

Marc Laporte- Hoffmann



im Unterricht

Anregungen für den Einsatz im Unterricht



1. Einleitung

LEGO Mindstorms EV3 ist schon die dritte Generation. Breits 1998 erschien der RCX-Baustein, gefolgt von dem NXT-Baustein 2005. Von Anfang an war LEGO Mindstorms auch für den Einsatz in Bildungskontexten gedacht. Das Prinzip hat sich im Wesentlichen nicht verändert: Es gibt „Ausgänge“, an die Motoren angeschlossen werden und „Eingänge“, an die „Messwerte“ von Sensoren geliefert werden. Das Zusammenspiel der Motoren und Sensoren wird entweder über die im Baustein (auch „Brick“) implementierte Software, oder komfortabler mittels Computer, programmiert.

Zu unterscheiden gilt übrigens zwischen der „Education“-Ausführung und der Home-Edition. Sie unterscheiden sich vor allem durch die Bestückung mit Motoren und Sensoren. Und natürlich im Preis: Die Home-Edition ist um die 300,- Euro zu haben, während für die Education-Ausführung ca. 450,- Euro gerechnet werden muss¹.

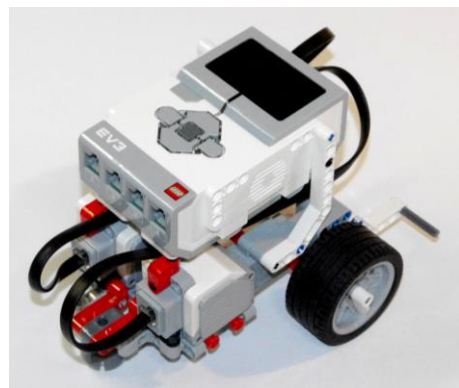
2. Einsatz im Unterricht: Vorüberlegungen

LEGO Mindstorms eignet sich für Schüler ab der vierten Klasse. Durch vielseitige Einsatzmöglichkeiten kann damit bis in hohe Klassen, auch Oberstufe, gearbeitet werden. Pro Kasten können maximal 4 Schüler arbeiten, sinnvoller jedoch ist eine Partnerarbeit. Da die Programmierung am Baustein selbst eher umständlich ist, bietet es sich an, die kostenlose Software zu benutzen, um die Programme zu entwickeln.

Sofern eine Schule ausreichend finanzielle Mittel hat, kann sie selbst die Kästen kaufen. Dabei sollte vorab überlegt werden, wie und wann es eingesetzt werden können. Für eine AG mit 16 Schülern bspw. wären 8 Kästen notwendig. Beim Einsatz im Klassenverband u.U. noch mehr Kästen. Auch der Einsatz in einer Projektwoche ist denkbar. Alternativ gibt es vielleicht die Kästen zu leihen, z.B. beim regionalen Medienzentrum². Dann ist aber zu beachten, dass selbst bei einem „oberflächlichen“ Einsatz ausreichend Unterrichtszeit eingeplant werden muss: Für die „Basics“ mindestens 5-10 Unterrichtsstunden.

3. Konzeption dieses Scripts

Das vorliegende Script richtet sich an Lehrende, die mit ihren Schülern in die Grundlagen der Robotik und Programmierung einsteigen wollen. Von daher liegt der Focus auf dem Einsatz des Roboters und nicht auf den „Baumöglichkeiten“. Auf Grund dessen sollten das Educator-Basismodell und sämtliche Sensoren als Baugruppen **vollständig vormontiert** sein (lt. Bauanleitung), sodass im Unterricht der Schwerpunkt auf das Verständnis der Motoren und Sensoren und dessen Zusammenspiel gelegt werden kann. Das Script ist entsprechend aufgebaut: Nach einander werden zuerst die Motoren dann unterschiedliche Sensoren ausprobiert und Aufgaben gestellt. Dabei wird auf die grundsätzliche Umsetzbarkeit in den Unterrichtsstunden geachtet.



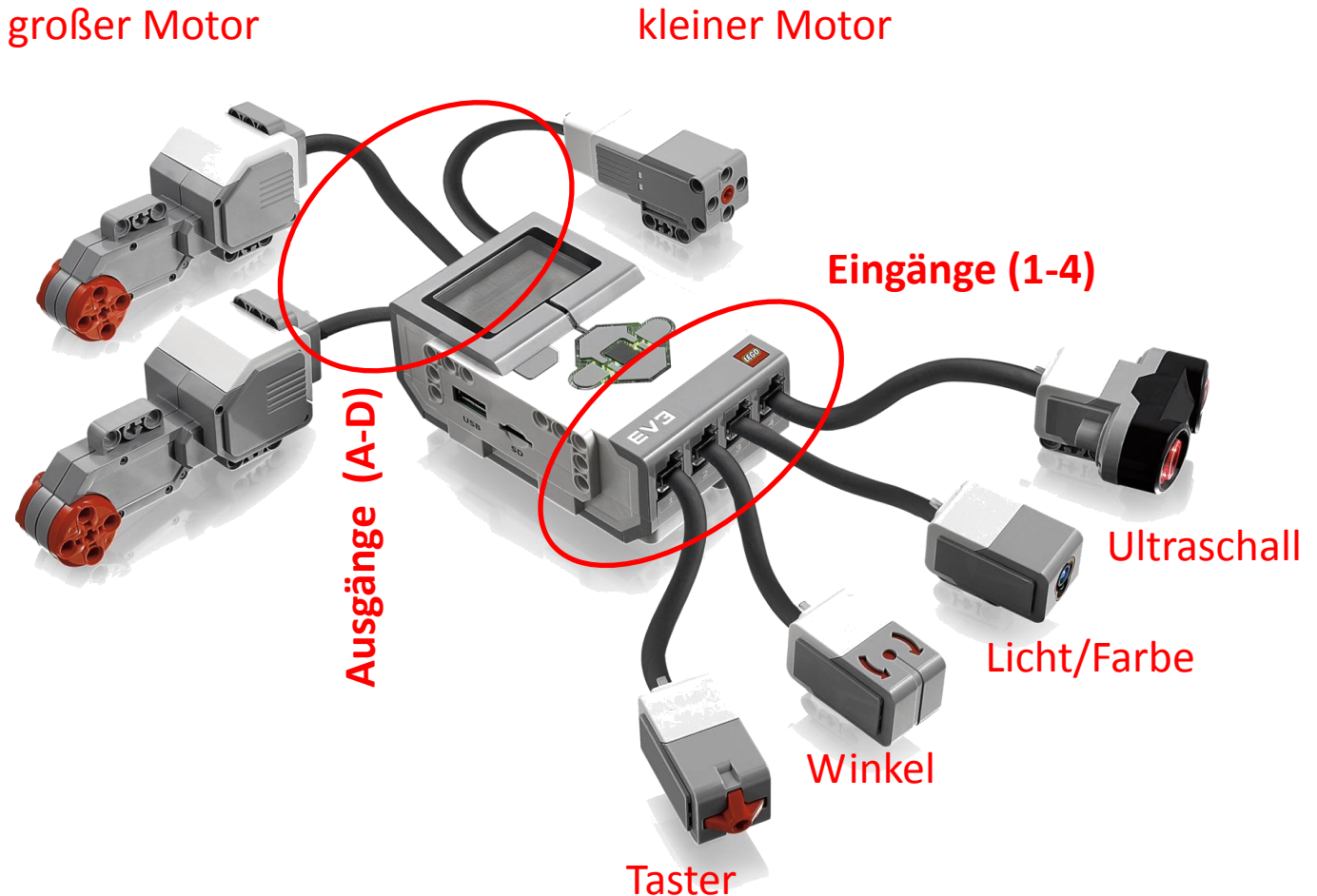
Die vorliegenden „Einheiten“ sind als **Anregungen** für Lehrkräfte zu verstehen und beziehen sich auf einen Einsatz in der vierten bis siebten Klasse, ggf. auch höheren Klassen.

