

DIPLOMARBEIT

Robotereinsatz im Schulunterricht

Institut für Softwaretechnologie
Technische Universität Graz

Martin Neppel



Betreuer: Univ.-Prof. Dipl.-Ing. Dr. techn. Franz Wotawa



Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

.....

(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....

date

.....

(signature)

Kurzfassung

Diese Diplomarbeit, im Rahmen des Lehramtsstudiums der Unterrichtsfächer Chemie und Informatik und Informatikmanagement, am Institut für Softwaretechnologie an der Technischen Universität Graz, befasst sich mit dem Robotereinsatz im österreichischen Schulunterricht. Neben einer Bündelung der fragmentarisch vorhandenen Informationen, betreffend den Robotereinsatz an Schulen, und der Darstellung der aktuellen Verwendungssituation soll diese Arbeit eine Informations- und Argumentationsgrundlage für Lehrerinnen und Lehrer sein. Nach der Definition des Systems Roboter, wird die derzeitige Verwendungsintensität von Robotern im Schulunterricht anhand einer im Rahmen dieser Diplomarbeit durchgeführten quantitativen Forschung dargelegt. Es konnte herausgefunden werden, dass die Durchdringung aktuell etwa bei einem Viertel der Stichprobe liegt und je nach Schultyp stark divergiert. Besonders an neuen Mittelschulen und allgemein bildenden höheren Schulen werden die Vorteile der Wissensvermittlung durch Roboter kaum erkannt und genutzt, weshalb anschließend Gründe für den Einsatz von Robotern im Unterricht im Detail dargelegt werden. Diese umfassen vor allem eine Steigerung der Motivation und der damit einhergehenden Lernleistung sowie berufswahlprägende Aspekte. Auch die Vorteile von projektorientiertem Lernen könnten durch den Einsatz von Robotern realisiert werden. Um den Robotereinsatz möglichst effektiv und effizient gestalten zu können, werden die Einsatzmöglichkeiten von Robotern im schulischen Alltag abgegrenzt. Diese Arbeit differenziert zwischen Lernen über Robotik sowie Lernen durch Robotik und konnte anhand aktueller Lehrplaninhalte des Bundesministeriums für Unterricht, Kunst und Kultur mögliche Einsatzgebiete für Roboter abstecken. Sinnvolle Lehrplanbezüge konnten insbesondere für den Physik-, den Mathematik- und den Informatikunterricht hergestellt werden. Auch konnte herausgearbeitet werden, warum die programmierbaren Automaten eine Bereicherung für fächerübergreifenden Unterricht sind. Diese Arbeit informiert ebenfalls über aktuelle Roboterwahlmöglichkeiten für den schulischen Einsatz. Es wird dargelegt, welche wichtigen Fragen die verantwortlichen Schulvertreter sich vor dem Kauf eines Roboters stellen sollten. Unterschiede zwischen praktischen und theoretischen Systemen werden aufgezeigt. Anhand des Aufwandes vor dem Erstbetrieb werden Roboter in Fertigmodelle, modulare Modelle und Bausatzmodelle unterteilt. Die Vor- und Nachteile der einzelnen Kategorien werden im Detail besprochen. Anschließend werden exemplarisch für jede Kategorie aktuell am Markt

erhältliche Robotermodelle vorgestellt, die aufgrund ihrer Eigenschaften für Schulen von Interesse sein könnten. Es wird erklärt, warum die Programmiersprache ebenfalls als Kriterium zur Systemauswahl herangezogen werden kann. Alsdann werden relevante grafische und textuelle Programmiersprachen vorgestellt. Abschließend wird ein Überblick über die vorgestellten Robotersysteme sowie die von ihnen unterstützten Programmiersprachen gegeben.

Abstract

This diploma thesis, which was written as part of the study program for becoming a teacher in Chemistry and Informatics at the Graz University of Technology, discusses the use of mobile robots in Austrian schools. Besides the bundling of the fragmentarily existing information about the use of robots in schools and an overview about the actual situation of their use, this work should be an argumentation and information basis for teachers. After the definition of robots, the actual use intensity of robots in Austrian schools will be analyzed through a quantitative research. It has been shown, that about a quarter of the interviewed schools use robots, but also that the use in the different school types diverges. Especially at the new middle schools and at the high schools, the advantage of the use of robots for teaching is hardly recognized, wherefore reasons for the use of robots for educating will be subsequently shown in detail. These include mainly the higher motivation and associated learning performance and also associated aspects for the career choice. Also, the advantages of the project-orientated learning theory can be realized through the use of robots in schools. For making the use of robots effective and efficient, possible applications of robots in schools are defined. This thesis differentiates between learning about robots and learning through robots and shows possible areas of application based on the actual curricula of the Austrian ministry of education. Appropriate options for the use of robots in schools were found in the curricula of the school subjects physics, mathematics and informatics. It has also been possible to show why robots enrich cross-curricular teaching. This diploma thesis also informs about the actual choices of robot models for the use in schools. It is also shown which important questions the responsible teachers in schools should ask themselves before buying a robot for the use in schools. Differences between theoretical and practical systems are shown. Based on the amount of time, which is needed before the first use of a robot model, the models were divided in the categories ready-to-use models, modular models and self-made models. The pros and cons of these categories were explained in detail. Of each category there are some exemplary models presented, which are probably interesting for the use in schools because of their properties. It is also explained why the programming language is a criterion for the choice of a robot. Compatible graphical programming languages as well as textual programming languages for the presented robot models

are explained. Finally an overview about all stated robot models and the compatible programming languages is shown.

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die durch ihre Zeit, ihre Geduld, ihre fachliche Kompetenz sowie ihre persönliche Unterstützung das Verfassen dieser Diplomarbeit erst ermöglicht haben.

Ganz besonderer Dank gilt meinem Betreuer, Herrn Univ.-Prof. Dipl.-Ing. Dr. techn. Franz Wotawa der mir ermöglicht hat mich dem interessanten Thema des Robotereinsatzes im Unterricht widmen zu können, und für seine fachliche Kompetenz mit welcher das Verfassen dieser Arbeit gestützt wurde.

Auch bei Dipl.-Ing. Bakk. rer. soc. oec. Martin Kandlhofer möchte ich mich für seine Zeit und seinen konstruktiven Rat bedanken.

Mein besonderer Dank gilt auch meiner Lebensgefährtin Evelyn, die mir zu jeder Zeit mit allen verfügbaren Mitteln zur Seite stand. Ebenso möchte ich mich bei meiner Familie und meinen Freunden bedanken, die mich immer wieder ermutigt haben an meiner Diplomarbeit weiterzuarbeiten und mich durch ihre Geduld in dieser wichtigen Phase meines Lebens unterstützt haben.

Schließlich möchte ich mich auch bei meinen Studienkollegen und Kolleginnen sowie meinen Professoren bedanken die mitverantwortlich für meine angenehme Studienzeit waren.

Inhaltsverzeichnis

1	Einführung.....	1
2	Ziele der Arbeit.....	2
3	Ausgangssituation.....	3
3.1	Definition Roboter	3
3.2	Aufbau der Erhebung	4
3.3	Ergebnis der Erhebung	6
3.4	Interpretation der Ergebnisse	9
4	Gründe für den Einsatz von Robotik im Unterricht	11
4.1	Einsatz von Robotern an Schulen zur Steigerung der Motivation	12
4.2	Schule, Roboter und die Berufswahl	14
4.3	Projektorientiertes Lernen	16
4.4	Fazit	17
5	Abgrenzung der Einsatzmöglichkeiten von Robotern im schulischen Alltag	19
5.1	Lernen über Robotik.....	19
5.2	Lernen durch Robotik.....	20
6	Abgrenzung möglicher Einsatzgebiete von Robotern durch Lehrplaninhalte	23
6.1	Lehrplanbezüge aus dem Unterrichtsfach Physik	23
6.2	Lehrplanbezüge aus dem Unterrichtsfach Mathematik	27
6.3	Lehrplanbezüge aus dem Unterrichtsfach Informatik	30
6.4	Einsatz von Robotern im fächerübergreifenden Unterricht.....	32
6.5	Fazit	33
7	Auswahl möglicher Roboter für den schulischen Einsatz.....	35
7.1	Wichtige Fragen, die vor dem Kauf eines Roboters gestellt werden können.....	35
7.2	Überblick über ausgewählte Fertigmodelle	39
7.3	Überblick über ausgewählte modulare Modelle.....	45
7.4	Überblick über ausgewählte Bausatzmodelle.....	58
8	Programmiersprache als Kriterium zur Systemauswahl.....	64

8.1	Grafische Programmiersprachen	65
8.2	Textuelle Programmiersprachen	83
8.3	Robotersysteme und unterstützte Programmiersprachen im Überblick.....	93
9	Zusammenfassung.....	94
10	Schlussbemerkung und Ausblick.....	96
11	Quellenverzeichnis	97
12	Anhang.....	99
12.1	Fragebogen der Schulumfrage	99
12.2	[25] E-Mail: danielbw@betterbots.com vom 07.01.2014.....	100
12.3	[33] E-Mail: Roger.Wolf@legoeducation.eu vom 07.02.2014	100

Abbildungsverzeichnis

Abbildung 1 – Schematische Darstellung eines Roboters	4
Abbildung 2 – Finch: Ansicht von oben, Sensorübersicht, USB Anschluss und LED	41
Abbildung 3 – Scribbler 2 von oben	43
Abbildung 4 – Scribbler 2 von unten	43
Abbildung 5 – NXT und Komponenten	48
Abbildung 6 – EV3 und Module	48
Abbildung 7 – ROBO TX Controller	51
Abbildung 8 – Fischertechnik Beispiel eines mobilen Roboters	51
Abbildung 9 – VEX IQ Controller	51
Abbildung 10 – Anwendungsbeispiel des VEX IQ Super Kit	53
Abbildung 11 – Auslieferungszustand	60
Abbildung 12 – Anwendungsbeispiel	60
Abbildung 13 – Boe Bot Basisausstattung	61
Abbildung 14 – Boe Bot Anwendungsbeispiel	61
Abbildung 15 – CREATE Lab Visual Programmer Expression Builder	67
Abbildung 16 – CREATE Lab Visual Programmer Sequence Builder	68
Abbildung 17 – Finch Dreams	70
Abbildung 18 – Snap! Oberfläche mit Beispielprogramm	72
Abbildung 19 – Scratch Beispielprogramm im offline Editor	74
Abbildung 20 – Greenfoot Programmbeispiel / Java Editor	76
Abbildung 21 – RoboRealm Finch Interface	77
Abbildung 22 – S2 GUI Beispielprogramm	78
Abbildung 23 – ROBO PRO Programmbeispiel	80
Abbildung 24 – Programmoberfläche – LEGO Mindstorms Education EV3 Software	81
Abbildung 25 – MODKIT for VEX, Entwicklungsumgebung mit Beispielprogramm	83

Tabellenverzeichnis

Tabelle 1 – Schulen der Umfrage und ihre prozentuelle Verteilung	5
Tabelle 2 – Schulen der Umfrage nach Typ und Bundesland	6
Tabelle 3 – Funktionsbeispiele für das Quadrieren der Zahl x	29
Tabelle 4 – Unterschiede von LEGO Mindstorms NXT und LEGO Mindstorms EV3	47
Tabelle 5 – Übersicht über Sensoren der verglichenen modularen Roboter Sets	56
Tabelle 6 – Robotersysteme und unterstützte Programmiersprachen im Überblick	93

Diagrammverzeichnis

Diagramm 1 – Roboterverwendung im Schulunterricht	7
Diagramm 2 – Roboterverwendung nach Schultypen	8
Diagramm 3 – Gründe gegen Robotereinsatz bei Interesse	9

1 Einführung

Die vorliegende Arbeit zielt darauf ab, einen Überblick über die verschiedenen Einsatzmöglichkeiten von Robotern im Unterricht zu geben. Einerseits sollen die fragmentarisch vorhandenen Informationen und Erkenntnisse über den Robotereinsatz in Schulen miteinander in Verbindung gebracht werden um einen Gesamtüberblick zu ermöglichen. Andererseits soll diese Arbeit als Hilfestellung dienen, den sinnvollen Einsatz von Robotern in Schulen begründen zu können und die Auswahl des richtigen Roboters für den entsprechenden Einsatz zu erleichtern. Diese Zusammenstellung soll vor allem auch Lehrer und Lehrerinnen als Adressaten ansprechen und soll unterstützend für die Verwendung von Robotik im Unterricht sein. Es werden auch grundlegende Fragen bezüglich der erhofften Ziele durch den Einsatz von Robotik-Inhalten im Unterricht behandelt.

Nach der Darstellung der gegenwärtigen Ausgangssituation betreffend den Einsatz von Robotersystemen zur Wissensvermittlung an österreichischen Schulen und der Auslegung des Begriffs Roboter soll eine Eingrenzung der Unterrichtsfächer, in denen der Einsatz der programmierbaren Maschinen sinnvoll erscheint, stattfinden. Dazu werden die Lehrpläne der österreichischen Schulen im Hinblick auf Unterrichtsinhalte untersucht, deren robotergestützte Vermittlung denkbar wäre. Auf Basis der Lehrpläne wird auch thematisiert, in welchen Schulstufen und somit auch in welchen Altersstufen der Schüler und Schülerinnen ein Einsatz von Robotern zur Wissensvermittlung umsetzbar wäre.

Anschließend folgt eine exemplarische Auflistung ausgewählter Roboter, deren Verwendung im Unterricht aufgrund unterschiedlicher Spezifikationen und der daraus resultierenden Einsatzmöglichkeiten in verschiedenen Unterrichtsfächern gewinnbringend für die Bildung von Schülern und Schülerinnen sein kann. Es wird dargestellt, welche Vor- und Nachteile die unterschiedlichen vorgestellten Systeme in verschiedenen Einsatzgebieten mit sich bringen.

Mögliche Kombinationen eines Roboters mit potentiell kompatiblen Programmiersprachen werden überblickshaft veranschaulicht. Vice versa kann auf diese Weise einerseits - ausgehend von den Lehrinhalten, welche zu vermitteln gewünscht werden - auf geeignete Programmiersprachen und somit auf passende Systeme geschlossen, und andererseits erkannt werden, für welche Einsatzgebiete sich ein als vielversprechend empfundener Roboter eignen kann.

2 Ziele der Arbeit

Aus dem vorangegangenen Kapitel lassen sich die Ziele dieser Arbeit ableiten, diese werden folgend kurz dargestellt:

- i. Die Abhandlung soll die vorhandenen fragmentarischen Informationen, betreffend den Robotereinsatz an Schulen, untereinander in Verbindung bringen.
- ii. Als Adressaten soll diese Arbeit sich auch an Lehrerinnen und Lehrer richten und diesen als Unterstützung für den Einsatz von Robotern zur Wissensvermittlung dienen.
- iii. Die aktuelle Situation bezüglich des Robotereinsatzes in österreichischen Schulen soll durch eine Umfrage erhoben und dargestellt werden.
- iv. Gründe, die für den Einsatz von Robotern im Schulunterricht sprechen, sollen aufgezeigt werden.
- v. Eine Abgrenzung der Fächer, die sich gut für eine inhaltliche Vermittlung durch robotergestützten Unterricht eignen, soll stattfinden und begründet werden.
- vi. Robotermodelle, die sich aufgrund verschiedener Eigenschaften für unterschiedliche schulische Einsätze eignen würden, sollen vorgestellt und verglichen werden.
- vii. Mit den vorgestellten Robotermodellen kompatible Programmiersprachen sollen bezüglich ihrer Einsatzmöglichkeiten erörtert werden.

3 Ausgangssituation

Dieses Kapitel befasst sich zunächst mit dem Begriff Roboter und dessen Abgrenzung. Im Anschluss wird die Frage nach der aktuellen Situation in Hinblick auf die Verwendung von Robotern in österreichischen Bildungseinrichtungen beleuchtet. Anhand der Daten einer Erhebung an 100 österreichischen Schulen wird die gegenwärtige Verbreitung der programmierbaren Automaten zur Stützung oder Widerlegung der These: „Roboter sind an österreichischen Schulen nur selten zur Wissensvermittlung in Verwendung“ dienen. Anschließend werden das Untersuchungsdesign dieser Erhebung sowie dessen Ergebnisse dargestellt.

3.1 Definition Roboter

Das ehemals für die Bezeichnung eines Arbeiters im Frondienst stehende Wort Roboter¹ lässt heute mehrere Interpretationen zu. Im deutschen Duden werden aktuell zwei Bedeutungen gefunden. Zum einen werden darunter Apparaturen verstanden, die der menschlichen Gestalt nachgebildet sind, um bestimmte Funktionen des Menschen zu übernehmen und auszuführen². Andererseits definiert das Wort heute auch einen Automaten, „...der ferngesteuert oder nach Sensorsignalen bzw. einprogrammierten Befehlsfolgen anstelle eines Menschen bestimmte mechanische Tätigkeiten verrichtet“³. Neben dieser allgemeinen Auslegung deutet die Richtlinie 2860 des VDI (Verein Deutscher Ingenieure) das Wort Industrieroboter wie folgt *„Industrieroboter sind universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d. h. ohne mechanischen Eingriff) programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen.“* [01 S 1].

Zusammenfassend können Roboter demnach als Maschinen bezeichnet werden, die in einem bestimmten Umfang mit ihrer Umwelt interagieren können und dies in Folge eines vordefinierten Programmablaufs und/oder anhand von Sensordaten tun. In Abbildung 1 soll schematisch eine Einteilungsmöglichkeit der Hauptelemente von Robotersystemen dargestellt werden.

¹ vgl. Duden, <http://www.duden.de/rechtschreibung/Roboter>, (27.01.2014)

² vgl. ebenda

³ vgl. ebenda

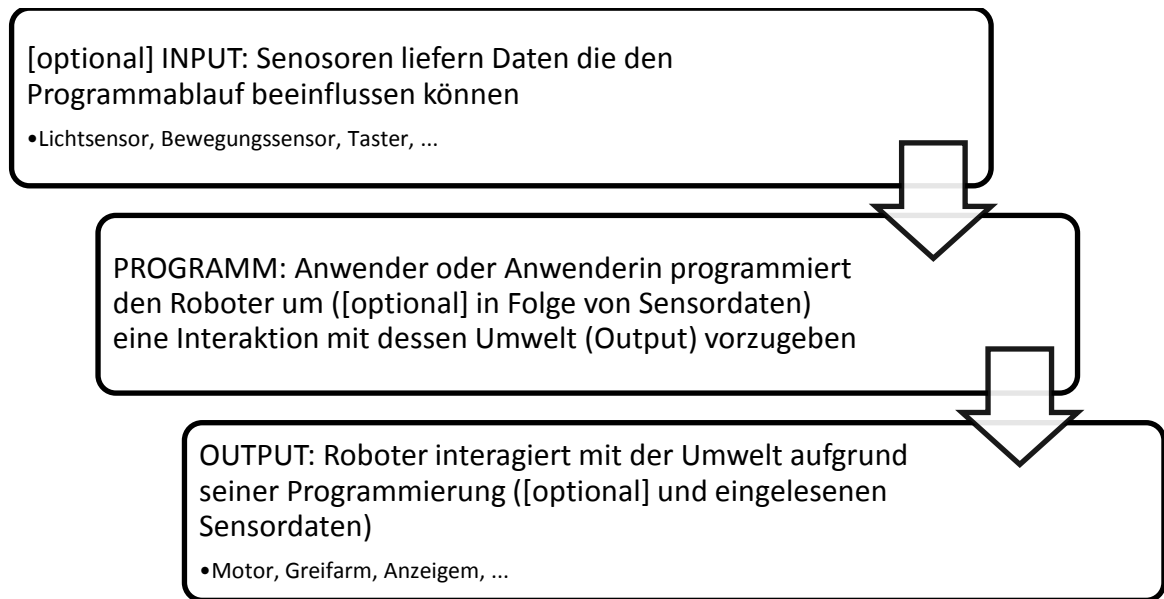


Abbildung 1 - Schematische Darstellung eines Roboters⁴

Demnach handelt es sich um ein System, das bestimmte Kriterien erfüllen muss, um als Roboter gesehen zu werden. Es muss mit seiner Umwelt interagieren können, also einen auf diese Einfluss nehmenden Output liefern. Diese Ausgabe kann zum Beispiel durch die Bewegung von mechanischen Teilen wie beispielsweise Motoren erfolgen. Welche Interaktion im Rahmen des Handlungsspielraums des Roboteroutputs tatsächlich erfolgt, wird im Laufe der Programmierung festgelegt. Auf den Programmablauf können eventuell auch äußere Faktoren Einfluss nehmen. Für eine derartige Einflussnahme würden Sensoren benötigt werden, um so einen äußeren Zustand zu messen und das Messergebnis an das Programm zu übermitteln. Die Teilaspekte, die das System Roboter erfüllen muss, lassen sich dementsprechend verschiedenen Disziplinen wie der Messtechnik oder der Softwareentwicklung zuordnen. Demzufolge könnten auch diese Disziplinen in verschiedenen Unterrichtsfächern zum Thema gemacht werden. Mit entsprechenden Zuordnungsmöglichkeiten setzt sich ein folgender Abschnitt dieser Arbeit im Detail auseinander.

3.2 Aufbau der Erhebung

Um die aktuelle Verbreitung von Robotern an österreichischen Schulen in Erfahrung zu bringen, wurde eine Umfrage durchgeführt. Auf ihren Aufbau wird in der Folge eingegangen. Die Erhebung erfolgte per E-Mail; im Fall von Unklarheiten, fehlender Rückmeldung oder zweideutigen Antworten wurden weitere Details telefongestützt

⁴ Eigene Darstellung

ermittelt. Die zugrunde liegenden Daten, auf deren Basis diese Umfrage und die damit in Verbindung stehenden Berechnungen erfolgten, wurden der Internetseite des Bundesministeriums für Unterricht, Kunst und Kultur⁵ sowie der Webseite der Statistik Austria⁶ entnommen.

Da sich diese Arbeit in Inhalt, Umfang und Komplexität an Lehrerinnen und Lehrer von allgemein bildenden höheren Schulen (AHS), neuen Mittelschulen (NMS) und höheren berufsbildenden Schulen (BHS) in Österreich richtet, ist die Bezugsgruppe der Untersuchung auch aus diesen Schultypen zusammengesetzt. Um ein repräsentatives Umfrageergebnis zu erhalten, wurde zunächst errechnet, wie die prozentuelle Verteilung der verschiedenen untersuchten Bildungsinstitutionen in Österreich ist, um auf Basis dessen errechnen zu können, wie viele Schulen jedes Typs befragt werden müssen, um diese prozentuelle Verteilung - bezogen auf die einhundert benötigten Schulen - beizubehalten. In Tabelle 1 ist die Berechnung der Umfrageverteilung dargestellt. Daraus resultierte, dass - auf der Grundlage der 1336 Schulen der Bezugsgruppe⁷ - eine Stichprobe von 100 Schulen derart zusammengesetzt sein muss, dass österreichweit 51 NMS, 26 AHS und 23 BHS darin vertreten sind.

Schulen in der Bezugsgruppe: 1 336		Schulen in der Umfrage: 100	
Gesamt je Typ:		Prozent:	Schulen:
NMS:	683	51,12	NMS: 51
AHS:	344	25,75	AHS: 26
BHS:	309	23,13	BHS: 23

Tabelle 1 – Schulen der Umfrage und ihre prozentuelle Verteilung

Um einen österreichweiten Durchschnittswert der befragten Schulen zu erhalten, wurde weiteres ermittelt, wie die prozentuelle Verteilung der untersuchten Schultypen je nach Bundesland ist, und auf Basis dieser Verteilung berechnet, wie viele Schulen jedes Schultyps pro Bundesland befragt werden müssen, um zu gewährleisten, dass das Umfrageergebnis die österreichweite Durchschnittsverteilung widerspiegelt. Diese Berechnung ist in der folgenden Tabelle 2 dargestellt.

⁵ Schuldatei online, <http://www.schulen-online.at/>, (27.01.2014)

⁶ Statistik Austria, <http://www.statistik.at/>, (20.02.2014)

⁷ Stand 16.01.2014

	NMS ges.	NMS% /BL	NMS Umfrage	AHS ges.	AHS% /BL	AHS Umfrage	BHS ges.	BHS% /BL	BHS Umfrage
Österreich:	683	100,00	51	344	100,00	26	309	100,00	23
Burgenland:	40	5,86	3	11	3,20	1	15	4,85	1
Kärnten:	48	7,03	4	23	6,69	2	24	7,77	2
Niederösterreich:	149	21,82	10	57	16,57	4	63	20,39	5
Oberösterreich:	123	18,01	9	48	13,95	4	54	17,48	4
Salzburg:	27	3,95	2	26	7,56	2	27	8,74	2
Steiermark:	107	15,67	8	48	13,95	4	43	13,92	3
Tirol:	74	10,83	6	25	7,27	2	31	10,03	2
Vorarlberg:	51	7,47	4	14	4,07	1	13	4,21	1
Wien	64	9,37	5	92	26,74	6	39	12,62	3
* Bundesland									

Tabelle 2 – Schulen der Umfrage nach Typ und Bundesland (ges. = gesamt, BL = Bundesland)

Die je Bundesland und Schultyp benötigte Anzahl an Schulen wurde dann zufällig aus dem jeweiligen Schulkontingent ausgewählt. Diese wurden anschließend per E-Mail befragt. Im Fokus dieser Befragung stand die Hauptfrage: „Werden in irgendeiner Form Roboter im Unterricht verwendet?“. Wurde diese beantwortet, so konnte das Ergebnis gewertet werden. Ist diese Frage mit ja beantwortet worden, wurde des Weiteren nach den Schulfächern und –stufen gefragt, in denen diese Anwendung finden, sowie nach den Robotermodellen und den verwendeten Programmiersprachen. Nach einer verneinenden Antwort wurde ergänzend gefragt, ob jemals über den Einsatz von Robotern nachgedacht wurde, und falls ja, was der Grund gegen eine Umsetzung war.

