenkung (Motoren links – rechts, vorwärts – rückwärts)

## Bausteine

Bevor ihr das Problem der Gärtnerei *Spichtig* lösen könnt, lernt ihr in den drei Baustein-Posten die Funktionen und Einsatzmöglichkeiten von Fahrblöcken und Greifaktionen kennen. Dieses Wissen hilft euch bei der Problemlösung des Schubkarrenroboters.

- Baustein 1: Das Fahren
- Baustein 2: Das Greifen

# Baustein 1: Das Fahren

## Ziele

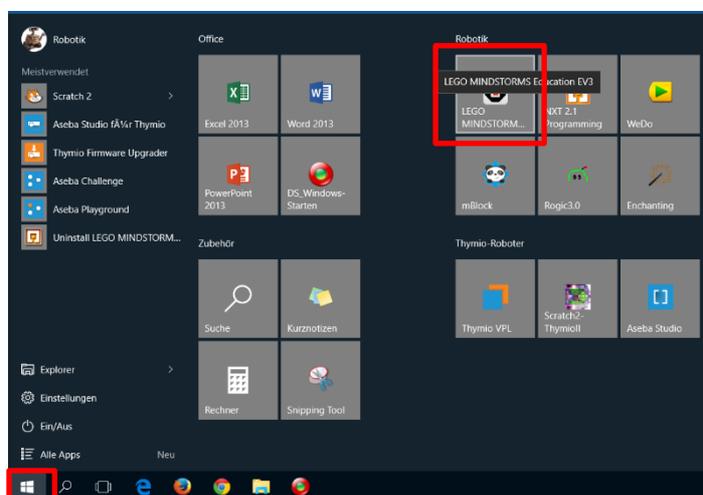
Ich kann ein Programm sequentiell programmieren, sodass ein Roboter eine bestimmte Strecke fährt.

## Material

- 1 EV3 Lego Mindstorms Roboter
- 1 Notebook mit der Software LEGO Mindstorms EV3
- 1 USB Verbindungskabel



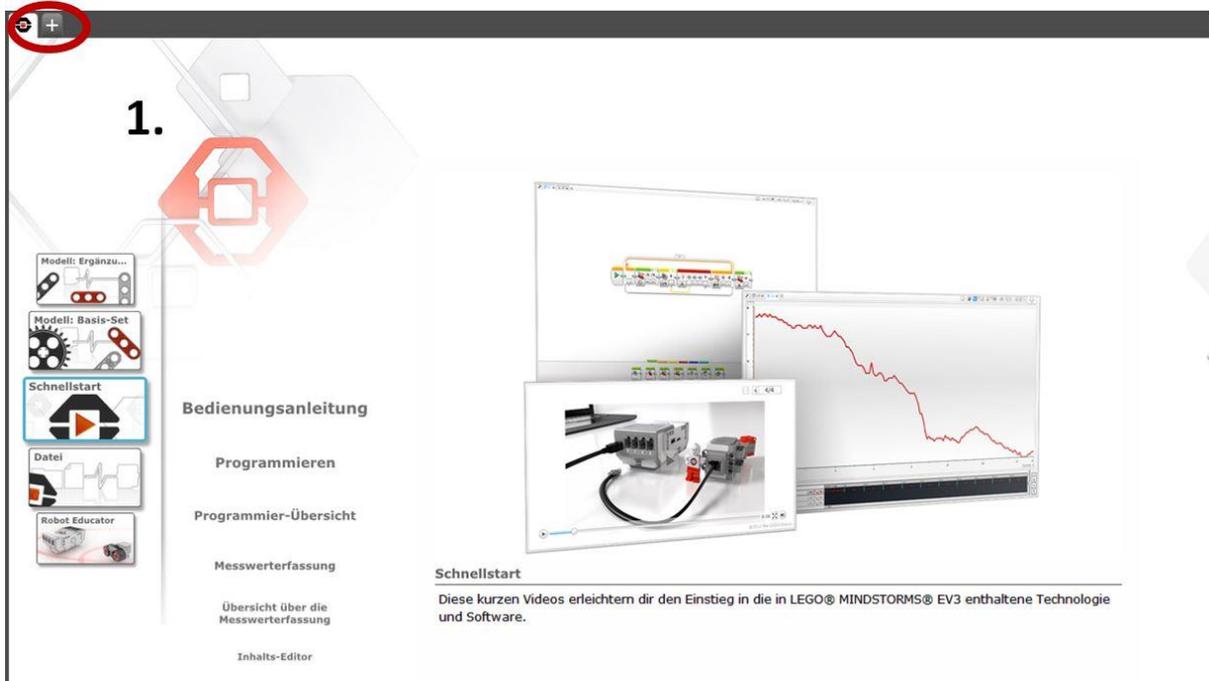
**Hinweis zum Starten der Software LEGO Mindstorms EV3 starten:**