¹ Stand: August 2015

² An den Medienzentren kann man sich im Verleih nach LEGO Mindstorms-Baukästen erkundigen. Das LMZ hat am Standort Stuttgart insgesamt 8 Baukästen, die ausgeliehen werden können.

4. Grundlagen: Hard- und Software

Der EV3-Baustein verfügt über vier **Eingänge** (erkennbar an den Zahlen 1-4) und vier **Ausgänge** (erkennbar an den Buchstaben A-D). Im Education-Kasten sind folgende Motoren und Sensoren verfügbar (der Taster 2x):



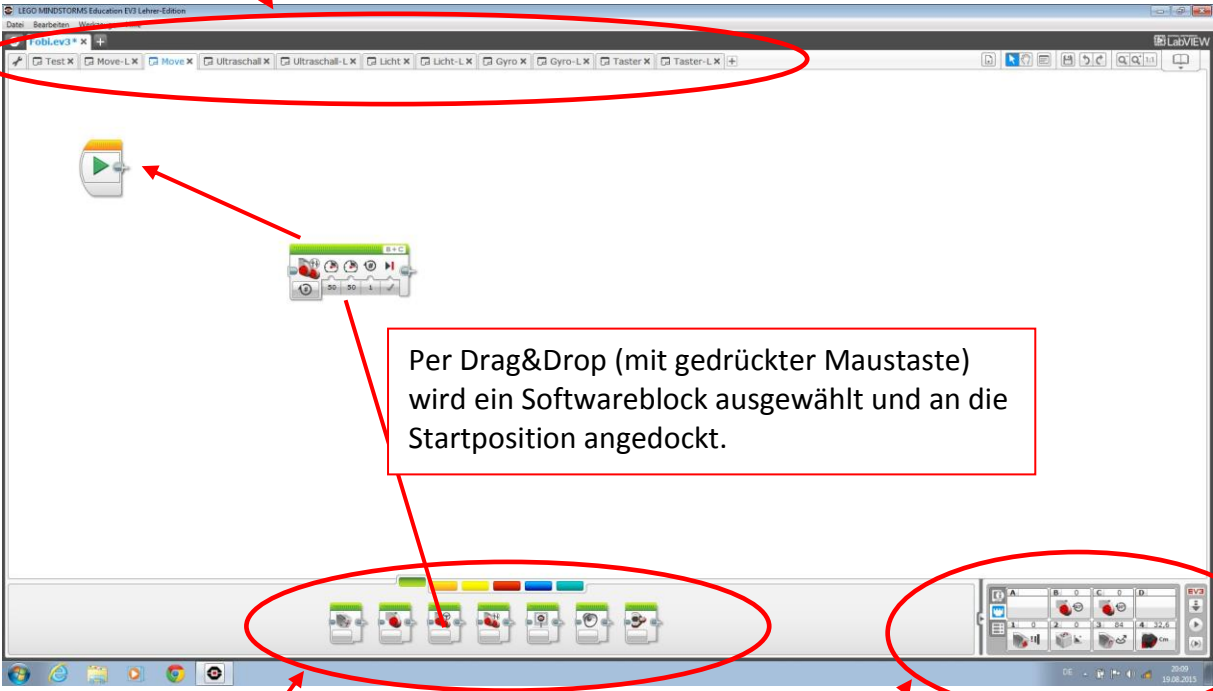
Für die Programmierung ist es wichtig, zu wissen, an welchem Eingang (bzw. Ausgang) welcher Sensor (bzw. Motor) angeschlossen ist, um die Befehle und Abfragen korrekt zu geben.

Alle Sensoren liefern Werte, die sowohl am Baustein auf dem Display als auch in der Software (unten rechts) abgefragt werden können. Für die Anzeige in der Software muss der Baustein mit dem USB-Kabel am Computer verbunden sein. Diese Werte werden für die spätere Programmierung gebraucht. So zeigt z.B. der Ultraschall-Sensor die Distanz zum nächsten Objekt an oder der Farbsensor liefert bspw. unterschiedliche Werte für weiß und schwarz. Auch die Motoren liefern Werte über die Stellung.

Wichtig: Eventuell muss ein Programm manuell beendet werden. Ist der Baustein noch via USB verbunden, kann das in der Software unten rechts (Stopp-Symbol) erfolgen. Ist der EV3 nicht mehr mit dem PC verbunden, dann muss man auf den grauen Knopf drücken, der sich links oben, separat vom Bedienungsfeld, befindet. Gelegentlich kann auch der komplette Baustein (Software) abstürzen. Hier hilft nur ein Reset: Den grauen Knopf oben links, den mittleren (dunkelgrauen) Knopf und den linken, grauen Knopf im Bedienungsfeld gleichzeitig einige Sekunden lang gedrückt halten. Um den EV3 auszuschalten, den grauen Knopf links oben drücken, Haken auswählen und bestätigen.

Die Software ist übersichtlich aufgebaut: Am unteren Rand befinden sich die verschiedenen Gruppen (Abteilungen) mit den jeweiligen „Softwareblöcken“, die per Drag&Drop auf die Programmierfläche gezogen werden. Zu beachten sind dabei die korrekten Ein- und Ausgänge sowie diverse Parameter.

In einem **Projekt** können mehrere **Programme** verwaltet werden (über Karteireiter). Diese sind dann auch unter dem Projektordner auf dem EV3-Baustein zu finden, sobald sie runter geladen wurden.