3.3 Ergebnis der Erhebung

Im Rahmen der vorgenommenen Erhebung wurden 100 österreichische Schulen (51 NMS, 26 AHS, 23 BHS) bezüglich des Einsatzes von Robotern im Unterricht befragt. Im Folgenden werden die Ergebnisse der Erhebung dargestellt und interpretiert.

Von den einhundert befragten Schulen gaben insgesamt nur 26 Prozent an, Roboter in irgendeiner Form im Rahmen des Normunterrichts zu verwenden. Von den 74 Prozent der Schulen, die Roboter nicht im Normunterricht verwenden, bieten allerdings etwa sieben Prozent einen robotikbezogenen Wahlgegenstand an. Etwa fünf Prozent der Schulen ohne Robotereinsatz im Pflichtunterricht haben bereits Erfahrung mit deren Verwendung, nutzen diese allerdings nicht (mehr) im normalen Unterricht. Drei Prozent der befragten Schulen gaben an, dass gegenwärtig eine Anschaffung von Robotern im Raum stehe. Die Ergebnisse sind in Diagramm 1 zusammengefasst.

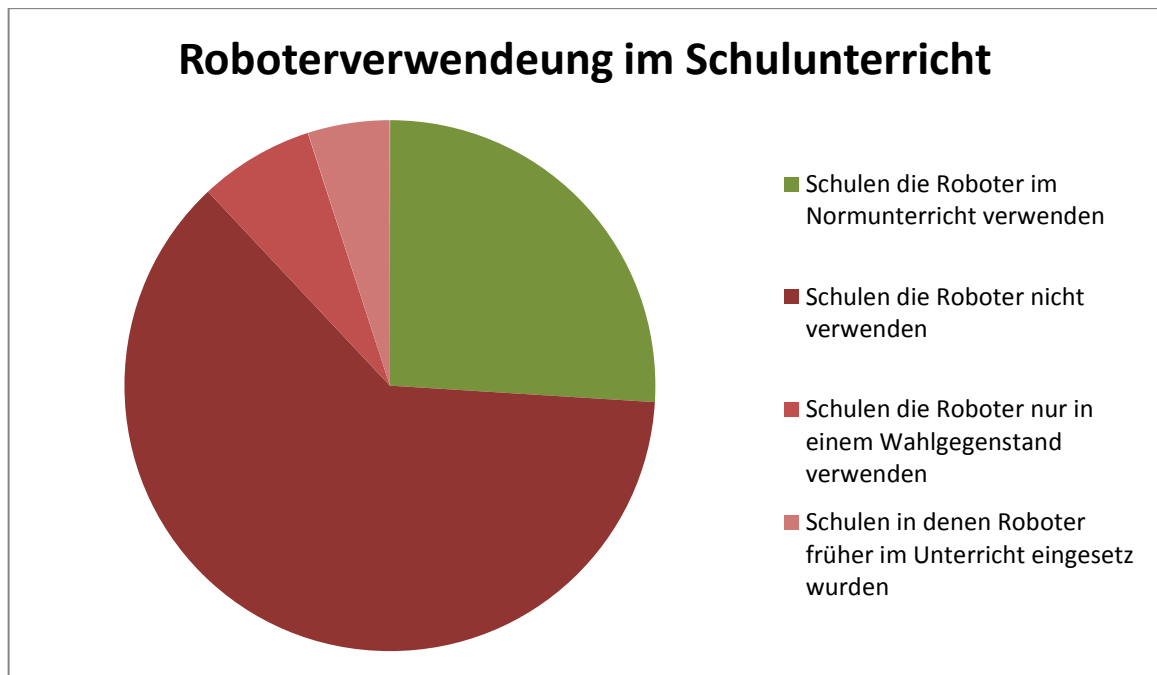


Diagramm 1 – Roboterverwendung im Schulunterricht

Hier ist bereits ersichtlich, dass der Einsatz von Robotern zur Wissensvermittlung an österreichischen Schulen ausgeprägter sein könnte. Nur etwa ein Viertel der Schulen bedient sich der programmierbaren Automaten im normalen Schulunterricht.

Bei der Analyse der Ergebnisse stellte sich heraus, dass der Prozentsatz der österreichischen Schulen, die Roboter einsetzen, je nach Schultyp sehr unterschiedlich ist. Anwendung im Unterricht finden Robotersysteme vor allem an höheren technischen Lehranstalten - etwa 70 Prozent dieser Schulen verwenden Roboter zur Wissensvermittlung. An allgemeinbildenden höheren Schulen finden die mechanischen Helfer nur zu etwa 27 Prozent Einsatz. Das Schlusslicht bilden die neuen Mittelschulen. An Schulen dieses Typs werden Robotersysteme nur zu etwa sechs Prozent eingesetzt. Im folgenden Diagramm 2 sind, jeweils farblich nach Schultypen getrennt, die Ergebnisse der einzelnen Schultypen veranschaulicht.

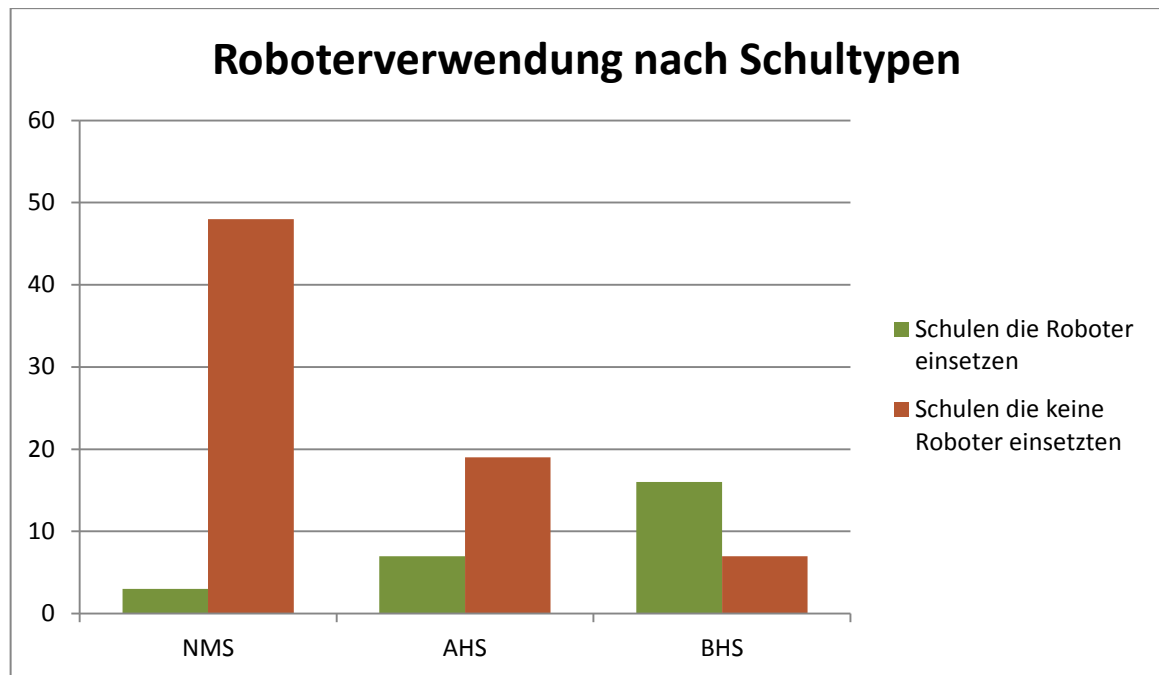


Diagramm 2 – Roboter Verwendung nach Schultypen

An den allgemeinbildenden Schulen (AHS, NMS), an denen Roboter verwendet werden, ist dies meist im Rahmen des Informatikunterrichts der Fall. Vereinzelt wurde auch angegeben, dass diese im Fach Physik Verwendung finden. Bezüglich der Schulstufen, in denen die programmierbaren Systeme eingesetzt werden, ließ sich lediglich eine Tendenz Richtung Oberstufe feststellen, allerdings wurden alle Schulstufen genannt. Die verwendeten Robotermodelle sind an Schulen dieses Typs beinahe immer LEGO-Mindstorms-Systeme (siehe Kapitel 7.3.1: „LEGO Mindstorms Education EV3-Basis-Set“), nur eine dieser Bildungseinrichtungen gab an, ein anderes Modell einzusetzen. Bei den verwendeten Programmiersprachen ließ sich keine klare Tendenz erkennen, sowohl unterschiedliche grafische als auch textuelle Sprachen (siehe Kapitel 8: „Programmiersprache als Kriterium zur Systemauswahl“) kommen diesbezüglich zur Anwendung.

An berufsbildenden Schulen, an denen verhältnismäßig oft Roboter verwendet werden, finden diese in verschiedensten Fächern Verwendung. Oft wurden in diesem Zusammenhang die Fächer Mechatronik, Elektrotechnik und Informatik genannt. Wie auch bei den allgemeinbildenden Schulformen konnte keine verstärkte Anwendung in bestimmten Schulstufen festgestellt werden. Im Gegensatz zu den an AHS und NMS eingesetzten Robotern finden an Schulen des Typs BHS hauptsächlich Industrieroboter und Eigenentwicklungen Anwendung. Hinsichtlich der

Institut für Softwaretechnologie

Programmiersprachen konnte keine Tendenz festgestellt werden, auffällig war lediglich, dass beinahe nur textuelle Sprachen eingesetzt werden.

Bei den Schulen, von denen die Rückmeldung kam, dass gegenwärtig keine Robotersysteme eingesetzt werden, dies aber gewünscht wäre, konnten die Gründe dafür in drei Kategorien eingeteilt werden. Vereinzelt Nennungen, die etwa ein Drittel der Rückmeldungen betreffen, wurden in der Kategorie „anderer Grund“ zusammengefasst. Etwa ein weiteres Drittel der befragten Schulvertreter gab an, dass für die Anschaffung von Robotersystemen das benötigte Budget fehlen würde. Ein weiterer Grund, der vom verbleibenden Drittel angegeben wurde, war der Mangel an Lehrpersonal, das einerseits Interesse daran hat, Roboter einzusetzen, und andererseits die benötigten Qualifikationen erfüllt. Die Verteilung dieser Begründungen soll in Diagramm 3 veranschaulicht werden.

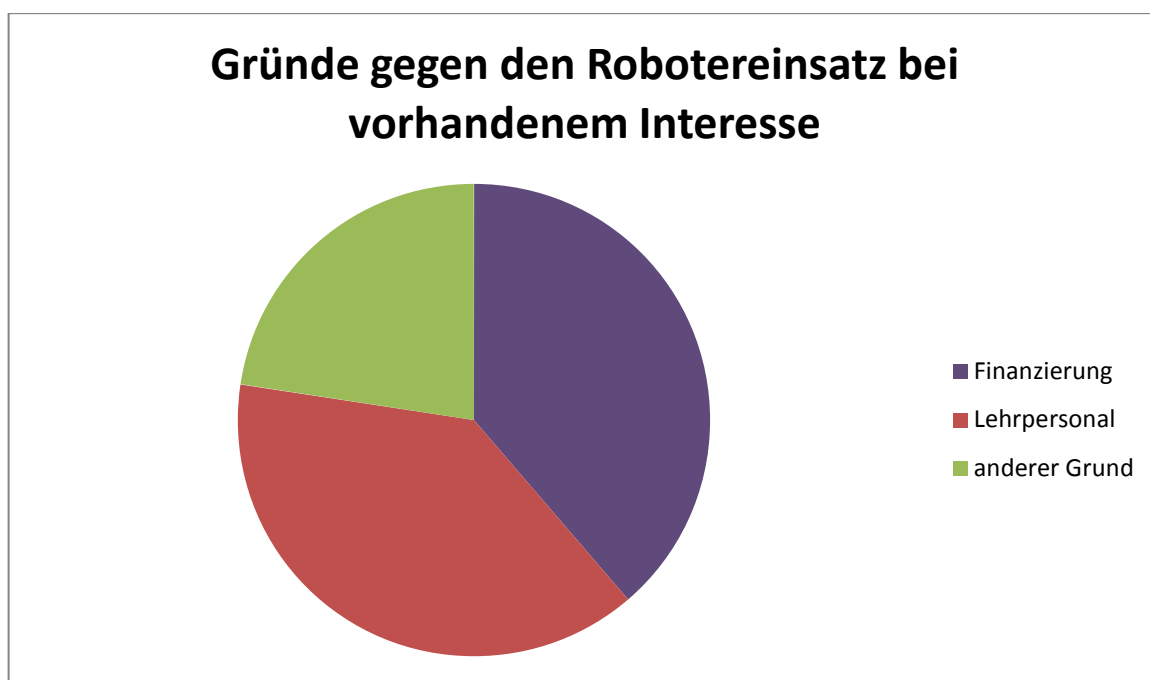


Diagramm 3 – Gründe gegen Robotereinsatz bei Interesse

3.4 Interpretation der Ergebnisse

Die Erhebung konnte die These „Roboter sind an österreichischen Schulen nur selten zur Wissensvermittlung in Verwendung“ zum Teil bestätigen, da nur an etwa einem Viertel der Schulen, die im Laufe der Erhebung befragt wurden, Robotersysteme verwendet werden. Nur im Bereich von neuen Mittelschulen kann die These als vollständig bestätigt angesehen werden. Hier kommen Roboter

Institut für Softwaretechnologie

besonders selten zum Einsatz. Bei den beiden anderen Schultypen könnte der Einsatz jedoch weitaus ausgeprägter sein. Etwa 70 Prozent der Befragten gab an, dass darüber noch nie nachgedacht wurde, die verbleibenden 30 Prozent scheiterten meist an fehlenden finanziellen Mitteln oder dem Mangel an passendem Personal.

Auffällig ist, dass zum Teil ein offensichtlicher Informationsmangel besteht. Es kam zu Rückmeldungen wie „Zu wenig Information!“ oder „Keine entsprechenden Materialien“. Weitere Angaben wie „In einer AHS nicht erforderlich“ oder „Zu wenig Lehrplanbezug“ lassen ebenfalls Rückschlüsse auf einen Informationsmangel zu. Dieser Informationsmangel stellt den Hauptgrund für die Verfassung der vorliegenden Arbeit dar. Im Vordergrund stehen besonders Gründe für den Einsatz an Schulen, vor allem auch an allgemeinbildenden Institutionen (siehe Kapitel 4: „Gründe für den Einsatz von Robotik im Unterricht“), sowie entsprechende Lehrplaninhalte, die sich besonders für den Einsatz von Robotersystemen zur Wissensvermittlung eignen.

Es konnte auch festgestellt werden, dass eingesetzte Roboter beinahe nur im Informatikunterricht Verwendung finden; dies könnte ebenfalls an fehlenden Informationen liegen. Die Lehrplanbezüge sind nicht immer offensichtlich. Um einen erweiterten Einsatz von Robotern in anderen Schulgegenständen und besonders auch im Rahmen eines fächerübergreifenden Unterrichts zu unterstützen, wird versucht, ein möglichst breites Feld an lehrplanbezogenen Verwendungsmöglichkeiten zu veranschaulichen.

Bezüglich der meist genannten Gründe, die gegen eine angedachte Unterrichtsintegration von Robotern sprechen (Mangel an qualifiziertem Lehrpersonal und finanziellen Mitteln), soll diese Arbeit durch Untermauerung der Sinnhaftigkeit der Verwendung entsprechender Systeme auch als Argumentationsgrundlage für eine entsprechende Finanzierung dienen. Auch Lehrpersonal mit (zumindest scheinbar) fehlender Qualifikation soll durch die Erklärung einfacher Robotersysteme und grafischer Programmiersprachen geholfen werden zu erkennen, dass nicht zwingend Vorkenntnisse benötigt werden.

4 Gründe für den Einsatz von Robotik im Unterricht

Dieses Kapitel der Arbeit befasst sich mit der grundlegenden Frage, warum gerade Roboter im Unterricht zur Vermittlung von Lerninhalten in Schulen geeignet sein sollen und welche Vorteile ihr Einsatz gegenüber einem roboterfreien Unterricht haben kann. Neben dem Aspekt der Wissensvermittlung werden auch weitere Bereiche des Schulunterrichts, die durch den Robotereinsatz beeinflusst werden könnten, thematisiert.

Um diese Fragen beantworten zu können, wird zunächst die Frage nach dem Sinn des Schulunterrichts an sich gestellt. Per Definition ist die Schule eine „*Lehranstalt, in der Kindern und Jugendlichen durch planmäßigen Unterricht Wissen und Bildung vermittelt werden*“⁸ soll. Umfassender schrieb der deutsche Informationswissenschaftler Harald H. Zimmermann über die Rolle der Schule: „*Ihre Aufgabe ist es (wenn auch nicht notwendig die einzige), Grundlagenwissen zu vermitteln, möglichst in Richtung auf mehr Wissen und weniger Unsicherheit.*“ [02]. Folglich soll Schule, zumindest zum Teil, (Grundlagen-)Wissen vermitteln. Neben dieser Hauptaufgabe nimmt die Bildungseinrichtung Schule auch weitere Funktionen wahr. In einem weiter gefassten Rahmen schreibt der Erziehungswissenschaftler der Georg-August-Universität Göttingen, Hermann Giesecke, „*Aufgabe der Schule ist ... mit ihren besonderen Möglichkeiten – nämlich denen des Unterrichts im weitesten Sinne – jedem Kind die Chance zu geben, seine Fähigkeiten in optimalem Maße zu entfalten, damit es in einer Gesellschaft voller Optionen eine individuell befriedigende Balance zwischen objektiven Anforderungen und subjektiven Bestrebungen finden und darauf seine persönliche Lebensplanung, z. B. in beruflicher Hinsicht, gründen kann.*“ [03 S 97]. Für die Berufswahl spielt die Institution Schule nach wie vor eine wichtige Rolle, da diese die Berufsorientierung und Identitätsbildung von Kindern und Jugendlichen ermöglicht oder begrenzt [vgl. 04 S 7]. Neben der Funktion als wissensvermittelnde Institution nimmt der Schulalltag demnach auch Einfluss auf den Berufsweg junger Menschen.

Da sich der Ingenieursbedarf in Europa im Laufe der letzten sechs Jahren fast verdreifacht hat, die Anzahl an Hochschulabsolventen in technischen Bereichen hingegen zurückgegangen ist, wäre es wünschenswert, das Technikinteresse von

⁸ vgl. Duden, <http://www.duden.de/rechtschreibung/Schule>, (27.01.2014)

Schülerinnen und Schülern zu steigern [vgl. 05 S 1]. Neben dem Hauptziel von Schulen - Wissen und Bildung zu vermitteln - sollen diese dementsprechend auch möglichst das Interesse an technischen Berufen wecken. Aus diesem Grund gibt es inzwischen Förderungsprogramme wie „Sparkling Science“ des BMWF⁹, ein MINT¹⁰- und Nachwuchsförderungsprogramm mit dem Ziel, das Interesse von Jugendlichen am Thema Forschung zu fördern. Durch dieses und ähnliche Programme, wie der Kinderuniversität, wird angestrebt, nachhaltig wirksame Impulse für die spätere Ausbildungs- und Berufswahl der jungen Teilnehmer/innen zu geben¹¹.

Aus der Darstellung grundlegender Funktionen von Schulen leiten sich neue Fragestellungen ab. Zunächst gilt es zu beleuchten, welche Faktoren den schulischen Lernprozess von Kindern und Jugendlichen beeinflussen, sowie herauszufinden, ob der Einsatz von Robotik das technische Interesse von Schülerinnen und Schülern weckt. Die folgenden Kapitel versuchen nicht nur diese Fragestellungen zu beantworten, sondern auch auf weitere positive Aspekte, die durch den Einsatz von Robotern an Schulen zu erwarten sind, einzugehen.

4.1 Einsatz von Robotern an Schulen zur Steigerung der Motivation

„Verantwortlich für den Ablauf des Lernprozesses sind kognitive Komponenten und das Motivationsniveau, die wiederum von einer metakognitiven Kontrolle geleitet werden. Daraus wird unter anderem gefolgert, dass nur durch eine permanent vorhandene Motivation der Lernprozess aufrechterhalten werden kann.“ [06 S 24]. Motivation kann dementsprechend als eine der Schlüsselkomponenten für den erfolgreichen Ablauf eines Lernprozesses gesehen werden.

Auch wenn vermutet werden kann, dass es weitere, im Rahmen dieser Arbeit nicht behandelte Faktoren gibt, die den Lernprozess beeinflussen, so lässt sich die obige Grundaussage unter Einbeziehung weiterer Fachliteratur bestätigen. Steffi Zander, Mitarbeiterin der Forschungsgruppe „Lernen und neue Medien“ der Universität Erfurt, äußerte sich diesbezüglich wie folgt: *„Die Motivation Lernender wirkt sich ... folgendermaßen auf kognitive Belastung aus: Die Regulation der*

⁹ Bundesministerium für Wissenschaft und Bildung

¹⁰ MINT: Zusammenfassender Begriff für die (Unterrichts-) Fächer der Bereiche: Mathematik, Informatik, Naturwissenschaften und Technik

¹¹ vgl. BMWF Sparkling Science/Kinderuniversitäten, <http://www.bmwf.gv.at/startseite/das-ministerium/gender-und-diversitaet/programme-und-initiativen/mint-und-nachwuchsfoerderung/> (29.01.2014)

Aufmerksamkeitsprozesse in Anpassung an die Aufgabenanforderungen resultiert in höherer kognitiver Belastung der hoch motivierten Lernenden (im Vergleich zu niedriger motivierten) unter hoher Anforderungen. Es kann also mehr Informationsverarbeitung in derselben Zeit stattfinden (Salomon 1983). Die erhöhte Bereitstellung und Intensität der Nutzung von Aufmerksamkeits- und Anstrengungsressourcen wirkt sich dann positiv auf den Lernerfolg aus. Hoch motivierte haben höheren Lernerfolg, sowohl unter den moderaten als auch unter den schwierigen Bedingungen.“ [07 S 87]. Basierend auf diesen Grundlagen findet des Weiteren eine Auseinandersetzung mit Einflussfaktoren auf die Motivation und die Motivationssteigerung von Schülerinnen und Schülern statt.

Bereits seit den sechziger Jahren, als Seymour Papert sich in Zusammenarbeit mit Jean Piaget in Genf dem Thema „Kinder und Computer“ widmete, gibt es die ersten Überlegungen für einen Robotereinsatz zu Bildungszwecken¹² [vgl. 08]. Heute nehmen Fachleute wie Peter Protzel, der Roboter-Experte der Universität Chemnitz, an, dass Roboter zur Vermittlung schulischen Wissens eingesetzt werden könnten. Durch ihren Einsatz soll vor allem auch die Motivation, sich mit den Lehrinhalten auseinanderzusetzen, gesteigert werden: *“Studies suggest that certain teaching strategies may foster the STEM¹³ participation and achievement and give some evidence that using scientific equipment and hands-on activities are related to higher science and mathematics achievement“* [05]. Ähnlich äußerte sich in diesem Kontext auch Peter Protzel, in einem Interview: *„Die Arbeit am PC vermittelt entweder das Bedienen fertiger Programme wie z. B. der Textverarbeitung oder abstrakte Informatikkenntnisse. Beides ist wichtig, ersetzt aber nicht die greifbaren Erfahrungen, die man beim Umgang mit einem komplexen technischen System wie einem Roboter gewinnt. Auch das Feedback beim Lernen unterscheidet sich: Auf dem Bildschirm des PCs steht ein falsches Ergebnis, der Roboter fährt gegen die Wand oder über die Tischkante. Letzteres führt zu einem höheren emotionalen Engagement und fördert die Motivation und damit den Lernerfolg. Wenn der eigene Roboter losfährt, fiebert man mehr mit, als wenn Zahlenkolonnen über den Bildschirm laufen.“* [09].

¹² vgl. Seymour Papert, <http://www.papert.org/>, (08.01.2014)

¹³ STEM ist das englische Pendant zur deutschen Abkürzung MINT

Infolge der, mit der Unterrichtsintegration von Robotern einhergehenden, Motivationssteigerung sollten Lernprozesse dementsprechend effektiver ablaufen. Der Einzug der programmierbaren Automaten in den Alltag von Schülerinnen und Schülern weckt folglich nicht nur deren Begeisterung, sondern führt durch den Kompetenzausbau im Zuge der praktischen Anwendung auch zu einer höheren Leistungsfähigkeit.

Neben dem Anstieg der Lernleistung von Schülerinnen und Schülern durch die Erhöhung der Motivation führt also auch die Verwendung von Robotern an sich durch die praktische Auseinandersetzung mit ihnen zum gewünschten Ziel.