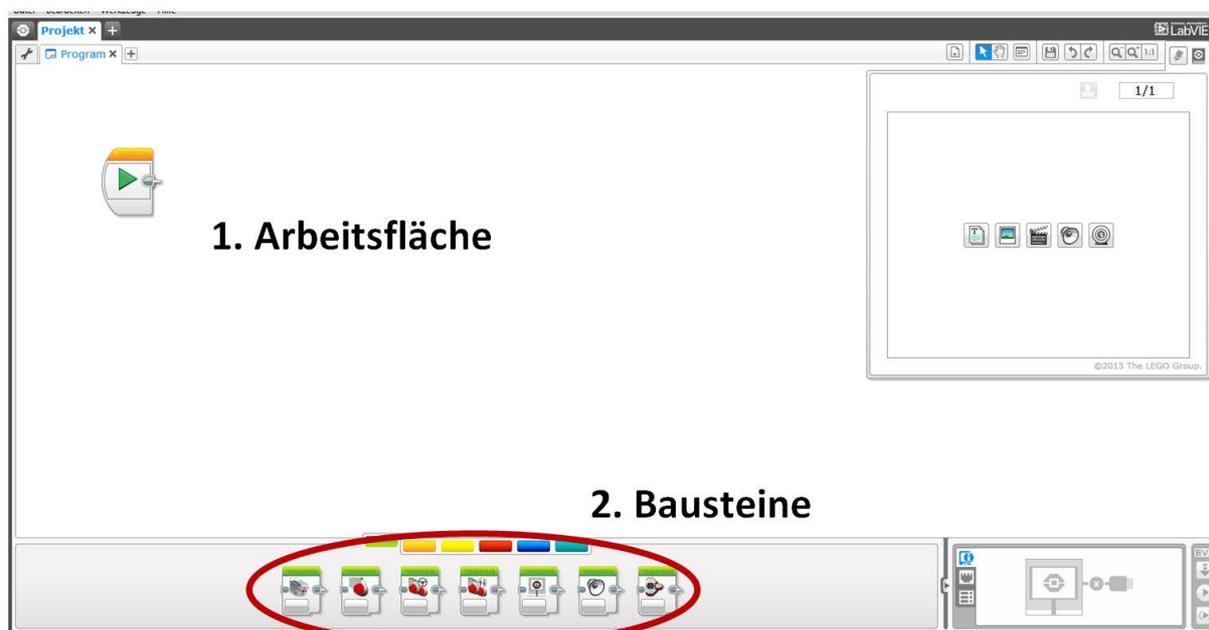


## Vorgehen

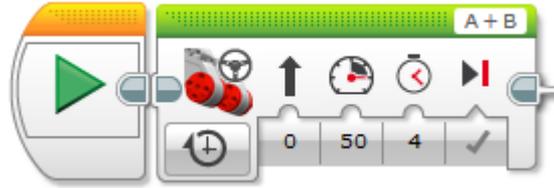
Um einen Roboter zu programmieren, braucht ihr ein entsprechendes Programm. Öffnet die Software „LEGO MINDSTORMS EV3“ und erstellt ein neues Projekt. Die Abbildung hilft euch dabei.



Per „Drag & Drop“ könnt ihr Bausteine in die Arbeitsfläche ziehen. Indem ihr die verschiedenen Bausteine aneinanderreicht, schreibt ihr ein Programm. Dieses Programm kann vom Roboter anschliessend ausgeführt werden.



