

Your EdVenture into Robotics

10-Lektionen-Plan *Begleitkommentar*



Die Unterrichtspläne wurden in Zusammenarbeit mit RoboticsWPS erstellt.

ROBOTICS
Robotic Workshops
coding for schools

Übersetzt wurde sie durch das Institut für Weiterbildung und Medienbildung, PHBern

Inhalt

Einleitung.....	3
EINFÜHRUNG	4
LEKTION 1: KENNEN LERNEN UND STARTEN	5
Edison kennen lernen (Arbeitsblatt 1.1)	6
Strichcode-Programmierung (Arbeitsblatt 1.2)	7
EdWare kennen lernen (Arbeitsblatt 1.3)	8
Test-Programm (Arbeitsblatt 1.4)	9
LEKTION 2: ROBOTERBEWEGUNG - FAHREN	11
Programm 1 Vorwärts fahren (Arbeitsblatt 2.1)	11
Programm 2 Rückwärts fahren (Arbeitsblatt 2.2).....	11
Programm 3 Vorwärts- & Rückwärtsfahren (Arbeitsblatt 2.3)	12
Programm 4 Geschwindigkeit anpassen (Arbeitsblatt 2.4)	12
LEKTION 3: ROBOTERBEWEGUNG - DREHEN.....	13
Programm 1 Rechtsdrehung (Arbeitsblatt 3.1)	13
Programm 2 Linksdrehung (Arbeitsblatt 3.2).....	13
Programm 3 Rechts- & Linksdrehung (Arbeitsblatt 3.3)	14
Programm 4 Mini-Labyrinth (Arbeitsblatt 3.4).....	14
LEKTION 4: FAHR-CHALLENGE UND LA-OLA-WELLE.....	15
Programm 1 Fahr-Challenge.....	15
Programm 2 La-Ola-Welle.....	15
LEKTION 5: PROJEKT 1 – MEIN ERSTES PROGRAMM.....	16
LEKTION 6: KLATSCH-SENSOR	18
Programm 1 LED reagiert auf Händeklatschen (Arbeitsblatt 6.1).....	18
Programm 2 Mit Händeklatschen fahren (Arbeitsblatt 6.2).....	18
Programm 3 Tanzen nach Händeklatschen (Arbeitsblatt 6.3)	18
LEKTION 7: HINDERNISSEKKNUNG	19
Wie funktioniert die Infrarot-Hindernis-Erkennung? (Arbeitsblatt 7.1).....	19
Programm 1 Ein Hindernis erkennen und stoppen (Arbeitsblatt 7.2).....	19
Programm 2 Ein Hindernis erkennen und ausweichen I (Arbeitsblatt 7.3) ...	19
Programm 3 Ein Hindernis erkennen und ausweichen II (Arbeitsblatt 7.4) ..	20
Programm 4 Ein Hindernis erkennen und umfahren (Arbeitsblatt 7.5)	20
LEKTION 8: EINE LINIE ERKENNEN UND FOLGEN	21
Wie funktioniert der Linien-Sensor? (Arbeitsblatt 8.1)	21
Programm 1 Bis zu der schwarzen Linie fahren (Arbeitsblatt 8.2).....	21
Programm 2 Innerhalb einer Grenze fahren (Arbeitsblatt 8.3).....	21
Programm 3 Einer Linie folgen (Arbeitsblatt 8.4).....	21
Video – Humans need not apply	22
LESSON 9: AUF LICHT REAGIEREN	23
Wie funktioniert der Licht-Sensor? (Arbeitsblatt 9.1)	23
Programm 1 Helligkeitsalarm (Arbeitsblatt 9.2)	23
Programm 2 Automatische Beleuchtung (Arbeitsblatt 9.3).....	23
Programm 3 Einem Licht folgen (Arbeitsblatt 9.4).....	23
Animal behaviours in robotics.....	24
LEKTION 10: PROJEKT II – MEIN ZWEITES PROGRAMM	25
Edison Pass	27

Einleitung

Dieser 10-Lektionen-Plan wurde in Workshops von Robotics WPS in den letzten fünf Jahren mit einer Vielzahl von Robotern entwickelt und getestet. Nun wurden die Lektionen mit dem Edison-Roboter weiterentwickelt und haben sich als sehr erfolgreich erwiesen.

Als Lehrperson können sie diesen Lektionen-Plan so übernehmen wie er ist oder Sie nutzen ihn als Ausgangspunkt, um Ihren eigenen angepassten Plan zu entwickeln. Der Lehrpersonen wird empfohlen, Anpassungen bei der Wahl der Arbeitsblätter vorzunehmen um einzelne ausgewählte Kompetenzen zu unterrichten.


Der Unterricht beginnt mit einfachen Programmier-Icons und endet mit den aus Erfahrung gewonnen Favoriten der Klasse, wie z. B. Linienverfolgung und Sumo-Wrestling. Lernende aller Altersgruppen haben Spass an dieser Form der Programmierung, Problemlösung und Zusammenarbeit, welche mit dem Lektionen-Plan erreicht werden kann.

Sie werden feststellen, dass einige Lernende die ersten Konzepte leicht erfassen und für die nächste Aktivität bereit sind. Der Lektionen-Plan ermöglicht somit ein unabhängiges und selbstständiges Lernen und bietet einen Übergang zur nächsten Lektion. Sie entwickeln Ihr eigenes Bewusstsein, wie Sie am besten Ihre Lernenden von Lektion zu Lektion begleiten.

VO.2 Your EdVenture into Robotics / O Lesson Plans

Lesson 2 Worksheet 1 - Drive the robot forward

Write the following program to drive the Edison robot forward. Use the activity sheet or marked lines on a desk or as start and stop markers.



Set the 'Seconds pass' (replace?) to an amount of time in seconds to have the robot start from behind the start line and stop before the finish line.
Minimum amount of time is 0.01 seconds.
Maximum amount of time is 327.67 seconds.
Try different amounts of time until your robot drives forward and stops just before the finish line.

What is the correct amount of time to get your robot from the start to the finish?
Describe what the robot is doing and why?

www.meetedison.com www.roboticswps.com.au page 20

Arbeitsblätter

Für jede Lektion ist eine Reihe von Arbeitsblättern bereitgestellt. Die Arbeitsblätter erlauben den Lernenden, selbstständig zu arbeiten, da sie Anweisungen für die Lernenden beinhalten, welchen ohne grosse Unterstützung selber gelöst werden können.