4.2 Schule, Roboter und die Berufswahl

Im Folgenden wird auf den bereits thematisierten Zusammenhang zwischen Schule und der Berufswahl von Schülerinnen und Schülern näher eingegangen und beleuchtet, welchen positiven Einfluss Roboter darauf nehmen können.

Da es aufgrund des hohen Ingenieursbedarfs und der geringen Anzahl an Absolventen und Absolventinnen im technischen Bereich erstrebenswert ist, das Technikinteresse von Kindern und Jugendlichen zu fördern, wird auf die Sinnhaftigkeit des dahingehenden, unterrichtsbezogenen Robotereinsatzes eingegangen.

Diesbezüglich antwortete der Leiter des Naturwissenschafts- und Technikdidaktik-Zentrums der Fachhochschule Nordwestschweiz, Peter Labudde, auf die Frage, was man tun kann, damit sich Jugendliche für MINT-Studien entscheiden: *„Der Unterricht sollte einen stärkeren Alltagsbezug herstellen und sich stärker an Fragen und Interessen der Schüler und Schülerinnen ausrichten. Und das selbstständige, entdeckende Lernen sollte auf allen Schulstufen verstärkt werden.“*¹⁴ Durch die zunehmende Integration von Robotern in den menschlichen Alltag und das Interesse von Schülern wie Schülerinnen, das durch den Robotereinsatz gesteigert werden kann, ist denkbar, dass ihre Verwendung, aus oben genannten Gründen auch zu einer Erhöhung der Anzahl an MINT-Studieninteressierten führt. Durch das direkte Feedback das Kinder und Jugendliche bei der Nutzung eines Roboters bekommen, können diese Systeme auch das selbständige Lernen fördern beziehungsweise erleichtern.

¹⁴ Interview Peter Labudde mit science.orf.at, <http://science.orf.at/stories/1703048/> (28.01.2014)

Wegen des geringen Anteils an (vor allem auch) Frauen in technischen Studienfächern wurde im Rahmen des Roberta Projektes untersucht, ob der Einsatz von Robotern das Interesse speziell von weiblichen Personen an technischen Ressorts steigern kann. *„Ziel des Vorhabens war, das Interesse von Mädchen und Frauen für Informatik und Naturwissenschaften zu wecken und Verständnis für technische Systeme zu fördern, sowie der digitalen Spaltung zwischen Mädchen und Jungen (Frauen und Männern) entgegenzuwirken. Dafür sollten Roboterkurse als Teil des Bildungsangebots etablieren werden.“* [10 S 16].

Durch das von 2002 bis 2007 laufende Projekt, in dessen Rahmen Begleitforschung zur Bestätigung der Ergebnisse von Kursen für Schülerinnen und Schüler betrieben wurde, ließen sich vier Faktoren des Erlebens der Teilnehmerinnen indizieren: Freude, Inhalt, Lernumgebung und Lernerfolg [vgl. 10 S 30].

„Die Roberta Kurse schienen für die Entwicklung einer positiveren Einstellung der Kinder gegenüber der Informatik – sowohl in der Einschätzung der eigenen Fähigkeiten als auch in der beruflichen Orientierung - geeignet zu sein.“ [10 S 30].
„Die Ergebnisse des Projekts „Roberta – Mädchen erobern Roboter“ hatten gezeigt, dass Robotik sehr gut geeignet ist, bei Mädchen Interesse für Naturwissenschaften, Technik und Informatik zu wecken und ihr Selbstvertrauen in ihre diesbezüglichen Fähigkeiten zu stärken.“ [10 S 54].

Auf Basis dieser und vorangegangener Ergebnisse kann bezüglich des beruflichen Interesses von Kindern und Jugendlichen in MINT-Bereichen einerseits bestätigt werden, dass dieses durch die Verwendung von Robotern steigt, und andererseits, dass dies für beide Geschlechter gilt.

Zusammenfassend kann gesagt werden, dass der Schulunterricht einen wesentlicher Einflussfaktor für die Berufswahl von Schülerinnen und Schülern darstellt. Infolge des Techniker- und Technikerinnenmangels gilt es, das Interesse an technischen Studien und Berufen zu stärken. Diese Interessenssteigerung kann aus den oben ausgeführten Gründen durch den Einsatz von Robotern zur Wissensvermittlung im Unterricht gelingen.

4.3 Projektorientiertes Lernen

Im Fall des projektorientierten Lernens handelt es sich um ein Unterrichtskonzept, das entwickelt wurde, um das Engagement von Schülerinnen und Schülern bei der Behandlung authentischer Probleme zu steigern [vgl. 11 S 369]. Da diese Konfrontation mit eben diesen authentischen Inhalten aus oben genannten Gründen zu einer Motivationssteigerung sowie einem höheren Lernerfolg führen kann, wird dieses Konzept im Folgenden kurz behandelt.

Diese Lehrmethode basiert auf modernen Lerntheorien, in denen argumentiert wird, dass Wissen, Denken und Handeln untrennbar miteinander verbunden sind. Projektorientiertes Lernen ist eine Unterrichtsmethode, die auch darauf abzielt, Distanz zu einem isolierten und um den Lehrer strukturierten Unterricht zu gewinnen. Durch das Anwenden von projektorientiertem Lernen ist es möglich, interdisziplinär zu unterrichten und im Rahmen schülerzentrierten Unterrichts reale und alltagsnahe Themen in den Vordergrund zu stellen. Es unterstützt das Erstellen von Querverweisen zwischen dem Unterrichtsinhalt und der unterrichtsexternen Welt und führt somit zu ersichtlicherem Sinn, der hinter dem Erlernen geforderter Schulinhalte steht. Des Weiteren unterstützt projektorientiertes Lernen die Fähigkeit zu Teamwork, das Ergreifen der Initiative und das Lösen komplexer Probleme [vgl. 12 S 104].

In der Umsetzung von projektorientiertem Lernen kann die Verwendung von Technologien eine zentrale Rolle spielen, um motivierend auf die Implementierung von entsprechenden Unterrichtsprojekten zu wirken. Der Technikeinsatz kann auch dazu beitragen, das Interesse von Schülerinnen und Schülern am Projekt zu steigern, vor allem wenn es sich dabei um eine Auseinandersetzung mit realistischen und anspruchsvollen Inhalten handelt [vgl. 11 S 384].

Für den Einsatz als Medium bei der Umsetzung der Bildungsmethode projektorientierten Lernens eignen sich Roboter aus mehreren Gründen gut. Sie bieten ein breites Feld an Anwendungsoptionen, veranschaulichen den realen Nutzen des Lernprojekts und geben vor allem auch ein Feedback an die User zurück, anhand dessen die Korrektheit der Projektumsetzung ersichtlich ist. Der Umstand, dass dies zu einem höheren emotionalen Engagement führen kann und

dadurch auch der Lernerfolg, als Konsequenz der geförderten Motivation, steigt, wurde im vorangegangenen Kapitel „Einsatz von Robotern an Schulen zur Steigerung der Motivation“ bereits behandelt.

4.4 Fazit

In den obigen Kapiteln (4.1 bis 4.3) wurden unterschiedliche Vorteile, die mit dem Robotereinsatz an Schulen einhergehen, behandelt. Es folgt eine zusammenfassende Darstellung dieser positiven Aspekte. Diese Übersicht können Lehrerinnen und Lehrer als Argumentationsgrundlage verwenden, um den Nutzen der Anschaffung von Lernrobotern fundamentierte zu begründen.

Aus den Inhalten der vorangegangenen Kapitel kann geschlussfolgert werden, dass der Einsatz von Robotern in Schulen Einfluss auf diverse Aspekte der Wissensvermittlung nimmt. Besonders die Effizienz des Lernens, das Erwägen einer technischen Karriere sowie die Freude an der Weiterbildung können positiv beeinflusst werden.

Auf die Frage nach den Gründen für den Einsatz von Robotern im Schulunterricht – welche diesem Kapitel zu Grunde liegt – können folgende Aspekte argumentiert werden:

- i. Roboter steigern die Motivation von Schülerinnen und Schülern im Unterricht
 - a. Permanente Motivation ist die Grundlage zur Aufrechterhaltung von Lernprozessen
 - b. Eine Steigerung der Motivation führt zu einem höheren Lernerfolg
- ii. Roboter erhöhen die Leistung von Schülerinnen und Schülern durch die Praxisnähe
- iii. Roboter eignen sich gut für den Einsatz im projektorientierten Lernen
 - a. Erhöhung des Engagement von Schülerinnen und Schülern
 - b. Herstellung realer Bezüge, Steigerung der Motivation
- iv. Roboter eignen sich dazu, das Interesse an Naturwissenschaften, Technik und Informatik zu erhöhen (vor allem auch von Mädchen und Frauen). Durch die Interessenssteigerung in den MINT-Unterrichtsfächern kann eine Interessenssteigerung an technischen Berufen gefolgert werden

Dementsprechend gehen mit der Verwendung von Robotern im Schulalltag viele positive Auswirkungen auf Jugendliche und Kinder einher. Auf dieser Basis und nach Beantwortung des „Warum?“ setzt sich das folgende Kapitel mit den möglichen Unterrichtsfächern für den Einsatz von Robotern auseinander.

5 Abgrenzung der Einsatzmöglichkeiten von Robotern im schulischen Alltag

Nachdem verschiedene Gründe für die Verwendung von Robotern im Unterricht im vorangegangenen Abschnitt begründet werden konnten, befasst sich das aktuelle Kapitel mit einer Abgrenzung von Unterrichtsfächern, in denen diese Verwendung gewinnbringend sein kann.

Grundsätzlich können zwei schulische Haupteinsatzgebiete der Robotik anhand des zu vermittelnden Inhaltes unterschieden werden. Kategorisiert werden kann einerseits „Lernen über Robotik“ als ein Teilgebiet, das sich mit dem Aufbau und der Programmierung von Robotern befasst. In diesem Fall ist der Unterrichtsgegenstand der verwendete Roboter selbst. Dieses Teilgebiet kann daher eher zur Vermittlung von technischen Inhalten, wie dem Programmieren oder dem Aufbau der mechanischen Helfer, eingesetzt werden.

Das zweite Haupteinsatzgebiet kann unter dem Begriff „Lernen durch Robotik“ zusammengefasst werden. In diesem Teilgebiet dient die Verwendung von Robotern nicht primär der Erlernung von robotikbezogenen Lerninhalten. Vielmehr sollen in diesem Fall Roboter als unterrichtsunterstützendes Material dienen, um Lerninhalte ohne direkten Roboterbezug, wie beispielsweise mechanische Bewegungsabläufe, anschaulicher darzustellen und die Motivation der Schülerinnen und Schüler zu steigern.

Anhand der unterrichteten Fächer an allgemeinbildenden und berufsbildenden höheren Schulen wird nun erörtert, welches der beiden Haupteinsatzgebiete sich für den jeweiligen Schultyp beziehungsweise dessen Fächer eignet.

5.1 Lernen über Robotik

Abhängig von den angebotenen Fachrichtungen von höheren technischen Lehranstalten (HTL) lassen sich an diesen technischen Schulen logische Verwendungszwecke in verschiedenen Zweigen identifizieren. Beispielsweise sind im Bereich der Elektrotechnik an der HTL Bulme Graz Roboter und deren Programmierung grundlegende Kompetenzen, die unter anderem als Schwerpunkte

der Fachrichtungen erlernt werden sollen¹⁵. Ergänzend dazu kann auch die Ausbildungsrichtung Fahrzeugtechnik, die an mehreren HTLs in Österreich angeboten wird, genannt werden. Einerseits stellt die Automobilindustrie den Hauptabnehmer von industriellen Robotern weltweit dar¹⁶, andererseits wird gegenwärtig verstärkt Forschung im Sektor autonomes Fahren von Fortbewegungsmitteln im Alltagsverkehr betrieben [vgl. 13]. Die Nachvollziehbarkeit der Einsatzmöglichkeiten der programmierbaren Maschinen an HTLs resultiert vor allem daraus, dass es sich hier um das Teilgebiet „Lernen über Robotik“ handelt. Dadurch dass der Roboter selbst im Unterrichtsmittelpunkt steht, lässt sich der sinnhafte Einsatz ableiten.

Anders verhält sich dies in Bezug auf Unterrichtsfächer an allgemeinbildenden höheren Schulen, Realschulen und ähnliche Schulformen. Das Teilgebiet „Lernen über Robotik“ ist an Schulen dieses Typs meist nur im Rahmen des Informatikunterrichts integrierbar, da dieser einen direkten Bezug zulässt. Im Bereich der Informatik, vor allem im Wahlpflichtfachumfang, sind die Einsatzmöglichkeiten von Robotern infolge ihrer Programmierbarkeit naheliegend. Dies kann durch den zu vermittelnden Lehrstoff des Wahlpflichtfachs Informatik, an allgemein bildenden höheren Schulen, durch Lehrplaninhalte wie „Konzepte von Programmiersprachen“ [14 S 80] und „künstliche Intelligenz“ [14 S 80] bestätigt werden.

Da in den weniger oder nicht-technischen Fächern der direkte Roboterbezug kaum gegeben ist, um „Lernen über Robotik“ gewinnbringend in den Unterricht zu integrieren, stellt sich die Frage, in welchen Schulfächern „Lernen durch Robotik“ einen positiven Einfluss auf die Schulbildung ausüben kann.

5.2 Lernen durch Robotik

Die Analyse der Lehrpläne verschiedener Schulformen der Unter- und Oberstufe an österreichischen Schulen lässt den Schluss zu, dass die Verwendung von Robotern zur Wissensvermittlung in vielen Schulfächern möglich wäre. Allerdings gibt es Fächer, die sich für den Einsatz der Maschinen besser eignen als andere.

¹⁵ vgl. HTL Bulme Graz – Schwerpunkte Elektrotechnik, <http://www.neu.bulme.at/bulme258/index.php/ausbildung/fachricht/elektrotechnik/schwerpunkteelektrotechnik>, (30.01.2014)

¹⁶ vgl. International Federation of Robotics, <http://www.ifr.org/industrial-robots/statistics/>, (07.01.2014)

Welchen vorteilhaften Aspekt die Verwendung von Robotern zum Beispiel im Religionsunterricht bringt, scheint schwerer verständlich. Laut Lehrplan für den katholischen Religionsunterricht wären die programmierbaren Helfer aber durchaus thematisierbar, um Schülern wie Schülerinnen die Kompetenz „*Reflektieren von Wertvorstellungen und ethischen Fragen im Zusammenhang mit Natur und Technik*“ [15 S 3592] näher zu bringen. Im Folgenden werden dennoch Schulfächer mit einem direkteren Zugang im Vordergrund stehen. Dabei werden MINT-Fächer näher betrachtet. Diese umfassen Mathematik, Informatik, Biologie und Umweltkunde, Chemie und Physik.

Informatik ist das einzige der fokussierten Fächer, das nur in der Oberstufe und nur für ein Jahr eine feste Lehrplanverankerung hat. Allerdings kann Informatik an vielen Schulen auch als Wahlpflichtgegenstand gewählt werden. An einigen österreichischen Bildungsinstituten werden eigene Schulzweige mit Informatik als Schwerpunkt angeboten, in diesem Fall ist die Informatik meist als alternativer Pflichtgegenstand zu betrachten. Obwohl der Teilaspekt Softwareentwicklung im Lehrplan der Informatik keine spezielle Erwähnung findet, bietet sich die Robotik an, um Lehrplaninhalte wie „*Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern kennen lernen...*“ [14 S 71] zu verdeutlichen.

Durch den Einsatz der programmierten Maschinen könnten in der Physik zum Beispiel Stromkreise und elektrische Schaltungen erklärt werden. Anhand von Sensoren wäre es möglich, ablaufende Prozesse beim Drücken eines Tasters zu veranschaulichen. Ihr Einsatz bietet den Schülern auch einen möglichen Zugang zur Funktionsweise elektrischer Messgeräte. Der Elektromotor, der im Lehrplan der Physik ebenfalls Erwähnung findet, ist Bestandteil einer Vielzahl von Robotern. Diese können dementsprechend als Anschauungsmaterial dienen.

Bezüglich der Mathematik lassen sich im Lehrplan besonders viele Inhalte finden, zu denen die Verwendung von programmierbaren Robotern neben einem rein theoretischen auch einen praktischen, transparenten Zugang liefern könnte.

Um die Einsatzmöglichkeiten in diesen näher betrachteten Unterrichtsfächern herauszufiltern und eine weitere Abgrenzung vorzunehmen, befasst sich das folgende Kapitel mit den Lehrplänen der MINT-Unterrichtsfächer.

6 Abgrenzung möglicher Einsatzgebiete von Robotern durch Lehrplaninhalte

In diesem Kapitel wird versucht, die Einsatzmöglichkeiten der programmierbaren Maschinen im Unterricht anhand der Lehrpläne abzugrenzen und zu begründen. Dafür wurden die Lehrpläne der Pflichtfächer Mathematik, Informatik, Biologie und Umweltkunde, Chemie und Physik nach relevanten Inhalten untersucht. Es konnte festgestellt werden, dass die Lehrpläne der Unterrichtsfächer Chemie sowie Biologie und Umweltkunde kaum Lehrplaninhalte aufwiesen, von denen ein direkter, gewinnbringender Robotereinsatz zur Wissensvermittlung ableitbar ist [vgl. 14]. Durch den Lehrern und Lehrerinnen gegebenen Handlungsspielraum in der Gestaltung ihres Unterrichts ist es jedoch grundsätzlich möglich, ausgewählte Inhalte vieler, hier nicht speziell erwähnter Fächer, durch einen demonstrativen Robotereinsatz zu verdeutlichen. Im Laufe des Roberta-Projektes wurden zum Beispiel auch Analogien zum Schulfach Biologie und Umweltkunde hergestellt. Diese befassten sich mit dem Feuerkäfer und einem passenden Robotikexperiment, in dem Temperaturfühler zum Einsatz kamen [10 S 18].

Da im Rahmen dieser Arbeit aber Schulfächer im Vordergrund stehen sollen, bei denen auch ohne flexible Lehrplaninterpretation der Nutzen von Roboterverwendung im Unterricht ersichtlich ist, wird im Folgenden nur auf die Lehrpläne der Schulfächer Informatik, Mathematik und Physik im Detail eingegangen.

Neben der Abgrenzung von den Lehrplaninhalten der entsprechenden Schulfächer für den jeweils vorteilhaften Einsatz von Robotern soll auch untersucht werden, ob es eine altersbezogene Abgrenzungsmöglichkeit für deren Verwendung auf Basis der Lehrpläne gibt. Auch wenn die Vermittlung gewisser robotikbezogener Inhalte in einem Alter sinnvoll(er) wäre, das nicht der Schulstufe des entsprechenden Lehrstoffs laut Lehrplan entspricht, so wäre diese im Rahmen des Schulunterrichts nicht möglich, da dieser größtenteils an bestimmte Klassen und somit bestimmte Altersstufen der Schüler gebunden ist.

6.1 Lehrplanbezüge aus dem Unterrichtsfach Physik

Im Folgenden werden die Lehrplanabschnitte der Unter- und Oberstufe des Schulfachs Physik gelistet und abgehandelt. Die Auszüge, die auf einen Nutzen

durch die Verwendung von Robotern schließen lassen, werden mit Verwendungsmöglichkeiten in Verbindung gebracht.

6.1.1 2. Klasse, „Ausgehend von unterschiedlichsten Bewegungsabläufen im Alltag ... in der Technik sollen die Schülerinnen und Schüler ein immer tiefergehendes Verständnis der Bewegungsmöglichkeiten, der Bewegungsursachen und der Bewegungshemmungen von ... unbelebten Körpern ihrer täglichen Erfahrungswelt ... gewinnen.“ [16 S 1207]

Bewegungsabläufe im technischen Bereich werden meist von Robotersystemen ausgeführt. Eine Maschine interagiert infolge eines Eingabewertes oder eines bestimmten Programmes durch Bewegung mit der Umwelt. Es beginnt bei einfachen Robotern wie ferngesteuerten Autos, bei denen der Benutzer über die Sensoren der Fernsteuerung, die Steuerungstasten, einem Micro Controller im Auto vermittelt, welche Motorbewegung ausgeführt werden soll. Zur Veranschaulichung könnte zum Beispiel auch ein Roboterarm dienen, der nach dem Aktivieren auch ohne weitere Eingaben - einem bestimmten Programm folgend - einen Bewegungsablauf, zum Beispiel eine Greifbewegung, ausführt.

Für die Demonstration von „*Bewegungsmöglichkeiten, der Bewegungsursachen und der Bewegungshemmungen von ... unbelebten Körpern*“ [16 S 1207] eignen sich Roboter sehr gut, besonders die Modelle, die ein breites Spektrum an Bewegungsmöglichkeiten abdecken. Dies sind besonders Robotertypen, deren Aufbau und Funktion verändert werden kann. Es gibt beispielsweise Roboter, die sich je nach Anordnung einzelner Elemente auf Rädern, spinnenähnlichen Beinanordnungen, Ketten oder als humanoide¹⁷ Gestalt auf zwei „Beinen“ fortbewegen können und auf diese Weise einen umfassenden Einblick in technische Bewegungsmöglichkeiten und –ursachen geben.

¹⁷ humanoid: „*menschenähnlich (besonders von Robotern)*“. Duden, <http://www.duden.de/rechtschreibung/humanoid>, (06.01.2014)

6.1.2 3. Klasse, „Ausgehend von Alltagserfahrungen sollen die Schülerinnen und Schüler immer intensiver mit grundlegenden elektrischen Vorgängen im technischen Alltag ... vertraut gemacht werden. ... verschiedene Spannungsquellen ... und einfache Stromkreise verstehen; Gleichstrom und Wechselstrom, Stromstärke, Spannung, Widerstand, das Ohm'sche Gesetz; elektrische Erscheinungen in Technik und Natur erklären können.“[16 S 1208]

Im Alltag werden Schülerinnen und Schüler in vielerlei Hinsicht mit Robotern konfrontiert. Hierbei kann es sich um sehr einfache Systeme handeln, die nicht gleich als Roboter erkannt werden. Exemplarisch kann hier eine elektronische Schiebetür in einem Einkaufszentrum genannt werden, die infolge eines Sensorwertes (zum Beispiel eines Infrarotsensors) entscheidet, wann die Tür geöffnet wird, und dies dann, zum richtigen Zeitpunkt (Programmierung), auch umsetzt (Output). Allerdings haben mittlerweile auch als Roboter ausgewiesene, autonom funktionierende Staubsauger Einzug in das tägliche Leben vieler Familien gefunden, über welche ein direkterer Alltagsbezug zu Robotern hergestellt werden kann. Die für das Verhalten solcher Roboter mitverantwortlichen Sensoren und Motoren, sowie die für ihre Einsetzbarkeit notwendige Spannungsquelle stellen eine gute Grundlage für die Vermittlung obigen Lehrplaninhaltes dar.

Die Funktionsweise der Roboterkomponenten und deren elektrische Vernetzung ermöglichen eine Veranschaulichung von Stromkreisen und Schaltungen. Besonders gut eignen sich hier Systeme, die als Bausatz erhältlich sind (siehe Kapitel 7: „Auswahl möglicher Roboter für den schulischen Einsatz“), da bei diesen Modellen eben genannte Schaltungen konstruiert werden müssen, um ein angestrebtes Ziel zu erreichen. Es gibt allerdings auch gängige Systeme, die nicht von Grund auf konstruiert werden müssen, wie den Lego Mindstorms EV3 (siehe Kapitel 7: „Auswahl möglicher Roboter für den schulischen Einsatz“), die eine Verwendung von selbstgebauten Schaltungen ermöglichen. Diesbezüglich lassen sich online auch Bauanleitungen finden, die als Hilfestellung für den Eigenbau kompatibler Schaltungen dienen können.¹⁸ Die Vor- und Nachteile verschiedener Systeme werden später in dieser Arbeit im Detail besprochen.

¹⁸ vgl. LEGO Mindstorms NXT Hardware Eigenbau, <http://legostorms.wordpress.com/category/hardware/eigenbau/>, (08.02.2014)

6.1.3 4. Klasse, „Einsichten in Funktionsprinzipien technischer Geräte aus dem Interessensbereich der Schülerinnen und Schüler gewinnen (Elektromotor).“ [16 S 1208]

Dass sich gerade Roboter für die Vermittlung dieses Lehrplaninhalts eignen, kann natürlich nicht als pauschale Tatsache gesehen werden, da diese programmierbaren Maschinen nicht unbedingt im „*Interessensbereich der Schülerinnen und Schüler*“ [16 S 1208] liegen müssen. Dieser Interessensbereich kann je nach Klasse sehr divergieren. Auf jeden Fall aber können Roboter als inhaltliche Option zur Auswahl stehen, da diese sich eignen, „*Funktionsprinzipien technischer Geräte*“ [16 S 1208] zu vermitteln. Als Beispiel für eine technische Komponente, die dem Lehrplaninhalt entspricht, wird darin auch der Elektromotor genannt. Dieser kann zwar auch ohne die Verwendung eines Roboters erklärt und demonstrativ im Unterricht eingesetzt werden, die programmierbaren elektronischen Helfer bieten aber eventuell einen motivationsreicheren und anwendungsbezogenen (fahren durch Elektromotor) Zugang (siehe Kapitel 4.1: „Einsatz von Robotern an Schulen zur Steigerung der Motivation“).