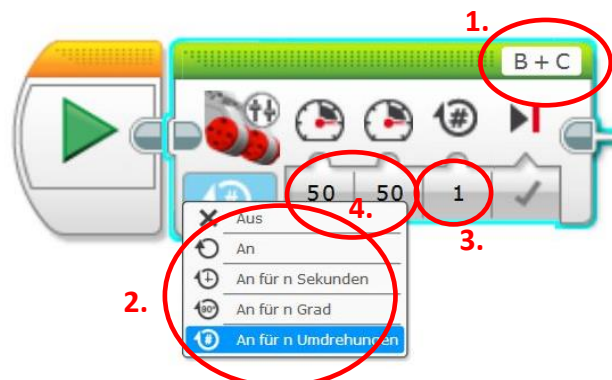
Per Drag&Drop (mit gedrückter Maustaste) wird ein Softwareblock ausgewählt und an die Startposition angedockt.

Hier findet sich, farblich sortiert, verschiedene Softwareblöcke, mit denen die Motoren und Sensoren gesteuert werden können. Außerdem auch Softwareblöcke zur Programmierung wie Schleifen und Verzweigungen (Wenn-Dann).

Hier können die Werte der Sensoren ausgelesen und außerdem die Programme auf den EV3-Baustein runter geladen werden.

Bei den Softwareblöcken muss ganz besonders auf die Einstellungen geachtet werden, wie am Block für die Motoren exemplarisch erläutert:

1. Werden die richtigen Ausgänge angesprochen?
2. Welche Einheit wurde gewählt (Sekunden / Grad / Umdrehungen)?
3. Stimmt der Wert?
4. Wie ist die Leistung eingestellt (negative Werte bedeuten, dass sich die Motoren rückwärts drehen!)



5. Einheiten

Die nachfolgenden Einheiten sind als Anregungen für den Unterricht gedacht. Dabei ist eine ungefähre Einschätzung der benötigten Unterrichtszeit ebenfalls angegeben. Jede Einheit beschäftigt sich mit einem anderen Schwerpunkt, wobei die Reihenfolge so gewählt wurde, dass die Komplexität zunimmt (sowohl in der Nutzung der Sensoren als auch in der Programmierung).

Inhalt

Einheit 1 - Einführung in Robotik	6
Einheit 2 - <i>Motoren</i> – Einführung.....	8
Einheit 3 - <i>Motoren</i> - Vertiefung	10
Einheit 4 - <i>Ultraschall-Sensor</i>	12
Einheit 5 – <i>Farb-/Licht-Sensor</i>	14
Einheit 6 – <i>Taster</i>	18
Wichtige Links:	19

Für Anregungen, Korrekturen und Hinweise zu diesem Script bin ich dankbar:
marc@laporte-hoffmann.de



Hinweis zum Urheberrecht: Dieses Script darf ohne Veränderung, bei Namensnennung nur im nichtkommerziellen Einsatz weiter gegeben werden.

Einheit 1 - Einführung in Robotik

ca. 1 Unterrichtsstunde

Ziel(e): Verstehen, dass „Bewegungsabläufe“ aus vielen einzelnen Schritten bestehen und jeder Schritt in einem „Programm“ vorher bestimmt werden muss = der Roboter macht nur das, was man ihm (in einem „Programm“) vorher mitteilt.

Kennenlernen des Motorblocks und erkennen, dass ein Programm aus „Abfolgen“ besteht.

- Ggf. mit PowerPoint oder im Unterrichtsgespräch: Einführung
 - *Sch. machen sich Gedanken, wie etwas oder jemand vom Ruhezustand in Bewegung kommt – welche „Schritte“ notwendig sind^{*)}.*
 - Demo-Programm (im „Robot Educator Modell“) starten
 - UG³: Beschreibung der Schritte (des „Programms“) / Robotertätigkeit
 - **Impulsfragen:**
 - *Was macht der Roboter?*
 - *Welche Einzelschritte sind zu erkennen?*
 - *Welche „Befehle“ erhält der Roboter?*
 - ggf. Hinweise zur Benutzung des Lego-Kastens, bzw. „Robot Educator Modell“ (USB-Kabel oder Bluetooth-Übertragung, Software usw.)
 - ggf. erste Erläuterungen zur Software
 - Erstes Programm: **Aufgabe 1a: Geradeaus fahren** – im UG erörtern, was dem Roboter „gesagt“ werden muss
 - Lösung frontal (mittels Beamer) erklären (dabei Hinweise zur Software geben)
 - Umsetzung durch die Schüler
- Aufgabe 1b für die Schnellen:** *Den Roboter ein Stück vorwärts und dann rückwärts fahren lassen*

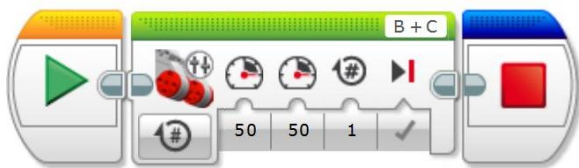
Lösungen

^{*)} **Beispiel für die Aufgabe, welche Schritte notwendig sind, um mit dem Auto loszufahren:**

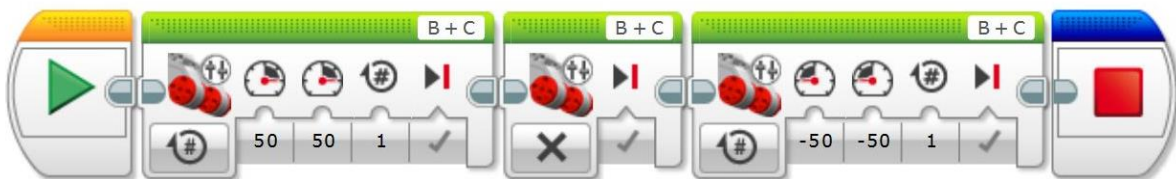
1. Fahrertür entriegeln
2. Tür öffnen
3. Auf dem Fahrersitz Platz nehmen
4. Tür schließen
5. Schlüssel ins Zündschloss stecken
6. Kupplungspedal treten
7. eventuell Bremspedal treten
8. Zündschlüssel zum Anlassen umdrehen
9. eventuell Feststellbremse („Handbremse“) lösen
10. Gang einlegen
11. Bremse loslassen
12. Kupplung vorsichtig loslassen
13. Gaspedal treten
14.



³ UG = Unterrichtsgespräch

1a: Gerade aus fahren

Anmerkung: Zum Ende des Programms sollte man sich angewöhnen, einen „Stopp-Block“ einzufügen, damit die Programme ordentlich beendet werden.