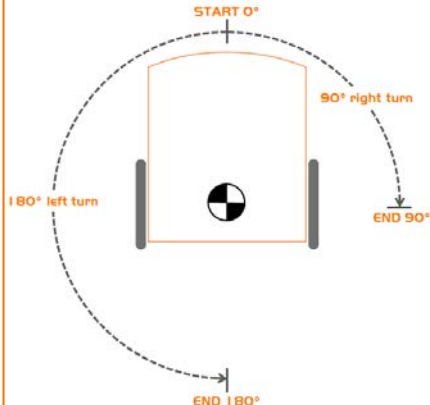
Die Arbeitsblätter enthalten zusätzliche Fragen für die Lernenden, welche dazu dienen, das jeweilige Thema zu vertiefen.

VO.5 Your EdVenture into Robotics / O Lesson Plans

Lesson 3 Activity sheet 3.1 - Turning

Place the robot down as per the outline then play your turning program. Your robot should:

- Program 1 - Turn right from start 0° to end 90°
- Program 2 - Turn left from start 0° to end 180°
- Program 3 - Turn right from start 0° to end 90°, then turn left 270° to end 180°



www.meetedison.com www.roboticswps.com.au page 31

Activity-Aufgaben

Einige der Lektionen beinhalten zudem Activity-Aufgaben. Sie bieten den Lernenden einen Arbeitsbereich, um ihre eigenen Programme schnell zu testen und Experimente durchzuführen.

EINFÜHRUNG

Edison ist ein kleiner Bodenroboter mit welchem Lernende Elektronik, Programmierung und Robotik auf eine lustige und spielerische Art erforschen und lernen können.

Edison ist mit den notwendigen Sensoren, Lampen und Motoren ausgerüstet, um die Lernenden in die faszinierende Welt der Robotik einzuführen.

Das tönt ja gut, aber was ist Robotik? Nun, diese Frage hat keine einfache Antwort. Der Erfinder von Edison, Brenton O'Brien sagt *“ein Roboter ist eine Maschine, die selbständig etwas tun kann”*. Das heisst, ein Roboter kann überlegen, selber Entscheidungen fällen und diese dann ausführen. Andere Leute würden das anders erklären, aber wir finden diese Beschreibung gut, weil einfach ist und zu dem passt, was die Lernenden hier gleich lernen werden.



Edison, der LEGO-kompatible Roboter

Robotik wäre nicht möglich ohne Elektronik. Beim Edison kann man sie durch die transparente Oberseite sehen. Dort befinden sich Widerstände, Kondensatoren, Transistoren, Motoren und andere Bauteile. Die wichtigste Komponente ist aber der sogenannte "Mikrocontroller".



Der Mikrocontroller des Edison

Der Mikrocontroller ist so etwas wie das Gehirn des Edisons. Dort finden seine *“Denkprozesse”* statt. Der Mikrocontroller des Edisons gleicht dem Prozessor in einem Computer, nur ist er viel kleiner. Wie beim Prozessor in einem Computer besitzt der Mikrocontroller vom Edison auch Programme. Die Programme lassen den Edison Entscheide fällen und

selber *“denken”*.

Edison enthält einige fest installierte Programme, die aufgerufen werden können, indem man ihn über einen speziellen Strichcode fahren lässt. Hier ist ein Beispiel:



Dieser Strichcode aktiviert das *“An der Grenze abprallen”*-Programm des Edison

Das faszinierendste beim Edison ist, dass man ihm eigene Programme beibringen kann. Die Lernenden können Edison vorgeben, wie er denken, sich verhalten und auf seine Umgebung reagieren soll. Programmieren ist so überraschend einfach.

LEKTION 1 : KENNEN LERNEN UND STARTEN

Technologischer Aspekt: Die Lernenden machen sich mit der Programmierumgebung vertraut und lernen, wie ein Programm auf den Edison heruntergeladen, werden kann.

Bevor man den Edison programmieren kann, sollten folgende Vorbereitungsschritte vollzogen werden:

1. Edison startklar machen
2. Edison kennen lernen (Arbeitsblatt 1.1)
3. Edison mit den Strichcodes programmieren (Arbeitsblatt 1.2)
4. Software "EdWare" installieren und sich mit den Icons vertraut machen (Arbeitsblatt 1.3)
5. Funktionstest mit einem Programm (Arbeitsblatt 1.4)

Edison startklar machen



Die Batterien richtig einlegen

Das Batteriefach öffnen und das Programmierkabel heraus nehmen. 4 Stück 1.5 V Batterien der Grösse AAA einlegen. Auf der Deckelaussenseite steht angeschrieben, in welche Richtung die Batterien eingelegt werden müssen. Dann den Deckel aufs Batteriefach klicken um zu schliessen.

Um den Edison einzuschalten, muss der Hauptschalter auf der Unterseite zum Strich hin geschoben werden. Die roten LED-Lampen beginnen dann zu blinken.

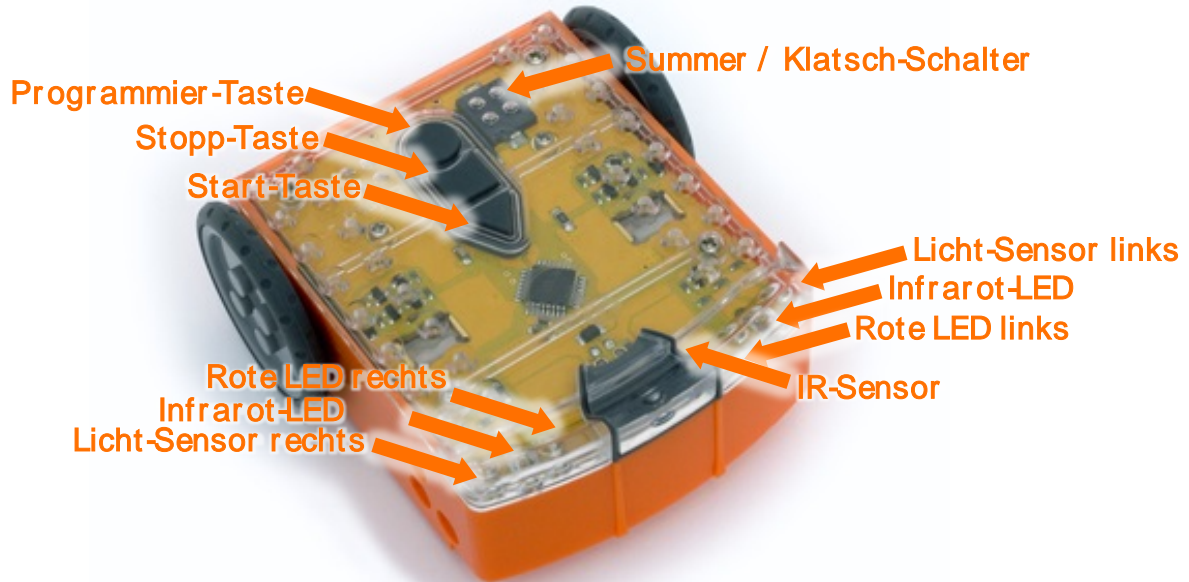
Edison ist jetzt startklar!