6.1.4 5. und 6. Klasse, „Die Schülerinnen und Schüler sollen ... mittels einfacher Schülerexperimente insbesondere die Fähigkeit zum ... Planen, Durchführen und Auswerten entwickeln ... Grundlagen der Elektrizitätslehre (einfacher Stromkreis, Spannung, Strom, elektrischer Widerstand, elektrische Energie und Umgang mit elektrischen Messgeräten) anwenden ... an Hand von Grundeigenschaften mechanischer Wellen Verständnis für Vorgänge, beispielsweise aus Akustik oder Seismik, entwickeln und als Mittel für Energie- und Informationsübertragung verstehen“ [14 S 60]

Im Zusammenhang mit dem Lehrplaninhalt der 5. und 6. Klasse Physik wäre ein sinnvoller Robotikeinsatz in mehrfacher Hinsicht denkbar. Einerseits dienen die programmierbaren Automaten als umfangreiche Systeme, mit denen die „*Fähigkeit zum ... Planen, Durchführen und Auswerten*“ [14 S 60] vermittelt werden kann. Anhand einer Problemstellung, wie beispielsweise einer Temperaturmessung, können die Abläufe der Umwandlung von erfassten Messdaten durch einen Temperatursensor in elektronische Signale veranschaulicht werden. Dabei erfolgt eine Umwandlung der analogen, physikalischen Größe Temperatur, die der Sensor misst, in einen digitalen Wert eines bestimmten Wertebereichs [vgl. 17 S 190]. Die

Messung des analogen Temperatursignals kann aufgrund der Messung des elektrischen Widerstandes erfolgen, um so ein digitales Signal zu erhalten, das vom Controller des Roboters ausgewertet werden kann [vgl. 18 S 379 ff.]. Dieser Zugang bietet sich in weiterer Folge als Überleitung zum Thema elektrische Messgeräte an. Anhand von Robotern, die einen akustischen In- und Output unterstützen, lassen sich theoretisch Daten durch Geräuschabfolgen von einem Lautsprecher zu einem Mikrophon übertragen, was einen interessanten Einstieg in das - nicht ausnahmslos beliebte - Thema der wellenbasierten Informationsübertragung ermöglicht.

6.1.5 7. und 8. Klasse, „Die Schülerinnen und Schüler sollen folgende physikalische Bildungsziele erreichen: ... verstärkt Querverbindungen mit anderen Bereichen knüpfen können“ [14 S 60]

Kaum ein anderer Lehrplanauszug ist so beispielhaft für die Sinnhaftigkeit des Einsatzes von robotikbezogenen Inhalten im Unterricht wie dieser. Alle vorangegangenen Optionen des Robotereinsatzes im Physikunterricht lassen sich über das Bindeglied Roboter in einen schlüssigen Zusammenhang zu den Unterrichtsfächern Mathematik und Informatik bringen. In manchen Themenbereichen können auch die Inhalte in eine Relation zum Unterricht der Gegenstände Biologie und Umweltkunde sowie Chemie gebracht werden.

In der Chemie wäre beispielsweise eine Verbindung über das Thema elektrische Messgeräte herstellbar, die in der Physik wie in der Chemie ein wichtiges Anwendungsgebiet haben. Für eine Querverbindung zwischen der Physik und der Biologie und Umweltkunde wären Roboter gut einsetzbar, um Bewegungsabläufe aus der Natur mit ihrer Hilfe nachzuahmen und zu analysieren. Hier zeichnet sich bereits ab, dass der Einsatz von Robotern eine Bereicherung in der fächerübergreifenden Vermittlung von Wissen darstellt. Auf diesen Aspekt wird im Folgenden noch ausführlicher eingegangen werden.

6.2 Lehrplanbezüge aus dem Unterrichtsfach Mathematik

Da die Mathematik als Grundlage der Informatik gesehen werden kann und eine Roboterprogrammierung ohne Mathematik gar nicht möglich wäre, kann hier bereits festgehalten werden, dass Roboter für eine Verwendung im Mathematikunterricht heranziehbar sind. Im Folgenden werden relevante Lehrplaninhalte dargelegt und in Verbindung mit Anwendungsmöglichkeiten für den Einsatz von Robotern gebracht.

Da die entsprechenden Lehrplanbezüge in der Mathematik während der ersten vier Schulstufen sehr ähnlich sind, werden dementsprechende Verwendungsmöglichkeiten dieser vier Jahre zusammenfassend dargelegt.

6.2.1 1. Klasse, „Mit Variablen allgemeine Sachverhalte beschreiben können... Formeln bzw. Gleichungen aufstellen ... Formeln anwenden und interpretieren können.“ [16 S 1195]

2. Klasse, „mit Variablen allgemeine Sachverhalte beschreiben ... Gleichungen und Formeln aufstellen, insbesondere auch in Sachsituationen ... Formeln interpretieren.“ [16 S 1195]

3. Klasse, „Formeln in Sachsituationen ... aufstellen können“ [16 S 1195]

4. Klasse, „Sicherheit beim Arbeiten mit Variablen, Termen, Formeln und Gleichungen steigern“ [16 S 1195]

Das Arbeiten mit Variablen, Formeln und Gleichungen stellt die Grundlage der Mathematik sowie der Informatik dar. Ein logischer Zusammenhang, da sich die Informatik als eigenständige Wissenschaft von der Mathematik ableitet [vgl. 19]. Da grundlegende Mathematikinhalte als Basis für das Erlernen der meisten Programmiersprachen benötigt werden, ist es möglich, mathematisches Wissen durch die Auseinandersetzung mit einer Programmiersprache zu vertiefen. Dieser Umweg über die Informatik bringt allerdings kaum Vorteile, es sei denn, die Programmierung bezieht sich auf Roboter, da durch deren Verwendung eine Steigerung der Motivation im Mathematikunterricht erwartet werden kann.

Gerade in der Mathematik haben Schülerinnen und Schüler oft das Problem, den schulexternen Nutzen zu erkennen, der durch das Erlernen des Schulfachs erlangt werden kann. Dies wirft oft die direkte Frage von Schülern und Schülerinnen im Unterricht auf, wozu sie „das“ denn lernen sollen [vgl. 20 S 9]. In diesem Zusammenhang stellen Roboter, als unterrichtsunterstützende Objekte, ein Bindeglied dar, durch welches die Inhalte der Mathematik - über den Umweg der Roboterprogrammierung der Disziplin Informatik - einen für Schüler greifbareren Nutzen bekommen. Dies ist einerseits durch die mechanische Ausführung der mathematikbasierten Programmbefehle gegeben, andererseits durch den

Alltagsstellenwert, den Roboter in der modernen Gesellschaft mittlerweile erhalten haben.

6.2.2 5. Klasse, „Arbeiten mit Funktionen in anwendungsorientierten Bereichen“ [14 S 47].

Der Funktionsbegriff des Programmierens und der Funktionsbegriff in der Mathematik beschreibt in beiden Fällen ein Konstrukt, das infolge eines bestimmten Inputs einen bestimmten Output liefert. Dies wird am Beispiel der folgenden Funktionen veranschaulicht, diese sollen die eingegebene Zahl x quadrieren.

Funktionsbeispiele für das Quadrieren der Zahl x .	
Funktion Mathematik	Funktion Informatik (Pseudocode)
$f(x) = x^2$	function (x) {xQuadrat = x * x; return xQuadrat;}
Der Wert x wird in der Funktion f quadriert.	Der Wert x wird an die Funktion übergeben, quadriert und dann über xQuadrat returniert.

Tabelle 3 - Funktionsbeispiele für das Quadrieren der Zahl x

Entsprechend ist es möglich, die Struktur mathematischer Funktionen mit einem Computerprogramm darzustellen. Um davon im Unterricht zu profitieren, sollte eine textuelle Programmiersprache gewählt werden, die das Ziehen von Rückschlüssen auf mathematische Funktionen erlaubt (Siehe Kapitel 8: „Programmiersprache als Kriterium zur Systemauswahl“). Beispielsweise kann durch ein Programm, das einen Roboter x Felder geradeaus fahren lässt, und danach x^2 Felder nach rechts fahren lässt, eine anwendungsorientierte Verwendung von Funktionen durch den Einsatz von Robotern veranschaulicht werden.

6.2.3 6. Klasse, „Kennenlernen von Verallgemeinerungen des Funktionsbegriffs ... Verketteten von Funktionen“ [14 S 48].

Nachdem im vorangegangenen Lehrplanbezug zum Mathematikunterricht der fünften Klasse bereits dargelegt wurde, warum sich Roboter dazu eignen, die Einsatzmöglichkeiten von Funktionen anschaulicher darzustellen, wird darauf hier nicht erneut eingegangen.

Zur Demonstration der „Verallgemeinerungen des Funktionsbegriffs“ [14 S 48] könnte die Informatik herangezogen werden, da der Funktionsbegriff hier allgemeiner

Institut für Softwaretechnologie

eingesetzt wird. In der Mathematik stellt eine Funktion eine Relation von zwei (oder mehr) Zahlen oder Mengen zueinander dar, die sich gegenseitig beeinflussen. In der Informatik kann durch eine Funktion ebenfalls ein Verhältnis zwischen zwei Zahlen oder Mengen hergestellt werden, es sind aber auch andere Relationen möglich. Dabei kann es sich zum Beispiel um eine Funktion handeln, die einen eingegebenen Datensatz um einen zusätzlichen Datensatz erweitert und das gesamte Datenkonstrukt dann als Ausgabe zurückgibt. In diesem Zusammenhang ist es denkbar, durch die Programmierung eines Roboters eine allgemeine und realitätsnahe Definition des Begriffs „Funktion“ zu präsentieren, die Rückschlüsse auf die mathematische Konkretisierung einer Funktion ermöglicht.

Die meisten Computerprogramme bestehen aus einer Vielzahl von Funktionen, die miteinander interagieren und oft einen eingabeabhängigen Rückgabewert liefern, um die Gesamtaufgabe des Programms zu erfüllen. Im Fall eines programmierten Roboters können in einzelnen Funktionen Programmsegmente für Bewegungsabläufe wie geradeaus fahren, am Stand drehen oder stoppen zusammengefasst werden. Es handelt sich dementsprechend um ein Konstrukt von Funktionen, mit dessen Hilfe es möglich wäre, das „*Verketteten von Funktionen*“ [14 S 48] demonstrativ zu behandeln.

6.3 Lehrplanbezüge aus dem Unterrichtsfach Informatik

Der Informatikunterricht, der erst seit dem Schuljahr 1985/86 als Pflichtfach verankert ist, umfasst zwei Wochenstunden in der fünften Klasse. Im Rahmen der Oberstufenreform 1989/90 wurde Informatik zusätzlich als Wahlpflichtgegenstand für die Klassen sechs bis acht eingeführt [vgl. 21 S 6 ff.]. Informatik ist der naheliegendste Unterrichtsgegenstand für den Einsatz von Robotern, da durch die Notwendigkeit ihrer Programmierung die Informatik als ein Teilgebiet der Robotik gesehen werden kann. Im Folgenden werden die Einsatzmöglichkeiten von Robotern bezogen auf den Pflichtfach- und Wahlpflichtfach-Lehrplan des Informatikunterrichts untersucht.

6.3.1 5. Klasse, „Die Schülerinnen und Schüler sollen ... Grundprinzipien von Automaten, Algorithmen und Programmen kennen lernen ...

Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern kennen lernen und somit in ihrer Berufsorientierung Unterstützung finden“ [14 S 71]

Um Schülerinnen und Schülern die Grundprinzipien von Automaten beizubringen, bietet sich der Einsatz von Robotern verständlicher Weise an, da diese selbst im Rahmen der Definition eines Automaten enthalten sind. Durch die Programmierung dieser speziellen Maschinen kann der Lehrplaninhalt *„Grundprinzipien von Automaten, Algorithmen und Programmen kennen lernen“* [14 S 71] zusammengefasst - beispielsweise im Laufe eines Projekts - vermittelt werden, da diese Themenbereiche sich durch die Verwendung eines Roboters gewissermaßen gegenseitig ergänzen.

Ebenfalls können Roboter eingesetzt werden, um Schülerinnen und Schülern *„Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern“* [14 S 71] näher zu bringen, und somit auch, um das Interesse an technischen Berufen zu wecken beziehungsweise zu intensivieren.

6.3.2 6. bis 8. Klasse (Wahlpflichtgegenstand), „Konzepte von Programmiersprachen ... grundlegende Algorithmen und Datenstrukturen ... künstliche Intelligenz ... Informatik, Gesellschaft und Arbeitswelt“ [14 S 80]

Für die Schulstufen sechs bis acht, in denen der Informatikunterricht an den meisten allgemeinbildenden höheren Schulen nur als Wahlpflichtgegenstand gewählt werden kann, ist der Lehrplan nicht nach Schulstufen getrennt. Es ist dem Lehrpersonal überlassen, wie und wann die entsprechenden Inhalte vermittelt werden. Grundsätzlich kann hier bereits festgehalten werden, dass durch diese Flexibilität das Arbeiten an Schüler- und Schülerinnenprojekten unter Verwendung von Robotern möglich wäre. Entsprechend vorangegangener Begründungen lassen sich Roboter zur anschaulichen Darstellung von Programmierinhalten verwenden und können demnach für die Vermittlung der Inhalte *„Konzepte von Programmiersprachen“* [14 S 80] sowie *„grundlegende Algorithmen und Datenstrukturen“* [14 S 80] eingesetzt werden.

Der Themenbereich der künstlichen Intelligenz stellt als eigene Disziplin der Informatik ein weitläufiges Gebiet dar. Unter den vielen vorhandenen Definitionen des Begriffs soll beispielhaft jene von Elaine Rich dienen: „*Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better.*“[22] Dementsprechend sollen im Bereich der künstlichen Intelligenz Computern dafür programmiert werden, Lösungen für Aufgaben zu finden, die gegenwärtig nur, oder besser, von Menschen ausgeführt werden können. Auch wenn es im Rahmen des Schulunterrichts nur schwer möglich ist, neue Lösungen in diesem Bereich zu entwickeln, können Beispiele der gegenwärtigen Forschung behandelt werden. Da dies die Entwicklung autonomer Fahrzeuge beinhaltet, bietet sich dieser Kontext an, um anhand fahrzeugähnlicher Roboter Grundsätze der künstlichen Intelligenz zu thematisieren.

Ausgehend vom Thema „*künstliche Intelligenz*“ [14 S 80], vor allem bezüglich autonomer Fahrzeuge, bietet sich eine Überleitung zum Lehrplaninhalt „*Informatik, Gesellschaft und Arbeitswelt*“ [14 S 80] an, da das vorangegangene Thema sich gut eignet, um die zu vermittelnden Teilaspekte Informatik, Gesellschaft und Arbeitswelt in den Unterricht zu integrieren. Abgesehen davon eignen sich Roboter, sowie deren Programmierung, demonstrativ für eine Vielzahl von Themen, den Lehrplaninhalt „*Informatik, Gesellschaft und Arbeitswelt*“ [14 S 80] betreffend, da diese in allen drei Bereichen heute wichtige Aufgaben übernommen haben.

6.4 Einsatz von Robotern im fächerübergreifenden Unterricht

Fächerübergreifender Unterricht findet an vielen Schulen Einsatz, weil dadurch eine Vernetzung der erlernten Inhalte erleichtert werden kann. Es wird ermöglicht, eine Problemstellung aus den verschiedenen Perspektiven der differenzierten Wissenschaften zu sehen und Querverweise zwischen den einzelnen Sichtweisen herzustellen. Exemplarisch sollen weitere Begründungen für fächerübergreifenden Unterricht gegeben werden [vgl. 23 S 333]:

- i. Durch das Einbeziehen des Vorwissens und des Interesses von Schülern werden gute Voraussetzungen für den Ablauf von Lernprozessen geschaffen. Diese Interessen lassen sich meist nicht einem einzelnen Fach zuweisen, weshalb der Einsatz von fächerverbindendem Unterricht in diesem Kontext als Voraussetzung gesehen werden kann.

- ii. Fächerübergreifender Unterricht unterstützt die Vorbereitung auf den Berufsalltag, in dem Fächergrenzen oftmals überschritten werden. Die Grenzen und Chancen der jeweiligen Unterrichtsfächer werden veranschaulicht.
- iii. Für die Umsetzung von Schulprojekten (deren Vorteile in Kapitel 4.3: „Projektorientiertes Lernen“, bereits Erwähnung fanden) ist fächerübergreifender Unterricht oft notwendig, da im Rahmen von Projekten meist die Fachgebiete einzelner Unterrichtsfächer überschritten werden.
- iv. Durch die Verknüpfung von Unterrichtsinhalten verschiedener Schulgegenstände, zum Beispiel im Bereich der MINT-Fächer, kann ein gendergerechter Unterricht ermöglicht werden, da ein breiteres Spektrum an Interessensgrundlagen vereint wird. Dies kann besonders bei Schülerinnen den Zugang zu technischen Berufen fördern (was auch dem Interesse der Wirtschaft entspricht - siehe Kapitel 4.2: „Schule, Roboter und die Berufswahl“).

Für die Planung fächerverbindenden Unterrichts, speziell im Rahmen eines Unterrichtprojekts, würden sich Roboter hervorragend als zentrales Thema eignen. Dies resultiert aus den vorhergehend genannten Einsatzmöglichkeiten in den verschiedenen Schulgegenständen, die sich teilweise gegenseitig ergänzen. Beispielsweise könnte in den Gegenständen Biologie und Umweltkunde sowie Physik eine in der Natur vorkommende Fortbewegungsart wie etwa die der Spinnen mechanisch, für die roboterbezogene Fortbewegung, imitiert werden. Nach der mathematischen Betrachtung der Beincoordination wäre es denkbar, diese im Rahmen des Informatikunterrichts in einem Programm umzusetzen, welches den Spinnenroboter in Bewegung setzen soll.

6.5 Fazit

In den vorangegangenen Kapiteln wurde ersichtlich, dass sich Roboter für den Einsatz in den Schulgegenständen Physik, Mathematik und Informatik in mehrerlei Hinsicht eignen. Ebenfalls wurde auf Einsatzmöglichkeiten und damit einhergehende positive Aspekte des fächerübergreifenden Unterrichts eingegangen, wodurch ein noch breiteres Anwendungsgebiet erschlossen werden könnte.

Entsprechend der Lehrpläne der im Detail behandelten Schulgegenstände ist eine Verwendung von der ersten bis zur achten Klasse möglich. Aufgrund der Curricula ist

es demnach nicht möglich, das Alter von Schülerinnen und Schülern, die als Bezugsfeld dienen, einzugrenzen, da für jede Schulstufe - und somit auch für jede Altersstufe - der Kinder und Jugendlichen sinnvolle Lehrplanbezüge hergestellt werden konnten.

Zu berücksichtigen gilt es allerdings, dass ein Robotermodell zum Einsatz kommen soll, welches den Schülern und Schülerinnen der jeweiligen Altersgruppen zugänglich ist, und dass besonders die Programmiersprachen, die zur Roboterprogrammierung des angedachten Modells herangezogen werden können, für das Zielgruppenalter geeignet sind. Um die Auswahl eines dementsprechenden Robotermodells zu erleichtern, werden in den folgenden Kapiteln Robotermodelle für den Schuleinsatz und die jeweils nutzbaren Programmiersprachen vorgestellt.

7 Auswahl möglicher Roboter für den schulischen Einsatz

Robotersysteme sind bereits sehr weit verbreitet und tief in den Alltag wirtschaftsstarker Länder integriert. Zu Hause fahren Staubsauger und Rasenmäher und erfüllen selbstständig ihre Aufgabe, während deren Besitzer oder Besitzerin schon in der Arbeit ist und einen Teil einer neuen Fertigungsanlage programmiert - einen Industrieroboter.

Aus der Wirtschaft sind die mechanischen Helfer kaum mehr wegdenkbar, aus vielen Kinderzimmern allerdings auch nicht. Bei einer unendlich lang erscheinenden Liste an angebotenen Robotern (rund 11,500 Ergebnisse bei einer Produktsuche auf www.amazon.com in der Kategorie „Toy & Games“ – Suchbegriff „robot“ vom 05.01.2014) spielt die Modellwahl, gerade für die Wissensvermittlung, eine primäre Rolle. Es gibt viele Auswahlkriterien, welche die Entscheidung für ein bestimmtes System beeinflussen können. Im Folgenden werden einige wichtige Auswahlpunkte angesprochen und anschließend ein Überblick darüber gegeben, welche Produkte sich unter anderem für den Einsatz zur Wissensvermittlung in Schulen eignen. Die Vor- und Nachteile der ausgewählten Systeme für den Schuleinsatz stehen im Fokus und sollen überblicksweise angegeben werden. Diese Betrachtung bezieht sich auf Aspekte, die für den schulischen Einsatz relevant sind, auf detaillierte Aspekte wurde im Rahmen dieser Darstellung deshalb kein Bezug genommen.

7.1 Wichtige Fragen, die vor dem Kauf eines Roboters gestellt werden können

In diesem Kapitel wird im Folgenden ein Überblick über Fragen gegeben, die vor der Auswahl eines Robotermodells für den Unterricht gestellt werden sollten, um so die Wahl eines Modells zu erleichtern und die Anzahl in Frage kommender Modelle einzugrenzen.

- Praktisches System – theoretisches System – was wird benötigt?
- Aufwand vor dem Erstbetrieb – wie wird der Roboter geliefert?
 - o Handelt es sich um einen Bausatz?
 - Wird Werkzeug benötigt?
 - Welches technische Knowhow wird benötigt?
 - Ist der Bau Schülern zumutbar?

- Wie viel Unterrichtszeit/Vorbereitungszeit muss vor dem Erstbetrieb investiert werden?
- Input und Output des Roboters – was kann der Roboter?
 - Kann dieser erweitert werden?
 - Wie weit kann dieser erweitert werden?
 - Welche Interaktion mit seiner Umwelt beherrscht der Roboter?
 - Welche Bewegung kann er ausführen?
 - Welche Sensoren hat er?
- Preis – wieviel kostet der Roboter?
 - Was ist im Kaufpreis enthalten?
 - Was wird zusätzlich benötigt?
 - Wie viel kosten zusätzliche Komponenten?
- Programmierung – wie kann ich den Roboter befehligen?
 - Welche Programmiersprachen können verwendet werden?
 - Muss eine Änderung am Roboter vorgenommen werden, um eine andere Sprache zu verwenden?

7.1.1 Praktisches System – theoretisches System

Es gibt neben den bereits besprochenen praktischen Robotersystemen auch theoretische Umgebungen, in denen eine Robotersimulation nach einer Programmabfolge virtuell Roboterbewegungen darstellt. Viele der Vorteile, die der Einsatz von Robotik im Unterricht mit sich bringt - und welche im Kapitel 4: „Gründe für den Einsatz von Robotik im Unterricht“ bereits angesprochen wurden - sind mit diesen virtuellen Umgebungen aber nicht gegeben. Beispielsweise fehlt die Praxisnähe und es lässt sich infolge obiger Ausführungen vermuten, dass der Einfluss auf die Motivation geringer ausfällt als bei dem tatsächlichen Einsatz von Robotern. Aus diesem Grund werden rein theoretische Systeme im Rahmen dieser Arbeit nicht speziell thematisiert, auch wenn virtuelle Systeme eventuell die Möglichkeit bieten, abstrakte Programmierabläufe anschaulicher zu vermitteln. Es gibt aber auch Umgebungen, in denen beide Darstellungsformen möglich sind. Das geschriebene Programm kann in diesem Fall per virtuellem oder realem Roboter ausgeführt werden. Diese hybriden Entwicklungsumgebungen haben rein praktischen Umgebungen gegenüber keine Nachteile, da die reale Roboterprogrammierung zusätzlich möglich ist. Hingegen bietet die Anwendung einige Vorteile.

Hat eine Schule nicht das Budget, sich für jeden Schüler, oder zumindest jede Kleinstgruppe, einen Roboter zu kaufen, so können Programme zunächst virtuell getestet werden, um diese anschließend auf einem realen Roboter auszuführen. Dies senkt die Anzahl an gleichzeitig benötigten Robotern und somit die Anschaffungskosten. Ein weiterer Vorteil dieser kombinierten Darstellungsmöglichkeit ist, dass Schüler und Schülerinnen auch zu Hause, ohne einen realen Roboter, an Projekten und Programmen arbeiten und so auch versäumte Inhalte nachholen können. Diesen Vorteil können sich auch Lehrerinnen und Lehrer zunutze machen, um Unterrichtsstunden vorzubereiten, oder gegebenenfalls Programme von Schülern zu testen und zu korrigieren. Dieser Nutzen kann anhand eines Praxisbeispiels bestätigt werden. Im Rahmen des Roberta-Projekts wurde von vielen Lehrpersonen der Wunsch geäußert, sich zu Hause in die Roboterprogrammierung einzuarbeiten zu können. Da nur selten zu Hause ein Roboter zur Verfügung stand, wurde unplanmäßig sogar ein eigener Simulator für die verwendeten Roboter entwickelt [vgl. 10 S 18 f.].