1b: Vorwärts und rückwärts fahren

Wie in Aufgabe 1a schon verwendet, wird der vorgefertigte Motorenblock „Hebelsteuerung“ genutzt – übrigens: Die Blöcke lassen sich anklicken, kopieren (Strg+C) und einfügen (Strg+V). Danach wird der gleiche Block genutzt, allerdings hier mit der Einheit „aus“. Dann wird der Block nochmals verwendet, allerdings wird die Motorleistung negativ eingetragen und damit fährt der Roboter rückwärts.

Es hätte auch der Motorenblock „Standardsteuerung“ gewählt werden können. Vorteil bei der Hebelsteuerung ist jedoch, dass die Schüler gleich merken, dass beide Motoren angesteuert werden können, auch mit unterschiedlichen Werten.

Einheit 2 - Motoren – Einführung

ca. 1 Unterrichtsstunde

Ziel(e): Parametrisierung erkennen und verstehen.

- Wiederholung der Inhalte aus Einheit 1 (ggf. im UG)
- **Aufgabe 1a:** (um die verschiedenen Parameter kennen zu lernen)
 - Hinweis: Die Motorleistung sollte auf 25% reduziert werden**
 - Der Roboter soll genau 20 cm vorwärts fahren (ggf. Lineale bereit legen!)
 - Dabei sollen die verschiedenen Parameter ausprobiert werden:
Umdrehungen, Grad, Zeit > mit was ist es am genauesten?
- Zusatzaufgabe 1b für die Schnellen: Den Roboter auf der Stelle drehen lassen**
- Besprechung der Lösung(en)
- **Aufgabe 2:** Der Roboter soll eine Kurve fahren, bzw. sich um 90 Grad drehen
- Besprechung der Lösung
- **Aufgabe 3:** Der Roboter soll einmal im Viereck fahren
- Besprechung der Lösung

Lösungen

Aufgabe 1a: Verschiedene Parameter ausprobieren; 20cm fahren

Die Schüler sollen verschiedene Parameter ausprobieren. Wichtig dabei ist, dass die 360 Grad für die Umdrehung des Motors stehen, also 1 Umdrehung. Wer rechnerisch daran gehen möchte, kann zuerst messen, wie weit (cm) bei Motorenleistung von 25% der Roboter bei 360 Grad fährt und mit dem Dreisatz den Wert für Grad errechnen. Die anderen probieren einfach fleißig aus (müsste ca. 410 Grad sein). Mit der Einheit „Umdrehungen“ lässt es sich viel schlechter einstellen, ebenso mit der Einheit „Zeit“ (da die ja auch abhängig von der Geschwindigkeit ist).

Zusatzaufgabe 1b: Den Roboter auf der Stelle drehen lassen

Die Lösung ist denkbar einfach: Im Motorenblock „Hebelsteuerung“ wird die Leistung des einen Motors auf bspw. „50“ eingestellt, bei dem anderen Motor auf „-50“, bei der Einheit „Umdrehungen“ (eine).

Aufgabe 2: Eine Kurve fahren / 90 Grad Drehung

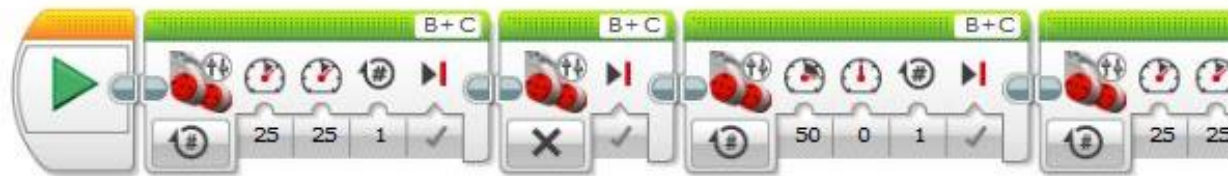
Eine mögliche Lösung wäre, wieder mit dem Motorenblock „Hebelsteuerung“ zu arbeiten und bei einem Motor die Leistung „50“, bei dem anderen Motor die Leistung „0“ einzustellen. Wenn die Einheit „Umdrehung“ benutzt wird, ist zu beachten, je niedriger die Motorleistung, desto kleiner der Winkel (also weniger als 90 Grad Drehung des Roboters). Alternativ kann auch mit dem Motorenblock „Standardsteuerung“ gearbeitet werden. Allerdings ist es vielleicht verwirrend, dass für eine 90 Grad Drehung der Wert „55“ eingegeben werden muss.

Aufgabe 3: Im Viereck fahren

Nachdem so langsam die Funktionsweise der Blöcke durchdrungen wurde, ist nun ein, zunächst, komplexes Programm nötig:



Ein Ausschnitt jedoch zeigt, dass sich (die ersten) drei Schritte viermal wiederholen:



Zuerst fährt der Roboter bei einer Leistung von 25% (bei beiden Motoren) eine Umdrehung gerade aus. Dann werden die Motoren gestoppt. Dann wird nur ein Motor mit einer Leistung von „50“ gestartet, während der andere auf „0“ steht. Nun wiederholen sich die Schritte.

Einheit 3 - Motoren - Vertiefung

ca. 1-2 Unterrichtsstunde(n)

Ziel(e): Erkennen, dass komplexe Programme vereinfacht werden können, vor allem dann, wenn sich Befehle / Strukturen wiederholen (> Schleifen).

Einführung von Variablen und deren Einbindung.

- Wiederholung Einheit 2 / letzter Abschnitt: Im Viereck fahren
- Ggf. im UG erarbeiten, dass das Programm sehr aufwändig ist. Da sich die ersten drei Schritte wiederholen, kann auch eine „Schleife“ benutzt werden
 - ggf. klären, was unter einer *Schleife* zu verstehen ist
- Erläuterung Softwareblock „Schleife“ und Verwendung
- **Aufgabe 1:** Umsetzung (Reproduktion) durch die Schüler
- Besprechung der Lösung (Fragen?)

- **Aufgabe 2 (für jüngere Schüler):**
Die Schüler sollen den Roboter durch einen kleinen Parcours fahren lassen – möglichst mit Hindernissen im gleichen Abstand (kleine Bauklötze o.ä.).
- Besprechung der Lösung

- **Aufgabe 2 (für höhere Klassen):**
 - a) Der Roboter soll ganz langsam mit einer Motorenleistung von 5% anfahren und sich in 5er-Schritten bis 25% steigern.
 - b) Das Programm aus Aufgabe 2a mit einer Schleife und der variable „Leistung“ automatisieren.
- Besprechung der Lösung

Lösungen

Aufgabe 1: Roboter im Viereck fahren (mit einer Schleife)

Der Schleifenblock ist in der orangenen Abteilung zu finden. Er wird ebenso per Drag&Drop auf die Fläche gezogen. Dann werden die ersten drei Schritte (aus Aufgabe 3 der letzten Einheit) in die Schleife reingezogen. Die Zählung der Schleife (rechts) muss von „unendlich“ auf „Zählen“ eingestellt und die Anzahl „4“ eingetragen werden.