Den Hauptschalter zum Symbol hinschieben

Edison kennen lernen (Arbeitsblatt 1.1)

Die Lernenden müssen mit dem Edison vertraut sein und wissen, wo alle Sensoren sind und was die drei Tasten bewirken. Diese Arbeitsblatt sollte somit immer in Griffnähe sein, damit die Lernenden bei Unklarheiten die Bedeutungen der Tasten und Schalter nachschauen können.



Edisons Sensoren und Tasten

Start-Taste: Das Programm starten

Stopp-Taste: Das Programm beenden

Programmier-Taste: 1x drücken = Programm vom Computer laden
3x drücken = Strichcode lesen



Hauptschalter und Linien-Sensor

Der Linien-Sensor vom Edison besteht aus zwei Teilen, nämlich einem roten LED-Licht und einem Helligkeits-Sensor (Fototransistor). Das rote Licht scheint auf die Unterlage. Falls diese weiss ist und das Licht reflektiert, dann bekommt der Helligkeits-Sensor ein starkes Signal. Wenn die Unterlage schwarz ist und kein Licht reflektiert, dann misst der Sensor kein Signal.

Über das Kabel können Programme auf den Edison geladen werden. Das Kabel wird beim Kopfhörer-Ausgang von Computers oder Tablets eingesteckt.



Programmierkabel

Strichcode-Programmierung (Arbeitsblatt 1.2)

Um die Lernenden für die Robotik bzw. für den Edison zu begeistern und um zu zeigen, was damit gelernt werden kann, können sie mit den vorprogrammierten Programmen bzw. Strichcodes arbeiten.

Edison enthält einige fest installierte Programme, die aufgerufen werden können, indem man ihn über einen speziellen Strichcode fahren lässt.

Das Arbeitsblatt enthält vier Strichcodes, welche der Edison scannen kann:

- Strichcode – Fahrt mit Händeklatschen steuern
- Strichcode – Hindernisse umfahren
- Strichcode – An der Grenze abprallen
- Strichcode – Einer Lampe folgen

Der Edison hat noch weitere fest installierte Programme. Passende Abenteuer/Aufgaben hierzu sind im Dossier „*Strichcode-Programmierung*“ oder im eBook2 „*Du bist Steuermann*“ zu finden:

- Strichcode – Ringkampf
- Strichcode – Einer Linie folgen
- Strichcode – Fahren mit Fernsteuerung

Die Lernenden sollen den Edison die Strichcodes scannen lassen und dann beschreiben, welche Befehle der Roboter ausführt.

Alle Strichcodes sind ebenfalls im Dossier „*Strichcode-Programmierung*“ aufgeführt.

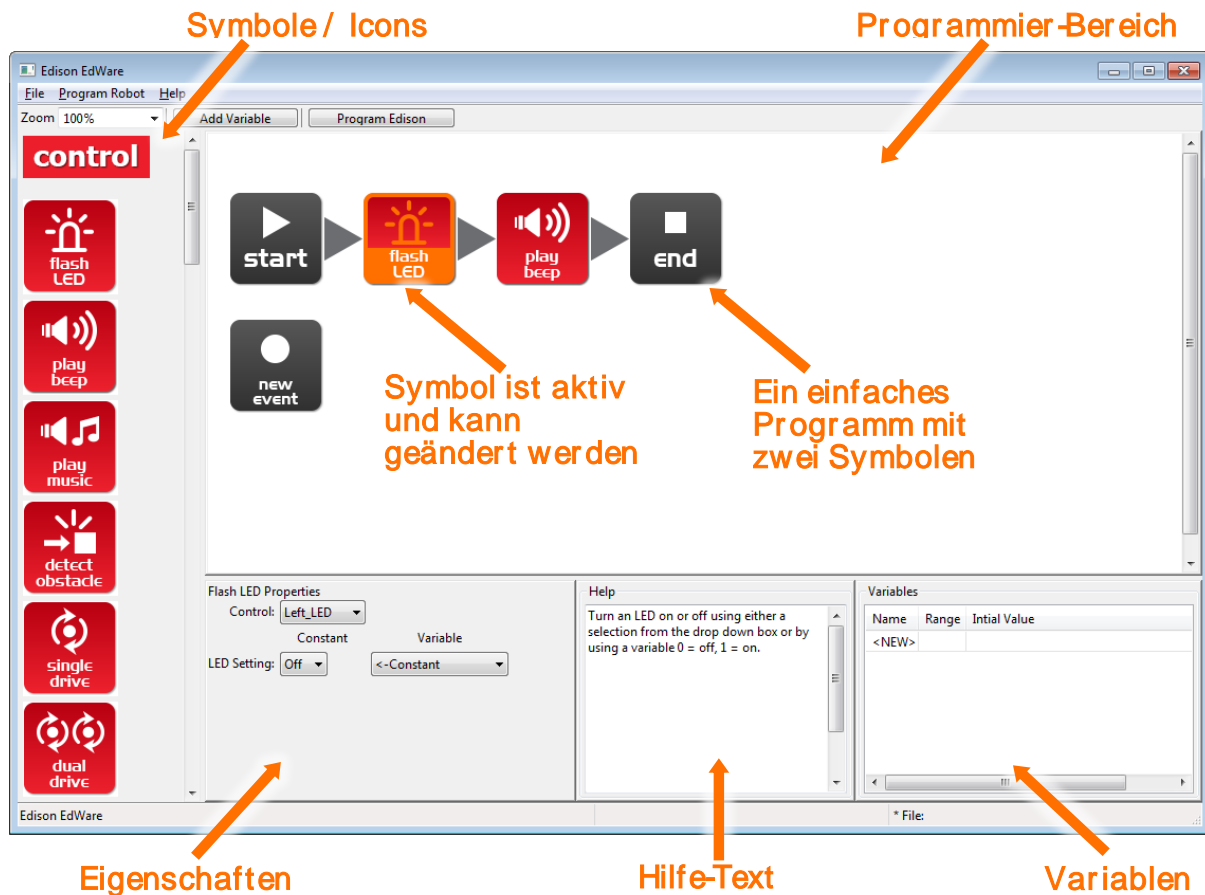
Strichcode scannen – So wird der Strichcode gelesen

1. Stelle Edison rechts vom Strichcode hin, so dass er nach links schaut
2. Drücke die runde Programmier-Taste 3x
3. Edison fährt ein Stück vorwärts und liest den Strichcode ein

EdWare kennen lernen (Arbeitsblatt 1.3)

Sobald die Lernenden begeistert und motiviert sind, können sie mit EdWare beginnen.

