

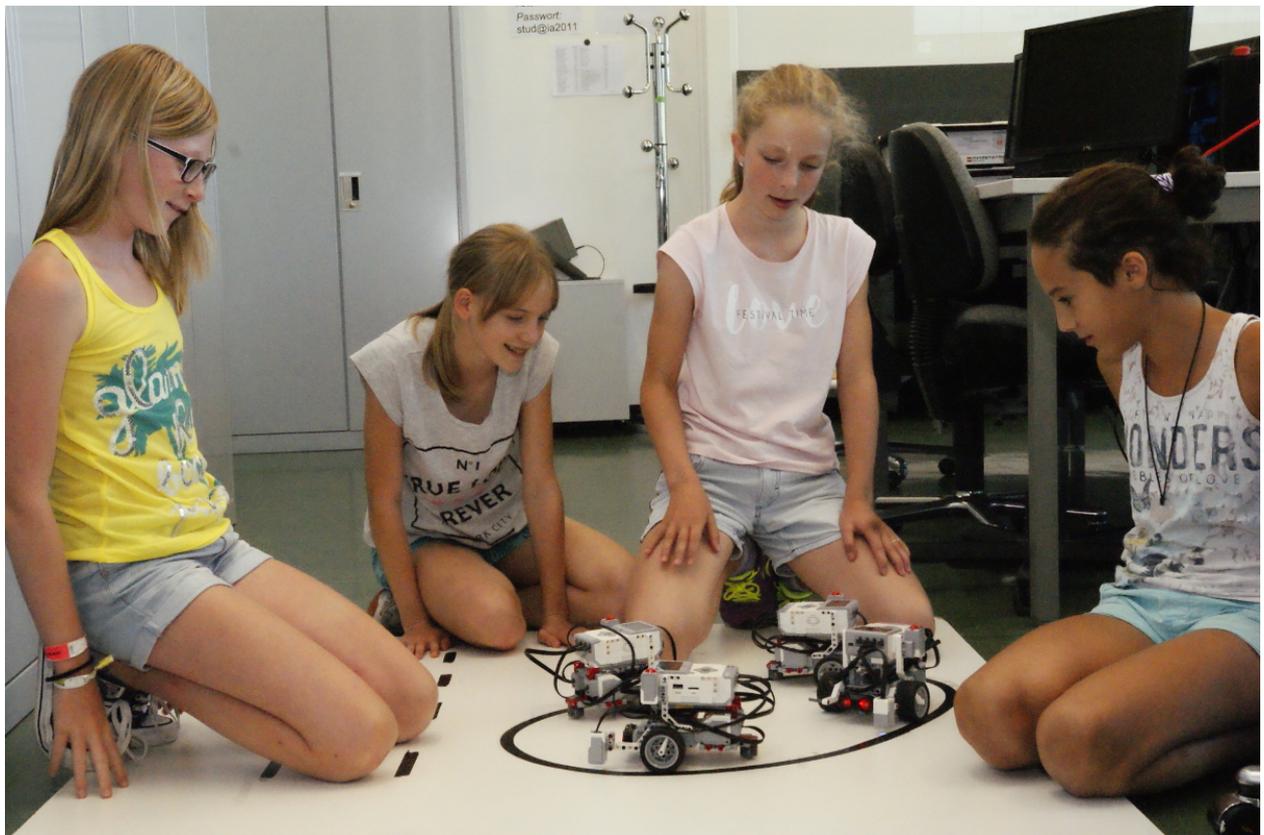


MINT Wahlpflichtfach BL/BS

Modul «Robotik»

– Jürg Keller und Matthias von Arx –

Dokumentation für die Lehrperson





Inhaltsverzeichnis

Vorwort.....	3
Didaktisches Konzept.....	3
Block 1	8
Aufbau des Basisroboters.....	10
Sensorparcours	11
Block 2: Lektion 1:.....	16
Wie rede ich mit einem Roboter?.....	16
Aufgabe 1 - sequentieller Programmablauf.....	22
Aufgabe 2 - Programmablauf mit explizitem 'Warten'.....	24
Aufgabe 3 - Programmablauf mit Schleife.....	26
Aufgabe 4 - Programmverzweigungen.....	28
Block 2: Lektionen 2-4.....	30
Die Programmierumgebung kennenlernen (Arbeitsblatt 3).....	30
Programmierübung 2: Fahren mit Kurve (Arbeitsblatt 4).....	32
Programmierübung 3: Fahren um Objekt (Arbeitsblatt 5).....	33
Programmierübung 4: Fahren einer Figur (Arbeitsblatt 6).....	34
Programmierübung 5: Sensorabhängiger Ton (Arbeitsblatt 7).....	35
Programmierübung 6: Sensorabhängiger Ton (Arbeitsblatt 8).....	35
Programmierübung 7: einfache Zusatzaufgabe (Arbeitsblatt 9).....	36
Programmierübung 8: Zusatzaufgabe (Arbeitsblatt 10).....	37
Block 3: Realisierung einer weiterführenden Aufgabe	38
Block 4: Roboter und Gesellschaft	39



Vorwort

Roboter kommen heute in sehr vielen Bereichen zum Einsatz. Von der robotergesteuerten Produktion (z.B. in der Autoindustrie) über den Operationsroboter im Spital und den Staubsaugerroboter zu Hause, bis hin zum Erkundungsroboter in Gebieten, die für Menschen nicht zugänglich oder gefährlich sind (z.B. auf dem Mars oder in Fukushima). In Zukunft werden solche Systeme vermehrt auch in unserem Alltag Einzug halten. Zudem gibt es eine ganze Reihe von Berufen oder Forschungs- und Entwicklungsbereichen, die sich mit dem Bau, der Steuerung und der Optimierung von Robotern beschäftigen. Genügend Gründe also, um sich mal direkt mit einem einfachen Roboter und seiner Steuerung auseinanderzusetzen.

Die bestehenden Lego-EV3 Bausätze eignen sich sehr für einen ersten Kontakt mit dem Thema. Das vorliegende Modul hat zum Ziel, die Schülerinnen und Schüler in relativ kurzer Zeit dazu zu befähigen, einen Roboter zu bauen und eine einfache Aufgabe ausführen zu lassen. Eine zentrale Rolle in unserem didaktischen Konzept spielt die Sprache. Lässt sich eine Aufgabe, die ein Roboter bewältigen soll, präzise in Sprache fassen, ist die Umsetzung letztlich fast nur noch eine Übersetzung in Programmiersprache. Da diese bei Lego-Robotik im Wesentlichen aus grafischen Bausteinen besteht, die aneinandergesetzt werden können, ist das gar nicht mehr so schwierig. Wir hoffen, dass Sie beim Ausprobieren zusammen mit Ihrer Klasse durchaus auch Spass haben werden. Und: Keine Angst! Sie werden selber mit jeder Lektion etwas dazulernen und nach 2-3 Klassen bereits selber ein kleiner Experte / eine kleine Expertin sein.

Für die ausserordentlich wertvollen Inputs bei der Überarbeitung des Moduls möchten wir Simone Meier, Sekundarschule Therwil herzlich danken.

Didaktisches Konzept

Im Folgenden werden die Ideen zur Einführung in die Robotik erläutert.

Thema	Beschreibung
Einführung in die Robotik	Die SuS sollen realisieren, dass die Robotik in ihrer Welt an vielen Orten anzutreffen ist und ein faszinierendes Gebiet ist. Sie sollen emotional engagiert werden.
Aufbau des Roboters, Material kennen lernen	Konstruktive und feinmotorische Fähigkeiten werden geübt und den SuS auch Hürden weggeräumt, selber etwas zu konstruieren
Sensorparcours	Wichtig zu erkennen ist, dass intelligente Systeme, die flexibel auf die Umwelt reagieren sollen, mit Sensoren ausgestattet sein müssen. Wenn dieses Bewusstsein fehlt, dann sind die späteren Lösungen recht simple Abfolgen von Fahrbefehlen. Kennen die SuS aber die Möglichkeiten und Tücken der Sensoren, so werden Sensoren auch eingesetzt.



Einführung in die Programmierung	<p>Die Programmierung eines Roboters ist eine sprachliche/bildliche Beschreibung der gewünschten Funktion. Was man nicht beschreiben kann, kann man auch nicht programmieren. Im Unterschied zur menschlichen Kommunikation ist die ‚Robotersprache‘ äusserst beschränkt. Die Gesetzmässigkeiten der Robotersprache können einzeln mit einem Arbeitsblatt oder mit einer Gruppenübung erkundet werden. Die Idee ist, dass die ‚Robotersprache‘ unabhängig von der Programmierumgebung eingeführt wird. Dies bietet den Vorteil, dass die Tücken der Bedienung der EV3-Applikation nicht mit den logischen Problemen der Roboterprogrammierung vermischt werden.</p> <p>Mit den Arbeitsblättern 3 bis 5 lernen die SuS ein gegebenes, abgebildetes Programm in EV3 Classroom nach zu programmieren. Dabei stehen die Programmieretechnik und der Umgang mit dem Roboter im Vordergrund.</p> <p>Im Arbeitsblatt 6 soll ein sehr einfaches Problem nun selbständig gelöst werden, ohne dass eine Vorlage dazu besteht.</p> <p>In den Arbeitsblättern 7 und 8 werden weitere Elemente der Programmierung kennen gelernt, die für mittelschwere Programme nötig sind.</p> <p>Arbeitsblatt 9 und 10 beinhalten Zusatzaufgabe für begabtere SuS. Darin werden fortgeschrittenere Programmiermöglichkeiten erlernt.</p>
Eigene Aufgabenstellung	<p>Eine eigene Idee umzusetzen, ist sehr motivierend. Dies soll im Block 3 erfolgen. Da es für SuS oft schwierig ist, die Komplexität eines Vorhabens abzuschätzen, soll mit Arbeitsblatt 11 eine Aufgabenanalyse erfolgen. Dies kann auch für eine spätere Benotung sehr nützlich sein.</p>

Ablauf der Unterrichtseinheit, Infrastruktur, Leitfaden

Sie finden in diesem Dossier alle Hintergrundinformationen und Beschreibungen, die Sie als Lehrperson brauchen. Planen Sie die das Modul (was die Materialien, die benötigten Laptops und die Software-Installation betrifft) ca. 2 Monate im Voraus. Setzen Sie sich zudem vor der ersten Durchführung des Moduls einen halben oder einen ganzen Tag mit den Blöcken 1 und 2 des Moduls auseinander (oder besuchen Sie das zugehörige Weiterbildungsangebot) und schon sind sie bereit, das Modul mit Ihrer ersten Klasse durchzuspielen.

Folgende Tabelle gibt Ihnen einen Überblick über die gesamte Unterrichtseinheit. Neben dem groben zeitlichen und inhaltlichen Ablauf wird auf die Dokumente hingewiesen, die Ihnen zur Verfügung stehen. Zudem wird auf besondere Material- oder Raumanforderungen hingewiesen, die Sie in der langfristigen Planung beachten sollten.