Für jüngere Schüler

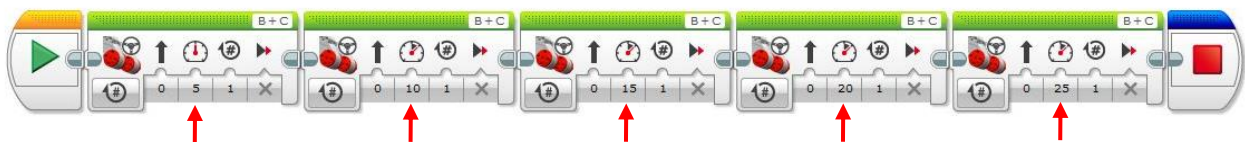
Aufgabe 2: Roboter im Parcours fahren lassen

Hier kann wieder der Motorenblock „Hebelsteuerung“ zum Einsatz kommen und beide Motoren unterschiedlich angesteuert werden. Im ersten Schritt hat der eine Motor die Leistung „30“ und der andere bspw. nur „10“. Somit fährt der Roboter eine Kurve und gleichzeitig vorwärts. Im nächsten Schritt werden die Werte vertauscht. Der Roboter fährt eine Kurve in die andere Richtung und vorwärts. So lässt sich der Roboter durch einen Parcours steuern. Dabei sollten die Hindernisse (z.B. kleine Bauklötze) regelmäßig verteilt sein. Ggf. muss zwischen die beiden Motorenblöcke noch ein weiterer Motorenblock eingefügt werden, bei dem beide Motoren die gleiche Leistung bekommen und der Roboter damit zwischendurch noch ein Stück vorwärts fährt.

Für ältere Schüler

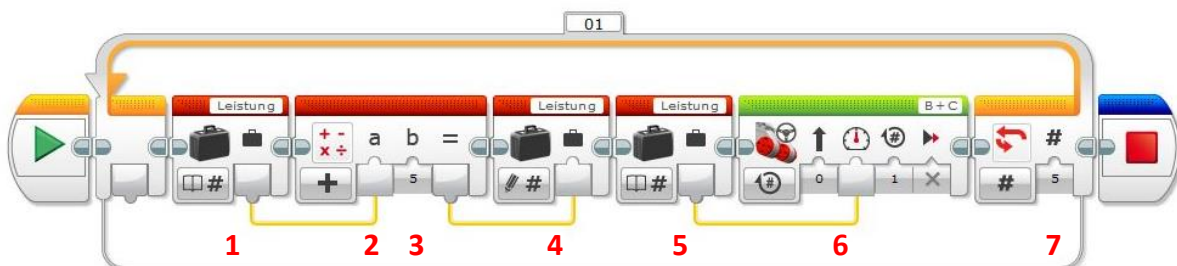
Aufgabe 2a: Roboter langsam starten und kontinuierlich schneller werden lassen

Diese Aufgabe ist eine Wiederholung der Kettenaufgabe. Es müssen fünf Motorenblöcke, in diesem Fall reicht der „Standardsteuerung“-Block aus, mit jeweils um den Wert 5 erhöhte Leistungswerte nacheinander geschaltet werden:



Aufgabe 2b: Roboter langsam starten und kontinuierlich schneller werden lassen, mit Hilfe einer Variablen und einer Schleife



Dieses Programm ist schon recht komplex und erfordert ggf. unterstützende Informationen, bspw. muss vorab geklärt werden, was eine Variable ist. Wichtigste Eigenschaft einer Variablen ist, dass sie den Wert ändern kann, was für die kontinuierliche Leistungssteigerung der Motoren notwendig ist.



Zunächst muss eine Variable eingefügt und erstellt (benannt) werden. Die Variablen sind in der roten Abteilung zu finden. Als erstes wird der Wert ausgelesen (1). Dann im Softwareblock „Mathematik“ auf die Position „a“ übergeben (2). In Position „b“ wird der Steigerungsfaktor, in diesem Fall „5“ festgelegt und addiert (3) und wieder an die Variable übergeben (4). Dabei muss die Funktion des Variablenblocks auf „schreiben“ (numerischer Wert) eingestellt werden. Um eine Verknüpfung zum Motorenblock herstellen zu können, wird im nächsten Schritt wieder die Variable ausgelesen (5) und übergeben, natürlich an die Position des Leistungswertes (6). Die Schleife wird auf den Zählen-Modus eingestellt und soll nach fünf Durchgängen enden (7).

Einheit 4 - Ultraschall-Sensor

ca. 1 Unterrichtsstunde

	<p>< vormontierter Ultraschallsensor</p>	
<p>Anbau an das Robot Educator Modell ></p>		
<p>Ziel(e): Den Ultraschallsensor und dessen Einstellungsoptionen kennenlernen. Verschiedene Messwerte zur Steuerung benutzen.</p>		
<ul style="list-style-type: none"> ➤ <i>Impulsfragen:</i> <ul style="list-style-type: none"> - Was ist ein Sensor? Was macht ein Sensor? - Wie kann ein Sensor auf Funktion des Roboters Einfluss nehmen? ➤ <i>Erarbeitung „Ultraschall“ - Impulsfragen:</i> <ul style="list-style-type: none"> - Was ist Ultraschall? Kennt ihr Tiere, die das benutzen? = Fledermäuse - Warum benutzen Fledermäuse Ultraschall? = Fliegen im Dunklen und müssen z.B. Hindernisse erkennen können. - ggf. Tafelbild: Funktion von Ultraschall (Reflektion von Schall) oder Film ➤ Aufgabe 1: Ultraschallsensor an „Robot Educator Modell“ anschließen und so programmieren, dass er vor einem Hindernis stoppt. ➤ Erklärung des Warteblocks und des Schwellwerts (Abstand zum Objekt) ➤ Umsetzung durch die Schüler ➤ Besprechung der Lösung ➤ Aufgabe 2: Roboter fährt zunächst schnell und nähert sich verlangsamt dem Objekt ➤ Besprechung der Lösung 		

Lösungen

Aufgabe 1: Roboter fährt und stoppt vor einem Objekt

Zunächst muss der Warteblock (orangene Abteilung) erklärt werden. Der Warteblock wartet auf ein Ereignis, einen Wert eines Sensors, wobei Sensor und Schwellwert eingestellt werden müssen. Wichtig: Beim Motorenblock, muss die Einstellung „an“ vorgenommen werden, d.h. der Roboter fährt solange vorwärts, bis der Warteblock feststellt, dass der eingestellte Schwellwert des Sensors überschritten (unterschritten) wird. Wichtig: Es muss die richtige Vergleichsoperation eingestellt werden!