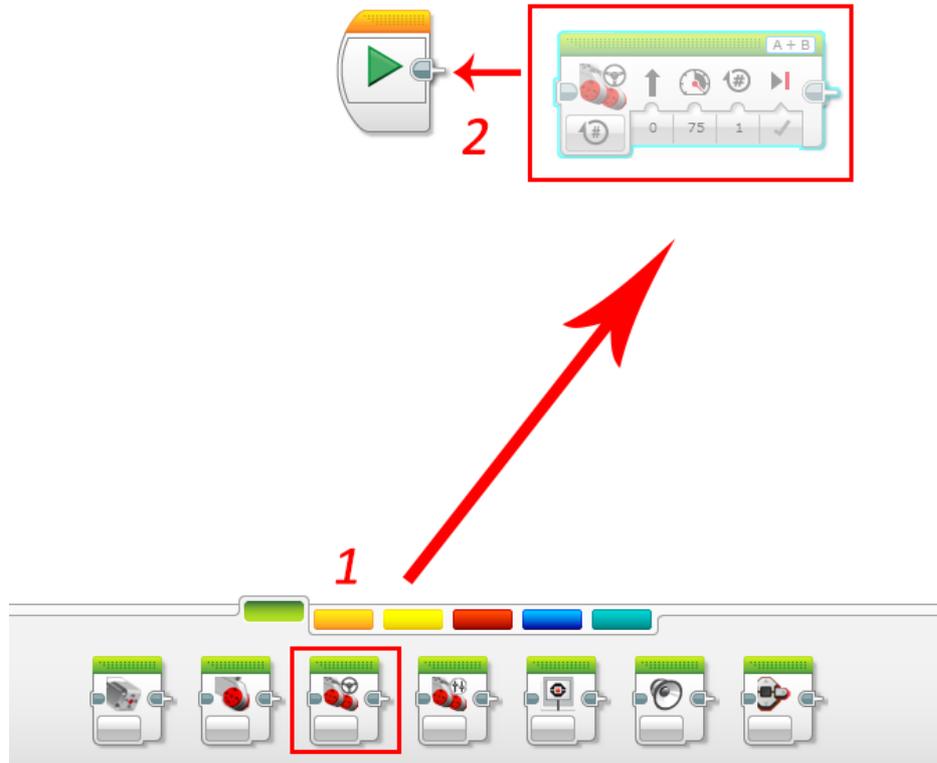
Zuerst lernt ihr einfache Fortbewegungen des Roboters kennen. Baut das folgende Programm nach. Überlegt euch dabei, was euer Roboter macht, wenn ihr das Programm auf ihn überträgt und startet.



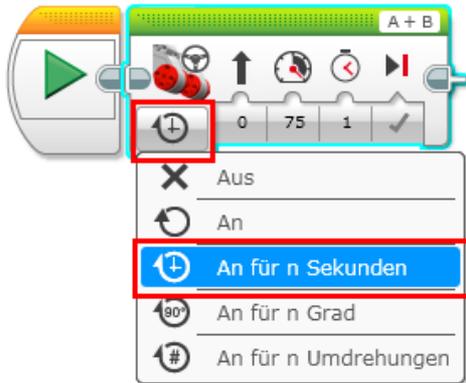
*Hinweis: Nachfolgend findet ihr eine kleine Programmierhilfe.*

Programmierhilfe

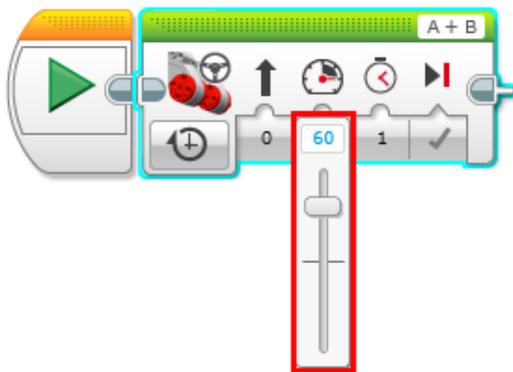
Zieht im Programm *LEGO Mindstorms EV3* einen Motorblock aus der unteren Leiste (1) in den Programmierbereich und setzt ihn an das Startfeld an (2).



Jetzt muss eigentlich nur noch der Motor richtig eingestellt werden:



Damit der Roboter eine bestimmte Zeit lang fährt, klickt ihr auf das erste Feld im *Motorblock* und wählt dort *An für n Sekunden* aus.



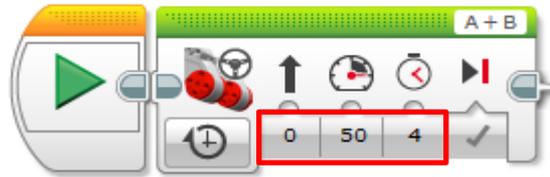
Im mittleren Feld mit der *Tachoanzeige* setzt ihr die Zahl von 100 runter auf 50. *Was bewirkt das wohl?*



Und rechts bei der kleinen *Uhr* erhöht ihr die Zahl eins auf vier.



**Programm ausführen.** Und? Was hat euer Roboter gemacht? Richtig, er ist vier Sekunden lang einfach nur geradeaus gefahren. Und das nicht so schnell, nur mit der Hälfte seiner Motorleistung. Warum ist das so?