Dauer (Lektionen)	Inhalte	Dokumente	Besondere Material- und Raum-anforderungen
2-4	Block 1: Einstieg: Präsentation sowie Arbeitsblatt 1: Anwendungen der mobilen Robotik weitere 2-3 Lektionen: Material kennenlernen, Aufbau des Basisroboters, Sensorparcours	<i>Dokumentation fuer die Lehrperson.docx</i> <i>Einfuehrung.pptx</i> <i>Bauanleitung_Basisroboter.pdf</i> <i>1.1b_Arbeitsblatt_1_Einfuehrung.docx</i> <i>1.4b_Arbeitsblatt_2_Sensorparcours.docx</i> <i>1.3_Infoblatt_1_Das Material kennenlernen_S.docx</i> <i>1.4a_Postenblätter_Sensorparcours.docx</i>	1 Laptop oder PC pro Gruppe mit installierter Software. Lego Mindstorms EV3 Bausätze
2-4	Block 2: Einführung in die Programmierung von Robotern.	<i>2.1_Robotersprache_Arbeitsblatt_S.docx</i> <i>2.2a_Arbeitsblatt_3_mein erstes Programm.docx</i> <i>2.3_Arbeitsblatt_4_Fahren mit Kurve_S.docx</i> <i>2.4_Arbeitsblatt_5_Fahren um Objekt.docx</i> <i>2.4_Arbeitsblatt_6_Fahren einer Figur.docx</i> <i>2.5_Arbeitsblatt_7_Sensorabhaengiger Ton.docx</i> <i>2.6_Arbeitsblatt_8_Ton und Anzeige.docx</i> <i>2.7_Arbeitsblatt_9_ZA1_Ereignisse.docx</i> <i>2.8_Arbeitsblatt_10_ZA2_Wand.docx</i>	Siehe oben
4-8	Block 3: Offenere Problemstellungen bearbeiten oder eigene Ideen formulieren, Roboter bauen und programmieren, Lösung testen und verbessern	<i>Dokumentation-fuer-die-Lehrperson_Aufgabenideen-Block-3_S.docx</i> <i>3.1_Arbeitsblatt-11_Aufgabenanalyse.docx</i>	Siehe oben
2-4	Block 4: Robotik und Gesellschaft: Recherche und Arbeitsblatt	<i>4.1_Arbeitsblatt-12_Roboter und Gesellschaft.docx</i>	



Wichtigste Lernziele

- Die Lernenden kennen einige wichtige Anwendungsfelder der Robotik.
- Die Lernenden verstehen die Funktionsweise der Sensoren des Lego-EV3 Bausatzes.
- Die Lernenden können mit EV3 Software umgehen und sie zur Steuerung der Roboter verwenden.
- Die Lernenden sind in der Lage, einfache Steueranweisungen in eine "maschinenverständliche" Sprache zu übersetzen.
- Die Lernenden können einen Roboter bauen (bzw. umbauen) und ihn selbständig so programmieren, dass er eine Aufgabe ausführt.
- Die Lernenden können die Wechselwirkungen zwischen Robotik und Gesellschaft analysieren, Zusammenhänge erkennen und Chancen und Risiken beurteilen.

Vorbereitung

ca. 1-3 Monate vor Unterrichtsbeginn

Stellen Sie sicher, dass die Lego-EV3 Baukästen an Ihrer Schule bereitstehen.

1. Installieren Sie in Absprache oder zusammen mit Ihrem ICT-Verantwortlichen auf den Computern die Lego-Mindstorms-Education Software (Eine detaillierte, und stets aktuelle Anleitung zur Software-Installation finden Sie unter <https://education.lego.com/de-de/downloads/mindstorms-ev3/software>). Sie brauchen pro Gruppe einen Computer. Für die Installation brauchen Sie Administratorenrechte. Nach der Erstinstallation ist es sinnvoll gleich noch die verfügbaren Updates durchzuführen. **Auf jedem PC ist die Installation mit einem Programmstart zu überprüfen.** Es kann sein, dass bei fehlender/falscher Lizenz oder gescheiterter Installation das Programm startet, aber während dem Start hängen bleibt und über den Taskmanager beendet werden muss.
2. Zur Nutzung der Help-Funktionen muss ein Web-Browser auf den Rechnern installiert sein. Die Funktionsfähigkeit und Kompatibilität muss vorher überprüft werden.
3. Starten Sie die Lego-Mindstorms Software und schliessen Sie den EV3 an den PC an. Falls die Firmware-Version auf dem EV3 ('Betriebssystem' auf dem EV3) nicht mit der aktuellen Version der Lego-Mindstorms Software übereinstimmt, werden Sie zu einer Aktualisierung der Firmware aufgefordert. Wählen sie hier 'herunterladen' und warten sie genügend lange, bis der EV3 wieder von selber gestartet ist (kann 10 Minuten dauern). Eine Firmwareaktualisierung darf auf keinen Fall unterbrochen werden. Führen Sie, falls nötig, für jeden EV3 das notwendige Firmware-update durch. Falls eine Aktualisierung misslingt, ist ein spezielles Vorgehen nötig, um den EV3 zu retten. Details siehe im Kapitel 'Firmwareaktualisierung' in der EV3-Anleitung.
4. Es wird empfohlen, die Programmierung über USB-Kabel zu realisieren. Dies funktioniert sehr robust. Bei Bluetooth-Verbindungen kann es zu Verwechslungen des



Roboters und entsprechendem Chaos kommen. **Testen Sie auf jedem PC/Notebook**, ob die Verbindung zum angehängten EV3 klappt.

5. Gehen sie die ganze Unterrichtseinheit mit Hilfe der Lehrerunterlagen und der Schülerarbeitsblätter **im Schnellauf durch**. Nehmen sie sich die Zeit, die sie brauchen, um die Grundidee der einzelnen Teilschritte nachvollziehen zu können. Sie müssen nicht alles selber vorgängig ausprobieren. **Das Dossier hilft Ihnen den Unterricht zu gestalten, ohne selber bereits Experte oder Expertin zu sein.**

Bestimmen Sie, an welchem Ort auf dem PC oder auf einem USB-Stick die SuS ihre Programme abspeichern sollen.

Die Punkte 2 und 3 müssen nur einmal, vor der allerersten Durchführung des Moduls, am besten gleich für das ganze Schulhaus durchgeführt werden.

Ca. 1 Woche vor Unterrichtsbeginn

- Laden Sie die Akkus aller Roboter mit dem Zugehörigen Netzteil (Den Anschluss finden Sie unterhalb der Buchsen für die Sensoren).
- Nun kann es losgehen...



Block 1

Einstieg: Anwendungen der Robotik

Gehen Sie zusammen mit den Schülerinnen und Schüler die Power-Point Präsentation *Einführung* durch. Erkunden Sie evtl. zuvor, was die Lernenden schon wissen, bzw. welche Vorstellungen sie haben. Einige Leitfragen finden sie auf dem *Arbeitsblatt_1_Einführung*. Die Links zu den Filmausschnitten finden Sie in der Leiste mit den Bemerkungen zu den einzelnen Folien. Bei den Filmausschnitten genügt es jeweils eine Sequenz von 3 - 4 Minuten zu zeigen. Die Schülerinnen und Schüler sollten ein Gefühl dafür bekommen was ein Roboter ist, was er kann und wo er eingesetzt wird.

Benötigtes Material:

Beamer und PC

Musterlösungen zum Arbeitsblatt:

a) Was ist für dich ein Roboter? (Definition Roboter)

Bei Robotern handelt es sich um Maschinen. Sie zeichnen sich dadurch aus, dass sie sich selbstständig bewegen und verschiedene Tätigkeiten erledigen können. Ferngesteuerte Maschinen brauchen direkt Befehle von Menschen (z.B. ein ferngesteuertes Spielzeugauto), während Roboter mit Hilfe von Programmen Situationen beurteilen und entsprechend handeln können. Automaten sind keine Roboter, da sie nur eine einzige Arbeit ausführen. Da Computer sich nicht bewegen können, gelten auch sie nicht als Roboter.

c) Wo werden Roboter überall eingesetzt?

Einige Beispiele:

Autonome mobile Roboter (Erkundungsroboter, Minensuchroboter)

Humanoide Roboter (Spielzeuglaufroboter, Asimo, „menschliche“ Diener)

Industrieroboter (Schweisroboter)

Portalroboter (Werkzeugmaschinen, Positionierroboter)

Serviceroboter (Staubsaugerroboter)

Spielzeugroboter (Roboterhund, Fussballroboter)

und viele mehr

d) was können Roboter nicht, und e) Gefahren?

Diese Fragen können an dieser Stelle angediskutiert werden mit der Klasse. Was denken die Lernenden dazu? Eine vertiefte Auseinandersetzung mit diesen Fragen findet dann an konkreten Beispielen im Block 4 statt. Mögliche Punkte, die diskutiert werden können, sind: fehlende oder nur rudimentäre Intelligenz, fehlende Emotionen, keine moralischen Vorstellungen, Verlust von Arbeitsplätzen etc.



2-3 Lektionen: das Material kennenlernen; den Basisroboter bauen

Ein LEGO-EV3-Set besteht grundsätzlich aus dem EV3 Block (dem Gehirn des Roboters), verschiedenen Sensoren (sozusagen den Augen und Ohren des Roboters) und Motoren (sozusagen den Händen und Füßen des Roboters). Auf die LEGO-Technik-Bauteile, die für den Aufbau des Roboters gebraucht werden wird hier nicht näher eingegangen.

Die folgenden Unterkapitel geben einen kurzen Einblick in die Funktion der verschiedenen Teile.

Das Herzstück des Roboters: Der EV3-Block

Der weiße EV3-Block enthält einen Prozessor, der für die Steuerung des Roboters verwendet wird. Wurde ein Programm für den Roboter am Computer geschrieben, kann es auf den EV3-Block übertragen und gespeichert werden. Von da aus kann das Programm ausgeführt werden. Auf der grauen Rückseite befindet sich ein Batteriefach für 6 AA-Batterien oder einen Akku. Auf der unteren Seite sind vier nummerierte Anschlüsse für die Sensoren und auf der oberen Seite sind vier Anschlüsse für die Motoren.