Wegen der Vorzüge einer zusätzlichen Simulation werden immer mehr virtuelle Lösungen entwickelt¹⁹ ²⁰. Da in der vorliegenden Arbeit primär die Unterrichtsintegration von realen Robotern thematisiert wird, werden keine separaten Simulationsprojekte im Detail angesprochen. Simulationsmöglichkeiten im Rahmen des Funktionsumfangs von Entwicklungsumgebungen, die für die Programmierung von realen Robotern im weiteren Verlauf dieser Arbeit noch thematisiert werden, werden in den entsprechenden Kapiteln behandelt.

Durch den erweiterten Anwendungsumfang einer hybriden Entwicklungsumgebung sollte vor einer Anschaffungsentscheidung eine Auseinandersetzung mit dem gewünschten Funktionsumfang stattfinden. Ob ein Roboter eine Programmierumgebung unterstützt, in der eine virtuelle Darstellung neben einer realen Programmierung ermöglicht wird, kann dementsprechend eine der Entscheidungsgrundlagen für ein bestimmtes System sein. Welches der verglichenen Robotersysteme mit welcher Endwicklungsumgebung kompatibel ist, wird bei den einzelnen Modellen weitestgehend genannt. Ein Gesamtüberblick über

¹⁹ vgl. RobotVirtualWorlds, <http://www.robotvirtualworlds.com/>, (14.01.2014)

²⁰ vgl. Universität Paderborn – Lego Mindstorms Simulator: <http://ddi.uni-paderborn.de/software/lego-mindstorms-simulator.html>, (14.01.2014)

die Kompatibilität von Robotern und Entwicklungsumgebungen wird im Kapitel 8.3: „Robotersysteme und unterstützte Programmiersprachen im Überblick“ gegeben.

7.1.2 Aufwand vor dem Erstbetrieb

Grundsätzlich können die Robotermodelle, die am freien Markt erhältlich sind, in unterschiedliche Kategorien des Auslieferungszustandes eingeteilt werden. In der vorliegenden Arbeit wurden, je nach Aufwand vor Erstinbetriebnahme, drei Kategorien gewählt - Fertigmodelle, modulare Modelle und Bausatzmodelle.

Fertigmodelle: Unter Fertigmodellen wird hier verstanden, dass keine oder nur minimalste Aufbauarbeiten nötig sind. Diese Kategorie Roboter ist dadurch am schnellsten einsetzbar und kann, ohne Verlust an Unterrichts- oder Vorbereitungszeit verwendet werden. Bei den Produkten der anderen beiden Sparten ist dies nicht der Fall. Allerdings bieten fertig erhältliche Modelle dem Anwender nur begrenzte Möglichkeiten zur Nutzung. Ihre fehlende, beziehungsweise eingeschränkte, Erweiterbarkeit stellt für den schulischen Einsatz den Hauptnachteil dar. Kann ein Roboter dieser Sparte nicht von vornherein alles, was für den folgenden Unterricht benötigt wird, kann dieser für den gewünschten Einsatz nicht (oder nur in Verbindung mit großem Aufwand) verwendet werden. Wird das das weite Feld von denkbaren Anwendungen für die erleichterte Wissensvermittlung in unterschiedlichen Schulfächern betrachtet, könnte eine Erweiterungsoption in vielen Fällen sinnvoll sein.

Modulare Modelle: In der Kategorie der modularen Roboter wurden Modelle zusammengefasst, deren einzelne Komponenten betriebsfertig geliefert werden, die allerdings noch zu einem funktionsfähigen System kombiniert werden müssen. Diese einzelnen Teile des Roboters haben meist bestimmte Funktionen und können beispielsweise Sensoren, Motoren oder Verbindungsstücke sein. Modulare Roboter bieten dem Anwender und der Anwenderin die Möglichkeit, das verwendete System für unterschiedliche Anforderungen anzupassen. Dadurch erweitert sich das Einsatzgebiet immens gegenüber den Fertigmodellen. Die Verwendungsmöglichkeiten hängen bei diesem Typ hauptsächlich von der gesamten angebotenen Produktpalette kombinierbarer Teile ab. Je nach Art des flexibel zusammenfügbaren Roboters wird aber auch entsprechendes Wissen über die Kombinationsmöglichkeiten benötigt. Der Einsatz von Werkzeug ist bei modularen

Maschinen meist nicht notwendig. Diese Roboterklasse bildet, im gegebenen Kontext, das Bindeglied zwischen Fertigmodellen mit ihrem eingeschränkten Funktionsumfang und den Bausatzrobotern mit dem breitesten Anwendungsfeld.

Bausatzmodelle: Als Bausatzmodelle werden hier alle Roboter verstanden, die in (mehr oder weniger) einzelnen Komponenten geliefert werden. Im Fall dieses Typs muss der Roboter vor der Verwendung erst zusammengebaut werden. Dies kann je nach Komplexität des Systems viele Stunden dauern. Im Fall des Bausatzes „ScoutWalker III“ von solarbotics zum Beispiel werden laut Herstellerangaben etwa fünf Stunden pro Modell benötigt [24 S 1]. Für eine Fertigstellung von Robotern aus Bausätzen müssen meist auch Werkzeuge wie Lötkolben zum Einsatz gebracht werden. Oft wird auch fundamentiertes Hintergrundwissen benötigt. Für den Aufbau des eben genannten Roboters sollten der Konstrukteur oder die Konstrukteurin in der Lage sein, elektronische Schaltpläne zu lesen und zu interpretieren [24 S 15]. Die Verwendung eines Bausatzroboters setzt einen sehr komplexen Zugang zur Robotik voraus. Durch die Erweiterbarkeit, die bei Bausatzrobotern meist gegeben ist, werden der Anwender und die Anwenderin in den Einsatzgebieten der entsprechenden Roboter kaum eingeschränkt. Die Anwendungsmöglichkeiten von Bausatzrobotern sind beinahe unbegrenzt, da viele Sensoren und Motoren mit diesen Systemen kombinierbar sind. Dadurch vergrößert sich das Einsatzgebiet. Die unterstützten Programmiersprachen stellen bei diesem Robotertyp oft den limitierenden Faktor bezüglich der Anwendungsmöglichkeiten in Schulen dar.

Der Robotikexperte Peter Protzel: *„Selbst gebaute Roboter sind deshalb so ideal, weil dort das Wissen aus Mechanik, Elektrotechnik und Informatik zu einem System integriert wird und man sofort sieht, ob es funktioniert. Anschaulicher kann man Kindern die Aufgaben eines Ingenieurs nicht vermitteln.“* [09]. Für den Einsatz an höheren technischen Lehranstalten eignen sich diese Modelle dementsprechend gut, da an diesen Einrichtungen die technische Ausbildung im Vordergrund steht. An allgemeinbildenden höheren Schulen hingegen könnte es schwer werden, diese Techniktiefe im Rahmen des Normunterrichts unterzubringen.

7.2 Überblick über ausgewählte Fertigmodelle

Dieser Abschnitt befasst sich mit zwei Fertigrobotermodellen, die aufgrund ihrer Spezifikationen gut für den Schulunterricht geeignet sein sollten. Diese Modelle

wurden unter anderem gewählt, weil beide im unteren Preissegment angesiedelt sind, einen akzeptablen Funktionsumfang ab Werk mitbringen und bereits im Schulalltag Verwendung fanden. Für beide Modelle gibt es auch große Communitys, die den Umgang durch eine Vielzahl bereits gelöster Probleme und auch zum Teil unterrichtsrelevanter Lehrmaterialien unterstützen.

7.2.1 Finch

Bei dem Finch handelt es sich um einen Roboter, der fertig für den Einsatz geliefert wird, und der bewusst für die Vermittlung schulischer, computerbasierter Bildung entworfen wurde. Er ist das Ergebnis einer vierjährigen Entwicklungsstudie des CREATE Lab der Carnegie Mellon Universität in Pittsburgh, USA. Die Grundidee hinter der Plattform war es, als Unterstützung für die Einführung in die Programmierung zu dienen²¹. Einer der Hauptvorteile dieses Fertigroboters ist, dass er bereits für umgerechnet etwa 73 Euro erhältlich ist, beim Kauf von mehreren Geräten reduziert sich dieser Preis je Modell auf bis zu umgerechnet etwa 58 Euro. Es handelt es sich also um ein recht preiswertes Robotiksystem²².

Beim Design des Finch standen fünf zu erfüllende Aspekte im Vordergrund. Er sollte:

- i. überall einsetzbar sein
- ii. viel Möglichkeiten zur Interaktion bieten
- iii. ein ansprechendes und praktisches Design haben
- iv. robuste Hardware besitzen
- v. leicht für unterschiedliche Lehrinhalte einsetzbar sein

Er arbeitet ohne eigene Spannungsquelle und erhält die benötigte Betriebsspannung über einen USB-Anschluss, über den er auch programmiert wird und der sich am hinteren Ende des Geräts befindet (siehe Abbildung 2)²³. Dieser Aspekt macht das Modell zwar weniger mobil, allerdings wird dadurch auch eine Fehlerquelle, die Spannungsquelle, vermieden. Der Roboter kommt ausgestattet mit zwei Lichtsensoren (LIGHT SENSORS), einem Temperatursensor (TEMPERATUR SENSOR), zwei Hindernissensoren (OBSTACLE DETECTION SYSTEM) und einem Drei-Achsen-Beschleunigungssensor (ACCELEROMETER), der seine Ausrichtung entlang der Längsachse misst (siehe Abbildung 2).

²¹ vgl. Finch, <http://www.finchrobot.com>, (06.01.2014)

²² vgl. Finch, <http://www.finchrobot.com/purchase>, (07.01.2014)

²³ vgl. Finch, <http://www.finchrobot.com>, (06.01.2014)

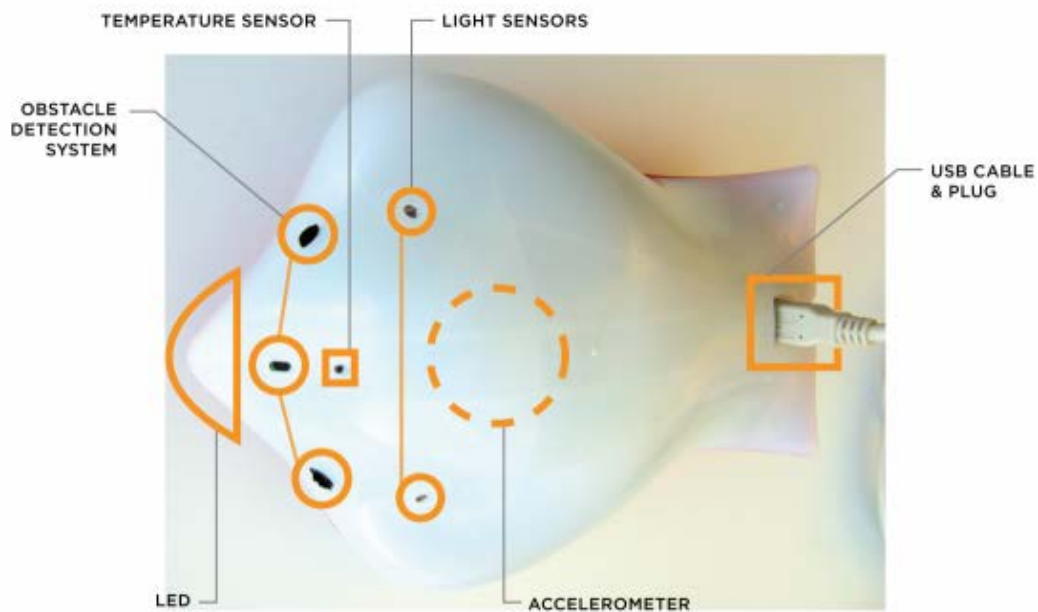


Abbildung 2 – Finch: Ansicht von oben, Sensorübersicht, USB Anschluss und LED²⁴

Das Modell ist mit zwei separaten Motoren bestückt, die neben einem Summer und einem RGB LED (siehe Abbildung 2) seine Ausgabeoptionen darstellen²⁵. Dieser Roboter unterstützt die Programmierung mit einer Vielzahl von Programmiersprachen, von denen Scratch, Snap!, Python, Java, C++, C# und Visual Basic exemplarisch genannt werden sollen²⁶. Ein Gesamtüberblick ist im Kapitel 8.3: „Robotersysteme und unterstützte Programmiersprachen im Überblick“ zu finden.

Obwohl es sich bei dem Finch um einen Fertigeroboter handelt, kann er für das Vermitteln von Lehrinhalten bezüglich elektrischer Schaltkreise herangezogen werden, da die Schaltpläne des Roboters öffentlich zugänglich sind. Es gibt bereits Erweiterungsoptionen für diesen Roboter, die vom Hersteller selbst angeboten werden. Beispielsweise existiert die Möglichkeit, den Roboter mit einem Raspberry Pi²⁷ zu koppeln, wodurch sich eine Vielzahl weiterer Anwendungsmöglichkeiten eröffnet und der Finch auch mit anderen Computern und dem Internet vernetzt werden kann²⁸.

²⁴ <http://www.finchrobot.com/finch-hardware>, (22.01.2014)

²⁵ vgl. Finch, <http://www.finchrobot.com/finch-hardware>, (07.01.2014)

²⁶ vgl. Finch, <http://www.finchrobot.com/software-overview>, (07.01.2014)

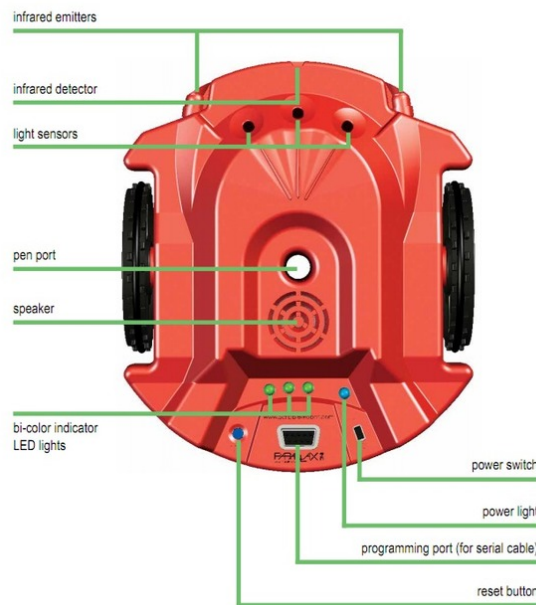
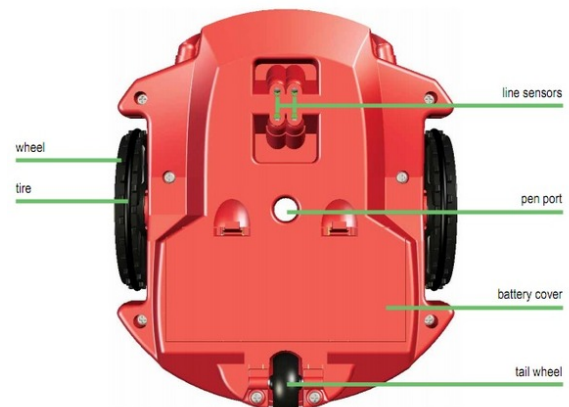
²⁷ vgl. Raspberry Pi – Ein Linux basierter Kleincomputer mit ARM Prozessor, <http://www.raspberrypi.org/>, (08.01.2014)

²⁸ vgl. Finch, <http://www.finchrobot.com/raspberry-pi>, (07.01.2014)

Als Fertigeroboter kann der Finch für ein breites Anwendungsfeld herangezogen werden, vor allem da er bereits gut bestückt mit Sensoren geliefert wird. Außerdem bieten sich auch Erweiterungsmöglichkeiten an. Seine geringen Anschaffungskosten (verglichen mit Konkurrenzprodukten) sprechen für diesen Roboter, der speziell für den Einsatz im Bildungssektor konzipiert wurde.

7.2.2 Scribbler 2

Der Scribbler 2 ist ebenfalls ein betriebsfertig gelieferter Roboter, der von der Firma Parallax Inc. aus Rocklin in Kalifornien, USA, entwickelt wurde. Im Auslieferungszustand ist dieses Modell bereits vorprogrammiert mit acht Programmen, mit denen er in der Lage ist, zum Beispiel Objekte zu detektieren und Linien zu folgen. Dieses Modell hat eine integrierte Spannungsquelle, die aus handelsüblichen Batterien des Typs AA gespeist wird, wodurch ein kabelloser Betrieb ermöglicht wird. Zur Programmierung hat dieses Modell eine serielle Anschlussstelle. Mit Hilfe eines Adapters kann diese auch mit einem USB-Anschluss verbunden werden. Mit Abmessungen von 188 x 158.8 x 81 mm ist auch dieses Modell problemlos transportierbar. Preislich liegt es etwas über dem Finch und kostet etwa 96 Euro, ab zehn Stück wird der Preis auf etwa 86 Euro reduziert. Ausgestattet ist dieser Roboter mit Hindernissensoren (INFRARED EMITTERS, INFRARED DEDECTOR), Lichtsensoren (LIGHT SENSORS) und Liniensensoren (LINE SENSORS) (siehe Abbildung 3 und Abbildung 4). Für die Ausgabe bietet dieses Modell zwei Motoren, einen Lautsprecher (SPEAKER) sowie drei zweifarbige LEDs (BI-COLOR INDICATOR LED LIGHTS) (siehe Abbildung 4). Um den Programmablauf grafisch zu verfolgen, lässt sich ein Stift in seiner Mitte positionieren (PEN PORT).

TOP VIEWAbbildung 3 - Scribbler 2 von oben²⁹**BOTTOM VIEW**Abbildung 4 - Scribbler 2 von unten³⁰

Für die Programmierung bietet die Firma Parallax eine eigene grafische und eine eigene textbasierte Programmiersprache an. Durch Verwendung des Propeller GCC Compilers kann dieses Robotermodell auch mit den Programmiersprachen C und C++ programmiert werden und eignet sich laut Herstellerangaben von Parallax ab dem 14ten Lebensjahr³¹.

Bei dem Scribbler 2 handelt es sich um einen Roboter, der sich (verglichen mit Konkurrenzprodukten) im unteren Preissegment befindet. Da dieses Modell bereits vorprogrammiert ausgeliefert wird, werden für seinen ersten Einsatz nur sechs AA-Batterien benötigt, wodurch ein schneller Einstieg ermöglicht wird. Für die Programmierung mit Programmiersprachen wie C und C++ wird aber bereits ein Compiler eines Drittanbieters gebraucht. Da dieses Modell mit weniger Sensoren verkauft wird als manche Alternativprodukte, kann es wünschenswert sein, diesen mit weiteren Sensoren zu bestücken. Dies wird laut Herstellerangaben durch einen integrierten „hacker port“ ermöglicht. Es existiert auch die Möglichkeit, externe Motoren anzuschließen³².

²⁹ <http://www.pololu.com/product/1610/>, (22.01.2014)

³⁰ vgl. ebenda

³¹ vgl. Parallax, <http://www.parallax.com/product/28136/>, (06.02.2014)

³² vgl. ebenda

Durch den, im Vergleich zu Alternativprodukten wie etwa dem Finch, geringen Funktionsumfang des Roboters und seine eingeschränkten Programmieroptionen eignet sich dieses Produkt nur für ein kleineres Einsatzgebiet als andere Roboter. Der Roboter lässt sich durch den Einsatz von Elektronikkomponenten wie dem IPRE Fluke2 Add-On Board von betterbots [vgl. 25] zwar erweitern, da dieses Zusatzprodukt aber weitere etwa 73 Euro kostet, verschiebt sich die Preislage dann in ein weniger vorteilhaftes Segment³³. Diese Platine bietet zusätzliche Infrarotsensoren zur Hindernisdetektion, eine Kamera und weitere LEDs für die Ausgabedarstellung und einen Bluetooth-Chip zur kabellosen Datenübertragung.

7.2.3 Fertigmodelle Fazit

Die beiden Robotermodelle, der Finch und der Scribbler 2, die aus dem Produktsegment der fertigen Modelle vorgestellt wurden, werden jeweils in einem betriebsfertigen Zustand geliefert. Mit einem Preis von je unter 100 Euro sind sie im unteren Preissegment angesiedelt.

Für die Verwendung als Lehrmittel ist die Auswahl an angebotenen Programmiersprachen vordergründig, vor allem für den Einsatz zur Vermittlung von Inhalten der Softwareentwicklung. In diesem Vergleichspunkt bietet der Finch gegenüber dem Scribbler 2 eine viel größere Zahl an unterstützten Sprachen an, wodurch sich ein breiteres Einsatzgebiet abdecken lässt. Ein weiterer Unterschied der Modelle ist die Stromversorgung, die bei dem Finch per USB-Verbindung erfolgt, welche ohne Erweiterung für den Betrieb notwendig ist. Der Scribbler 2 wird bereits mit autonomer Stromversorgung geliefert.

Eine Entscheidungsgrundlage für eines der beiden Modelle kann auch der Sensorumfang sein. Beide Modelle kommen mit Licht- und Hindernissensoren, ein Beschleunigungs- und ein Temperatursensor wird nur vom Modell Finch geboten. Der Scribbler 2 beinhaltet dafür ab Werk bereits zwei Liniensensoren, welche der Finch nicht hat.

Ein weiterer Unterschied ist die Erweiterungsmöglichkeit. Der Scribbler 2 kann durch das IPRE Fluke2 Add-On Board in seinem Funktionsumfang erweitert werden, einer Platine, die für den Einsatz mit dem Scribbler 2 entwickelt wurde. Sie bietet

³³ vgl. BetterBots, <http://www.betterbots.com/>, (07.01.2014)

spezifische Zusatzmöglichkeiten, die für den Robotikeinsatz konzipiert sind. Der Finch kann durch die Koppelung mit einem Raspberry Pi einen größeren Funktionsumfang erhalten. Im Gegensatz zum IPRE Fluke2 Add-On Board handelt es sich dabei um einen eigenständig verwendbaren Kleinstcomputer, der mit einem eigenen Betriebssystem betrieben werden kann³⁴. Dieser bietet dem Anwender beziehungsweise der Anwenderin auch außerhalb der Robotik eine Vielzahl unterschiedlicher Verwendungsmöglichkeiten und kann zur Wissensvermittlung in vielen Bereichen der Informatik eingesetzt werden [vgl. 26 S 63 ff.]. Da beide Modelle in unterschiedlichen Bereichen Vor- und Nachteile aufweisen, ist das passendere Produkt vom Einsatzgebiet abhängig, wodurch keine allgemein gültige Empfehlung abgegeben werden kann. Vor einer Entscheidung für eines der beiden Robotersysteme sollte unbedingt auf die unterstützten Programmiersprachen geachtet werden, sowie auf die benötigten Sensoren für die Umsetzung bestimmter Lernaufgaben.

7.3 Überblick über ausgewählte modulare Modelle

Die modularen Robotersysteme stellen in diesem Kontext ein Bindeglied zwischen den Bausatz- und den Fertigmodellen dar. Sie sind (weitestgehend) ohne einen werkzeuggestützten Aufbau zusammensetzbar, sind aber im Aufbau flexibel und können mit den gerade benötigten Sensoren und Motoren ausgestattet oder erweitert werden. Drei modulare Robotersets werden in dieser Arbeit näher betrachtet, das LEGO Mindstorms Education EV3-Basis-Set, das Fischertechnik ROBO TX Training Lab und der VEX Robotics Super Kit.

Diese Baukastensets wurden ausgewählt, da sie entweder für den Bildungsbereich vorgesehen sind oder einem entsprechenden Set vom Umfang her möglichst nahe kommen. Im Fall des Lego-Systems handelt es sich bei dem LEGO Mindstorms EV3 um den Ende 2013 erschienenen Nachfolger des weit verbreiteten und im Bildungsbereich jahrelang erprobten LEGO Mindstorms NXT, der bereits seit 2006 auf dem Markt ist. Unterschiede der beiden Systeme finden in diesem Zusammenhang ebenfalls Erwähnung.

³⁴ vgl. Raspberry Pi, <http://www.raspberrypi.org/>, (08.01.2014)

7.3.1 LEGO Mindstorms Education EV3-Basis-Set

Mit dem LEGO Mindstorms EV3 brachte LEGO Ende 2013 den dritten modularen Roboter seiner Klasse auf den Markt. Seine Vorgängermodelle, der LEGO Mindstorms RCX und der LEGO Mindstorms NXT, gehören zu den ersten vergleichbaren Modellen und können als mitverantwortlich für die Entwicklungen in dieser Robotersparte gesehen werden. Laut Herstellerangaben eignen sich beide Systeme für die Verwendung ab dem zehnten Lebensjahr, somit sind sie für jedes Alter der Schülerinnen und Schüler der Bezugsgruppe geeignet. Das neue Modell EV3 ist, wie schon seine Vorgänger, als Education Basis-Set erhältlich, das für die Verwendung in Schulen zugeschnitten ist³⁵.