In diesem Lösungsbeispiel startet der Roboter mit einer Motorenleistung von 50%. Der Warteblock vergleicht den Sensorwert des Ultraschallsensors, bis der Schwellwert (Abstand zum Objekt) kleiner als 25 (25 cm) ist. Dann werden die Motoren gestoppt. Achtung: Es besteht die Möglichkeit bei der Maßeinheit zwischen cm und Inch zu wählen!

Aufgabe 2: Roboter fährt und stoppt verlangsamend vor einem Objekt

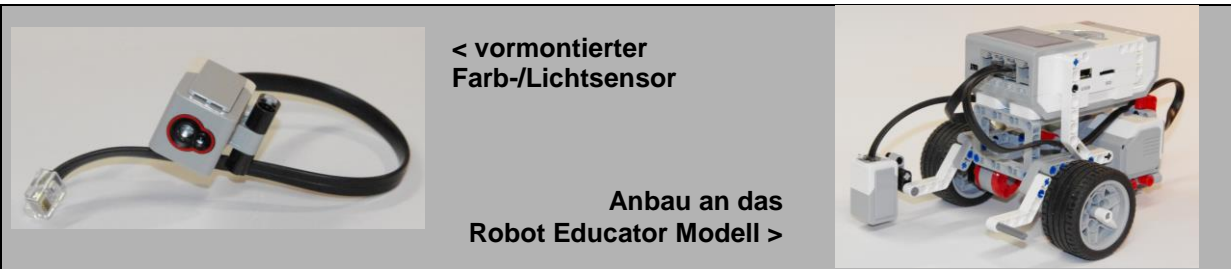
Als Kettenprogramm (ohne Schleife) wäre das eine mögliche Lösung:



Im Prinzip besteht das Programm aus zwei Schritten: Motoren einstellen, Warteblock mit Distanz, dann Leistung der Motoren senken, aber weiter fahren lassen, wieder Warteblock mit nun geringerer Distanz usw.

Einheit 5 – Farb-/Licht-Sensor

ca. 2-3 Unterrichtsstunden



< vormontierter
Farb-/Lichtsensor

Anbau an das
Robot Educator Modell >

Ziel(e): Den Roboter je nach Messwertergebnis unterschiedlich reagieren lassen.

Wiederholung der Schleifenfunktion.

Unterschiedliche Messverfahren beim Farb-/Lichtsensor kennenlernen.

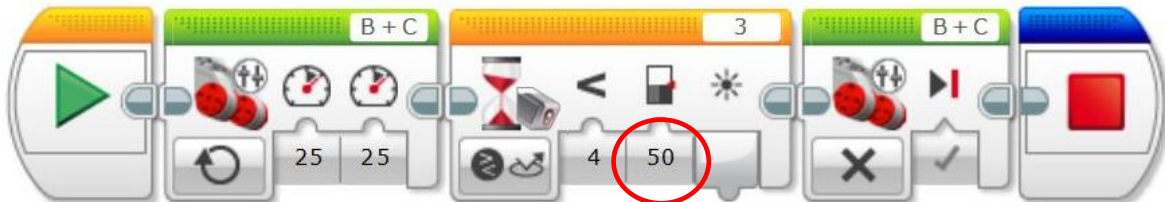
- *Erarbeitung „Farb- / Lichtsensor“ - Impulsfragen:*
 - *Vermutungen, was dieser Sensor macht*
 - reflektierendes Licht messen (weiß = hoher Wert, schwarz = kleiner Wert)
 - anhand des Reflektionswertes die Farbe bestimmen
 - anhand eines weißen Papiers mit einer (dicken) schwarzen Linie zeigen (EV3 per USB mit PC verbinden; in der Software unten rechts werden die aktuellen Werte (live) angezeigt > Farb-/Lichtsensor auf weiß und schwarz bewegen)
 - im UG Lösung der Aufgabe vorbesprechen:
- **Aufgabe 1:** Den Roboter vorwärts fahren und an der schwarzen Linie stoppen lassen
 - Motoren starten (an)
 - Warteblock mit Farb-/Lichtsensor
 - Schwellenwert hoch = weiß = fahren; Schwellenwert niedrig = schwarz = Motoren stoppen

Anmerkung: Der Schwellwert ist ein Wert in der Mitte der beiden Messwerte, d.h. wenn weiß=75 ist und schwarz=18, so kann der Schwellwert bei 50 eingestellt werden. Ausprobieren: Farbsensor langsam über die schwarze Linie schieben und Werteveränderung beobachten!
- **Zusatzaufgabe 1b für die Schnellen:** Den Roboter vorwärts fahren und an einem blauen Papier stoppen lassen (es bietet sich an, z.B. ein blaues und ein rotes Papier auszugeben, die beide in den Fahrweg des Roboters gelegt werden)
- Besprechung der Lösung(en)
- **Aufgabe 2:** Den Roboter in einem schwarzen Viereck fahren lassen
 - **Hinweis Material:** Am besten ein schwarzes Viereck mit einer dicken (!) schwarzen Linie auf DIN A4 ausdrucken und auf DIN A3 vergrößern (kopieren).
 - **Tipp:** Ggf. muss die Leistung der Motoren reduziert werden, denn wenn der Roboter zu schnell ist, kann er über die Linie „hinausfahren“ und weiter fahren. Empfehlung: Leistung auf 25% reduzieren; die Linie sollte ca. 1cm dick sein. Als kleiner Tipp kann den Schülern auch der Hinweis auf den Schleifenblock gegeben werden.
- **Zusatzaufgabe 2b für die Schnellen bzw. höhere Klassen:** Der Roboter fährt wie in Aufgabe 2 innerhalb des Vierecks. Wenn er über blaues Papier fährt, stoppt er.
- Besprechung der Lösung(en)

Lösungen

Aufgabe 1: Roboter vorwärts fahren und an einer schwarzen Linie stoppen lassen

Wie in der Vorbesprechung ist hier der Warteblock entscheidend und dessen korrekte Einstellung.