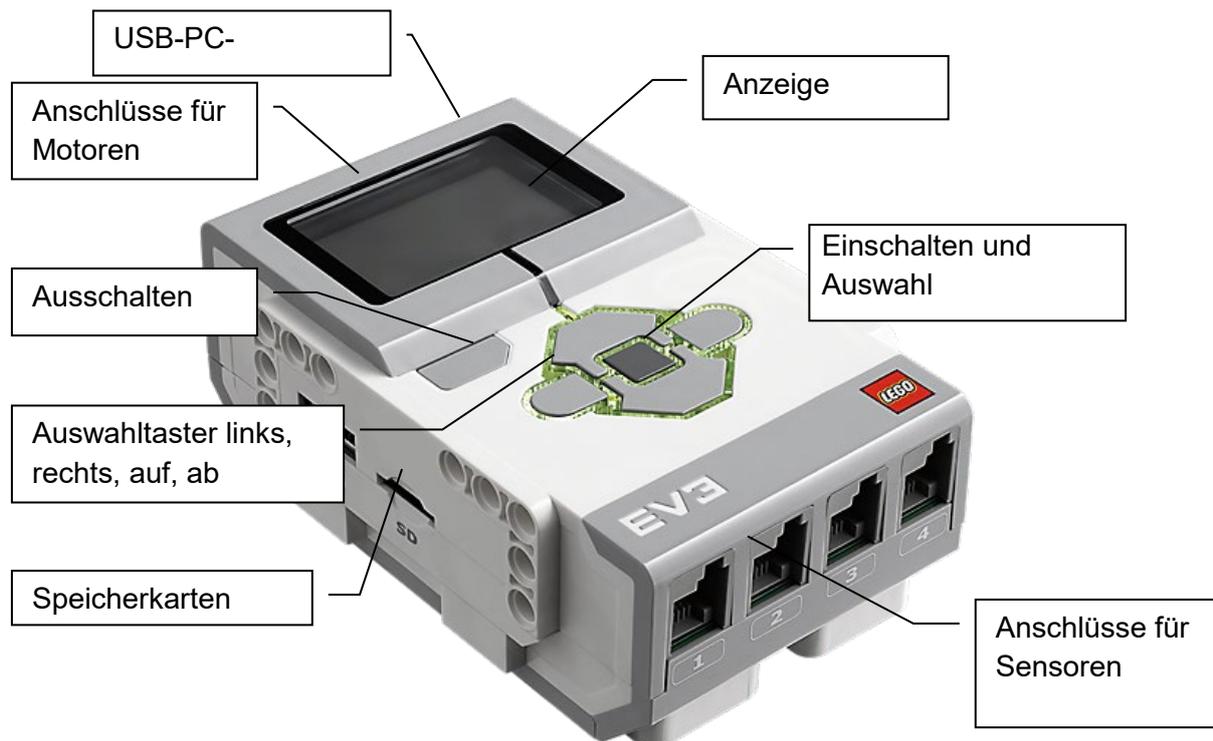
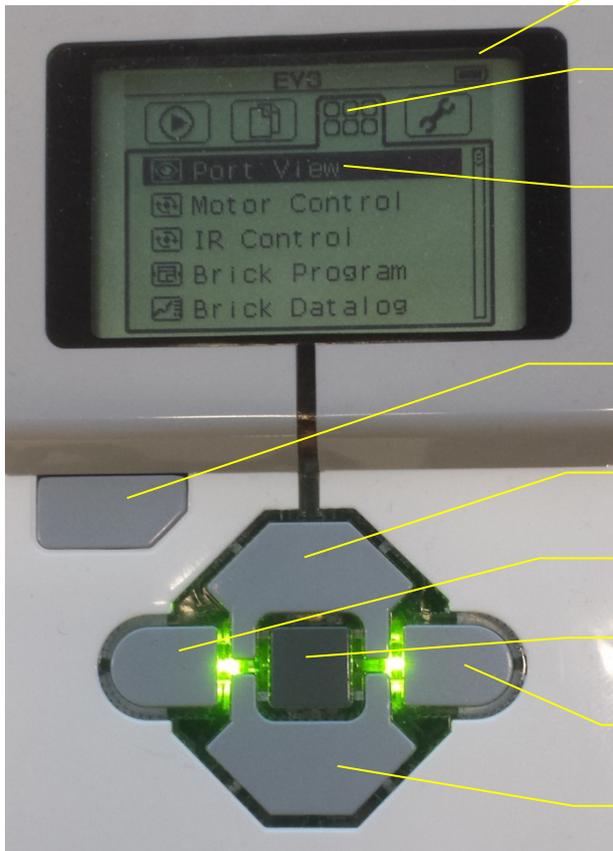


Abbildung 1: EV3-Block



Anzeige und Bedienung des EV3-Blocks:



- Batterieladestand
- Ordner: Programme, Files, Eingaben, Einstellungen
- Auswahl des Sensoreingangs
- Eingabe rückgängig machen & ausschalten
- Menu-Navigation nach 'oben'
- Menu-Navigation nach 'links'
- bestätigen, Auswahl
- Menu-Navigation nach 'rechts'
- Menu-Navigation nach 'unten'

Abbildung 2: EV3-Bedienung

Aufbau des Basisroboters

Eine Anleitung für den Aufbau des Basisroboters befindet sich in der Lego-Mindstorms-EV3-Software.

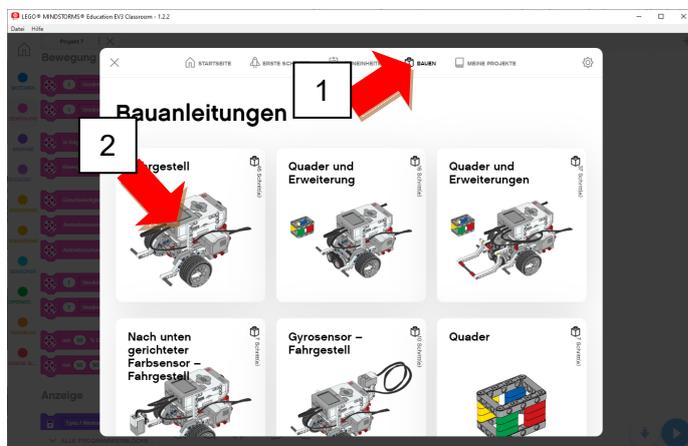


Abbildung 3: Weg zur Bauanleitung Basisroboter (Quelle ScreenShot aus Education EV3 Programm)



Die gleiche Anleitung finden Sie auch in gedruckter Form in jedem EV3-Bausatz und ist auch in den Unterrichtsunterlagen verfügbar.

HINWEIS: Mit dem Bau des Roboters kann sehr viel Zeit verloren gehen. Hier ist zu beachten, dass sich die SuS nicht beliebig in eigene Variationen verlieren. Es wird dringend empfohlen, den Roboter genau nach Plan zu bauen. Der Basisroboter zeichnet sich durch eine optimale Kombination von Funktionalität und Ausbaufähigkeit bei gleichzeitiger Einfachheit aus. Für den Sensorparcours muss der Roboter nicht fertiggestellt sein. Es ist also auch möglich, die Aufbauarbeit etwas aufzulockern und den Sensorparcours (oder einen Teil davon) (siehe nächster Abschnitt) dazwischen zu absolvieren.

Sensorparcours

Hier stehen vor allem das Kennenlernen des Materials und die Funktion der Sensoren im Vordergrund. Mit Hilfe eines Sensorparcours wird der Einsatz von verschiedenen Sensoren auf spielerische Weise getestet und die Eigenheiten der Sensoren werden untersucht. Nur wenn die SuS die Möglichkeiten der Sensoren kennen, werden sie diese im weiteren Verlauf auch richtig anwenden und einsetzen können.

Der Sensorparcours besteht aus sechs verschiedenen Posten. Die SuS werden in sechs Gruppen aufgeteilt und auf die Posten verteilt. Die Stationen werden nacheinander durchgearbeitet. Es reicht, wenn jede Gruppe 3 oder 4 Posten durcharbeitet. Am letzten Posten fasst die Gruppe die Ergebnisse dieses Postens anhand folgender drei Leitfragen zusammen:

- 1) Was kann der Sensor?
- 2) Wozu kann man diese Fähigkeit nutzen?
- 3) Welche Tücken muss man beachten?

Die Antworten werden auf ein A3-Blatt geschrieben und an der Tafel aufgehängt. So haben alle jederzeit den Überblick über alle Sensoren.

Für den Sensorparcours sollten nicht mehr als 45 Minuten verwendet werden.

Benötigtes Material:

- Pro Posten 1x EV3-Basis Roboter
- Pro Posten 1 Verbindungskabel
- Pro Posten 1 Sensor
- Pro Posten 1 Postenblatt (aus *Postenblätter_Sensorparcours.docx*)
- Pro Posten 1 leeres A3-Blatt für die Zusammenfassungen
- Pro SuS 1 *Infoblatt_1_Das Material kennenlernen.docx*.
- Pro SuS ein *Arbeitsblatt_2_Sensorparcours.docx* mit den zu beantwortenden Fragen
- farbige Legosteine.



Hinweise:

1. Zur Bearbeitung des Sensorparcours brauchen die SuS auch das *Infoblatt_1*. Sie müssen dieses lesen, damit sie die Posten absolvieren können (also z.B. Werte am EV3-Block ablesen können, etc).
2. Machen sie die SuS auch darauf aufmerksam, dass man bei den drei Lichtsensor-Posten immer mit dem gleichen Sensor arbeitet, durch die Einstellungswahl am EV3-Block aber dann entscheiden kann, in welchem Modus (mit LED, ohne LED, Farbe) der Sensor arbeiten soll. Wie das geht ist auf dem jeweiligen Postenblatt beschrieben.

Technische Tipps zur Frage 3:

Die Sensoren haben z. T. auch gewisse Tücken, welche die SuS beim Sensorparcours erkennen sollten:

- Der Ultraschallsensor kann Probleme haben, wenn er nicht nahe einem rechten Winkel zu einer Wand (oder ähnlichem) sendet. Wenn er z. B. schräg zu einer Wand sendet, erkennt er die Wand nicht, weil das Signal nicht zurückgestreut wird (siehe Skizze unten). Auch sehr schmale Objekte, wie z.B. ein Stuhlbein werden nicht erkannt.
- Die hell-dunkel-Unterscheidungen der Lichtsensoren funktionieren nur dann zuverlässig, wenn starke Kontraste auftreten. Zwischen weiss und schwarz kann z.B. gut unterschieden werden, bei Grautönen wird es aber schwierig. Wichtig ist der Abstand des Sensors zur Oberfläche
- Beim Gyrosensor ist die richtige Kalibrierung sehr wichtig. Wird der Nullpunkt für die Drehgeschwindigkeit zu Beginn nicht richtig festgelegt, so "läuft die Winkelmessung davon". Das heisst, der Sensor nimmt eine Drehung wahr, obwohl keine Drehung vorhanden ist. Auch misst der Gyrosensor nur Drehungen um eine Achse.

Die Einsicht in diese "Tücken" ist wichtig. Später werden bei den Versuchen mit den selbstgebauten Robotern oft Probleme auftreten, die genau mit diesen Tücken zu tun haben. Deshalb sollte man sie kennen und bei der Planung von Anfang an mitdenken.



In der Folge sind die Hauptkomponenten (EV3-Block) und die Sensoren kurz beschrieben. Die gleichen Angaben (etwas anders gruppiert) finden die Schülerinnen und Schüler auf ihren Posten-, Arbeits- und Infoblättern.

Berührungssensor



Der Roboter „fühlt“. Der Kontaktschalter besitzt vorne ein rotes bewegliches Köpfchen, das hineingedrückt werden kann. Fährt der Roboter mit diesem Sensor gegen eine Wand oder einen anderen Gegenstand, wird das Köpfchen eingedrückt und der Roboter merkt, dass etwas im Weg ist.

Licht- und Farbsensor



Der Roboter „sieht“. Der Lichtsensor misst die RGB-Farben oder umgerechnet auch die Lichtstärke (Helligkeit) in seiner Umgebung. Auf der Front sieht man zwei kleine Linsen. Die obere ist der Sensor, welcher die RGB-Farben misst und die untere ist ein LED (also quasi eine kleine Taschenlampe). Der Sensor kann in 3 verschiedenen Varianten gebraucht werden: 1. Messung der Lichtstärke, die in den Sensor fällt, ohne eigene Beleuchtung, als nur Umgebungslicht. 2.

Messung der Lichtstärke mit eigener LED-Beleuchtung. 3. Messung der Farbe eines Objekts. Mit Hilfe der LED-Beleuchtung kann der Sensor eine dunkle Oberfläche von einer hellen Oberfläche unterscheiden. So kann zum Beispiel einer Spur von schwarzem Klebeband auf einer weissen Unterlage gefolgt werden. Der Sensor sagt dem Roboter nicht nur ob es hell oder dunkel ist, sondern gibt einen Zahlenwert aus, der die aktuelle Helligkeit beschreibt. Die beste Distanz zur Oberfläche ist etwa 0.5 cm. In Port View oder mit einem Programm wird festgelegt, in welchem Modus der Sensor betrieben werden soll. Der Sensor kann Farben richtig erkennen, falls die Objektoberfläche nicht zu glänzend ist und nicht farbige Lichtquellen die Messung stören.