Um Sicherheit zu erlangen, sollten die Lernenden mit der Software experimentieren dürfen (ohne den Roboter). Dazu müssen die Lernenden die Software öffnen und die Symbole aus der Ikonenpalette auf der linken Seite in den Programm-Bereich auf der rechten Seite ziehen. Es ist nicht die zentrale Aufgabe bereits zu diesem Zeitpunkt ein funktionierendes Programm zu erstellen. Es geht lediglich darum, die Lernumgebung und die Symbole kennen zu lernen.



Um ein Programm zu entwickeln, müssen Symbole von der Ikonenpalette auf der linken Seite in den Programm-Bereich nach rechts gezogen werden. Diese müssen auf den Pfeilen zwischen den Symbolen "Beginn" (Start) und "Ende" platziert werden.

Nun kann ein Symbol ausgewählt (aktiviert) werden und die Einstellungen können bei "Eigenschaften" angepasst werden. Mit diesem Schritt wird festgelegt, wie sich Edison bei diesem Symbol verhalten soll.

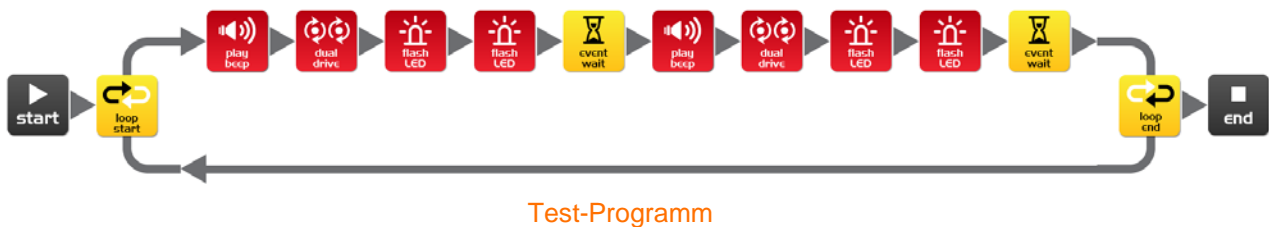
Der Hilfe-Text gibt Tipps zum Programmieren. Alles, was über ein Symbol relevant ist, befindet sich in diesem Feld.

Im Bereich "Variablen" kann man kleine Bereiche von Edisons Speicher verwalten und anschauen. Mehr dazu in Lektion 9.

Test-Programm (Arbeitsblatt 1.4)

Wenn EdWare auf dem Computer oder Tablet installiert ist, können nun Programme auf den Edison übertragen werden.

Zum Ausprobieren: Die Datei **TestProgram.edw** öffnen (Datei > öffnen: EdWare/Meine Programme). Das Programm sieht etwa so aus:



Hier könnte mit den Lernenden die „Denkweise“ oder das „Aussehen“ von Edison diskutiert werden:

- Die Pfeile zeigen die Richtung, in der Edison die Symbole liest.
- Edison schaut der Reihe nach auf jedes Symbol und führt den Symbolbefehl aus.
- In diesem Programm gibt es eine Endlos-Schleife, so dass das Programm immer weiterläuft.

Die Lernenden können nun das Testprogramm auf ihren Roboter herunterladen.

Das Programmierkabel wird in den Kopfhörer-Anschluss des Computers gesteckt und die Lautstärke auf den Maximalwert eingestellt.

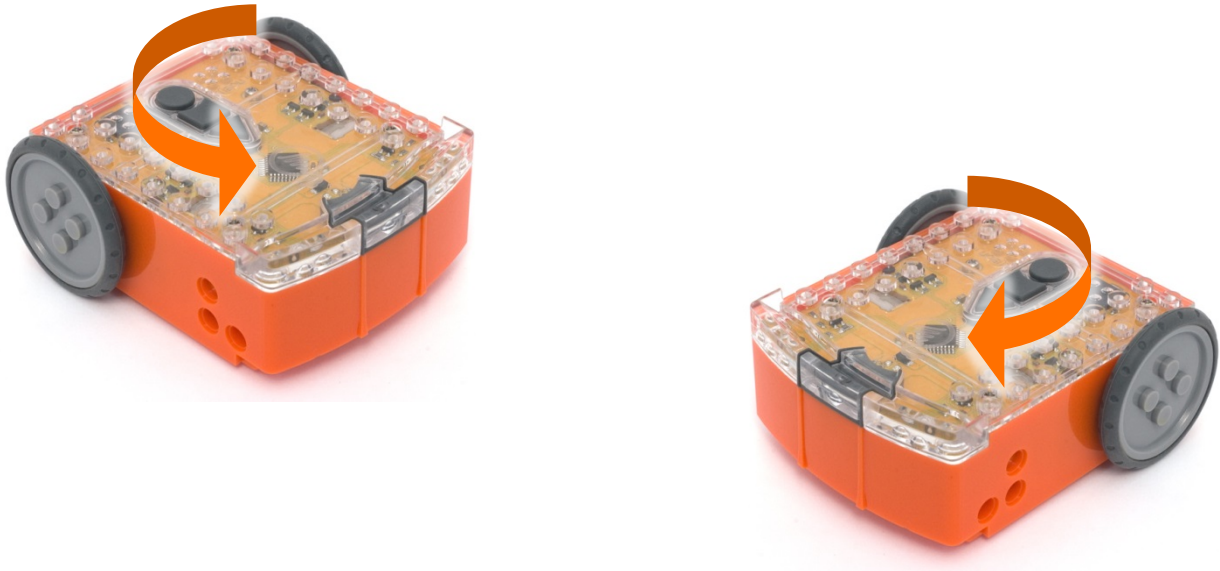
Das andere Ende des Programmierkabels auf der Unterseite von Edison einstecken.



Um das Testprogramm auf Edison zu übertragen, sind die folgenden Schritte auszuführen:

1. Die runde Programmier-Taste beim Edison 1x drücken (Edison vorbereiten)
2. In EdWare "zu Edison übertragen" wählen und "Übertrage Programm..."

Edison startet das Testprogramm, dreht sich nach links und rechts, blinkt mit den LED-Lampen und piept.



Mit den Lernenden kann anschliessend besprochen werden, was der Roboter nun wann und warum tut. Erklären Sie den Lernenden, dass der Roboter jedes Symbol einzeln und in der durch die Pfeile angezeigten Reihenfolge liest. Jedes Symbol sagt dem Roboter, was er tun soll.

Erklären Sie den Lernenden auch, wie das Programm vom Computer oder Tablet auf den Edison geladen wird: Das Programm wird durch das Kabel vom Computer / Tablet zum Edison geladen. Das Kabel wandelt die Töne von der Kopfhörerbuchse in Licht um, das der Roboter empfängt und speichert das Programm in seinem Gedächtnis, das heisst auf seinem Mikrocontroller.