Zunächst wird mit dem Motorblock „Hebelsteuerung“ (oder Standard) beide Motoren mit einer Leistung von 25% gestartet (dauerhaft an). Im Warteblock wird der Farb-/Lichtsensor ausgewählt. Einstellung hierbei ist: „Vergleichen“ > „Stärke des reflektierenden Lichts“. Der mathematische Operator ist „kleiner als“ (4). Als Schwellwert wird hier „50“ eingegeben. Danach werden die Motoren gestoppt.

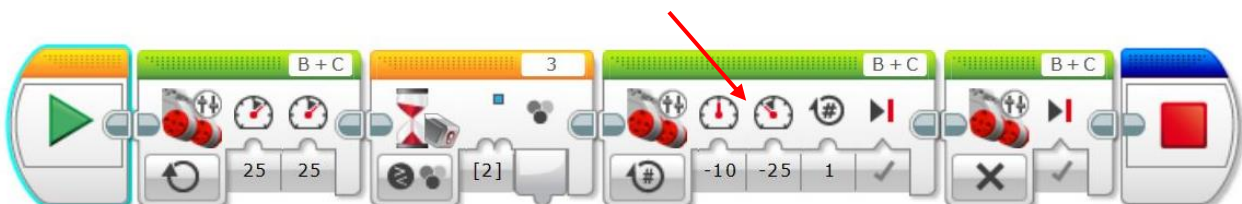
Aufgabe 1b: Roboter vorwärts fahren und bei einem blauen Papier stoppen lassen

Der Anfang ist wie bei Aufgabe 1. Beim Warteblock muss beim Sensor jedoch bei den Einstellungen „Vergleichen“ > „Farbe“ eingestellt werden. Dann wird statt eines Schwellwertes direkt die Farbe ausgewählt, danach die Motoren gestoppt.



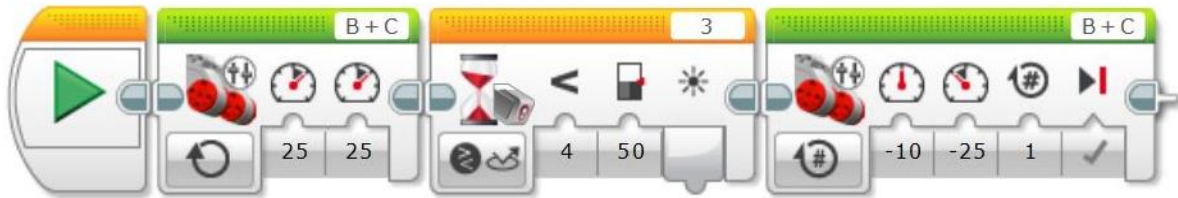
Hinweis:

In Vorbereitung auf Aufgabe 2 kann hier bspw. noch eingebaut werden, dass sich der Roboter, nachdem er das blaue Papier erfasst hat, ein Stück dreht. Zu empfehlen sind hier Werte für beide Motoren (bspw. „-10%“ und „-25%“).

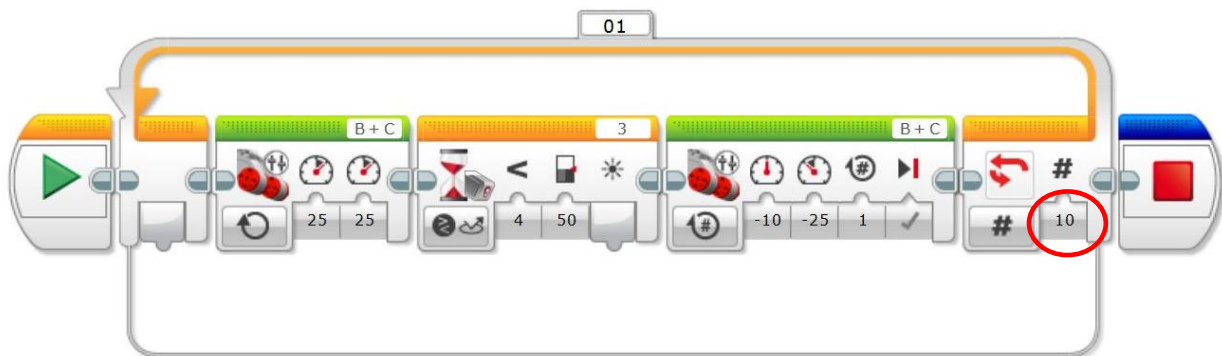


Aufgabe 2: Roboter innerhalb eines schwarzen Vierecks fahren lassen

Der Aufbau ist wie in Aufgabe 1: Der Roboter fährt vorwärts und erkennt die schwarze Linie (Warteblock). Statt die Motoren zu stoppen, bekommen beide unterschiedliche, negative Werte zugewiesen, damit der Roboter sich ein Stück rückwärtsfahrend dreht:



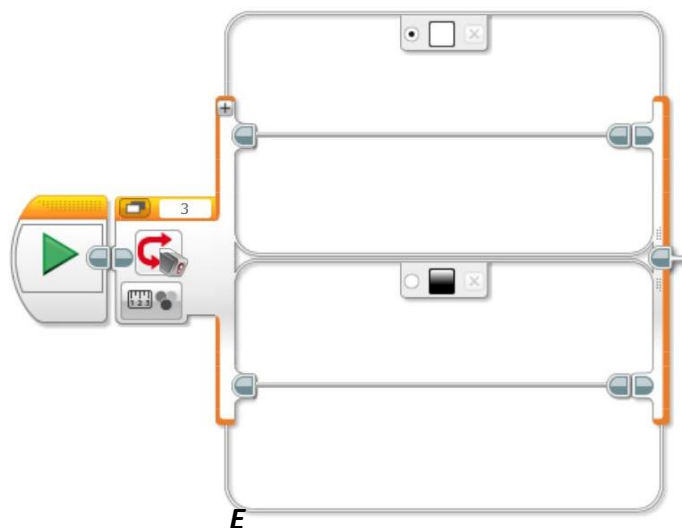
Jetzt soll der Roboter wieder vorwärts fahren, die schwarze Linie erkennen, rückwärts drehen usw. Es ist offensichtlich, dass hier wieder eine Schleife benötigt wird. Nach dem Startblock also eine Schleife einfügen (aus dem orangenen Abteil) und die drei Schritte (vorwärts fahren, Warteblock, Rückwärtsdrehung) in die Schleife einfügen:



Achtung: Bei der Schleife sollte ein Zähler verwendet werden, bspw. mit dem Wert „10“.

Zusatzaufgabe 2b: Roboter innerhalb eines schwarzen Vierecks fahren lassen und die Schleife erst dann unterbrechen, wenn der Roboter über blaues Papier fährt.