Distanzsensoren/Ultraschallsensoren



Mit dem Ultraschallsensor kann der Roboter Entfernungen zu bestimmten Objekten oder zur Wand messen. Er besteht aus einem Sender im linken Auge und einem Empfänger im rechten Auge. Das Prinzip ist dasselbe wie bei einer Fledermaus, die durch die Nacht fliegt. Der Sender sendet ein Ultraschallsignal aus, das auf einen Gegenstand oder eine Wand trifft und dabei reflektiert wird. Der



Empfänger empfängt das wiederkehrende Signal und misst die Zeit zwischen Senden und Empfangen. So kann die Entfernung des Gegenstandes berechnet werden.

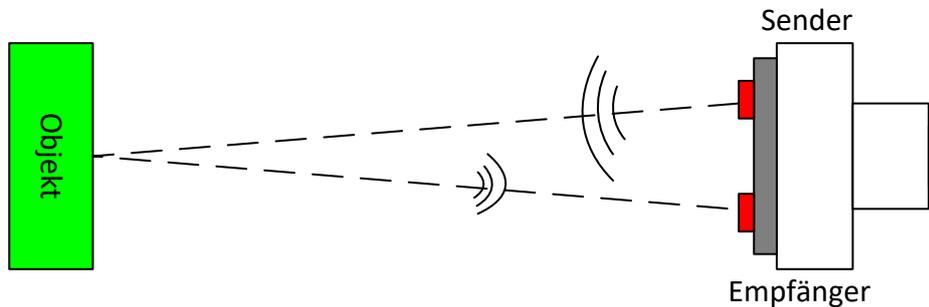


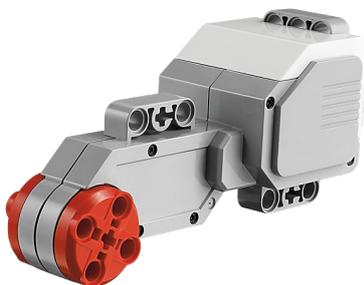
Abbildung 4: Prinzip Ultraschall

Gyro- oder Drehsensor



Der Gyrosensor misst die Drehgeschwindigkeit. Er kann also messen, wie schnell ein Roboter sich um diejenige Achse dreht, die senkrecht auf dem roten Kreis des Sensors steht (die von oben nach unten durch den roten Punkt geht). Aus der Drehgeschwindigkeit kann auch der Drehwinkel berechnet werden. Damit dies möglichst exakt erfolgen kann, darf die Drehgeschwindigkeitsmessung keinen Fehler des Nullpunkts (keine Drehung) haben. Es muss dem Sensor beigebracht werden, für welchen gemessen Wert keine Bewegung stattfindet. Dies nennt man Sensor-Kalibrierung. Der Sensor wird beim Einschalten des Roboters kalibriert. Damit dies richtig funktioniert, muss der Roboter beim Einschalten absolut ruhig auf dem Tisch stehen.

Grosse Motoren



Diese Motoren sind die wichtigsten Akteure in der Lego-Robotik, denn sie erlauben kontrollierte Bewegungen. Sie können mit unterschiedlichen Geschwindigkeiten drehen oder auch nur eine ganz bestimmte Anzahl Drehungen ausführen. (Zum Beispiel nur eine halbe Drehung). Ein Motor kann auch messen, um welchen Winkel seine Achse gedreht wurde. Dies ist nützlich, wenn man z.B. eine gewünschte Distanz fahren will.



Kleine Motoren



Die mittleren Motoren können etwas einfacher in einen Roboter eingebaut werden, haben weniger Kraft, drehen aber schneller und präziser als die grossen Motoren

Kabel



Die Kabel verbinden Sensoren und Motoren mit dem EV3-Block. Es wird dasselbe Kabel für Sensoren und Motoren verwendet.



Block 2: Lektion 1:

Wie rede ich mit einem Roboter?

Ein Mensch kann einem anderen Menschen eine Anweisung geben, z. B. ‚nimm ein Blatt Papier hervor‘ oder ‚hole in der Küche ein Glas Wasser‘. Der Mensch ist schon in jungen Jahren fähig, die Aufgabe zu lösen. Geht man mit dieser Vorstellung an die Programmierung eines Roboters, so wird man damit viel Mühe haben. Die wesentlichen Unterschiede sind:

- Der Roboter versteht nur ein minimales Vokabular, je nach Zählweise etwa 20-30 Wörter
- Die Wörter bezeichnen nur elementare Befehle und sehr, sehr einfache Aufgaben
- Der Roboter kann unpräzise, ähnliche Bezeichnungen nicht verstehen
- Das Verhalten des Roboters, d.h. das Roboterprogramm muss im Voraus komplett erstellt werden. Er kann auf unvorhergesehene Situation nicht sinnvoll reagieren.
- Die Sensoren eines Lego-Roboters sind sehr einfach und die Anzahl ist auf 4 beschränkt
- Einem Menschen ist der Aufbau einer Anweisung nicht bewusst, da vieles aus dem Kontext heraus klar ist.

Folgende Lernschritte müssen die SuS erreichen:

1. Die SuS müssen verstehen, dass eine Tätigkeit des Roboters vorkonfiguriert, dann gestartet und wieder gestoppt oder geändert werden muss. Der Start erfolgt entweder durch den Programmablauf oder durch ein Ereignis. Das Ende einer Tätigkeit ist immer durch das Erfüllen einer Bedingung gegeben. Dies ist ein grundlegendes Prinzip, das nicht nur in der Robotik gilt. Die meisten Programmierfehler entstehen nach den ersten einfachen Programmen dadurch, dass einige Bewegungsblöcke (auch Anzeige und Soundeffekte) die gestartete Tätigkeit stur zu Ende führen und mit einfachen Programmiermethoden nicht unterbrochen werden können. Der Programmablauf bleibt z.B. im Bewegungsblock '100 Umdrehung vorwärts bewegen', bis der Motor 100 Umdrehungen gemacht hat, dies auch wenn im folgenden Schritt ein 'Bewegung anhalten' programmiert wird.



Die Bewegungsblöcke sind bereits eine intelligente Sequenz von elementaren Anweisungen, welche natürlich das Programmieren stark vereinfachen. Dies wird im Folgenden erläutert.



Vereinfachte Programmierung mit Bewegungen



'Elementare' Programmierung



Abbildung 5: Vergleich 'intelligenter' Programmierblock mit elementarer Programmierung

In Abbildung 5 wird der Bewegungsbefehl links '1 Umdrehung mit Geschwindigkeit 50%, geradeaus fahren' mittels einer Sequenz elementarer Bausteine nachprogrammiert. Das Programm funktioniert wie folgt: Mit dem Sensor im Motor B wird die Umdrehungszahl gemessen. Darum muss der Wert auf 0 gesetzt werden. Danach müssen die Motoren 'B' und 'C' mit 50% Geschwindigkeit gestartet werden. Der Programmablauf wartet dann bis der Radwinkel grösser als 360 Grad ist. Danach werden beide Motoren gestoppt. Sobald eine Aufgabe nicht mehr mit intelligenten Programmierblöcken gelöst werden kann, wird die Aufgabe ziemlich schwieriger.

2. Die SuS müssen die Möglichkeiten kennen, wie sie den Programmablauf steuern können. Sie müssen die Strukturen erkennen können, diese benennen und auch zielgerichtet anwenden können. Mit zunehmender Komplexität sind dies:
 - a. im Programmablauf 'warten'
 - b. eine Abfolge von Programmierblöcken wiederholen
 - c. bedingte Verzweigungen im Programmablauf realisieren.

Sprachlich gibt es zwei Hürden:

- a. Programmierblock 'warten bis ...' führt nicht dazu, dass der Roboter selber wartet, sondern nur der Programmablauf nicht weitergeht, bis eine Bedingung erfüllt ist.
 - b. Die Kategorie der Programmierblöcke 'Steuerung' betrifft nicht die Steuerung des Roboters sondern die Steuerung des Programmablaufs.
3. Die SuS müssen verstehen, dass der Programmablauf über Bedingungen entschieden wird. Sie müssen diese richtig zur Realisierung des gewünschten Verhaltens einsetzen können. Erschwerend ist, dass die Bedingungen an unterschiedlichen Orten zu finden



sind. In einigen Bewegungen sind sie integriert, für die Programmablaufsteuerung müssen sie als Zähl- oder Zeitbedingung angegeben werden oder aufgrund von Sensorsignalen berechnet werden.

Als Methodik wird vorgeschlagen, die vorangehend beschriebenen Elemente eines Roboterprogramms losgelöst von einer Programmierung zu bearbeiten. Dies kann mit Hilfe des Arbeitsblatts '2.1_Robotersprache_Arbeitsblatt_S' in Einzelarbeit oder als Postenaufgabe instruiert werden.

Die Lösungen zum 'Arbeitsblatt 2.1_Robotersprache_Arbeitsblatt_S' sind:

Aufgabe 1:

- Mache alles leise
- Stehe auf
- Gehe seitlich bis in den Gang
- Laufe 10 Schritte vor
- Drehe um 90 Grad
- Laufe vorwärts bis zur Türe

Quiz 1:

Zu welcher Art gehören die folgenden Anweisungen? Mache ein Kreuz in der richtigen Spalte.

Anweisung	Voreinstellung	Tätigkeit	Schaltanweisung
Drehe dich um 90 Grad		X	
Benutze das Trottinet	X		
Schalte das Licht an			X
Mache einen Ton für 2 Sekunden		X	
Brauche den Motor mit einer Geschwindigkeit von 50%	X		
Motor mit Geschwindigkeit 50% und Linkskurve starten.			X
Nimm's gemütlich!	X		
Schreibe 'Juhui' auf Zeile 1 des Displays			X
2 Motorumdrehungen mit 50% Geschwindigkeit und starker Rechtskurve machen		X	
Laufe weg			X



Quiz 2:

Um welche Programmablaufsteuerung geht es hier? Markiere die Tätigkeit und die Programmsteuerung mit einer unterschiedlichen Farbe

Anweisung	Wiederholen	Verzweigen	Warten
Renne 5 mal um das Haus	X		
Warte bis das Wasser heiss ist, dann kannst du die Spaghetti hineingeben.			X
Drücke 5 mal auf die Eingabetaste	X		
Wenn du auf 'OK' drückst, dann wird das Bild gespeichert, sonst wird es gelöscht.		X	
Mache die Hausaufgaben. Wenn du fertig bist, kannst du spielen gehen			X
Wenn der Velopneu noch genügend Druck hat, fahre nach Hause, sonst musst du das Velo schieben		X	
Nach 8 Minuten kannst du das Spaghettiwasser abschütten			X
Fahre bis die Distanz kleiner als 10 cm ist. Danach halte an.			X
Fahre nach links, wenn der Helligkeitssensorwert kleiner 20 ist, sonst nach rechts		X	

Alternativ kann dies auch in einem Rollenspiel in der Klasse vermittelt werden. Jemand aus der Klasse oder die Lehrperson übernimmt die Rolle des Roboters. Dieser wird als der Robotermensch bezeichnet. Der Rest der Klasse wird in drei Gruppen aufgeteilt. Die SuS der ersten Gruppe spielen die Programmierer. Die zweite Gruppe beurteilt die Programmierer und die dritte Gruppe beurteilt das Verhalten des Roboters. Dies muss natürlich den Vorkenntnissen angepasst werden, die evtl. im Informatikunterricht vermittelt wurden.

Der Ablauf des Spiels ist wie folgt:

- Die Klasse vereinbart das Robotervokabular. Als Vorlage dient das Robotervokabular gemäss '2.1_Robotersprache_Arbeitsblatt_S' und der Präsentation '2.1a_PPP_AB-3_Robotersprache_SM.pptx'. Dieses ist analog den Möglichkeiten konzipiert, die dann bei der Programmierung vorhanden sind. Weitere Anweisungen können hinzugefügt werden, sollen aber nicht zu ‚intelligent‘ sein.