Der LEGO NXT, der seit 2006 am Markt ist, ist ein weit verbreitetes System, das im Bildungsbereich jahrelang erprobt wurde. Seine Motoren und Sensoren können mit dem Nachfolgermodell EV3 verwendet werden, umgekehrt ist die Verwendung der EV3-Sensoren mit dem NXT nicht möglich. Die Motoren sind allerdings auch rückwärts kompatibel³⁶. Obwohl der LEGO Mindstorms NXT weiter verbreitet ist und gegenwärtig mehr Programmiersprachen unterstützt, wurde der LEGO Mindstorms EV3 für den Vergleich im Rahmen dieser Arbeit gewählt, weil der NXT mutmaßlich nicht mehr lange erhältlich sein wird (Geschäftsverkauf bereits eingestellt, Verkauf und Support nur noch über LEGO Education³⁷) und gerade für Neuanschaffungen der EV3 deswegen sinnvoller erscheint. Bei einem Preisunterschied von nur etwa 20 Euro (LEGO Mindstorms Education NXT Basis-Set etwa 370 Euro³⁸, LEGO Mindstorms Education EV3 Basis-Set etwa 390 Euro³⁹) spricht der Preis des LEGO Mindstorms Education EV3 Basis-Sets auch für die Anschaffung des aktuellen Modells. Für Schulen bietet LEGO auch Klassen-Sets an, diese kosten für 16 Auszubildende etwa 4250 Euro und für 30 Schülerinnen und Schüler etwa 7550 Euro.

³⁵ vgl. Lego, <http://education.lego.com/de-de/lego-education-product-database/mindstorms-ev3/45544-lego-mindstorms-education-ev3-core-set>, (28.01.2014)

³⁶ vgl. Robot², <http://robotsquare.com/2013/07/16/ev3-nxt-compatibility/>, (28.01.2014)

³⁷ vgl. Lego, <http://education.lego.com/>, (28.01.2014)

³⁸ vgl. Lego, <http://shop.legoeducation.com/de/product/lego-mindstorms-education-nxt-basis-set-9797-29/>, (28.01.2014)

³⁹ vgl. Lego, <http://shop.legoeducation.com/de/product/lego-mindstorms-education-ev3-basis-set-45544-198/>, (28.01.2014)

Da dem LEGO Mindstorms NXT kein eigenes Kapitel gewidmet wurde, folgt ein kurzer Überblick über die prägnantesten Unterschiede zwischen NXT und EV3 sowie deren Ausstattungsmerkmale.

	LEGO Mindstorms NXT	LEGO Mindstorms EV3
Education Set Umfang	431 Teile, darunter: - Recheneinheit - drei Motoren (gleich) - fünf Sensoren (Ultraschall-, 2 Tast-, Lautstärken-, Lichtsensor(en)) (siehe Abbildung 5)	541 Teile, darunter: - Recheneinheit - drei Motoren (2 groß, 1 klein) - fünf Sensoren (Ultraschall-, 2 Tast-, Beschleunigungs-, Licht/Farb-sensor(en)) (siehe Abbildung 6)
Recheneinheit	Atmel 32-Bit ARM (48 MHz) 256 KB FLASH-RAM 64 KB RAM	ARM9 (300MHz) 16 MB Flash 64 MB RAM
Bildschirm	LCD Matrix, monochrome 100 x 64 Pixel	LCD Matrix, monochrome 178 x 128 Pixel
Sensoranschlüsse	4	4
Motoranschlüsse	3	4
Tasten	4	6 (mit Hintergrundlicht)
Akku	nicht enthalten	enthalten
LEGO Software	EV3 und ältere	EV3
Kommunikation	USB 2.0, Bluetooth	Micro SD (bis zu 32 GB), USB 2.0, Bluetooth

Tabelle 4 – Vergleich wichtiger Unterschiede von LEGO Mindstorms NXT und LEGO Mindstorms EV3^{40 41}

Der Grundfunktionsumfang kann durch die Anschaffung weiterer Motoren und Sensoren erweitert werden. Neben den im LEGO Mindstorms Education EV3 Basis-Set enthaltenen Sensoren bietet LEGO den NXT-Geräuschsensor und einen Temperatursensor an. Diese Sensoren sind mit dem EV3 verwendbar⁴². In Kooperation mit der Firma Vernier Software & Technology wurde für den LEGO Mindstorms NXT ein Vernier Sensor Adapter entwickelt, der den Anschluss einer Vielzahl von Vernier-Sensoren wie zum Beispiel ph-, Magnetfeld-, UVB- und

⁴⁰ vgl. Robot², <http://robotsquare.com/2013/07/16/ev3-nxt-compatibility/>, (28.01.2014)

⁴¹ vgl. Bot Bench, <http://botbench.com/blog/2013/01/08/comparing-the-nxt-and-ev3-bricks/>, (28.01.2014)

⁴² vgl. LEGO, <http://shop.legoeducation.com/de/category/lego-mindstorms-education-nxt-10/>, (28.01.2014)

Sauerstoffsensoren ermöglicht. Diese sind mit dem EV3 kompatibel⁴³. Durch die Vielzahl an verwendbaren Sensoren kann ein breites Unterrichtsfeld abgedeckt werden. Die Kosten für entsprechende LEGO-Sensoren liegen etwa im Bereich von 24 bis 42 Euro. Lediglich der Vernier-Adapter ist mit höheren Kosten verbunden. Er kommt auf etwa 83 Euro. Kompatible Vernier-Sensoren kommen separat auf etwa 27 bis 101 Euro^{44 45}. Der Aufbau sowie die meisten Sensoren des LEGO Mindstorms NXT und des LEGO Mindstorms EV3 sind unten dargestellt.



Abbildung 5 – NXT und Komponenten⁴⁶

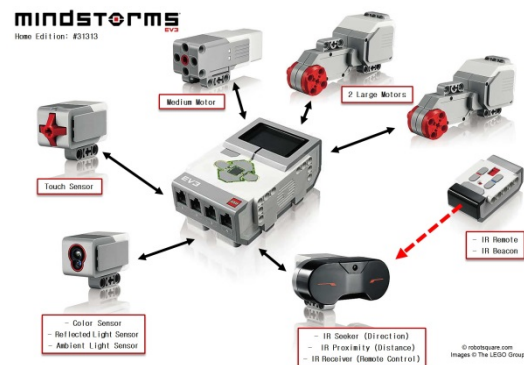


Abbildung 6 – EV3 und Module⁴⁷

Für die Entwicklung von Programmen steht den Anwendern und Anwenderinnen bei der Verwendung des LEGO EV3 die ikonische LEGO Mindstorms Education EV3 Software zur Verfügung. Diese muss allerdings zusätzlich erworben werden und kommt einzeln auf etwa 120 Euro. Eine Schullizenz ist für etwa 350 Euro erhältlich. In den Sets für 16 beziehungsweise 30 Schüler ist die Schullizenz bereits im Preis inbegriffen.

Auch durch Verwendung der Sprache ROBOT C kann bald auf C-Basis entwickelt werden, die Lizenzen dafür werden allerdings ebenfalls kostenpflichtig sein und sich im Bereich von etwa 60 (Einzelplatz) bis 440 (30 Plätze) Euro bewegen⁴⁸.

Durch die Verwendung der leJOS Firmware kann auch kostenlos auf Java Basis programmiert werden⁴⁹. Die Möglichkeit, den Roboter mit Python zu programmieren,

⁴³ vgl. Vernier, <http://www.vernier.com/til/3030/>, (28.01.2014)

⁴⁴ vgl. Vernier, <http://www.vernier.com/products/books/stem/>, (28.01.2014)

⁴⁵ vgl. Vernier, <http://www.vernier.com/products/books/stem2/>, (28.01.2014)

⁴⁶

http://www.educatec.ch/about/presse/LEGO_MINDSTORMS_NXT/LEGO_MINDSTORMS_NXT_Sys_, (09.02.2014)

⁴⁷ <http://mindstormsev3robots.com/wp-content/uploads/2013/02/EV3Hardware-large.jpg>, (09.02.2014)

⁴⁸ vgl. ROBOTC, <http://www.robotc.net/purchase/rvw/rvw-lego.php>, (28.01.2014)

ist ebenfalls gegeben. Vorausgesetzt, die LEGO EV3 Community entwickelt sich ähnlich wie die des LEGO NXT, so darf angenommen werden, dass bald mit weiteren Programmiersprachen gearbeitet werden kann.

Die LEGO Mindstorms Serie ist am längsten als Bindeglied zwischen Robotik und dem Schulunterricht existent und fällt, als jahrelang erprobtes System und von einer umfassenden Community gestützt, durch guten Support und Kompatibilität positiv auf. Vorteilhaft ist bei den LEGO-Modellen auch, dass LEGO den meisten Kindern und Jugendlichen ein bekannter Begriff ist und viele von ihnen schon Kontakt zu diesem bausteinbasierten Spielzeug hatten. Es kann angenommen werden, dass Schülerinnen und Schüler - da bereits im Umgang mit LEGO geübt - keine lange Vorbereitungsphase benötigen, und dass LEGO eher mit Spielzeug assoziiert wird als mit dem bekannten Schulalltag [vgl. 12 S 46].

7.3.2 Fischertechnik ROBO TX Training Lab

Das Fischertechnik ROBO TX Training Lab wurde für den Vergleich ausgewählt, weil es vom Umfang den anderen modularen Robotik-Sets am besten entspricht. Allerdings ist in diesem Baukastenset noch kein Akku-Set enthalten, dieses muss separat gekauft werden. Die Steuerungssoftware ROBO PRO von Fischertechnik ist für einen Arbeitsplatz enthalten, eine Schullizenz für die Verwendung an allen Computern einer Bildungseinrichtung ist ebenfalls getrennt erhältlich. Laut Hersteller kostet das ROBO TX Training Lab etwa 300 Euro⁵⁰ UVP⁵¹. Wird ein Akku-Set mitkalkuliert, erhöht sich der Preis um etwa 60 Euro⁵² UVP und kommt somit auf etwa 360 Euro pro Set. Genügt die enthaltene Softwarelizenz je Robotersystem nicht für den Unterricht, so kann die Schullizenz für etwa 110 Euro⁵³ UVP erworben werden.

Der Bausatz besteht aus 310 Einzelteilen und ist derart konzipiert, dass der Bau elf verschiedener Basismodelle ermöglicht wird (Beispielanwendung siehe Abbildung 8). Dabei handelt es sich um stationäre sowie mobile Roboter. Durch den modularen

⁴⁹ vgl. leJOS, <http://sourceforge.net/projects/lejos/>, (28.01.2014)

⁵⁰ vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-36/usetemplate-2_column_pano/, (28.01.2014)

⁵¹ unverbindliche Preisempfehlung

⁵² vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-36/40_read-6/usetemplate-2_column_pano/usetemplate-2_column_pano/, (28.01.2014)

⁵³ vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-40/usetemplate-2_column_pano/, (28.01.2014)

Aufbau ist es natürlich möglich, eine Vielzahl eigener Ideen umzusetzen. Obwohl es sich bei dem Fischertechnik ROBO TX Training Lab um ein modulares Modell handelt, müssen hier elektrische Komponenten durch Verbindung mitgelieferter Anschlusskabel komplettiert werden. Dazu ist es nötig, die Isolierung der Kabel zu entfernen und diese mit den entsprechenden Steckern zu verschrauben (entsprechendes Werkzeug ist im Kit enthalten). Einerseits hält sich dieser Fertigungsaufwand je Modell in einem überschaubaren Bereich, andererseits summiert sich dieser Aufwand beim Einsatz mehrerer Modelle. Dadurch, dass die Kabel selbständig angebracht werden müssen, bietet sich allerdings auch der Vorteil, abgenützte Kabel auszutauschen, Kabellängen zu variieren und leichter Komponenten von Drittanbietern anzuschließen.

Der Bausatz kommt ausgestattet mit zwei Getriebemotoren, einem kleineren XS-Motor, zwei Tastsensoren sowie einem Infrarot-Spursensor, in welchem zwei integrierte Sensoren verbaut sind. Dies ermöglicht beispielsweise einer Linie zu folgen. Allerdings kann durch diese Bauweise die Distanz zwischen den Sensoren nicht verändert werden, die Linienbreite muss daher entsprechen. Alternativ ist der Einsatz eines zweiten Sensors des gleichen Typs erforderlich.

Der ebenfalls enthaltene ROBO TX Controller (siehe Abbildung 7), der für die Steuerung des Robotermodells zuständig ist, bietet zur Programmierung und Kommunikation mit anderen Systemen zwei USB-Ports (normal und mini) und eine Bluetooth-Schnittstelle. Er verfügt über acht Universaleingänge für den Sensoranschluss, sowie vier Motorausgänge⁵⁴. Es folgt eine Detailübersicht über die technischen Spezifikationen:

- Gehäuse (90x90x15mm, Gewicht: 90g), Anbaumöglichkeiten an 5 Seiten
- 32-bit ARM 9 Prozessor (200MHz)
- Speicherkapazität: 8 MB RAM, 2 MB Flash
- 8 Universaleingänge: Digital/Analog 0-9VDC, Analog 0-5 kOhm
- 4 schnelle Zählengänge: Digital, Frequenz bis 1kHz
- 4 Motorausgänge 9V/250mA: Geschwindigkeit stufenlos regelbar, kurzschlussfest, alternativ 8 Einzelausgänge

⁵⁴ vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-36/usetemplate-2_column_pano/, (22.01.14)

- 2 Erweiterungsanschlüsse: I2C und RS 485 zur Kopplung weiterer ROBO TX Controller
- Display: 128x64 Pixel, monochrom
- USB-Schnittstelle: USB 2.0 (1.1 kompatibel), max. 12Mbit, inkl. Mini USB-Buchse⁵⁵

Zusätzlich kann der Bausatz mit weiteren Sensoren und Motoren ausgestattet werden. Dies soll auch bei Verwendung verschiedener Komponenten von Drittanbietern funktionieren. Fischertechnik selbst bietet neben den enthaltenen Sensoren auch, separat erhältlich, einen Ultraschallsensor und einen Farbsensor an. Zur Erweiterung können unterschiedliche Motoren erstanden werden⁵⁶. Der Preis der Erweiterungskomponenten (Sensoren, Motoren) von Fischertechnik bewegt sich im Bereich von etwa 15 bis 35 Euro⁵⁷.



Abbildung 7- ROBO TX Controller⁵⁸

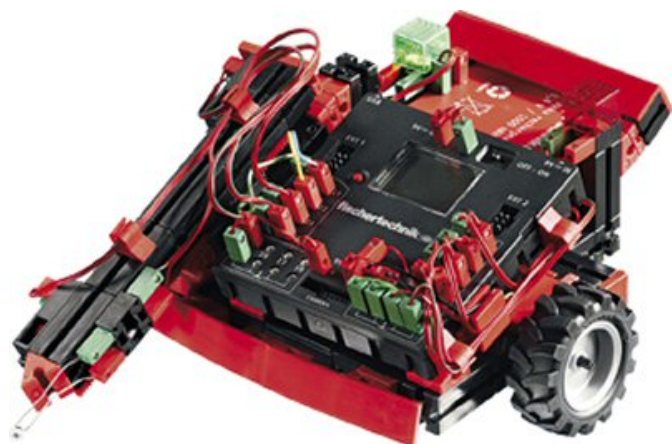


Abbildung 8 – Fischertechnik Beispiel eines mobilen Roboters⁵⁹

Zur Programmierung bietet Fischertechnik die - im Lieferumfang für einen Arbeitsplatz enthaltene - Oberfläche ROBO Pro an. Dabei handelt es sich um eine ikonische Programmiersprache, welche sich laut Herstellerangaben ab einem Alter

⁵⁵ vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-34/usetemplate-2_column_pano/, (22.01.14)

⁵⁶ vgl. Knobloch Electronic, <http://www.knobloch-gmbh.de/>, (28.01.2014)

⁵⁷ vgl. Knobloch Electronic, <http://www.knobloch-gmbh.de/wbc.php?&tpl=pr1010produktliste.html&rid=128>, (28.01.2014)

⁵⁸ http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-36/usetemplate-2_column_pano/, (22.01.2014)

⁵⁹ <http://stefanbrunner.com/fischertechnik-tx-outputs/>, (22.01.2014)

von zehn Jahren eignet. Des Weiteren bietet der Hersteller auf der Firmenseite Compiler an, durch die der ROBO TX Controller auch mit weiteren Sprachen programmiert werden kann. C, C++, C#, Visual Basic und Java seien hier exemplarisch genannt, eine vollständige Übersicht folgt im Kapitel 8.3: „Robotersysteme und unterstützte Programmiersprachen im Überblick“⁶⁰ [vgl. 27].

Bei diesem Modell von Fischertechnik handelt es sich um einen modularen Roboter, welcher dem Nutzer beziehungsweise der Nutzerin etwas handwerkliches Geschick abverlangt. Abgesehen davon, dass die Verbindungskabel selbstständig mit den Komponenten verbunden werden müssen, ist es bei diesem Roboter etwas kompliziert, bei vielen angeschlossenen Modulen den Überblick zu behalten (siehe Abbildung 8). Dies kann sowohl als Vor- wie auch als Nachteil gesehen werden. Der Roboter-Kit ist zwar etwas aufwendiger in seiner Verwendung als andere modulare Systeme, bietet dafür jedoch offensichtlichere Einblicke in den Aufbau der Schaltungen. Durch die Verwendung von Steckverbindern, in welche die entsprechenden Anschlusskabel geschraubt werden können, ist die Nutzung von Drittanbieter-Komponenten bei diesem Modell einfacher als bei anderen modularen Robotern. Dies kann sich auch positiv auf die Anschaffungskosten einzelner Sensoren und Motoren auswirken.

7.3.3 VEX Robotics Super Kit

Der VEX Robotics Super Kit der Produktreihe VEX IQ von der Firma VEX Robotics, der erst seit Ende 2013 auf dem Markt ist, ist im Vergleich der modularen Modelle das am umfangreichsten ausgestattete Set. Seine Entwicklung zielte auf die Anwendung im Bildungssektor ab und soll im Einsatz in MINT-Fächern die Wissensvermittlung unterstützen. Laut Hersteller eignet sich dieses Robotermodell für Schülerinnen und Schüler ab einem Alter von acht Jahren und kann dementsprechend in jeder Schulstufe der höheren Schulen eingesetzt werden. Der VEX Robotics Super Kit kommt mit allen erforderlichen Komponenten und umfasst über 850 Teile. Für einen Preis von etwa 280 Euro handelt es sich um das günstigste Modell der verglichenen modularen Systeme.

Das Set beinhaltet einen Beschleunigungssensor, zwei Tastsensoren, einen Ultraschallsensor, einen Farbsensor und zwei Berührungssensoren mit integriertem

⁶⁰ vgl. Fischertechnik, <http://www.fischertechnik.de/home/downloads/computing.aspx>, (28.01.2014)

dreifärbigem LED (zur Interaktion mit dem Anwender) und deckt somit ein großes Anwendungsfeld ab. Zur Fortbewegung und weiteren Interaktion mit der Umwelt werden vier Motoren mitgeliefert. Außerdem enthält der Roboter Kit als einziger der Modulare in seiner Basisausstattung eine Fernsteuerung in Form eines Gamepads (auch hierfür ist bereits ein Akku enthalten). Die Module dieses Roboters werden mit speziellen Kabeln, die aus je einem Strang bestehen, mit einem der zwölf Anschlussoptionen des Controllers (siehe Abbildung 9) verbunden. An diesen können jeweils Sensoren sowie Motoren angeschlossen werden, um flexibel Modelle zu konstruieren (siehe Abbildung 10)⁶¹. Weitere Sensoren und Motoren können für etwa 5 bis 28 Euro geordert werden⁶². Programmiert wird dieses Modell per USB-Anschluss.

Hier eine Detailübersicht über die technischen Spezifikationen:

- Gehäuse (110x76x42mm, Gewicht 322g)
- ARM Cortex-M4 Processor (100 MHz)
- Speicherkapazität: 256K Flash, 32K RAM
- 12 Smart Ports für Sensoren und Motoren
- USB 2.0 Port für die Programmierung
- Tether Port für die kabelgebundene Controllerverwendung/Controller laden⁶³



Abbildung 9 – VEX IQ Controller⁶⁴
Kit

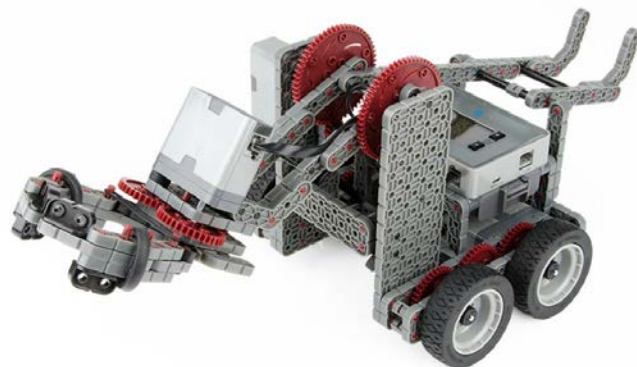


Abbildung 10– Anwendungsbeispiel des VEX IQ Super⁶⁵

⁶¹ vgl. Vex Robotics, <http://www.vexrobotics.com/vexiq/products/super-kit.html>, (28.01.2014)

⁶² vgl. Vex Robotics, <http://www.vexrobotics.com/vexiq/products>, (28.01.2014)

⁶³ vgl. Vex Robotics, <http://www.vexrobotics.com/vexiq/products/brain-g.html>, (28.01.2014)

⁶⁴ <http://www.vexrobotics.com/vexiq>, (09.02.2014)

⁶⁵ <http://www.vexrobotics.com/vexiq/products/super-kit.html>, (22.01.2014)

Für die Programmierung des VEX IQ Rechners bietet der Hersteller eine Umgebung an, die auf Scratch basierende, grafische Oberfläche Modkit für VEX, welche über die Herstellerseite kostenlos heruntergeladen werden kann⁶⁶. Die Firma Robomatter Inc. bietet des Weiteren die kompatible, C-basierte Programmiersprache ROBOTC an, welche ebenfalls von VEX Robotics beworben wird. Diese kostet allerdings für 30 Arbeitsplätze pro Jahr etwa 222 Euro und verlangt ein Firmware Update. Die eingeschränkte Auswahl kompatibler Programmiersprachen könnte daraus resultieren, dass der VEX IQ Kit erst seit Kurzem erhältlich ist. Aufgrund der Entwicklungsdynamiken bezüglich der Adaption weiterer existierender Programmiersprachen für ähnliche vorhandene Robotersysteme ist anzunehmen, dass in Zukunft auch für den VEX IQ weitere Programmiersprachen anwendbar werden. Gegenwärtig bietet dieses Modell durch die geringe Auswahl an Entwicklungssprachen nur eingeschränkte Einsatzoptionen im Schulunterricht. Genügen diese (zumindest vorerst), so bietet der VEX Robotics Super Kit ein umfangreiches Teilesortiment, das sich hardware-seitig für ein breites Feld von Verwendungen einsetzen lässt.

7.3.4 Modulare Modelle Fazit

Im Folgenden werden die Vor- und Nachteile der verglichenen modularen Robotermodelle von LEGO, Fischertechnik und VEX Robotics dargelegt. Auch wenn es sich bei allen drei Modellen um Robotersets derselben Kategorie handelt, sind die Unterschiede erheblich. Die Modelle von Fischertechnik und LEGO sind schon länger auf dem Markt, wodurch es jeweils große Communitys gibt, welche neben Beispielmustern und Programmen auch Lehr- und Lernmaterial für Schulen bereitstellen. Mit diesem Vorteil kann der VEX IQ (noch) nicht dienen, aber auch dessen Community scheint zunehmend größer zu werden.

Als Vorteil des Fischertechnik-Sets können seine anschaulichen Schaltungen gesehen werden. Werden diese aber nicht zur Veranschaulichung bestimmter Inhalte benötigt, so können sie auch zu unnötigen Verwirrungen führen. Dieses Modell ist auch das einzige Roboterset des Vergleichs, das nicht ohne die Verwendung von Werkzeug auskommt, auch wenn dieses nur selten benötigt wird. Der LEGO EV3 und der VEX IQ sind in ihrer Anwendung etwas weniger komplex, da hier einerseits

⁶⁶ vgl. Vex Robotics, <http://www.vexrobotics.com/vexiq/products>, (28.01.2014)

kein Werkzeug verwendet werden muss und andererseits die aufbaubaren Schaltungen übersichtlicher gestaltet sind (siehe Abbildungen 5, 6, 10).

Für die Programmierung bietet derzeit das Fischertechnik-Modell das umfangreichste Sortiment kompatibler Sprachen an. Dies dürfte daran liegen, dass dieses Modell am längsten am Markt ist. Vorausgesetzt, weitere Projekte (die jetzt schon erweiterte Programmieroptionen des LEGO NXT ermöglichen) folgen dem Beispiel des leJOS Projekts (das bereits für den Einsatz am LEGO EV3 angepasst wurde), so werden für den LEGO EV3 bald weitere Entwicklungsmöglichkeiten folgen. Welche Pläne es zur Erweiterung des Sortiments kompatibler Sprachen für den VEX IQ gibt, kann zu diesem Zeitpunkt noch nicht abgeschätzt werden. Einen Überblick über die meisten, aktuell verwendbaren Programmiersprachen der verglichenen Systeme ist im Kapitel 8.3: „Robotersysteme und unterstützte Programmiersprachen im Überblick“ zu finden.