Im roten Rahmen ist von links nach rechts zu erkennen:

- Beim **Pfeil** bedeutet die Zahl 0, dass der Roboter geradeaus fährt. Wäre die Zahl grösser als Null, würde der Roboter eine Rechtskurve fahren; bei einer Zahl kleiner als Null dementsprechend eine Linkskurve.
- Die Zahl bei der **Tachoanzeige** ist die aktuelle Motorenleistung. 50 bedeutet hier also, dass der Motor die Räder nur mit der Hälfte der Leistung dreht.
- Die Zahl unter der **Uhr** zeigt an, wie lange die Räder sich drehen – hier sind das vier Sekunden.

**Was tut er wohl?** Schaut euch nun diese vorbereiteten Programmierungen an. Welche Bewegungen wird der Roboter wohl ausführen? Überprüft eure Vermutungen, indem ihr das Programm auf dem Notebook nachbaut, auf euren Roboter lädt und ausführt.

### Schritt 1



### Schritt 2



### Auftrag 1: Weiterfahren

Alles klar? Dann programmiert euren Roboter nun so, dass er für 5 Sekunden mit einer Motorleistung von 60 geradeausfährt und anschliessend für 5 Sekunden, mit der Leistung von 70, wieder zurückfährt.

*Hinweis: Eine mögliche Lösung dazu findet ihr unter den Musterlösungen.*

### Auftrag 2: Drehen

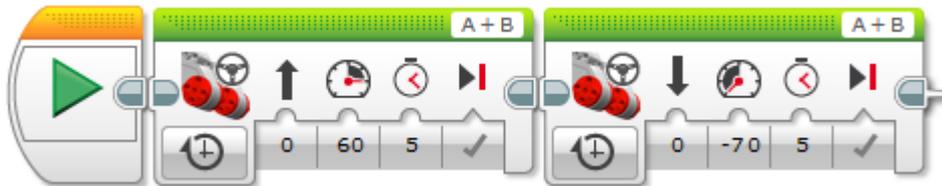
Versucht nun in einem nächsten Schritt einmal selbstständig euren Roboter so zu programmieren, dass er zuerst geradeaus fährt, sich dann um seine eigene Achse dreht und anschliessend wieder zurückfährt.

*Hinweis: Eine mögliche Lösung dazu findet ihr unter den Musterlösungen.*

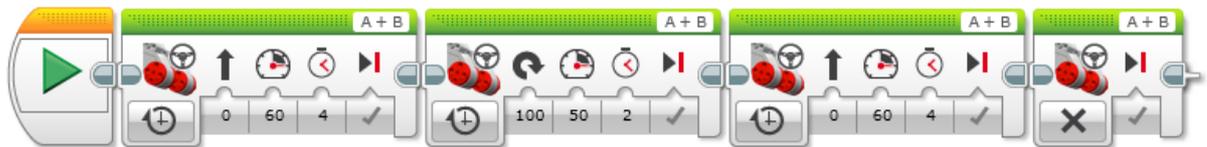
## Musterlösung

### Bausteinposten 1: Fahren

### Auftrag 1: Weiterfahren



### Auftrag 2: Drehen



# Baustein 2: Das Greifen

## Ziele

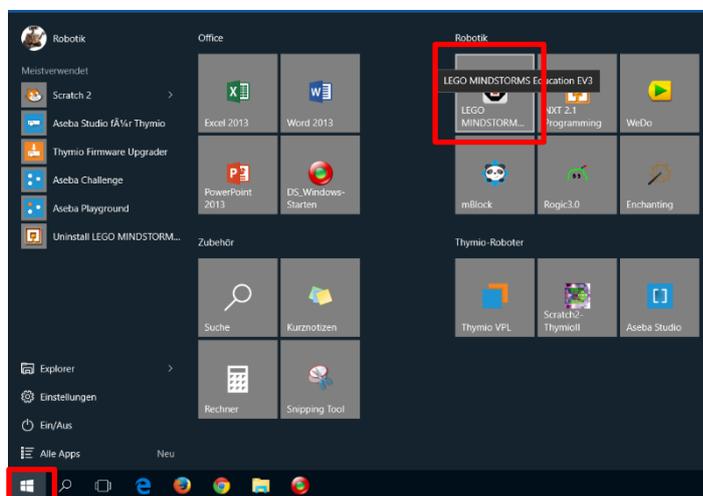
Ich kann ein Programm sequentiell programmieren, sodass ein Roboter einen bestimmten Gegenstand anhebt und wieder senkt.