- Die erste Gruppe der **Programmierer** erhalten eine Aufgabe. Beispiele sind unten angegeben. Die Gruppe berät, mit welcher Sequenz von Anweisungen die Aufgabe durch den Roboter gelöst werden kann. Ein SuS liest das Programm (soll einfach bleiben) dem Roboter vor.
- Der Roboter führt das Programm exakt so wie befohlen aus.
- Hat der Roboter das Programm beendet, so wird die Aufgabe besprochen. Die zweite Gruppe, **die Programmüberprüfungsgruppe**, beurteilt dabei das Roboterprogramm. Folgende Fragen sind zu beantworten: Wurden Begriffe verwendet, die nicht vorher abgemacht wurden? War das Programm klar? Wie könnte das Programm verbessert werden?
- Die dritte Gruppe, **die Roboterkontrolleure**, beurteilt das Verhalten des Roboters. Hat der Roboter das Programm stur durchgeführt? Hat er selber etwas dazu interpretiert?

Bemerkungen:

1. Bei der Erstellung eines Roboterprogramms trifft man auch auf die Aufgaben der Gruppen. Selbstverständlich muss ein Programm programmiert werden. Hat man in EV3 Classroom ein Programm erstellt, so wird im Hintergrund überprüft, ob das Programm formell richtig ist. So können Blöcke nur dort aneinandergehängt werden, wenn sie auch passen. Die dritte Aufgabe, nämlich das Analysieren der Robotertätigkeiten, ist von sehr grosser Bedeutung. Selten gelingt es, im ersten Anlauf ein Programm ohne Fehler zu schreiben. Die Analyse, was der Roboter wirklich macht, ist darum sehr wichtig für das Korrigieren von Fehlern. Nur wenn man beobachtet und verstanden hat, was der Roboter wirklich macht, ist eine Programmkorrektur möglich. In dieser Rolle sollen die SuS gezielt auf diese Analyse hin geschult werden. Sie müssen darum als Lehrperson in der Rolle als **Robotermensch** auch bewusst Programmierfehler oder Unklarheiten deutlich erkennbar machen.
2. Eventuell muss das Robotervokabular erweitert werden und eine Übung wiederholt werden.
3. Verwenden sie so viel Zeit für diese Aufgaben wie sinnvoll. Das kann je nach Klasse sehr unterschiedlich sein. Sie können diese Art von Übung auch zu jedem beliebigen Zeitpunkt (z.B. während des 3. Blocks) wieder durchführen, um bei den SuS das Bewusstsein für die Möglichkeiten und Grenzen von Programmcode zu erneuern.
4. Die in der Folge verwendeten Farben, violett für Aktionen und orange für die Ablaufsteuerung, haben ihre Entsprechungen in der Programmierumgebung, bereiten also ebenfalls bereits auf das Programmieren vor.

In der Folge wird detailliert dargestellt, welche Rolle Sie als Lehrperson übernehmen sollten und es wird erklärt welchen Aspekten Sie besonderes Gewicht beimessen müssen. Zudem geben wir Ihnen Hinweise zum Ablauf sowie methodisch-didaktische Tipps für die Umsetzung im



Klassenzimmer. Sie finden auf den nächsten Seiten eine Art Drehbuch für diese Lektion, auf das Sie sich abstützen können.

Benötigtes Material:

- PC und Beamer zur Vorstellung der Aufgaben (*Aufgaben_Robotersprache.pptx*)
- Für jeden SuS je ein *2.1_Robotersprache_Arbeitsblatt_S.docx*
- Wandtafel



Aufgabe 1 - sequentieller Programmablauf

Motivation: Mit dieser Aufgabe sollen die SuS erkennen, dass ein einfaches Programm aus einer Abfolge oder Sequenz von Befehlen besteht. Diese müssen in der Robotersprache befohlen werden, um für eine mit keinerlei Intelligenz zur Interpretation der Befehle ausgestatteten Maschine verständlich zu sein.

Ablauf: Erläutern Sie der Klasse die Aufgabe anhand dieser Abbildung (siehe auch *2.1a_PPP_AB-3_Robotersprache.pptx*) und weisen Sie nochmals auf die Spielregeln hin, bevor die SuS mit der Bearbeitung beginnen.

Im Klassenzimmer wird ein Spielfeld gemäss Abbildung vorbereitet. Am besten befestigen Sie im Start und Zielbereich je ein beschriftetes Blatt Papier am Boden.

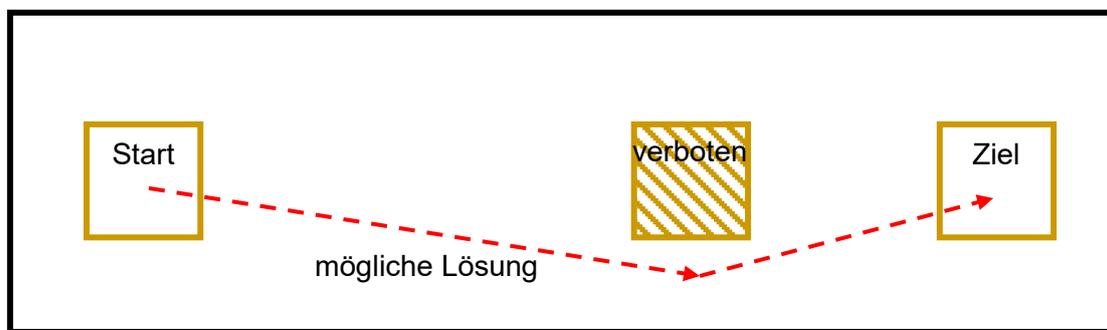


Abbildung 6: einfacher sequentieller Ablauf

Ziel ist es den Roboter mit möglichst wenigen und möglichst präzisen Befehlen in den Zielbereich zu führen. Die Aufgabe soll zuerst mit den intelligenteren Bewegungsbefehlen gelöst werden. Wenn dies gut geht, dann soll mit den elementaren Befehlen gearbeitet werden.

Lösung zu Aufgabe 1: sequentieller Programmablauf

Intelligente Blöcke (Farben entsprechen Scratch-Programmierungsumgebung)

Gehe 10 Schritte

Drehe nach links um 30 Grad

Gehe 7 Schritte

(EIN DETAIL: Natürlich verwendet der richtige Roboter nicht Schritte als Mass für die Distanz, sondern Motorumdrehungen.)

Merke I: Einem Roboter können nacheinander beliebig viele Befehle erteilt werden. Der Roboter erledigt diese Befehle in der gegebenen Reihenfolge. Dies ist ein **sequentieller Programmablauf**.



Merke II: Da die einfachen Legoroboter keine Orientierungsmöglichkeiten im Raum haben, ist das sehr präzise Festlegen der Startposition (und Orientierung) für jede Aufgabe sehr wichtig, dies im Speziellen bei Fahraufgaben.

Bei dieser ersten Aufgabe sollten unbedingt Sie als Lehrperson den Roboter spielen.

Dies bringt den entscheidenden Vorteil, dass Sie durch Ihre Reaktion (in Form einer Bewegung) auf die Befehle der Schülerinnen und Schüler auf die wichtigen Punkte hinweisen können.

Versuchen Sie dabei eine gute Mischung zwischen Spielform und ernsthafter Auseinandersetzung mit der Aufgabe bzw. mit der (Programm-)Sprache zu erreichen.

Ein paar mögliche Beispiele zur Erläuterung:

BEFEHL:	IHRE REAKTION:
- Roboter gehe	Sie gehen in eine beliebige Richtung los.
- Roboter gehe geradeaus	Sie gehen geradeaus bis in die Wand.
- Roboter gehe geradeaus Stopp!	Sie gehen geradeaus bis in die Wand, da Roboter nach dem Start des Programms keine Programmbefehle mehr entgegen nehmen.
- Roboter gehe geradeaus bis zum Ziel	Sie gehen geradeaus bis in die Wand, weil 'Ziel' nicht definiert ist
- Roboter drehe dich nach links	Sie drehen sich unendlich lange um ihre Körperachse.
- etc.	

Erkenntnis:

- Weil ein Roboter ohne Sensoren keine Kenntnisse von seiner Umgebung hat, müssen alle Aktionen vollumfänglich vorgängig im Programm festgelegt werden. Es ist dann kein Unterbruch mehr möglich. Das heisst, eine gestartete Bewegung läuft unendlich lang weiter, ausser das Ende der Bewegung wurde bereits im Voraus im Programm ebenfalls definiert (durch die Angabe einer **Bedingung**). Bei Bewegungen kann der Roboter z.B. Umdrehungen des Motors zählen (in unserem Spiel Schritte) oder auch Sekunden (also eine Zeitspanne), um anhand dieser inneren Information eine "Entscheidung" über das Ende der Bewegung zu treffen.
- Es können beliebig viele Befehle dieser Art aneinandergereiht werden. Der Roboter führt diese dann der Reihe nach aus. Man nennt dies einen **sequentiellen Programmablauf**.



Aufgabe 2 - Programmablauf mit explizitem 'Warten'

Motivation: Mit dieser Aufgabe sollen die SuS bewusst erfahren, dass die ‚Intelligenten Aktionen‘ nur mit Sensoren möglich sind. Dies erleichtert später das Programmieren und führt auch zu interessanteren eigenen Projekten, da die SuS nicht nur die Sensoren kennen, sondern auch bewusst erfahren, wie sie mit diesen einen Programmablauf steuern und so einen Roboter "intelligent" machen können. Mit „intelligent“ ist gemeint, dass der Roboter abhängig vom Erkennen seiner Umgebung reagieren kann.

Ablauf: Für diese Aufgabe hat der Roboter einen "Berührungssensor" erhalten (z.B. den nach vorne ausgestreckten Arm). Erläutern Sie der Klasse die Aufgabe (gemäß Abbildung, sowie 2.1a_PPP_AB-3_Robotersprache.pptx). Die Rollen 'Programmierer', 'Programmüberprüfer' und 'Roboterkontrolle' können abwechselnd wahrgenommen werden. So können alle drei Gruppen zuerst die Programmieraufgabe lösen und dann gegenseitig 'Programmüberprüfer' und 'Roboterkontrolleur' sein.

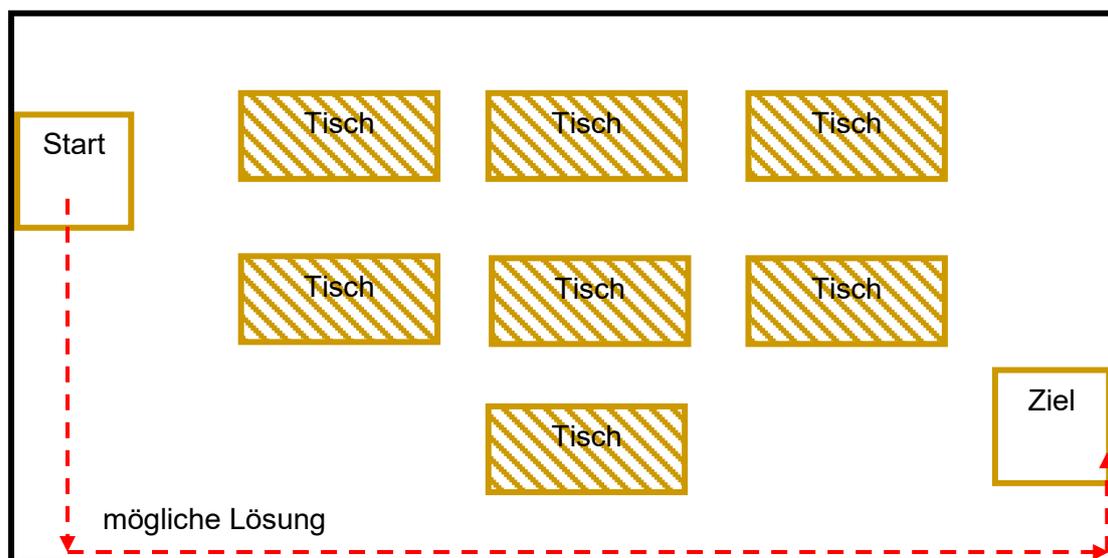


Abbildung 7: Programmablauf mit Warten



Lösung Aufgabe 2: Programmablauf mit Warten

Wenn der Roboter parallel zu Wand ausgerichtet worden ist:

mit elementaren Aktionen mit 'Warten'

Gehen

warten bis Berührung!