Das Ausmaß möglicher Aufbauten der nicht elektronischen Komponenten ist bei allen drei Modellen sehr umfangreich, da dem Anwender durch ihre Modularität kaum Grenzen auferlegt werden. Anders verhält sich dies bezüglich elektronischer Bauteile. Auch wenn sich im Internet für jeden der verglichenen Roboter Anleitungen für Eigenbauten finden, so unterscheidet sich diesbezüglich das Fischertechnik-Modell von seiner Konkurrenz. Durch die verschraubten Leitungen ist beispielsweise der Anschluss eines Motors von einem anderen Anbieter kein Problem, sofern die entsprechenden Leistungsdaten entsprechen. Auch handelsübliche Tastschalter können einfach verwendet werden, indem diese mittels Schrauben in Schaltkreise integriert werden, da diese nur zwei Zustände (Strom fließt, Strom fließt nicht) kennen. Für die Integration aufwändigerer Sensoren sollten entsprechende Kenntnisse vorhanden sein. Der Aufbau komplexer Schaltungen sowie der Eigenbau von Komponenten, die sich in Verbindung mit dem LEGO EV3 oder dem VEX IQ eignen würden, wird hier nicht im Detail behandelt. Ein Überblick über die Sensormodule, die von den jeweiligen Herstellern angeboten werden, inklusive einer Preisübersicht ist in der folgenden Tabelle 5 dargestellt.

	LEGO Mindstorms Education EV3- Basis-Set ⁶⁷	Fischertechnik ROBO TX Training Lab ⁶⁸	VEX Robotics Super Kit ⁶⁹
Set-Preis	386,74€ 16 Schüler Set 4236,39€* 30 Schüler Set 7556,49€*	299,95€	279,99€
Empfohlene (grafische) Basissoftware	118,99€ Schullizenz 416,49€	1 enthalten, 29,95€ Schullizenz 109,85€	Kostenloser Download
Berührungssensor ⁷⁰	2 enthalten, 23,79€	2 enthalten, 5,45€	2 enthalten, 4,99€
Berührungstaster ⁷¹			2 enthalten, 8,99€
Ultraschallsensor	1 enthalten, 41,64€	35,95€	1 enthalten, 27,99€
Lichtsensoren	1 enthalten (Farb- & Lichtsensor), 41,64€ (NXT) ⁷² Lichts. 22,60€	1 enthalten, 5,95€ 1 mit zwei Sensoren enthalten (Spurs.), 16,95€	1 enthalten (Farb- & Lichtsensor), 17,99€
Farbsensoren	(NXT) Farbs. 32,12€		
Beschleunigungssensoren	1 enthalten, 41,64€		1 enthalten, 27,99€
Geräuschsensoren	(NXT) 30,93€		
Temperatursensoren	(NXT) 44,02€	1 enthalten (liefert nur ungefähren Wert - NTC ⁷³), 1,75€	

*Schullizenz enthalten

Tabelle 5 – Übersicht über enthaltene und erhältliche Sensoren der verglichenen modularen Roboter Sets

In Kooperation mit der Firma Vernier wurde für den LEGO Mindstorms NXT eine Vielzahl an zusätzlichen Sensoren (ph Wert, UVB, Gasdruck) kompatibel gemacht. Dafür werden lediglich ein Vernier Sensor Adapter und ein Plug-In für die

⁶⁷ Preise beziehen sich auf die Angaben von <http://shop.legoeducation.com/>, (23.01.2014)

⁶⁸ Preise beziehen sich auf die Angaben von <http://www.knobloch-gmbh.de/>, (23.01.2014)

⁶⁹ Preise beziehen sich auf die Angaben von <http://www.vexrobotics.com/vexiq/products>, (23.01.2014)

⁷⁰ Berührungssensor bezeichnet in diesem Zusammenhang einen Schalter der durch eine Person oder ein Hindernis betätigt werden kann (Schalter)

⁷¹ Berührungstaster bezeichnet in diesem Zusammenhang einen Taster der durch eine Person betätigt werden kann, aber nicht von einem Hindernis ausgelöst wird (Tast-Knopf)

⁷² (NXT) bezeichnet Sensoren die für den LEGO NXT vorgesehen sind, aber auch mit dem EV3 Modell funktionieren

⁷³ NTC: Negativer Temperatur Koeffizient – hat bei 20°C einen Widerstand von 1,5Kilohm, dieser sinkt bei steigender Temperatur

entsprechende grafische LEGO-Software benötigt. Diese können über die Herstellerseiten beider Firmen bezogen werden. Vernier arbeitet bereits an der Kompatibilität dieser NXT-Sensoren für den LEGO Mindstorms EV3⁷⁴. Durch diese Erweiterungsmöglichkeiten eignet sich der LEGO EV3 auch für den Unterricht in den Fächern Chemie (pH-Sensor, Verfolgung chemischer Reaktionen), Physik (Magnetfeldsensor, Magnetismus von Gesteinsproben), sowie Biologie und Umweltkunde (Feuchtigkeitssensor, Bodenmessungen). Ein Vorteil, mit dem seine Konkurrenz nicht aufwarten kann.

Hinsichtlich der Anschaffungskosten für einen Roboter liegt das LEGO Mindstorms Education EV3 Basis-Set deutlich über den verglichenen Alternativprodukten und ist das einzige Set, das keine Software zur Programmierung enthält. Anders verhält sich dies bei der Anschaffung einer größeren Stückzahl. Diesbezüglich ist LEGO die einzige Firma, die Paketpreise anbietet. Bereits bei einem Set-Angebot für 16 Auszubildende ist die Schullizenz enthalten, dieses inkludiert acht Robotersets und vier Erweiterungssets. Preislich liegen die Robotersets von Fischertechnik und VEX Robotics auf einem ähnlichen Niveau. Bei der Wahl eines dieser Modelle sollten daher andere Aspekte im Vordergrund stehen.

Wird für den schulischen Einsatz ein modularer Roboter gewünscht, der den Bausatzmodellen möglichst nahe kommt und für den Aufbau von Schaltkreisen demonstrativ eingesetzt werden soll, dann ist das Fischertechnik ROBO TX Training Lab eine erwägenswerte Überlegung. Weitere Vorteile dieses Systems sind seine verstärkte Kompatibilität mit Produkten anderer Hersteller sowie die geringen Kosten für zusätzliche Sensoren und andere Elemente. Für dieses Set gibt es (gegenwärtig) auch das breiteste Feld an kompatiblen Programmiersprachen. Wird ein weniger komplexes aber modulares Modell gewünscht, so sollten die Modelle von LEGO und VEX Robotics im Fokus liegen. Beide Robotersets sind gut ausgestattet und bieten viele Optionen für Aufbauten an. Das LEGO-Modell bietet durch seine Kombinierbarkeit mit Vernier-Sensoren und das umfangreiche Sortiment separat erhältlicher LEGO-Bausteine das breiteste Anwendungsfeld, kostet aber auch etwas mehr als der VEX IQ Super Kit (das im Set-Umfang auch mehr Teile enthält). Abgesehen von den Anschaffungskosten lassen sich keine besonderen Vorteile des

⁷⁴ vgl. Vernier, <http://www.vernier.com/til/3030/>, (23.01.2014)

VEX IQ nennen. Ist das Budget vorhanden, sprechen mehr der obigen Aspekte für das LEGO EV3 Set.

Abschließend kann festgehalten werden, dass jedes der verglichenen Robotermodelle Vorzüge hat (Fischertechnik - sichtbare Schaltungen, VEX Robotics - umfangreich und billig, LEGO - Kompatibilität) und für das richtige Einsatzgebiet optimalen Nutzen gewährleistet. Durch das umfangreiche Sortiment existenter LEGO-Teile, die Bekanntheit des Systems unter Kindern und Jugendlichen, die große Community, dessen Möglichkeiten zur Koppelung mit Vernier-Sensoren und den geringeren Preis bei größeren Stückzahlen kann das LEGO Mindstorms Education EV3 Basis-Set allgemein empfohlen werden.

7.4 Überblick über ausgewählte Bausatzmodelle

IN diesem Kapitel der vorliegenden Arbeit werden zwei exemplarische Bausatzroboter vorgestellt und deren Vor- und Nachteile besprochen. Im Bereich der Bausatzmodelle existieren mit Abstand die meisten Produkte. Diese eignen sich besonders gut, um elektrische Schaltungen am Beispiel eines Roboters zu verdeutlichen. Wird der Roboter selbst aber nicht im Vordergrund stehen, sollte eher die Anschaffung eines Fertigmodells oder eines modularen Roboters überlegt werden, da diese durch ihre geringere Komplexität einfacher handzuhaben sind. Im vorangegangenen Kapitel wurde auch das Fischertechnik ROBO TX Training Lab vorgestellt, das zwar einen modularen Roboterbausatz darstellt, aber auch Möglichkeiten zur vereinfachten Veranschaulichung von Schaltungen bietet.

Die schaltplannahe Darstellung des elektronischen Aufbaus ist nur einer der Aspekte der Bausatzmodelle, die als vorteilhaft gesehen werden können. Diese Roboter sind meist kaum eingeschränkt hinsichtlich der Koppelung elektronischer Komponenten. Allerdings ist das entsprechende Fachwissen über Schaltungen und elektrische Bauteile Voraussetzung, es sei denn die Wahl fällt auf eine Erweiterung, die vom Hersteller durch bewiesene Kompatibilität und entsprechende Anleitungen unterstützt wird. Auch der Preis der Bausatzroboter kann Grund für ihre Anschaffung sein. Einfache Modelle sind schon für weniger als 50 Euro erhältlich (zum Beispiel ASURO PRO-BOT128K der Firma C-Control um 44,95€⁷⁵). Die Modelle, die verglichen

⁷⁵ vgl. Conrad Electronics, <http://www.conrad.com/ce/en/product/191919/C-Control-ASUROPRO-BOT128K-Programmable-Robot?ref=searchDetail>, (24.01.2014)

werden, sind etwas teurer. Sie wurden ausgewählt, da der Fokus auf der Eignung für den Schulunterricht und nicht auf dem Preis lag.

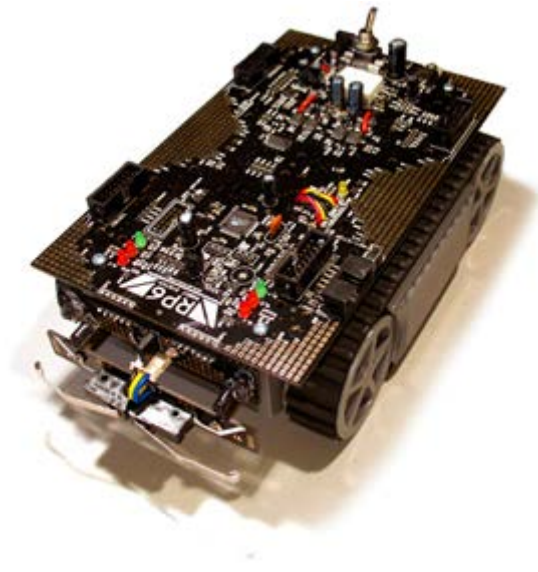
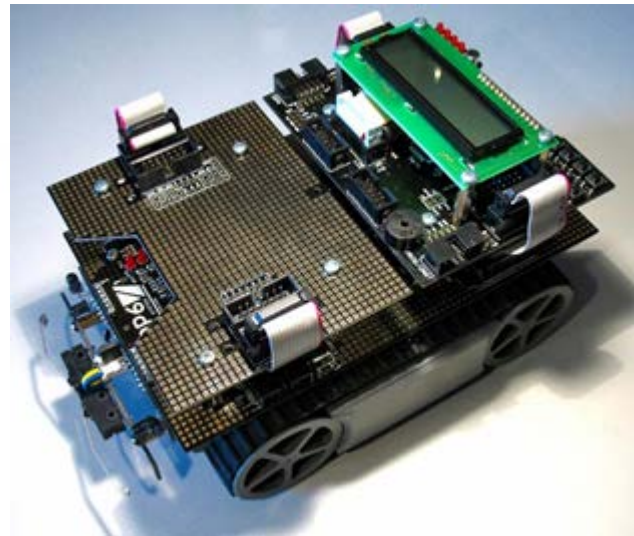
7.4.1 Arexx RP6 V2

Der Arexx RP6 V2 ist ein Produkt der niederländischen Firma Arexx Engineering und wird komplett gefertigt geliefert, um so einen baldigen Einsatz zu ermöglichen. Er wurde dennoch als Bausatz-Roboter kategorisiert, da er als Grundplattform für Erweiterungen konzipiert wurde. Das Modell RP6 V2 ist der Nachfolger des RP6, der bis Anfang 2012 vertrieben wurde. Dieser Roboter soll laut Herstellerangaben auch von Anfängern verwendbar sein. Eine altersbezogene Abgrenzung wird nicht angegeben.

Er liegt mit Anschaffungskosten von etwa 140 Euro⁷⁶ über dem Preis der oben behandelten Fertigmodelle, allerdings deutlich unter dem Preisdurchschnitt der verglichenen modularen Modelle. Da dieser Roboter mit C programmiert wird, ist keine spezielle Software notwendig. Die entsprechenden RP6-Funktionsbibliotheken für das Ansprechen der Sensoren und anderer Komponenten sind von der Herstellerwebseite kostenlos herunterladbar.

Im Auslieferungszustand hat der Roboter zwei Tastsensoren, die an der Vorderseite angebracht sind und Hindernisse detektieren, zwei Lichtsensoren, die sich ebenfalls an der Vorderseite befinden, sowie zwei Infrarot LEDs und einen passenden Empfänger, um Hindernisse vor dem Roboter zu detektieren. Zur Darstellung einer Ausgabe befinden sich sechs LEDs an der Oberseite. Für den Anschluss weiterer Module stellt das Robotersystem insgesamt vier Erweiterungsanschlüsse bereit, von denen jeweils zwei vollständig miteinander verbunden, sind um entsprechende Erweiterungen sowohl vorne wie auch hinten am Roboter anbringen zu können (siehe Abbildung 11) [vgl. 22 S 20 ff.].

⁷⁶ vgl. Conrad Electronics, <http://www.conrad.at/ce/de/product/191584/Arexx-RP6-V2-Robotersystem?ref=searchDetail>, (24.01.2014)

Abbildung 11 - Auslieferungszustand⁷⁷Abbildung 12 - Anwendungsbeispiel⁷⁸

Für die Programmierung steht dem Anwender und der Anwenderin nur die Sprache C zur Verfügung, eine häufige Einschränkung im Bereich der Bausatzroboter. Als Erweiterungen bietet der Hersteller diverse Platinen an, die meist mehrere Funktionen unterstützen. Diese reichen von einfacheren Modulen wie Displays bis hin zur rechenleistungserweiternden Platine mit integrierter Speichererweiterung, einem Mikrofon und weiteren LEDs, Anschlussports und ähnlichen Zusatzoptionen⁷⁹ (siehe Abbildung 11). Für den Schuleinsatz eignet sich vor allem auch die separat erhältliche Experimentierplatine, auf der Versuchsschaltungen aufgebaut werden können, ohne die Hauptplatine des Roboters direkt zu verwenden (oder zu beschädigen) (siehe Abbildung 11). Für einen Preis von etwa 21 Euro⁸⁰ kann diese nach Abnutzung oder Schaden kostengünstig ersetzt werden.

Dieses Bausatzrobotermodell bietet den Vorteil, dass es in einem funktionstüchtigen Grundaufbau ausgeliefert wird. Dadurch wird Vorbereitungszeit für eine erste Anwendung gespart. Durch die kompatiblen Platinen des selben Herstellers, die zusätzlich angebaut werden können, ist es möglich, den Funktionsumfang auf recht einfache Weise zu erweitern. Eine Vielzahl freier Lötstellen an der Hauptplatine (und der erhältlichen Experimentierplatine) erlauben den Aufbau unzähliger Schaltungen. Dadurch ist dieses Modell für den Schuleinsatz besonders gut geeignet.

⁷⁷ <http://www.arexx.com/forum/viewtopic.php?t=370>, (24.01.2014)

⁷⁸ <http://www.arexx.com/rp6/html/de/acc.htm>, (25.01.2014)

⁷⁹ vgl. Arexx Engineering, <http://www.arexx.com/rp6/html/de/acc.htm>, (28.01.2014)

⁸⁰ vgl. Conrad Electronics, <http://www.conrad.at/ce/de/product/191537/Arexx-RP6-Experimentierplatine/?ref=detview1&rt=detview1&rb=1>, (24.01.2014)

Eingeschränkt in dessen Anwendung werden die Nutzer und Nutzerinnen primär durch den Umfang der Programmiermöglichkeiten. Alternativen zur Sprache C sind bislang nicht bekannt.

7.4.2 Boe-Bot Robot Kit

Der Boe Bot Robot Kit der Firma Parallax ist ein Bausatz für einen kleinen, fahrenden Roboter. Er wird hauptsächlich in Einzelteilen ausgeliefert und erfordert vor der Erstverwendung eine Aufbauzeit von ein bis zwei Stunden. Er ist ab einem Alter von 13 Jahren geeignet (Herstellerangaben). Um den Aufbau und die ersten Schritte zu erleichtern, kann auf der Herstellerseite ein übersichtliches Skript heruntergeladen werden, das eine Schritt-für-Schritt-Anleitung enthält. Diese ist in mehreren Sprachen verfügbar, darunter Deutsch und Englisch. Die Firma Parallax vertreibt den Roboterbausatz für umgerechnet rund 124 Euro, dieser Preis sinkt gestaffelt auf bis zu rund 99 Euro.

An seiner Oberseite hat der Roboter eine Steckplatine, in welche Schaltungen ohne zu löten und ohne spezielle Verbindungsstücke einfach eingesteckt werden können (siehe Abbildung 13). Um diese verwenden zu können, befinden sich daneben zwei Anschlussleisten, eine für die Stromversorgung und eine mit 16 Eingabe/Ausgabe-Anschlüssen. Daneben befinden sich vier Anschlüsse für Servomotoren, von denen zwei für den Fahrbetrieb benötigt werden [vgl. 28 S 2 f.]. Im Bausatz enthalten sind neben den nötigen Komponenten für die Grundplattform auch noch zwei Tastsensoren, zwei Lichtsensoren und zwei Infrarot-Abstandssensoren, sowie zur Ausgabe zwei LEDs und ein Piezolautsprecher.

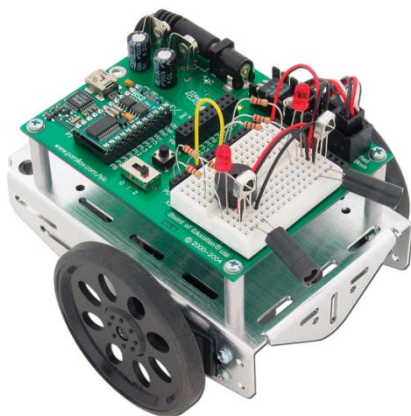


Abbildung 13 – Boe Bot Basisausstattung⁸¹

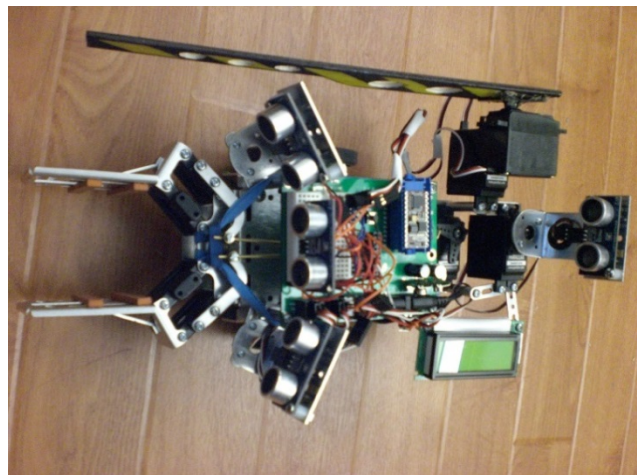


Abbildung 14 – Boe Bot Anwendungsbeispiel⁸²

⁸¹ <http://www.parallax.com/product/28832>, (25.01.2014)

Für die Programmierung des Boe Bots wird gegenwärtig nur PBASIC angeboten⁸³. PBASIC steht für Parallax BASIC und ist eine auf BASIC basierende Sprache, die um spezielle Eingabe- und Ausgabefunktionen erweitert wurde [vgl. 29 S 1].

Die Firma Parallax bietet für den umfangreicheren Einsatz des Robotermodells viele Erweiterungen an. Beispielhaft für optionale Sensoren seien zwei unterschiedliche Infrarot-Sensorsets zur Linienverfolgung^{84 85}, ein Ultraschallsensor⁸⁶ sowie ein Beschleunigungssensor⁸⁷ genannt. Für erweiterte Interaktionsmöglichkeiten mit der Umwelt kann das Robotermodell auch mit Ketten oder spinnenähnlichen Beinen⁸⁸ sowie mit einem Greifarm bestückt werden.

Die Stärke dieses Robotermodells liegt einerseits an seiner Steckplatine, die ohne zu löten den Aufbau von Schaltungen ermöglicht, welche anschließend einfach wieder entfernt werden können. Andererseits ist es vorteilhaft, dass die Herstellerfirma ein umfangreiches Sortiment an Erweiterungen anbietet, deren Kompatibilität gewährleistet ist. Als nachteilig kann die Einschränkung auf die Programmiersprache BASIC gesehen werden.

7.4.3 Bausatzmodelle Fazit

Die für den Vergleich ausgewählten Robotermodelle RP6 von Arexx und Boe Bot von Parallax haben sehr unterschiedliche Stärken. Welches dieser beiden Modelle sich für den Unterrichtseinsatz besser eignet, ist abhängig von dem zu vermittelnden Inhalt. Soll der Roboter von den Schülerinnen und Schülern selbst gebaut werden, spricht dies für den Boe Bot, der (teilweise) erst fertiggestellt werden muss. Soll dies nicht Teil des Unterrichts sein, so ist der RP6, der schon als fertig montierte Grundplattform geliefert wird, eventuell die bessere Wahl. Ein weiterer, wichtiger Unterschied ist die Vorgehensweise für den Aufbau von Schaltungen. Der Boe Bot bietet durch seine Steckplatine die Möglichkeit, etwaige Experimente ohne Löten umzusetzen, wohingegen beim RP6 gelötet werden muss (außer es wird auch hier eine Steckplatine integriert).

⁸² <http://classic.parallax.com/tabid/651/Default.aspx>, (25.01.2014)

⁸³ vgl. Parallax, <http://www.parallax.com/product/28832>, (28.01.2014)

⁸⁴ vgl. Parallax, <http://www.parallax.com/product/28108>, (28.01.2014)

⁸⁵ vgl. Parallax, <http://www.parallax.com/product/28034>, (28.01.2014)

⁸⁶ vgl. Parallax, <http://www.parallax.com/product/28015>, (28.01.2014)

⁸⁷ vgl. Parallax, <http://www.parallax.com/product/28017>, (28.01.2014)

⁸⁸ vgl. Parallax, <http://www.parallax.com/catalog/robotics/add-ons>, (28.01.2014)

Für beide Modelle bieten die Hersteller Erweiterungsplatinen mit speziellen Funktionen sowie ein umfangreiches Sortiment an Sensoren an. Soll ein bestimmtes Projekt umgesetzt werden, kann die Auswahl bereitstehender Komponenten eine Entscheidungsgrundlage darstellen. Durch die Möglichkeit, selbstgebaute Schaltkreise und Platinen zu integrieren, können beide Modelle mit dem notwendigen Knowhow auch ohne vorgefertigte Module in ihrem Funktionsumfang erweitert werden.

Beide Roboter bieten nur die Programmierung mit jeweils einer Sprache an. Im Fall des Arexx RP6 ist dies C, im Fall des Parallax Boe Bot handelt es sich um BASIC. Somit kann auch der Wunsch, eine bestimmte Programmiersprache zu verwenden, die Wahl des Modells beeinflussen.

Durch den komplexen Aufbau der verglichenen Bausatzrobotersysteme scheinen sich diese nur bedingt für den schulischen Einsatz zu eignen. Dies bestätigen auch die eingeschränkten Möglichkeiten zur Programmierung. Für eine Verwendung im Rahmen hardware-naher Projekte und eines Unterrichts in dessen Mittelpunkt der Roboter als Lernobjekt steht, können diese Modelle gut eingesetzt werden. Für einen Unterricht, in dessen Rahmen Roboter als unterstützendes Medium Verwendung finden sollen, wäre ein Modell einer weniger komplexen Kategorie vorteilhafter.

Es gilt grundsätzlich zuerst zu überlegen, welcher Inhalt über welchen Weg vermittelt werden soll, und anschließend erst über die Wahl eines geeigneten Roboters nachzudenken. Da für derartige Überlegungen meist die Programmiersprachen, die verwendet werden sollen, eine wichtige Rolle spielen, werden die jeweils nutzbaren Programmiersprachen der einzelnen Modelle behandelt.

8 Programmiersprache als Kriterium zur Systemauswahl

Das vorliegende Kapitel setzt sich mit den Programmiersprachen auseinander, die in Verbindung mit einem oder mehreren Robotern des vorangegangenen Vergleichs verwendet werden können (siehe Kapitel 8.3: „Robotersysteme und unterstützte Programmiersprachen im Überblick“). Diese können grundsätzlich in zwei Typen gegliedert werden: einerseits grafische Programmiersprachen und andererseits textuelle Programmiersprachen.