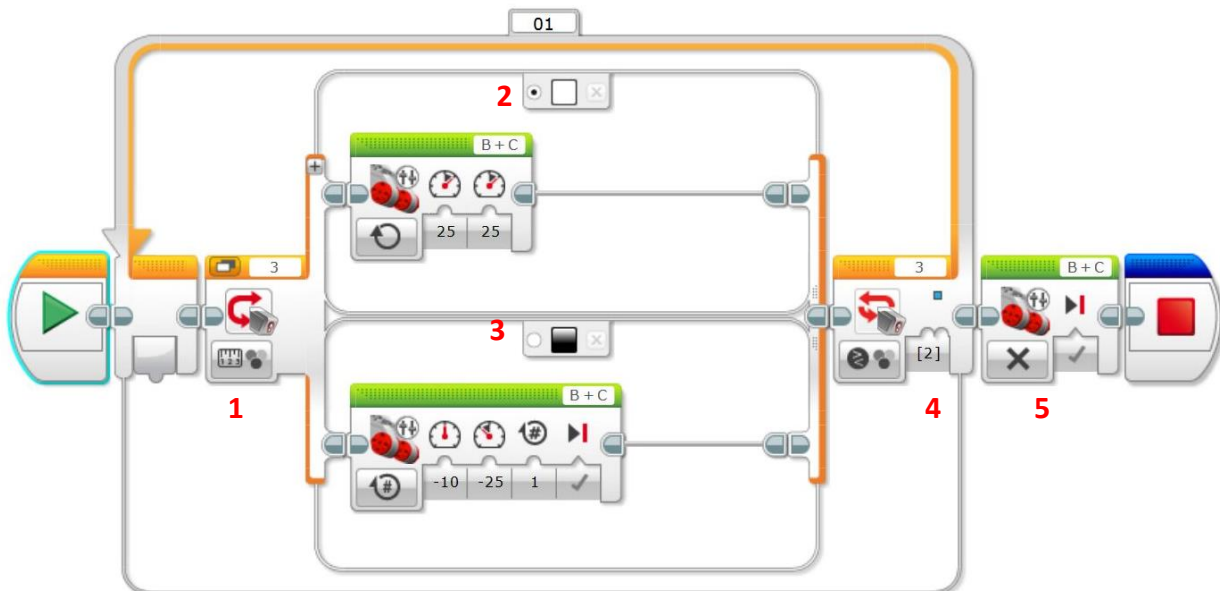
Dieses Programm ist etwas komplexer, daher u.U. auch eher für ältere Schüler. Denn mit dem Erkennen des blauen Papiers muss statt des Lichtsensors nun der Farbsensor eingesetzt werden. D.h. für die Bewegung wird nicht mehr die Stärke des reflektierenden Lichtes gemessen, sondern der konkrete Farbwert. Für die Programmierung bedeutet es, dass der Roboter bei „weiß“ fährt, bei „schwarz“ rückwärts dreht und bei „blau“ die Schleife verlässt. Damit ist die Einführung eines weiteren Programmierblocks, dem Schalter (Wenn-Dann-Logik), notwendig. Der befindet sich in der orangenen Abteilung rechts neben der Schleife.



Wenn-Dann-Funktion

Eine mögliche Lösung könnte so aussehen:

Das Programm startet mit der Schleife. In der Schleife befindet sich der Schalter. Diesem wird der Farbsensor zugewiesen, der die Farbe misst (Farbsensor > Messen > Farbe). Wenn „weiß“ gemessen wird (Standardfall), dann fährt der Roboter. Andernfalls, wenn „schwarz“ gemessen wird, dreht der Roboter rückwärts. Diese Schleife läuft so oft, bis der Farbsensor „blau“ misst. Erst dann wird die Schleife verlassen und die Motoren gestoppt.

**Beachten:**

1. Farbsensor + messen
2. Standardfall weiß
3. bei schwarz
4. Farbsensor messen + bei blau
5. Motoren stoppen

Einheit 6 – Taster

ca. 1 Unterrichtsstunde

	<p>< vormontierter Taster</p> <p>Anbau an das Robot Educator Modell ></p>	
<p>Ziel(e): Den Taster kennenlernen und eigenständig einen möglichen Einsatz überlegen und umsetzen. Erweiterte Funktionsmöglichkeiten der visuellen und akustischen Ausgabe kennenlernen.</p>		
<ul style="list-style-type: none"> ➤ Der Taster kann direkt an den Roboter gesteckt und verbunden werden. Alternativ kann eine kleine Halterung konstruiert und der Taster auf der Rückseite montiert werden. So kann die eine Fahrtrichtung mittels Ultraschallsensor kontrolliert werden, während beim Rückwärtsfahren der Taster die Berührung eines möglichen Hindernisses melden kann. ➤ Aufgabe 1: Den Taster nach eigenen Ideen einsetzen, bspw. <ul style="list-style-type: none"> - um ein Hindernis zu erkennen und zu stoppen - die Fahrtrichtung zu ändern - um ein Programm zu starten - um einen Ton / Melodie abzuspielen - ➤ Besprechung der Ergebnisse / Lösungen 		

Lösungen

Da hier verschiedene Ideen möglich sind, gibt es keine „pauschale“ Lösung. Am nachfolgenden Beispiel soll damit gezeigt werden, dass sich Klänge (Töne, Melodien, Sprache) wiedergeben lassen, ebenso Meldungen und Grafiken auf dem Display eingebunden werden können.



In dieser Beispiellösung wird beim Drücken des Tasters die Audiodatei „Fantastic“ abgespielt und die Bilddatei „Crazy 2“ (Augen) auf dem Display angezeigt. Um Audio- und Bilddateien auszuwählen, auf das weiße Feld oben rechts im Softwareblock klicken. Es gibt bereits eine große Auswahl bereitgestellter Dateien, aber es können auch eigene Dateien eingebunden werden.

WICHTIG: Um das Ergebnis zu hören und zu sehen, darf zum Abschluss KEIN Stopp-Block angeschlossen werden! Das Programm muss dann manuell am EV3-Baustein beendet werden!

Wichtige Links:

Für weitere Informationen und/oder zum Selbstlernen:

www.legoeducation.de/

<http://www.mindstormsforum.de/>

https://youtu.be/op_mgyUPges?list=PLdZ4eBU7Xk4p9hRz2rf_NJxMv_LemZGu-

Hintergrundinfos:

http://de.wikipedia.org/wiki/Lego_Mindstorms

Schule:

„Roberta“: Roboter und Mädchen:

<http://roberta-home.de>

<http://www.first-lego-league.org/de/>

Sonstiges:

<http://www.robotc.de/robotc-mit-dem-ev3/>

<http://worldrobotolympiad.de/>

<http://ldd.lego.com/de-de/>