Stopp

Drehen nach links

Warten bis Drehwinkel > 90 Grad (Hier wird klar, dass das ‚Warten‘ sich auf den Programmablauf bezieht und nicht, dass man still steht)

Gehen (Man beachte, dass hier das 'Drehen nach links' nicht mit programmiertem Stopp angehalten wurde. Wenn der neue Befehl 'Gehen' die Funktion der Beine komplett neu festlegt, so müssen die vorangehenden Aktionen nicht gestoppt werden.)

warten bis Berührung!

Stopp

drehen nach links

Warten bis Drehwinkel > 90 Grad

Gehen (wäre nicht besser: Gehe 2 Schritte?)

Warten bis 2 Schritte

Stopp

Merke: Das **Warten** ist die einfachste Möglichkeit, den Programmablauf mit Hilfe eines Sensors zu beeinflussen. Erreicht das Programm ein **Warten**, so wird das vorher eingeschaltete ausgeführt bis es nach Erfüllen der **Bedingung** wieder ausgeschaltet werden kann.

Bemerkung: Es ist offensichtlich, dass die Robotersprache mit den elementaren Befehlen viel aufwändiger ist. Die Lösung mit "intelligenteren" Aktionen ist unten angegeben. Allerdings ist die Programmierung mit elementaren Befehlen viel flexibler. Wird der Befehl 'gehe 20 Schritte' gegeben, so besteht keine Möglichkeit, bei einem Hindernis anzuhalten. Das Programm wird stur durchgeführt. Sind situationsabhängige Entscheide nötig, die von mehreren Sensoren abhängig sind, so kann dies nur mit elementaren Befehlen gelöst werden.



Lösung mit intelligenten Aktionen ohne 'Warten'

Programmblock	In Scratch möglich
gehen bis Berührung!	Nein
Drehen nach links, 90 Grad!	Ja
gehen bis Berührung!	Nein
drehen nach links, 90 Grad!	Ja
Gehen 2 Schritte	Ja

Erkenntnis:

Soll ein Roboter an irgendeiner Stelle einer Programmsequenz den aktuellen Befehl (z. B. eine Bewegung) so lange ausführen, bis er eine Rückmeldung eines Sensors erhält (z. B. Kontakt mit der Wand), so müssen zwei Programmschritte eingefügt werden:

- Mit einem Aktionsbefehl muss etwas gestartet oder eingeschaltet werden
- Direkt im Anschluss folgt im Programm der Befehl **Warte bis...** (z. B. der Berührungssensor gedrückt wird).
- Mit einem Aktionsbefehl muss das Gestartete wieder gestoppt werden oder mit einem neuen Aktionsbefehl für die nächste Aufgabe gestartet werden.

Mit Hilfe eines Sensors und dem Wartebefehl, kann der Roboter also mit der Umgebung interagieren.

Aufgabe 3 - Programmablauf mit Schleife

Motivation: Bei der Lösung von Robotikaufgaben ist man oft mit Aufgaben konfrontiert, in denen eine Sequenz von Befehlen öfters wiederholt werden muss, bis eine gegebene Anzahl Wiederholungen abgearbeitet wurde oder eine Sensorbedingung erfüllt ist. Dies soll an einem einfachen Beispiel gezeigt werden.

Ablauf: Durchführung der Aufgabe gemäss Abbildung, sowie *Aufgaben_Robotersprache.pptx*. Hier muss der Roboter um einen Tisch herum zufahren befohlen werden. Damit das Programm nicht unnötig lang wird, löst man dies mit der Ablaufsteuerung ‚**Schleife**‘.



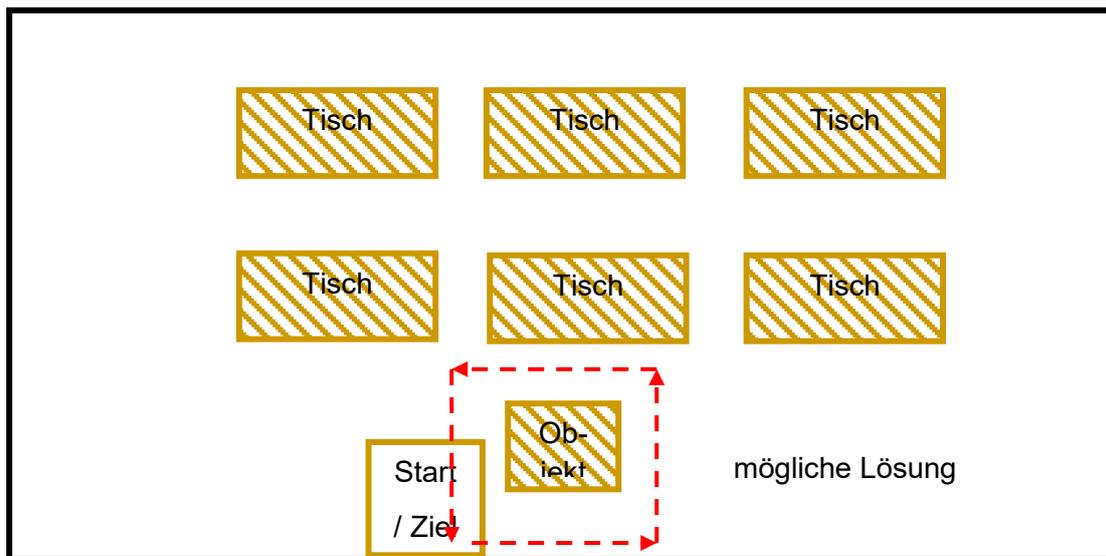


Abbildung 8: Programmablauf mit Schleife

Lösung Aufgabe 3: Programmablauf (Routine) mit Schleife

Start neben dem Tisch, Ausrichtung parallel zur Tischkante:

Wiederhole Schleife 4 mal

Gehe 3 Schritte

Drehe links

Warte bis Winkel >90 Grad

Ende Schleife

Merke: Befehle, die mehrfach wiederholt werden müssen, können in einer Schleife programmiert werden. Die Befehle werden so oft wiederholt, bis eine Bedingung erreicht ist. Die Bedingung definiert, ob die Befehlssequenz erneut durchgeführt wird oder ob der sequentielle Programmablauf fortgesetzt wird.

Erkenntnis

Ein Programm kann erheblich verkürzt werden, wenn man immer wiederkehrende Befehlsfolgen zusammenfasst und diese in einer **Schleife** so oft wie erwünscht wiederholt. In diesem Beispiel wird die Folge nach der ersten Ausführung noch 3 mal wiederholt.

Bei einer Schleife muss auch immer eine Bedingung angegeben werden, also festgelegt werden, wie oft oder wie lange eine Schleife wiederholt werden soll.



Lösung Aufgabe 4: Programmablauf mit Schalter

Gehe 3 Schritte

Prüfe ob Helligkeit > 50% (grau)

Falls wahr:

Drehen links 90 Grad

Falls falsch:

Drehen rechts 90 Grad

Ende Verzweigung

Gehe 4 Schritte

Merke: Der sogenannte **Schalter** ist die einfachste Möglichkeit für einen Roboter, eine Entscheidung zu treffen. Nach dem Ausführen des gewählten Programmablaufs wird zum gemeinsamen Programmablauf zurückgekehrt.

Zusammenfassung: Der Grundwortschatz für das Programmieren eines Roboters

Für das Programmieren eines Roboters brauchst du nur die folgende 4 Befehle und 2 Begriffe zu kennen:

Befehle:

- Aktion (Aufnahme/Abspielen, Klang und Anzeige funktionieren genau gleich)
- Warten
- Schleife
- Verzweigung/Weiche

Begriffe:

- Sequentieller Programmablauf, Stapel
- Bedingung

Mit diesen Befehlen kannst du schon eine ganze Menge programmieren. Der Kreativität sind da fast keine Grenzen gesetzt. Beachte, dass in den Lösungen zu den Aufgaben die gleichen Arten von Befehlen immer mit der gleichen Farbe gekennzeichnet sind. Bei der Programmierung des echten Roboters auf dem Computer werden die gleichen Farben zur Kennzeichnung verwendet.

Die wichtigsten Zusammenhänge:

- Die Befehle des Typs **Warten** und **Schalter** sind fast immer an die **Bedingung eines Sensors** gekoppelt.
- Mit Hilfe des Befehls vom Typ **Schalter** können "**entweder-oder-Entscheidungen**" getroffen werden.
- Mit Hilfe von **Schleifen** können Befehle zusammengefasst und wiederholt werden, was eine **effiziente Programmierung** erlaubt.



Block 2: Lektionen 2-4

Einleitung:

Die Einführung in die Programmierung erfolgt anhand von einfachen, aufbauenden Übungen. Der Aufbau ist der Folgende

- Einführung in das Mindstorms EV3 Programm (*Arbeitsblatt 3*):
 - a. Programme erstellen
 - b. Verbindung zum EV3 aufbauen
 - c. Bedienung der Oberfläche
- Programmierübungen:
 1. Fahren um eine bestimmte Strecke, anschliessend 90 Grad Drehung nach rechts (*Arbeitsblatt 4*)
 2. Um ein Objekt fahren durch viermaliges Wiederholen der Aufgabe 1 (*Arbeitsblatt 5*)
 3. Einsatz einer Schleife um das gleiche Problem zu lösen (*Arbeitsblatt 5*)
 4. Selbständiges Lösen einer Fahraufgabe (*Arbeitsblatt 6*)
 5. Sensor abhängiger Ton (*Arbeitsblatt 7*)
 6. Ton und Anzeige mit Hilfe von Variablen (*Arbeitsblatt 8*)
 7. Ereignisgesteuertes Programm (*Arbeitsblatt 9*)
 8. Fahren entlang einer Wand (*Arbeitsblatt 10*)

Der Aufbau ist in Lektionen gegliedert, muss aber dem Fortschritt der SuS angepasst werden.

Vorbereitung:

PC, Software und EV3 gemäss Kapitel Vorbereitung bereitstellen

Akku der EV3 aufgeladen

Vorlage ‚erstes Programm‘ auf Programmiergerät abspeichern. Diese Vorlage enthält eine minimale Anzahl Programmierblöcke. Dadurch sind die SuS nicht durch ungebrauchte Blöcke abgelenkt.

Benötigtes Material:

- Pro Gruppe 1x EV3-Basis Roboter
- Pro Gruppe 1 Verbindungskabel
- Pro Gruppe: 1x *Arbeitsblatt_3_Mein erstes Programm*
- Pro Gruppe 1x Arbeitsblatt zu jeder Übung (also die *Arbeitsblätter 3 bis 8, evtl. Zusatzaufgaben 9 und 10*)

Die Programmierumgebung kennenlernen (Arbeitsblatt 3)

Lernziele:

Die SuS lernen...