LEKTION 2 : ROBOTERBEWEGUNG - FAHREN

Einführung in die sequentielle Programmierung: Die Lernenden lernen wie der Roboter nacheinander auf Befehlssymbole reagiert und wie die Konzepte von Zeit, Geschwindigkeit und Distanz zusammenspielen.

In dieser Lektion verwenden die Lernenden die grundlegenden Steuerungs-Laufwerksymbole, um den Roboter vorwärts und rückwärts und um eine festgelegte Strecke zu fahren.

Programm 1

Vorwärts fahren (Arbeitsblatt 2.1)

Benutzen Sie entweder das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.

Geben Sie den Lernenden den Auftrag, ein Programm zu schreiben, dass den Roboter eine vorher festgelegte Strecke vorwärts abfahren lässt. Das heisst von einer Anfangsmarkierung zu einer Endmarkierung. Der Roboter muss stehen bleiben, bevor er über die Markierung fährt.

Programm 2

Rückwärts fahren (Arbeitsblatt 2.2)

Verwenden Sie entweder das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.

Geben Sie den Lernenden den Auftrag ein Programm zu schreiben, dass den Roboter eine vorher festgelegte Strecke rückwärts abfahren lässt. Das heisst von einer Anfangsmarkierung zu einer Endmarkierung. Der Roboter muss stehen bleiben, bevor er über die Markierung fährt.

Programm 3

Vorwärts- & Rückwärtsfahren (Arbeitsblatt 2.3)

Benutzen Sie entweder das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.

Geben Sie den Lernenden den Auftrag, ein Programm zu schreiben, das den Roboter eine vorher festgelegte Strecke vorwärts und dann rückwärts abfahren lässt. Das heisst von einer Anfangsmarkierung zu einer Endmarkierung und wieder zurück. Der Roboter darf keine Markierung überfahren.

Programm 4

Geschwindigkeit anpassen (Arbeitsblatt 2.4)

Fordern Sie die Lernenden auf, mit Geschwindigkeit, Zeit und Distanz zu experimentieren. Die Lernenden können ihre eigenen Start- und Stopp-Marker erstellen.

- Welches ist die **schnellste** Zeit, die der Roboter zur Stoppmarkierung und zurück fahren kann?
- Welches ist die **langsamste** Zeit, die der Roboter zur Stoppmarkierung und zurück fahren kann?

Als Erweiterung können die Lernenden ihre Programme mit Pieptönen und blinkenden LEDs ergänzen.

Jedes Arbeitsblatt bietet den Lernenden genug Platz, um zu beschreiben was und warum der Roboter etwas tut.



Eine mögliche Antwort zu oben abgebildetem Programm kann wie folgt aussehen:

Der Roboter fährt vorwärts, weil das Programm ein Bewegungssymbol hat, das vorwärts zeigt. Er fährt bis zur Ziellinie und stoppt da, weil die Zeit bei „Ereignis warten“ 0,5 Sekunden ist und dies der Zeit entspricht, die der Roboter braucht um die gewählte Strecke zu fahren.

LEKTION 3 : ROBOTERBEWEGUNG - DREHEN

Sequentielles Programmieren und grundlegende Geometrie: Die Lernenden lernen, wie der Roboter auf Zeit und Geometrie reagiert und wie sie den Roboter steuern können.

In dieser Lektion verwenden die Lernenden die grundlegenden Steuerungs-Icons, um den Roboter um verschiedene Winkel (90° , 180° und 270°) zu drehen. In Zusammenhang mit dem in Lektion 2 erworbenem Wissen können die Lernenden ein neues Programm, bzw. eine neue Fahrsequenz programmieren.

Programm 1

Rechtsdrehung (Arbeitsblatt 3.1)

Verwenden Sie entweder das Arbeitsblatt „Activity 3.1“ oder farbige Bänder auf einem Schreibtisch.

Geben Sie den Lernenden den Auftrag, ein Programm zu schreiben, welches den Roboter um genau 90° drehen lässt.

Programm 2

Linksdrehung (Arbeitsblatt 3.2)

Verwenden Sie entweder das Arbeitsblatt „Activity 3.1“ oder farbige Bänder auf einem Schreibtisch.

Geben Sie den Lernenden den Auftrag, ein Programm zu schreiben, welches den Roboter um genau 180° drehen lässt.

Programm 3

Rechts- & Linksdrehung (Arbeitsblatt 3.3)

Verwenden Sie entweder das Arbeitsblatt „Activity 3.1“ oder farbige Bänder auf einem Schreibtisch.

Geben Sie den Lernenden den Auftrag, ein Programm zu schreiben, welches den Roboter um genau 90° nach rechts und dann 270° nach links drehen lässt.

Programm 4

Mini-Labyrinth (Arbeitsblatt 3.4)

Verwenden Sie entweder das Arbeitsblatt „Activity 3.2“ oder farbige Bänder auf einem Schreibtisch.

Fordern Sie die Lernenden auf ein Programm zu schreiben, welches durch ein Mini-Labyrinth navigieren kann. Die Lernenden können das erworbene Wissen aus den Lektionen 2 und 3 nutzen, um diese Herausforderung zu bewältigen.

Jedes Arbeitsblatt bietet den Lernenden genug Platz an, um zu beschreiben, was und warum der Roboter etwas tut.

Der Roboter dreht sich, weil das Programm ein Bewegungssymbol hat, welches die Drehung auslöst. Es dreht sich um 90 Grad, weil die Zeit bei „Ereignis warten“ 0,35 Sekunden ist und dies der Zeit entspricht, die der Roboter braucht um die gewählte Drehung von 90° zu fahren.

Hinweis: Vorwärts fahren, Drehung rechts, vorwärts fahren, Drehung links, vorwärts fahren

LEKTION 4: FAHR-CHALLENGE UND LA-OLA-WELLE

Vertiefung des Gelernten: Die Lernenden nutzen ihre erworbenen Kenntnissen aus den Lektionen 1 bis 3, um zwei Projekte zu gestalten.

Programm 1 Fahr-Challenge

Die Lernenden wählen eine Herausforderung bzw. eine Problemstellung für ihren Roboter und schreiben dann ein Programm für ihn. Einige Beispiele könnten sein:

- Edison soll um ein Hindernis herum fahren, wie zum Beispiel eine Tasse oder ein Bleistift
- Edison soll am Rand eines Schreibtisches entlang fahren
- Die Lernenden zeichnen ein Labyrinth auf einem grossen Papier (Flipchart, Packpapier, etc.) welches Edison durchfahren muss
- Die Lernenden bauen ein Labyrinth mit Bausteinen, welches Edison durchfahren muss

Programm 2 La-Ola-Welle

Dies ist eine lustiges Klassen- bzw. Gruppenprojekt: Jeder Roboter führt das gleiche Programm aus, jedoch zeitlich etwas verschoben. Das Ergebnis ähnelt einer La-Ola-Welle oder einer Choreographie.