## Material

- 1 EV3 Lego Mindstorms Roboter (Greifarm muss offen sein)
- 1 Notebook mit der Software LEGO Mindstorms EV3
- 1 USB Verbindungskabel
- 1 Tennisball



Hinweis zum Starten der Software LEGO Mindstorms EV3 starten:





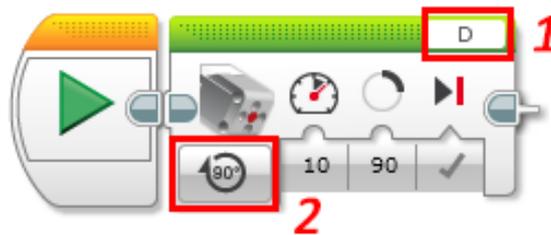
## Vorgehen

Am Anfang dieses Bausteinpostens liegt der Schwerpunkt auf einer einfachen Greifbewegung des Roboters. Und zwar ist es das Ziel, dass der Roboter einen Gegenstand, wie zum Beispiel einen *Tennisball* anheben und wieder senken kann.

*Hinweis: Nachfolgend findet ihr eine kleine Programmierhilfe.*

## Programmierhilfe

Dafür erstellt ihr ein neues Projekt in Lego Mindstorms EV3 und zieht einen *mittleren Motorblock* aus der unteren Leiste in den *Programmierbereich*. Diesen Block hängt ihr gerade an den Startblock an.

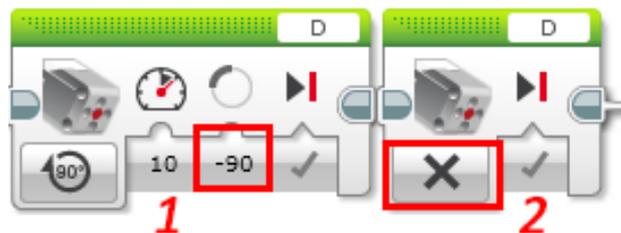


Als erstes ist es wichtig, dass ihr den richtigen Anschluss (Port) wählt (1). Ist euer *mittlerer Motor* zum Beispiel am Anschluss D, so wählt ihr diesen aus. Anschliessend stellt ihr den Motor so um, dass er sich nicht für eine bestimmte Zeit oder für eine bestimmte Anzahl an Umdrehungen, sondern um eine bestimmte Gradzahl dreht (2).

Mit den Einstellungen, wie sie oben auf dem Bild zu sehen sind, hebt der Roboter nun also den Tennisball – wenn man ihn auf den Greifarm stellt – um 90° an. Und das macht er mit einer Motorleistung von 10.

*Was ist jetzt wichtig zu beachten, wenn ihr den Greifarm nicht immer von Hand wieder nach unten ziehen möchtet?*

Genau. Ihr habt die Möglichkeit, den Motor genau das Umgekehrte machen zu lassen und den Tennisball nicht anzuheben, sondern zu senken.



Dafür ändert ihr ganz einfach den Wert von 90 in -90 (1). Da das Programm anschliessend zu Ende sein soll, fügt ihr wiederum einen *mittleren Motorblock* ein und schaltet den Motor mit *Aus* ab (2).



Jetzt geht das Anheben und Senken aber etwas schnell. Sobald der Greifarm oben ist, geht er auch schon wieder nach unten. Das nützt euch noch nicht so viel, möchtet ihr doch den Tennisball transportieren können. Doch die Lösung dieses Problems liegt nicht weit. Ihr fügt ganz einfach einen *Warten-Block* in euer Programm ein und ändert die Zeit des Wartens auf einen gewünschten Wert (3).



Und fertig ist euer *Heben-und-Senken-Programm* (vgl. nächstes Bild). Ladet es gleich einmal auf euren Roboter und führt es aus.



### Auftrag 1: Fahren und Aufheben

Stellt ein Programm zusammen, bei dem der Roboter von einem Meter auf eine Wand zufährt und den Tennisball, der gerade an der Wand liegt, aufhebt. Anschliessend soll der Roboter wieder einen Meter zurückfahren und den Tennisball dort absetzen.

*Hinweis: Eine mögliche Lösung dazu findet ihr unter den Musterlösungen.*

Musterlösungen

Bausteinposten 2: Greifen

Auftrag 1: Fahren und Aufheben



Autorenschaft: Andrea Maria Schmid, Urs Meier (PH Luzern)  
Unterstützt durch Studierende: Michael Wyrsh, Silvan Petermann  
Version: März 2017