- ein einfaches Programm im Lego-Mindstorms-Programm zu erstellen,
- mit dem EV3 über USB oder Bluetooth zu kommunizieren und das Programm laufen zu lassen.



Ablauf: Die Gruppen starten das Programm EV3 Classroom auf dem PC/Notebook und erstellen dort ein erstes Projekt nach Anleitung in Arbeitsblatt 3. Falls dies aus Installationsgründen nicht gelingen sollte, kann auch das Programm ‚erstes_Programm_reduzierte_Palette.lmsp‘ aus der Programmsammlung der Dokumentation geöffnet werden. Der gebaute Roboter wird mit dem PC/Notebook verbunden und die Verbindung hergestellt. Eine einfache Fahraufgabe wird programmiert. **Tipp:** Es kann sinnvoll sein, dass sie ihren PC mit dem Beamer verbinden, um die einzelnen Schritte des *Arbeitsblattes 3* bei der ersten Durchführung auch vorzeigen zu können.

Lösungen zu den Aufgaben im Arbeitsblatt 'Erstellen eines neuen Programms':

Das Programm ist bereits in der Abbildung 3 vollständig gezeichnet und muss nur noch richtig konfiguriert werden. Die in der Hilfe auffindbaren Texte sind:



Abbildung 10: neues Programm

Kommentare zum Programm

1: Programmstart, muss immer vorhanden sein

2: Damit der Roboter beim Drücken des Programmstartknopfs (dunkelgrau) nicht gleich losfährt, kann man 1 s oder mehr warten. Dies ist nicht zwingend nötig.

3: Eingabe der Anzahl Umdrehungen

4: Angabe der Masseinheit, es wäre auch möglich, den Drehwinkel in Grad anzugeben.

5: Auswahl der Bewegungsrichtung

Lösungen zu den Zusatzaufgaben:

1. Der Durchmesser eines Rades ist ca. 5.6 cm, der Umfang also $5.6 * \pi = 17.6$ cm. Somit ergeben sich für 40 cm etwa 2.3 Radumdrehungen. **ACHTUNG: Das Dezimaltrennzeichen ist . (Punkt) und nicht , (Komma)!**
2. Ohne Kommastellen ergibt sich in Grad: $2.3 * 360^\circ = 828^\circ$
3. Kurven ergeben sich aus unterschiedlichen Pneuradien. Da die Pneus elastisch sind, kann schon eine ungleiche Belastung der Räder zu einer leichten Kurve führen.



Programmierübung 2: Fahren mit Kurve (Arbeitsblatt 4)

Ziel: Die SuS lernen, dass Programm und Roboter zusammenhängen. Je nach Verkabelung des Motors ergibt eine Linkskurve im Programm eine Rechtskurve beim Roboter.

Diese Aufgabe ist relativ kurz, soll aber möglichst präzise gelöst werden, denn die Lösung wird für die nächste Aufgabe weiterverwendet. Das Prinzip der Steuerung muss erforscht werden. Wie bei allen Fahrzeugen gibt es unterschiedliche Möglichkeiten, eine 90-Grad Kurve zu fahren. In der folgenden Abbildung sind zwei mögliche Lösungen gezeichnet:



Bei der Kurve links ist der Kurvenradius klein, d.h. der Lenkwinkel gross. Rechts ist der Kurvenradius viel grösser und der Lenkwinkel klein. Für eine 90-Grad-Kurve mit grossem Lenkwinkel muss weniger weit gefahren werden als bei kleinem Lenkwinkel. Die SuS finden dies durch Probieren sehr schnell heraus.

Lösung:



Abbildung 11: Kurve Fahren

- 1 In diesem Programm wird konfiguriert, dass die Antriebsmotoren an den Anschlüssen B und C angeschlossen sind
- 2 Einfügen von Klängen hilft den SuS zu verstehen, ‚wo‘ der Programmablauf gerade ist. Dies ist für die Fehlersuche sehr nützlich. Achtung, Töne brauchen Zeit.
- 3 Diese Vorgaben stimmen nicht exakt, denn sie hängen vom Roboter ab
- 4 Das Programm wird nun am Ende ordentlich gestoppt.



Programmierübung 3: Fahren um Objekt (Arbeitsblatt 5)

Das in der vorangehenden Aufgabe erstellte Programm soll so ausgebaut werden, dass der Roboter um ein Objekt (z.B. Papierkorb) herumfahren kann. Das bestehende Programm wird 3 mal kopiert. Es entsteht ein längeres Programm. Die SuS müssen eventuell die Zoom-Funktionen nutzen, damit das ganze Programm auf dem Bildschirm angezeigt wird. Das ineffiziente, durch mehrmaliges Kopieren entstandene Programm wird in einem zweiten Schritt durch eine Schleife ersetzt.

Ziel:

Die Programmfertigkeiten werden verbessert und die erste Programmsteuerung wird erlernt.



Programm ohne Schleife: es wird 4 mal das Gleiche programmiert.

Abbildung 12: Programm ohne Schleife



- 1 Schleife: ‚wiederhole () mal‘. Die zu wiederholenden Blöcke müssen innerhalb des Rahmens der Steuerstruktur platziert werden.

Abbildung 13: Programm ohne Schleife



Programmierübung 4: Fahren einer Figur (Arbeitsblatt 6)

Die Figur 8 kann durch Aneinanderreihen von 2 Vollkreisen programmiert werden. Diese Aufgabe ist sehr einfach, fordert aber die SuS, etwas ohne Vorlage zu lösen.



Abbildung 14: Lösung Fahren einer Figur

Erläuterung zur Bedeutung der Fahrriechung.

Das äussere Rad in einer Kurve ist das Masterrad. Geschwindigkeit und auch Anzahl Umdrehungen oder Drehwinkel werden bezüglich dem Masterrad angegeben.

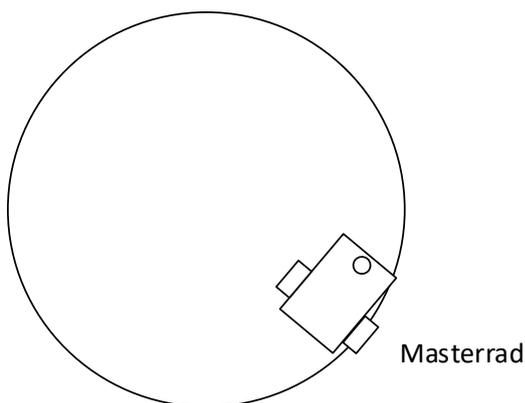


Abbildung 15: Masterrad Fahrriechung

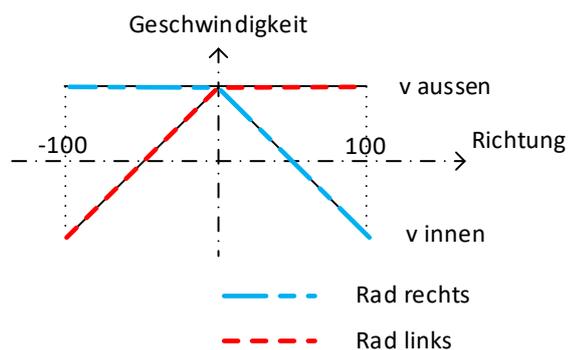


Abbildung 16: Fahrriechung

Die Vorgabe der Fahrriechung bestimmt, in welchem Verhältnis das innere Rad zum äusseren Rad dreht. Die Verhältnisse sind in Abbildung 16 dargestellt. Welches Rad nun das Linke ist, hängt vom Anschliessen der Motoren an den EV3 ab.

Ist die Fahrriechung gleich Null, so drehen beide Räder gleich schnell mit der vorgegebenen Geschwindigkeit. Ist die Fahrriechung gleich 50, so steht das innere Rad. Ist die Fahrriechung gleich 100 so dreht es gerade in entgegengesetzter Drehrichtung mit gleicher Geschwindigkeit. Das Vorzeichen gibt an, ob das linke oder das rechte Rad zum Masterrad wird, natürlich abhängig vom Anschluss der Motoren an die Ports.



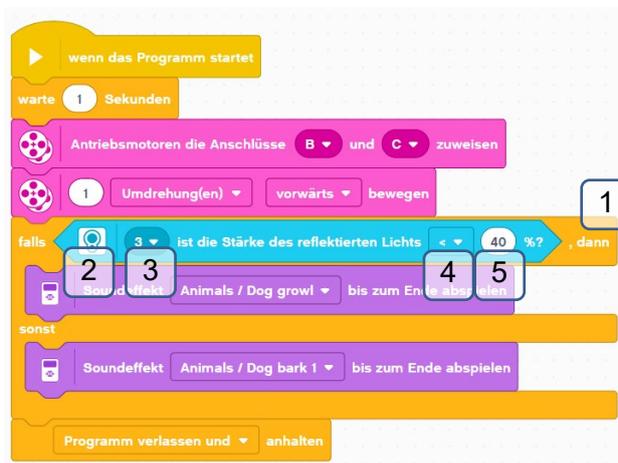
Programmierübung 5: Sensorabhängiger Ton (*Arbeitsblatt 7*)

Programmverzweigungen sind in Roboterprogrammen sehr nützlich. Nur so kann der Roboter situationsbedingt reagieren. Die Bedingung ‚falls < >, dann‘ braucht die Berechnung einer logischen Bedingung. In dieser Übung soll diese aufgrund des Helligkeitswerts des Lichtsensors bestimmt werden.

Ziele:

- Die SuS können ein einfaches Programm mit einer Verzweigung realisieren.
- Die SuS können die Bedingung für eine Verzweigung richtig konfigurieren.

Lösung



- 1 ‚falls < >,dann‘-Programmblock für die Verzweigung mit 2 Alternativen
- 2 Auswahl des Sensortyps: Icon beachten
- 3 Konfiguration des Sensoreingangs (Ort, wo das Kabel eingesteckt ist)
- 4 Wahl des Vergleichsoperators
- 5 Eingabe des Grenzwerts

Abbildung 17: Programm mit Verzweigung

Programmierübung 6: Sensorabhängiger Ton (*Arbeitsblatt 8*)

In Scratch gibt es keine Möglichkeiten, graphisch Informationen von einem Programmteil zu einem anderen Programmteil z. B. durch Signallinien zu übermitteln. Die einzigen Möglichkeiten sind die Verwendung von Variablen oder das Auslösen von Ereignissen. Variablen werden in dieser Übung eingeführt, die etwas weniger wichtigen Ereignisse in der ersten Zusatzaufgabe.

Variablen speichern in Scratch Zahlen. Ein natürlicher Zugang zu Zahlen ist das Zählen. Darum soll zuerst der Roboter so programmiert werden, dass er zählen kann. In der gleichen Aufgabe wird auch die Darstellung von Information auf dem Display eingeführt. Damit die Zahlen nicht nur abstrakte Größen sind, wird gelernt, wie mittels Zahlen Tonhöhen kodiert werden. Im EV3 Classroom sind die Tonhöhen folgendermassen mit Zahlen beschrieben:

C (60)	D (62)	E (64)	F (65)	G (67)	A (69)	H (71)	C (72)

Abbildung 18: Kodierung der Töne in EV3 Classroom



Ziele:

- Die SuS kennen das Konzept einer Variable als einen einfachen Koffer, in dem digitale Daten abgespeichert werden können.
- Die SuS können Variablenwert richtig initialisieren, Zahlen darin abspeichern und einfache Rechenoperationen durchführen.
- Die SuS verstehen mögliche Relationen zwischen Zahlenwerten und Tätigkeiten und können diese zur Erzeugung einer Tonleiter anwenden.