Die Lernenden schreiben ein kurzes Programm mit einer Abfolge von Roboterbewegungen. Sie können auch dazu ermuntert werden Licht und Ton in ihre Programme zu integrieren.

Die Klasse wählt nach der Präsentation der einzelnen Lernenden ihr Klassen-Programm aus. Dieses Programm wird dann mit allen Lernenden geteilt, damit alle auf ihrem Edison das Programm schreiben können. Aber: Alle fügen an den Anfang des Programms ein „Ereignis warten“ Symbol ein.

Um eine La-Ola-Welle mit Welleneffekt zu bilden, schreibt jeder Lernende eine etwas andere Zeit in das neue Ereignis warten Symbol. Beispielsweise:

Roboter 1 - Wartezeit 0 Sekunden
Roboter 2 - Wartezeit 0.2 Sekunden
Roboter 3 - Wartezeit 0.4 Sekunden
...

Die Roboter werden in der Reihenfolge ihrer Verzögerungszeit in einer Reihe angeordnet. Alle drücken gleichzeitig „Start“.

LEKTION 5: PROJEKT 1 – MEIN ERSTES PROGRAMM

Kreatives Denken und Problemlösen: Die Lernenden können durch ihre eigenen Lösungswege und deren Herausforderungen herausfinden, wie der Roboter eine Lösung finden kann. Die Lernenden können ihr eigenes Thema auswählen, den Zweck des Programms angeben und erläutern, wo er in der realen Welt eingesetzt werden kann.

Mit Hilfe vom Arbeitsblatt 5.1 können die Lernenden ihre Vermutungen, Beobachtungen, Hypothesen und ihre Gedanken dokumentieren.

1. Ein Problem auswählen, das der Roboter lösen kann oder eine lustige Bewegung zu Musik

In Zukunft werden Roboter unsere Helfer sein und uns im Alltag oder in Beruf unterstützen. Wir haben bereits Roboter Staubsauger und daher ist die Zukunft gar nicht mehr so weit weg. Welche weiteren hilfreichen oder unterhaltsamen Alltagsaufgaben könnten Lernende programmieren?

Beispiele sind:

- Tanz zur Musik - Roboter tanzt zur Lieblingsmusik des Lernenden
- Staubsauger - Schüler können ein Wohnzimmer-Boden-Grundriss auf einem grossen Papier zeichnen und den Roboter so programmieren, dass er die ganze Fläche abfährt
- Sicherheitsroboter - Lernende können einen sicheren Bereich definieren (Papier oder ein Objekt). Der Roboter hat die Aufgabe, den sicheren Bereich zu schützen, indem er patrouilliert und das Objekt umfährt.

Die Lernenden sollen sich für eine Problemstellung entscheiden und diese mit anderen Lernenden oder der Lehrperson besprechen.

2. Das Problem oder die Bewegungen beschreiben, die der Roboter machen muss

Bevor die Lernenden die Problemstellung bearbeiten, sollen sie **das Problem, welches ihr Roboter lösen soll, in eigenen Worten beschreiben**. Als weiterer Schritte versuchen sie eine mögliche Lösung zu beschreiben, **wie ihr Programm aussehen könnte**, welches dem Roboter sagt, wie er das Problem lösen soll.

Beispielsweise:

Das Problem ist: *Die Menschen können gut tanzen, aber ihre Bewegungen sind langsam und können nicht immer genau, auf die Sekunde gleich, mit anderen Tänzern synchronisiert werden.*

Mein Roboter wird dies lösen, indem er ... *Schnelle Bewegungen macht und mit den anderen Robotern immer genau synchronisiert ist und immer die gleiche Tanzabfolge macht.*

3. Das Programm schreiben und testen

Bevor die Lernenden ihr Programm in der Software schreiben, sollen sie ihre Gedanken und Überlegungen skizzieren. Anschliessend schreiben die Lernenden ihr Programm, indem sie die Icons nutzen, die sie bereits kennen gelernt haben. Abschliessend testen sie das Programm.

4. Fehlerbehebung

Nicht immer ist der erste Versuch erfolgreich. Einige Lernende können allenfalls zusätzliche Ermutigungen oder Anregungen, Tipps brauchen, um einen weiteren Versuch zu starten. Ein zentraler Punkt bei der Programmierung ist es, dass Fehler machen dazu gehört! Fehler sind ein normaler Teil der Programmierung (oder einer Ingenieurdisziplin). Thomas Edison ist 10.000 Mal gescheitert, bevor es ihm gelang, die Glühbirne zu erfinden. Die Lernenden können ihre Fehler auf dem Arbeitsblatt beschreiben.

5. Beschreiben der verwendeten Programmiersymbole und was sie bewirken

Auf dem Arbeitsblatt 5.2 beschreiben die Lernenden 3 wichtige, zentrale Icons, welche in ihrem Programm verwendet werden. Sie beschreiben, was das Icon im Programm bewirkt. Sie können zusätzliche Blätter ausdrucken, damit die Lernenden weitere Programmiersymbole beschreiben können.

6. Demo

Die Lernenden präsentieren den Roboter. Sie demonstrieren das Programm ihres Roboters und erklären ihre Idee, Überlegungen: das Ausgangsproblem, das Programm und die Stolpersteine, Fehler, denen sie begegnet sind und wie sie sie lösen konnten.

LEKTION 6 : KLATSCH-SENSOR

Einführung in die Eingaben (Sensoren): Die Lernenden lernen, wie der Roboter auf Äußere Reize reagieren kann (hier: Klatschen). Diese Lektion enthält auch eine lustige Klassen-Aktivität.

In dieser Lektion lernen die Lernenden, wie ein Roboter äussere Einflüsse aufnehmen, spüren kann und wie sie die Antwort des Roboters darauf programmiert können und so Reaktionen des Roboters bewirken können.

Programm 1

LED reagiert auf Händeklatschen (Arbeitsblatt 6.1)

Die Lernenden lernen, wie die Programmierung vom Roboter funktioniert, wenn er darauf wartet, dass ein Ereignis eintritt, bevor das Programm fortgesetzt wird. In diesem Fall ist das Ereignis ein lautes Geräusch wie zum Beispiel ein Klatschen.

Programm 2

Mit Händeklatschen fahren (Arbeitsblatt 6.2)

Die Lernenden integrieren bzw. kombinieren das in früheren gelernten Lektionen, um den Roboter als Reaktion auf ein Ereignis (hier: Klatschen) zu fahren.

Programm 3

Tanzen nach Händeklatschen (Arbeitsblatt 6.3)

Die Lernenden erstellen ihre eigenen Roboterbewegungen als Reaktion auf das Klatschen. Sie haben zwei Möglichkeiten:

1. Ein Programm, das den Roboter tanzen lässt, mit mehreren Tanzbewegungen
2. Ein Programm, das den Roboter durch ein Labyrinth in Reaktion auf Klatschen fahren lässt

LEKTION 7 : HINDERNISSEKLENNUNG

Die Lernenden programmieren den Roboter so, dass er Entscheidungen (künstliche Intelligenz) als Reaktion auf Hindernisse in der Roboterumgebung selber treffen kann.

Wie funktioniert die Infrarot-Hindernis-Erkennung? (Arbeitsblatt 7.1)

Die Lernenden werden mit der Infrarot-Hinderniserkennungstechnologie des Roboters vertraut gemacht. Sie lernen zudem wie der Roboter Hindernisse, die sich in seinem Weg befinden, erkennen kann.

Programm 1

Ein Hindernis erkennen und stoppen (Arbeitsblatt 7.2)

Die Lernenden schreiben ein Programm, welches den Roboter fahren lässt, bis ein Hindernis in seinem Weg erkannt wird. Der Roboter stoppt dann und vermeidet so eine Kollision.

Programm 2

Ein Hindernis erkennen und ausweichen I (Arbeitsblatt 7.3)

Die Lernenden schreiben ein Programm, welches den Roboter fahren lässt, bis ein Hindernis in seinem Weg erkannt wird. Der Roboter wendet sich dann ab und vermeidet so eine Kollision.

Programm 3

Ein Hindernis erkennen und ausweichen II (Arbeitsblatt 7.4)

Die Lernenden schreiben ein Programm, welches eine Dauerschleife enthält, die den Roboter fahren lässt, bis ein Hindernis in seinem Weg erkannt wird. Der Roboter wendet sich dann ab und vermeidet so eine Kollision. Der Roboter fährt dann allerdings wieder weiter, aufgrund der zuvor programmierten Dauerschleife.

Programm 4

Ein Hindernis erkennen und umfahren (Arbeitsblatt 7.5)

Die Lernenden schreiben ein Programm, welches ein „IF“-Abfrage enthält und so die Fähigkeit hat zu antworten, ob ein Hindernis auf der linken oder rechten Seite steht. Um dies zu programmieren brauchen die Lernenden das „Wenn?“ Icon.

Die IF-Abfrage verleiht dem Roboter die Fähigkeit Entscheidungen ohne menschliche Führung zu treffen. Wenn das in einem Roboter geschieht, wird er als **autonomer Roboter** bezeichnet, da er künstliche Intelligenz hat.

Wenn die Hinderniserkennung zu empfindlich oder zu unempfindlich ist, können die Lernenden sie mit den Anweisungen im Dossier „Info“ neu kalibrieren.

LEKTION 8 : EINE LINIE ERKENNEN UND FOLGEN

Industrielles Roboterverhalten: Die Lernenden lernen die grundlegende Robotererkennung und –Kontrolle kennen. Diese ist ähnlich wie in fortgeschrittenen automatisierten Fabriken und Lagerhallen.

Wie funktioniert der Linien-Sensor? (Arbeitsblatt 8.1)

Die Lernenden werden mit dem Roboter-Tracking-Sensor vertraut gemacht. Sie lernen wie der Roboter erkennen kann, ob die Fahrfläche schwarz oder weiß ist.

Programm 1

Bis zu der schwarzen Linie fahren (Arbeitsblatt 8.2)

Die Lernenden schreiben ein Programm, welches den Roboter fahren lässt, bis er auf eine schwarze Linie trifft.

Programm 2

Innerhalb einer Grenze fahren (Arbeitsblatt 8.3)

Die Lernenden schreiben ein Programm, welches den Roboter in eine Grenze fahren lässt. Der Linienvorgangssensor erkennt den Rand und der Roboter dreht sich um. Sie können dazu Arbeitsblatt „Activity 8.1“ verwenden. Oder sie machen sich eine eigene Vorlage mit einem grossen weissen/hellen Papier und einem schwarzen Marker oder schwarzen Klebeband, mit welchem sie den Umriss selber zeichnen können.

Programm 3

Einer Linie folgen (Arbeitsblatt 8.4)

Die Lernenden schreiben ein Programm, welches den Roboter einer schwarzen Linie folgen lässt. Sie können Arbeitsblatt „Activity 8.2“ verwenden. Alternativ kann mit einem grossen weissen/hellen Papier und einem schwarzen Marker oder schwarzen Klebeband ein eigener Weg gezeichnet werden. Ebenfalls gibt es Kopiervorlagen. Diese sind auf der Website beim IdeenSet Robotik 2 unter Zubehör zu finden.

Video – Humans need not apply

Line Tracking – das Folgen einer vorgegebenen Linie – ist ein sehr einfaches Roboterverhalten, welches in vielen automatisierten Fabriken und Lagerhäusern heute eingesetzt wird. Aber wie sieht es in der weiteren Zukunft aus? Wie sehen Arbeitsplätze der Zukunft mit Robotern aus? Werden sie überhaupt existieren? Das folgende Video kann als Anregung für eine Diskussion in der Klasse genutzt werden: Was bedeutet die weit verbreitete Aufnahme von Robotern für die Zukunft des Menschen?

<https://www.youtube.com/watch?v=7Pq-S557XQU> (15 Min.)



LESSON 9 : AUF LICHT REAGIEREN

Umgebungsmessung und Programmierung: Die Lernenden lernen, wie die Messung von Lichtniveaus, die Speicherung im Gedächtnis und die Durchführung unter dem Aspekt der Mathematik genutzt werden kann, um das Verhalten des Roboters zu kontrollieren.

Wie funktioniert der Licht-Sensor? (Arbeitsblatt 9.1)

Die Schülerinnen und Schüler lernen, was eine Variable beim Programmieren ist und wie man eine solche Variable beim Edison einsetzen kann. Sie lernen ebenfalls die zwei unterschiedlich grossen Variablen; 'bytes' (8 Bits - 0 bis 255) und 'words' (16 Bits - 32.767 bis +32.767) kennen.

Programm 1

Helligkeitsalarm (Arbeitsblatt 9.2)

Die Lernenden schreiben ein Programm, welches einen akustischen Alarm auslöst, wenn die Lichter im Raum eingeschaltet werden. Dieses Programm verbindet die Verwendung von Variablen und ein grundlegendes mathematisches Prinzip (einem "größer als" (>) Vergleich), als Abfrage.

Programm 2

Automatische Beleuchtung (Arbeitsblatt 9.3)

Die Lernenden schreiben ein Programm, welches den Roboter vorwärtsfahren lässt, währendem er die Lichtstärke überwacht. Fährt der Roboter in einen dunklen Bereich, werden die Frontlichter automatisch eingeschaltet. Dieses Programm verwendet einen 'weniger als' (<) Vergleich.