Die Lösung ist im Arbeitsblatt bereits detailliert vorgegeben. Variablen werden bildlich als Datenkoffer eingeführt und die Variablenoperationen schematisch dargestellt. Die SuS können die ersten Schritte nachprogrammieren. Programmierereigenleistung ist gefordert, wenn sie die Aufgabe so anpassen müssen, dass der letzte Ton länger ausgehalten wird.

Lösung:



- 1 Schlaufenwiederholung um 1 reduzieren
- 2 Letzter Ton einzeln programmieren

Programmierübung 7: einfache Zusatzaufgabe (Arbeitsblatt 9)

Ereignisgesteuerte Programmierung ist in vielen Bereichen sehr wesentlich. Die Übung ist eine Konfigurationsaufgabe, d.h. ein Programm wird vorgegeben. Dieses soll in seiner Funktion analysiert werden. Dabei soll ein spezielles Augenmerk auf den zeitlichen Ablauf gelegt werden. Steuern heisst zur richtigen Zeit auf eine Situation reagieren. Die zeitlichen Aspekte sind darum sehr wichtig. Den SuS kann auch das Zeitdiagramm instruiert werden.

Ziele:

- Die SuS lernen die Möglichkeiten der Ereignis-gesteuerten Programmierung kennen
- Die SuS lernen ein Programm hinsichtlich seines zeitlichen Verhaltens zu analysieren.



Lösung: Der zeitliche Ablauf kann im folgenden Zeitdiagramm anschaulich dargestellt werden. Die oberste Zeile im Diagramm zeigt die Veränderung der Anzeige durch den Stapel ‚Wenn das Programm startet‘. Alle 6 Sekunden wird das traurige Gesicht auf das Display geschrieben. Die zweite Zeile zeigt den Stapel für das Ereignis, dass der Taster gedrückt wurde. Immer wenn der Taster gedrückt wird, erscheint das Smiley. Wird der Taster gedrückt kurz nachdem das traurige Gesicht erschienen ist, so wird das Smiley lange angezeigt, im Beispiel unten 5.5s. Wartet man aber 5s (zählen auf 5), so erscheint das Smiley nur kurz.

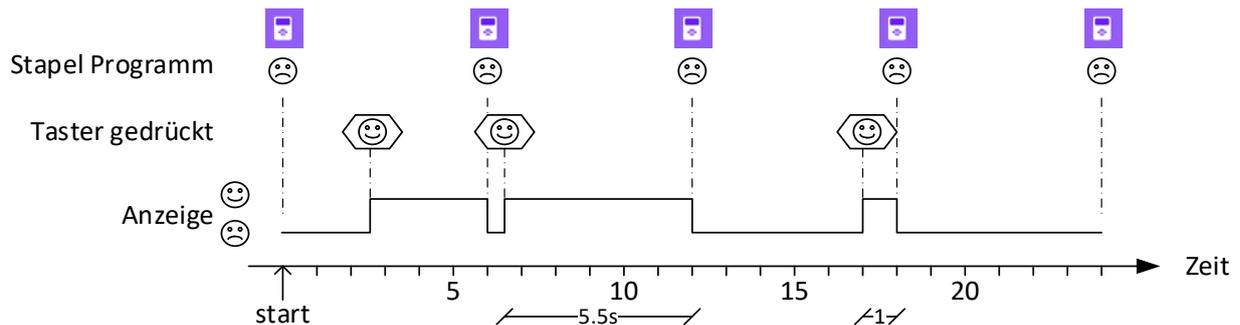


Abbildung 19: Zeitdiagramm für Ereignisse

Programmierung 8: Zusatzaufgabe (Arbeitsblatt 10)

Die Zusatzaufgabe ist eine klassische Regelaufgabe. Der Roboter wird mit dem Ultraschallsensor entlang einer Wand geführt.

Ziele:

- Die SuS können mit Sensorsignalen rechnen.
- Die SuS können die Funktion einer Regelung erklären.

Die Aufgabe ist eine Programmier- und Probieraufgabe, bei der man mit Tüfteln zu einer faszinierenden Lösung kommen kann. Sie erfüllt aus didaktischer Sicht auch den Zweck, **die "schnellen" mit einer interessanten Knobelaufgabe zu beschäftigen, während die etwas "langsameren" SuS noch an den Aufgaben 1-6 arbeiten.**



Block 3: Realisierung einer weiterführenden Aufgabe

In diesen vier bis 8 Lektionen soll jede Gruppe mindestens eine, besser mehrere weiterführende Aufgaben realisieren. Zu diesem Zweck besteht eine ganze Liste möglicher Aufgaben (*Dokumentation für die Lehrperson_Aufgabenideen-Block-3.docx*) Diese Liste liefert Ihnen Ideen, mögliche Lösungen, sowie Angaben zum Schwierigkeitsgrad, damit Sie die SuS begleiten können, auch ohne Experte/Expertin zu sein. Es ist sicher sinnvoll, wenn die SuS zuerst mit einer der einfacheren Aufgaben beginnen. Diese können dann auch kreativ und individuell erweitert werden.

- Grundsätzlich lohnt es sich für die Gruppen vom bestehenden Basis-Roboter auszugehen und diesen ggf. zu erweitern. Natürlich soll ein kompletter Neubau nicht verboten sein, dies braucht aber relativ viel Zeit. Und die Praktikabilität des Neubaus ist keineswegs sicher.
- Sorgen Sie dafür, dass die SuS die Motoren zu Beginn langsam laufen lassen (z.B. mit 25% Leistung). So bleibt genug Zeit, um zu beobachten und bei Irrläufen einzugreifen. Wenn ein Programm dann mal funktioniert, kann man die Motoren immer noch schneller einstellen.
- Scheuen Sie sich nicht vor diesem 3. Block, auch wenn Sie bisher keine Erfahrung mit Lego-EV3 haben. Die SuS finden sich mit der Programmierumgebung erfahrungsgemäss auf Anhieb zurecht und probieren gerne aus. Natürlich können auch Fragen auftreten, die Sie nicht beantworten können. Das müssen Sie aber auch nicht. Sie fungieren als Berater/Beraterin. Beobachten Sie zusammen mit den SuS ganz genau was bei Problemen passiert. Die Lösung des Problems findet man durch
 - Ausprobieren
 - Genau beobachten
 - Evaluieren
 - Anpassen/Optimieren
 - und erneutem Ausprobieren...
- Wenn Sie den SuS dabei helfen systematisch und kleinschrittig vorzugehen, finden sie meist selbst eine Lösung.

Alle weiteren Informationen inklusive Hinweisen zur Differenzierung und einem Vorschlag für eine Bewertungsvariante finden Sie in der *Dokumentation für die Lehrperson* im Kapitel zum Block 3.



Block 4: Roboter und Gesellschaft

Hier finden Sie ein paar generelle Hintergrundinformationen zum Auftrag an die SuS (*Arbeitsblatt_12_Robotik und Gesellschaft*). Im Auftrag wird offen gelassen, wie viel Zeit die SuS für den Auftrag haben und welches Produkt sie erstellen sollen. Das sollten Sie für Ihre Klasse noch festlegen.

Gibt es Arbeiten, welche Roboter nie erledigen werden können?

- Künstliche Intelligenz kann noch lange nicht mit menschlichem Denken gleichgesetzt werden. Gefühle, Intuition, Fantasie, abstraktes Denken und Erfahrung sind Eigenschaften die nicht einfach programmiert werden können. Ein Roboter verfügt somit immer noch über ein beschränktes Repertoire an Möglichkeiten, auch wenn er gewisse Arbeiten 1000 mal schneller und genauer erledigen kann als ein Mensch.

Welche Gefahren bergen Roboter?

- Trotz aller Sicherheitsvorkehrungen kann es vorkommen, dass Menschen durch Roboter verletzt werden. Je autonomer ein Roboter im Umfeld des Menschen agiert, desto grösser ist die Wahrscheinlichkeit, dass Lebewesen oder Gegenstände zu Schaden kommen werden.

Beispiele:

- Industrieroboter müssen mit Gittern und Lichtschranken vor jeglichen Eingriffen bewahrt werden. Der Roboter merkt sonst nicht, wenn sich jemand in der Nähe befindet und kann Personen verletzen.
- Auch ein Staubsaugroboter birgt Gefahren. Eine Person, die zufällig auf den Roboter tritt darf nicht ausrutschen, bzw. der Roboter darf nicht wegrutschen. Sobald mehr Gewicht auf dem Roboter lastet, blockieren die Räder und das Chassis wird an den Boden gedrückt und bremst so den Roboter.

Als ethische Richtlinie dienen die drei Robotergesetze (Isaac Asimov, 1942):

- Ein Roboter darf kein menschliches Wesen (wissentlich) verletzen oder durch Untätigkeit gestatten, dass einem menschlichen Wesen Schaden zugefügt wird.
- Ein Roboter muss den ihm von einem Menschen gegebenen Befehlen gehorchen – es sei denn, ein solcher Befehl würde mit Regel eins kollidieren.
- Ein Roboter muss seine Existenz beschützen, solange dieser Schutz nicht mit Regel eins oder zwei kollidiert.

Hinweise zu Informationsquellen

Das Internet ist voll mit Informationen unterschiedlichster Güte zu Robotern. Man kann auf unterschiedlichsten Wegen seine Informationen sammeln. Ohne passende Strategie ist es aber gut möglich, dass man entweder nur zweifelhafte Information findet, oder dann solche, die nur für Experten von Interesse sind.



Auf YouTube lassen sich aber gute Informationen z.T. aus Dokumentationssendungen finden, die einen guten dokumentarischen Wert und einen angemessenen Informations- bzw. Schwierigkeitsgrad haben. Zum Beispiel:

<https://www.youtube.com/watch?v=BGrIOVmicL0> (letzter Zugriff, 21.12.2020)

<https://www.youtube.com/watch?v=SG7DG3NIHKg> (die ersten 50 Sekunden, letzter Zugriff, 21.12.2020)

<https://www.youtube.com/watch?v=TFLJjxoUgkQ> (letzter Zugriff, 21.12.2020)

'Erfolgsgeschichte der Industrieroboter setzt sich fort', International Federation of Robotics., <http://www.ifr.org/news/ifr-press-release/the-continuing-success-story-of-industrial-robots-414/>, abgerufen 21.12.2020, Downloadlink für deutsche Fassung.

Ichbiah D., Roboter, Geschichte_Technik_Entwicklung, Kensebeck GmbH & Co, 2005
und vieles mehr...Suchen Sie (oder ihre SuS) selbst, was Sie am interessantesten finden



Abbildungsverzeichnis

Quelle: J. Keller, Screenshots aus Lego Mindstorms

EV3-Block	9
EV3-Bedienung	10
Weg zur Bauanleitung Basisroboter (Quelle ScreenShot aus Education EV3 Programm)	10
Prinzip Ultraschall.....	14
Vergleich 'intelligenter' Programmierblock mit elementarer Programmierung	17
einfacher sequentieller Ablauf	22
Programmablauf mit Warten.....	24
Programmablauf mit Schleife.....	27
Programmablauf mit Verzweigung.....	28
neues Programm.....	31
Kurve Fahren	32
Programm ohne Schleife	33
Programm ohne Schleife	33
Lösung Fahren einer Figur	34
Masterrad Fahrrichtung	34
Fahrrichtung	34
Programm mit Verzweigung	35
Kodierung der Töne in EV3 Classroom	35
Zeitdiagramm für Ereignisse.....	37