Programm 3

Einem Licht folgen (Arbeitsblatt 9.4)

Die Lernenden schreiben ein Programm, welches den Roboter in Richtung helles Licht fahren lässt, wie zum Beispiel das von einer Taschenlampe. Dieses Programm führt mehr Berechnungsmathematik ein, indem man Subtraktion und dann einen "weniger als" (<) Vergleich verwendet.

Animal behaviours in robotics

Das obige Programm zeigt ein Verhalten, welches demjenigen einer Motte ähnelt: Eine Motte wird in der Nacht vom Licht einer Strassenlaterne angezogen wird. Dies kann eine gute Diskussionsgrundlage sein, ob sich die Intelligenz der Insekten von der von Robotern unterscheiden, bzw. ob sie ähnlich/gleich ist. Warum ist ein Insekt, das vom Licht angezogen wird kein Roboter?

LEKTION 10: PROJEKT II – MEIN ZWEITES PROGRAMM

Kreatives Denken und Lösen von Problemen: In diesem zweiten Projekt entwerfen die Schülerinnen und Schüler ihre eigene Problemstellung und programmieren einen Lösungsansatz, der anschliessend vom Roboter ausgeführt wird.

Die letzte und abschliessende Lektion kann ebenfalls nach zwei Unterrichtsstunden abgeschlossen werden. Alternativ bietet es sich aber an mehrere Unterrichtsstunden dafür einzusetzen. So kann dieses Projekt ausgebaut werden und eine Art Abschlussarbeit darstellen.

Die Lernenden können ihr eigenes Thema, ihren eigenen Schwerpunkt auswählen, den Zweck des Programms angeben und erklären, wo das Programm in der realen Welt eingesetzt werden könnte. In diesem zweiten Projekt kann mehr von den Lernenden erwartet werden, da sie nun über (fast) alle Informationen/Icons betreffend dem Edison verfügen. Sie haben jetzt auch mehr Programmierkenntnisse und kennen die Sensoren vom Edison und können diese in ihren Programmen verwenden.

1. Ein Problem auswählen, das der Roboter lösen kann

Die Lernenden haben nun das Video "*Humans need not apply*" gesehen und erfahren, wie Roboter in Zukunft eingesetzt werden können. Dieses Video sollte die Lernenden auf neue Ideen bringen und ihre Kreativität anregen.

Beispiele sind:

- Rettungsroboter - Der Roboter fährt innerhalb einer Grenze, und sucht eine verlorene Person sucht (Objekt, Minifigur oder Puppe). Wenn der Roboter die Person lokalisiert hat, ertönt ein Alarm.
- Fahrerloses Auto - Der Roboter fährt auf einer bestimmten Strasse (Linie), ohne mit Menschen, anderen Autos oder Gebäuden (Spielzeug) zu kollidieren.

Die Lernenden sollen sich für eine Problemstellung/Aufgabe entscheiden und diese mit anderen Lernenden oder der Lehrperson besprechen.

2. Das Problem oder die Bewegungen beschreiben, die der Roboter machen muss

Die Lernenden beschreiben auf dem Arbeitsblatt das Problem, welches sie ihrem Roboter stellen. Weiter sollen sie das entsprechende Roboterprogramm erläutern und wie das Programm das Problem löst.

Das Problem ist: *Menschliche Fahrer sind leicht ablenkbar und können beim Fahren Fehler machen. Solche Fehler können tödlich enden.*

Die Lösung könnte sein: *Dieses Programm zeigt, wie ein Roboter mithilfe einer Hinderniserkennung eine Strasse entlangfahren und dabei Kollisionen mit Menschen, anderen Autos oder Gebäuden vermeiden kann. Daher kann er sicherer sein.*

3. Das Programm schreiben und testen

Bevor die Lernenden ihr Programm in der Software schreiben, sollen sie ihre Gedanken und Überlegungen skizzieren. Anschliessend schreiben die Lernenden ihr Programm, indem sie die Icons nutzen, die sie kennen gelernt haben. Abschliessend testen sie das Programm.

4. Fehlerbehebung

Nicht immer ist der erste Versuch erfolgreich. Einige Lernende können allenfalls zusätzliche Ermutigungen, Anregungen oder Tipps brauchen, um einen weiteren Versuch zu starten. Ein zentraler Punkt bei der Programmierung ist es, dass Fehler machen dazu gehört! Fehler sind ein normaler Teil der Programmierung (oder einer Ingenieurdisziplin). Thomas Edison ist 10.000 Mal gescheitert, bevor es ihm gelang, die Glühbirne zu erfinden. Die Lernenden können ihre Fehler auf dem Arbeitsblatt beschreiben.

5. Beschreiben der verwendeten Programmiersymbole und was sie bewirken

Auf dem Arbeitsblatt 5.2 beschreiben die Lernenden 3 wichtige, zentrale Icons, welche in ihrem Programm verwendet werden. Sie beschreiben, was das Icon im Programm bewirkt. Sie können zusätzliche Blätter ausdrucken, damit die Lernenden weitere Programmiersymbole beschreiben können.

6. Demo

Die Lernenden präsentieren den Roboter. Sie demonstrieren das Programm ihres Roboters sowie ihre Idee und Überlegungen. Folgenden Inhalt sollte die Demo beinhalten: Ausgangsproblem, Programm, Stolpersteine, Fehler und Fehlerlöseverhalten.

Edison Pass

Programm	Stamp
2.1 Vorwärts fahren	
2.2 Rückwärts fahren	
2.3 Vorwärts- und Rückwärtsfahren	
2.4 Geschwindigkeit anpassen	
3.1 Rechtsdrehung	
3.2 Linksdrehung	
3.3 Rechts- und Linksdrehung	
3.4 Mini-Labyrinth	
4.1 Fahr-Challenge	
4.2 La-Ola-Welle	
5 Projekt I: Mein erstes Programm	
6.1 LED reagiert auf Händeklatschen	
6.2 Mit Händeklatschen fahren	
6.3 Tanzen nach Händeklatschen	

Programm	Stamp
7.1 Ein Hindernis erkennen und stoppen	
7.2 Ein Hindernis erkennen und ausweichen I	
7.3 Ein Hindernis erkennen und ausweichen II	
7.4 Ein Hindernis erkennen und umfahren	
8.2 Bis zu der schwarzen Linie fahren	
8.3 Innerhalb einer Grenze fahren	
8.4 Einer Linie folgen	
9.2 Helligkeitsalarm	
9.3 Automatische Beleuchtung	
9.4 Einem Licht folgen	
10 Projekt II: Mein zweites Programm	
Eigene Projekte/Programme 1.	
Eigene Projekte/ Programme 2.	
Eigene Projekte/ Programme 3.	