Textuelle Programmiersprachen basieren größtenteils auf der menschlichen Sprache und ermöglichen die Erstellung eines Programms durch dessen Beschreibung, folgend einer bestimmten Syntax der jeweiligen verwendeten Programmiersprache.

Bei grafischen Programmiersprachen handelt es sich um Programmiermöglichkeiten in einer bestimmten Entwicklungsumgebung, in der durch das Anordnen und/oder Verbinden einzelner grafischer Segmente und Icons der gewünschte Programmablauf determiniert wird. Die meisten grafischen Sprachen benötigen im gesamten Entwicklungsprozess keine beziehungsweise kaum zusätzliche textuelle Eingaben, es gibt aber auch Entwicklungsumgebungen, in denen eine kombinierte Programmierung (grafisch und textuell) vorgesehen ist.

Ob sich eine textuelle oder eine grafische Programmierung besser für den Einsatz an Schulen eignet, ist hauptsächlich von den jeweiligen Vorkenntnissen, dem umzusetzenden Programm und dem gewünschten Unterrichtsziel, das durch die Programmierung erreicht oder gestützt werden soll, abhängig.

Sind zum Beispiel benötigte Vorkenntnisse für eine textuelle Programmierung nicht vorhanden und ist deren Erlangung auch nicht Ziel des Unterrichts, so kann es von Vorteil sein, eine einfacher zu erlernende, grafische Sprache einzusetzen, um das eigentliche Lernziel im Fokus zu behalten. Es ist nicht nur schwer, die menschlichen Sprachkonzepte in Algorithmen zu übersetzen, die ein Computer versteht, es muss auch auf die richtige Syntax des Programms geachtet werden. Grafische Programmiersprachen können dieses Syntaxproblem entschärfen und gleichzeitig Kindern eine ansprechendere Umgebung bereitstellen [vgl. 05 S 126 f.]. Ein weiterer wichtiger Punkt ist eine mögliche Entlastung des Gedächtnisses der Schülerinnen und Schüler durch die Verwendung grafischer Sprachen. „*In der Informationsphase*

ist darauf zu achten, die Kapazität des Arbeitsgedächtnisses der Lernenden nicht zu überlasten, worauf die Cognitive Load-Theorie aufmerksam macht. Der Entlastung des Arbeitsgedächtnisses dienen Superzeichen oder Schemata, aber auch Grafiken, Bilder und anschauliche Darstellungen. Nach Paivio hilft die doppelte Kodierung im verbalen und piktoralen Arbeitsgedächtnis, neuen Lehrstoff besser zu verankern.“ [30 S 105]

Icons prägen sich im Sinn einer Reduktion leichter ins Gedächtnis ein und können leichter abgerufen werden. Dieser Vorteil geht bei komplexen Programmen verloren, da infolge der Bildschirmbegrenzung oft nur Teile des Programms angezeigt werden können und dessen Änderung mühsam wird [vgl. 06 S 27]. Folglich kann es ab einem gewissen Umfang des zu erstellenden Programms nachteilig sein, dieses auf Basis einer grafischen Sprache umzusetzen. Je nach Komplexität ist es auch möglich, dass sich das gewünschte Programm kaum oder gar nicht mit einer zur Verfügung stehenden grafischen Umgebung erstellen lässt. In diesem Fall bieten textuelle Sprachen oft mehr Umsetzungsmöglichkeiten und Flexibilität.

Am wichtigsten ist allerdings das gewünschte Lernziel, da sich die Frage nach der geeigneten Sprache nur stellt, wenn das Erlernen der Programmiersprache selbst nicht im Vordergrund steht, da andernfalls keine sinnvolle Wahlmöglichkeit besteht.

8.1 Grafische Programmiersprachen

Die folgenden Kapitel befassen sich mit den grafischen Programmiermöglichkeiten, die in Verbindung mit vorgestellten Robotermodellen verwendet werden können. Da eine detaillierte Behandlung aller Sprachen aufgrund des inhaltlichen Ausmaßes im Rahmen dieser Arbeit nicht möglich ist, werden nur die wichtigsten Umgebungseigenschaften dargelegt. Neben allgemeinen Eckdaten werden auch für Schulen relevante Programmeigenschaften behandelt, wie empfohlenes Alter der Anwender und Anwenderinnen, unterstützte Systemsprachen und Anschaffungskosten. Es wird jeweils kurz beschrieben, wie die Programmierung in der behandelten Umgebung funktioniert, auf eventuelle Besonderheiten der Programme wird ebenfalls eingegangen.

8.1.1 CREATE Lab Visual Programmer

Alter: 8+⁸⁹

Systemsprachen: Englisch

Betriebssystem(e): Windows XP, Vista, 7, 8; Mac OSX 10.6-10.9; Linux Ubuntu⁹⁰

Entwickler/Lizenz: Carnegie Mellon University – CREATE Lab

Basiert auf: nicht bekannt/unabhängig entwickelt

Kosten: kostenloser Download⁹¹

Die Programmiersprache CREATE Lab Visual Programmer wurde, wie auch der Finch, an der Carnegie Mellon Universität entwickelt. Dieser Roboter ist der einzige im Vergleich der vorliegenden Arbeit, der mit dem Visual Programmer kompatibel ist, weshalb hier nur auf diesen Bezug genommen wird. Die ikonische Sprache des CREATE Lab lässt sich auf den meisten gängigen Betriebssystemen verwenden und eignet sich laut Hersteller ab einem Alter von acht Jahren. Da die Entwicklungsumgebung derzeit nur mit englischsprachiger Oberfläche angeboten wird, wären dementsprechende Grundkenntnisse von Vorteil. Durch die bildhafte Programmierung sind diese aber keine zwingende Voraussetzung.

Um mit dem CREATE Lab Visual Programmer einen Roboter zu programmieren, müssen zu Beginn im „Expression Builder“ (siehe Abbildung 15) der Software gewünschte Ausgaben (Expressions) festgelegt werden. Durch das Klicken auf die Darstellung der zu verwendenden Hardware öffnet sich ein entsprechendes Fenster, in dem die möglichen Parameter geregelt werden können. Nachdem eine Ausgabe festgelegt wurde, kann diese gespeichert werden. Diese ist dann für die Programmierung im Fensterbereich „Expressions“ zu sehen.

⁸⁹ vgl. Finch, <http://www.finchrobot.com/Creating%2C%20Compiling%2C%20and%20Running%20Programs%20for%20Finch/using-create-lab-visual-programmer>, (03.02.2014)

⁹⁰ vgl. Finch, <http://www.finchrobot.com/software/create-lab-visual-programmer>, (03.02.2014)

⁹¹ vgl. ebenda

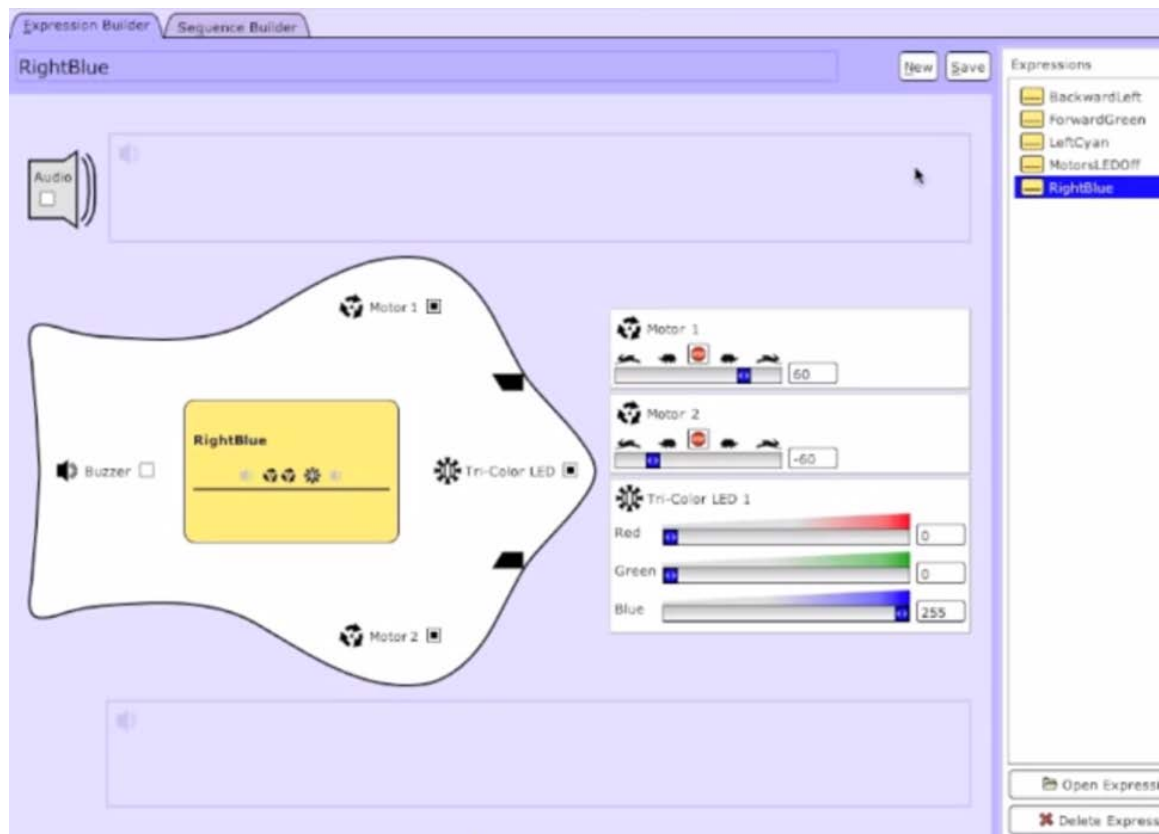


Abbildung 15 - CREATE Lab Visual Programmer Expression Builder⁹²

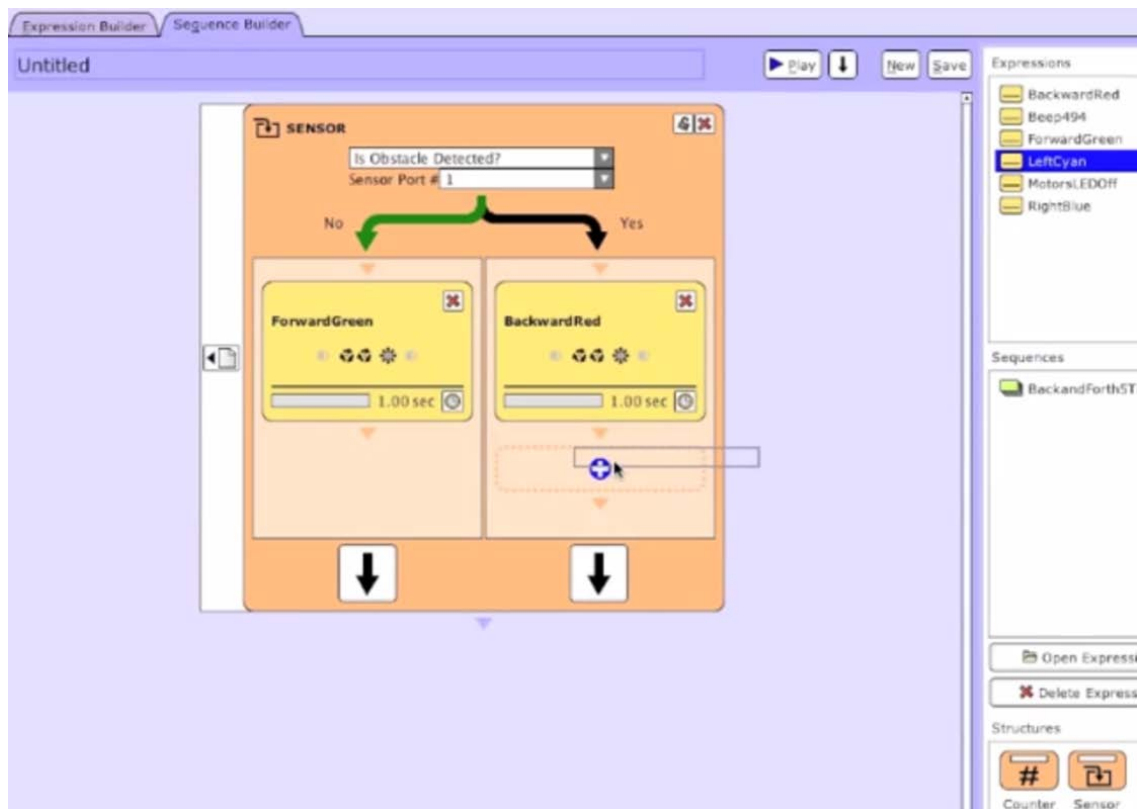
Nachdem die benötigten „Expressions“ erstellt wurden, können diese im „Sequence Builder“ in eine Programmsequenz integriert werden (siehe Abbildung 16). Dafür gibt es vorgefertigte Strukturen (Structures), die sowohl Schleifen als auch Sensoren umfassen und sich per Drag & Drop ineinander verschachteln lassen. Diese Programmsequenzen lassen sich, wie auch „Expressions“, speichern und sind dann im rechten Fensterbereich unter „Sequences“ auswählbar. Die Sequenzen können in andere Schleifen integriert werden. Das fertige Programm ist dann direkt am Finch ausführbar⁹³.

⁹²

<http://www.finchrobot.com/Creating%2C%20Compiling%2C%20and%20Running%20Programs%20for%20Finch/using-create-lab-visual-programmer#>, (03.02.2014)

⁹³ vgl. Finch,

<http://www.finchrobot.com/Creating%2C%20Compiling%2C%20and%20Running%20Programs%20for%20Finch/using-create-lab-visual-programmer>, (03.02.2014)

Abbildung 16 - CREATE Lab Visual Programmer Sequence Builder⁹⁴

CREATE Lab Visual Programmer bietet eine übersichtliche Entwicklungsumgebung, die durch Drag & Drop Programmierung einen schnellen Einstieg ermöglicht. Durch die spezielle Abstimmung des Programms auf den Finch können bei der Erstellung von Ausgaben die entsprechenden Komponenten an einer Finch-Darstellung gewählt werden (siehe Abbildung 15). Als nachteilig kann die sprachliche Beschränkung auf Englisch gesehen werden. Stellt diese kein Problem dar, so ist die Umgebung durchaus für den schulischen Einsatz geeignet.

8.1.2 Finch Dreams

Alter: 8+⁹⁵

Systemsprachen: Englisch

Betriebssystem(e): Windows, Mac OSX (keine Versionsangaben, erfolgreich getestet unter Windows 8)

Basiert auf: Alice 2.2⁹⁶

⁹⁴

<http://www.finchrobot.com/Creating%2C%20Compiling%2C%20and%20Running%20Programs%20for%20Finch/using-create-lab-visual-programmer#>, (03.02.2014)

⁹⁵ vgl. Finch, <http://www.finchrobot.com/software-overview>, (03.02.2014)

⁹⁶ vgl. Alice, <http://www.alice.org/>, (08.02.2014)

Entwickler/Lizenz: Carnegie Mellon University – entwickelt von Saad Farooq und Tom Lauwers⁹⁷

Kosten: kostenloser Download⁹⁸

Bei Finch Dreams handelt es sich um eine sehr spezielle Programmierumgebung, die auf der (ebenfalls an der Carnegie Mellon University entwickelten) Alice 2.2 Software basiert und für die Erstellung einer interaktiven, virtuellen, dreidimensionalen Welt entwickelt wurde. In dieser virtuellen Welt können sich Charaktere bewegen und miteinander sowie mit ihrer Umwelt interagieren. Im Fall von Alice 2.2 handelt es sich um eine Umgebung zur Schaffung einer rein virtuellen Welt, wohingegen Finch Dreams, durch die Erweiterung um den Roboter, auch eine reale Interaktion ermöglicht. Es ist möglich, den Finch in Folge eines Programmablaufs in der echten Welt fahren zu lassen oder ein Ereignis durch ein LED oder ein Geräusch darzustellen. Auch auf Objekte der virtuellen Welt kann auf Basis der Sensorwerte des Finch Einfluss genommen werden. Zum Beispiel ist es möglich, den Finch-Roboter als Controller zu verwenden, mit dem ein virtuelles Objekt auf Basis der Werte des Beschleunigungssensors gesteuert werden kann, indem der Roboter in der Hand bewegt wird.

Laut Hersteller ist die Umgebung für Kinder ab acht Jahren geeignet, durch die Komplexität und den großen Funktionsumfang sollte jedoch für den Einstieg eventuell ein höheres Alter gewählt werden (vor allem für Kinder ohne Englisch als Muttersprache).

Die Programmierung in der Finch-Dreams-Umgebung erfolgt per Drag & Drop, ist aber recht komplex, besonders wegen des großen Funktionsumfangs. Alle virtuell existenten Objekte (Welt, Finch, Stein, etc.) haben bestimmte Eigenschaften, die voreingestellt und verändert werden können. Diese sind (siehe Abbildung 17 links oben) in einer Baumstruktur zusammengefasst und können für ihre Programmierung ausgewählt werden. Darunter (siehe Abbildung 17 links unten) sind unter Details alle Möglichkeiten des ausgewählten Objekts zu sehen, zusammengefasst zu den entsprechenden Punkten Einstellungen, Methoden und Funktionen. Rechts davon (siehe Abbildung 17 rechts unten) befindet sich die Darstellung der aktuellen

⁹⁷ Finch Dreams - V1.01 vom 16.02.2012 (Help -> About Finch Dreams)

⁹⁸ vgl. Finch, <http://www.finchrobot.com/downloads#finchdreams>, (03.02.2014)

Hauptmethode, in die objektbezogene Methoden und Funktionen hineingezogen werden können. Eine Auswahl der benötigten/möglichen spezifischen Einstellungen erscheint automatisch. Im unteren Bereich des Hauptmethodenfensters können ebenfalls integrierbare Schleifen gewählt werden. Diese lassen sich mit den meisten Funktionen verknüpfen. Des Weiteren gibt es ein Fenster zur Voransicht der virtuellen Welt (siehe Abbildung 17 Mitte oben) und ein Eventfenster, in dem sich Ereignisse (Mausklick) erstellen lassen.

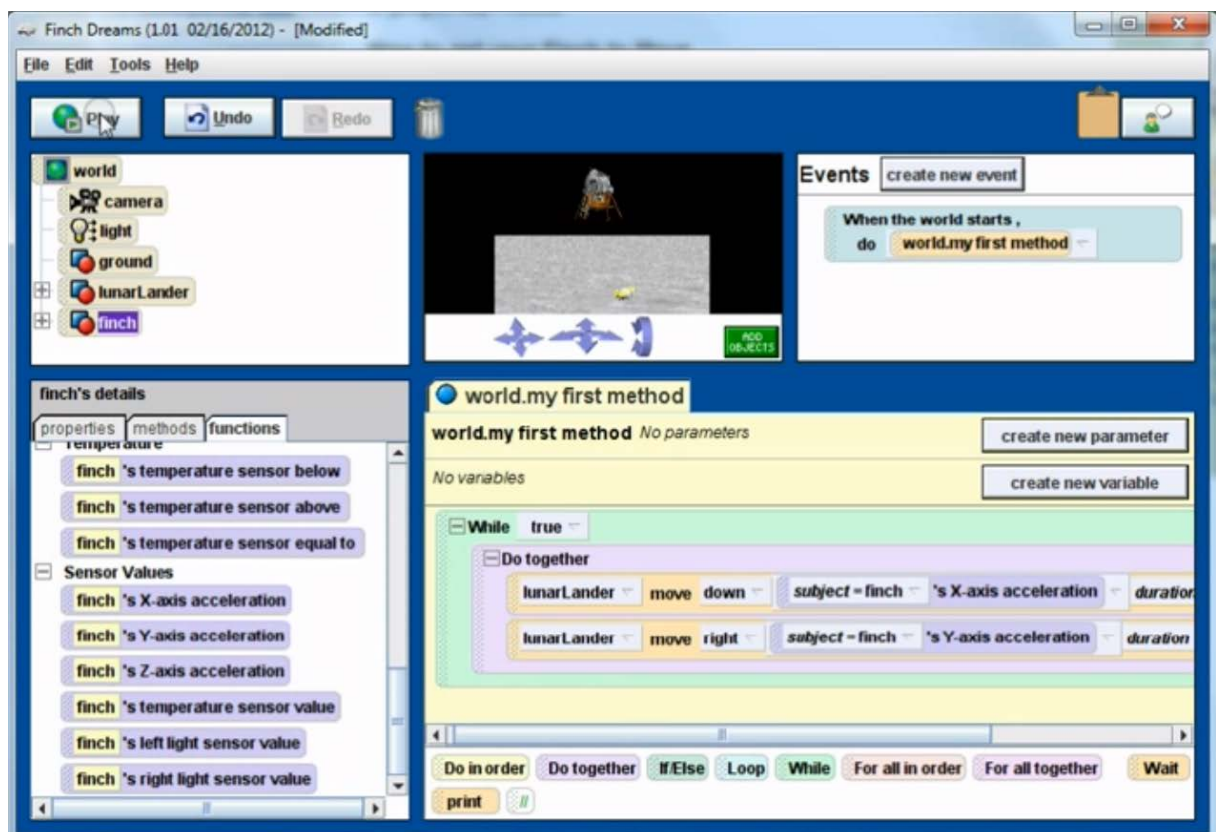


Abbildung 17 – Finch Dreams⁹⁹

Finch Dreams ist eine sehr umfangreiche Entwicklungsumgebung, die trotz der vielen Fenster und Möglichkeiten einen recht guten Überblick bietet. Trotz der grafischen Drag & Drop Oberfläche ist diese Software nicht so einfach, überschaubar und intuitiv wie alternative Umgebungen. Ist es nicht das Ziel, eine interaktive, virtuelle Welt zu programmieren, sondern soll vielmehr Wissen über oder durch einen Roboter vermittelt werden, so eignen sich alternative Umgebungen eventuell besser dafür.

⁹⁹

<http://www.finchrobot.com/Creating,%20Compiling,%20and%20Running%20Programs%20for%20Finch/finch-dreams-tutorial>, (03.02.2014)

8.1.3 Snap!

Alter: 8+¹⁰⁰

Systemsprachen: Deutsch, Englisch, Französisch, weitere¹⁰¹

Betriebssystem(e): unabhängig, browserbasiert; Scribbler 2 benötigt Python 2.6+¹⁰²;

Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁰³

Basiert auf: (hie früher BYOB), Scratch¹⁰⁴

Entwickler/Lizenz: Berkeley - University of California

Kosten: kostenlose online Umgebung (benötigt offline Schnittstelle – kostenlos)¹⁰⁵

Die grafische Sprache Snap! ist die weiterentwickelte Neuprogrammierung der Programmiersprache BYOB. Beide basieren ursprünglich auf Scratch (siehe Kapitel 8.1.4: „Scratch 2.0 (offline Editor)“). Snap! wurde an der Universität Berkeley in Kooperation mit Studenten anderer Hochschulen und unabhängigen Firmen entwickelt. Es handelt sich um eine browserbasierte Sprache, die für die Kreierung einer virtuellen Umgebung gedacht ist. Mittlerweile können damit aber auch verschiedene Roboter programmiert werden, unter anderem der Scribbler 2 und der Finch, die im obigen Vergleich behandelt wurden. Die Voraussetzung dafür ist eine zusätzliche Offline-Schnittstelle, die kostenlos über die Herstellerseite bezogen werden kann. Es ist auch möglich, den LEGO Mindstorms NXT mit Snap! zu programmieren. Ob sein Nachfolger, der LEGO Mindstorms EV3 auch unterstützt werden wird, ist noch unklar.

Die Programmerstellung erfolgt mittels Drag & Drop grafischer Bauelemente, die aneinandergesetzt werden, um so den Ablauf festzulegen. Dazu können aus acht Kategorien (Bewegung, Aussehen, Klang, Steuerung, Variablen, ...) jeweils passende Blöcke gewählt werden (siehe Abbildung 18 links), welche dann in den mittleren Bereich verschoben und aneinandergereiht werden. Zusätzlich gibt es ein Fenster, in dem die virtuelle Umsetzung erfolgt (siehe Abbildung 18 rechts oben) und

¹⁰⁰ vgl. Finch, <http://www.finchrobot.com/software-overview>, (03.02.2014)

¹⁰¹ vgl. Snap!, <http://snap.berkeley.edu/snapsource/snap.html>, (03.02.2014)

¹⁰² vgl. Technoboy10, <http://technoboy10.github.io/ssquared/>, (03.02.2014)

¹⁰³ vgl. Finch, <http://www.finchrobot.com/software/birdbrain-snap-v02-alpha>, (03.02.2014)

¹⁰⁴ vgl. Scratch, <http://scratch.mit.edu/>, (08.02.2014)

¹⁰⁵ vgl. Snap!, <http://www.finchrobot.com/software/snap>, (03.02.2014)

eine Übersicht über die erstellten Umgebungen und Objekte (siehe Abbildung 18 rechts unten).

Eine beliebige Anzahl an Schleifen und Funktionen kann ineinander geschachtelt und bei Bedarf auch als übergeordneter Block abgespeichert werden. Da nur Blöcke kombinierbar sind, die auch in Kombination Sinn ergeben, ist es kaum möglich, Programmfehler zu integrieren [vgl. 31].

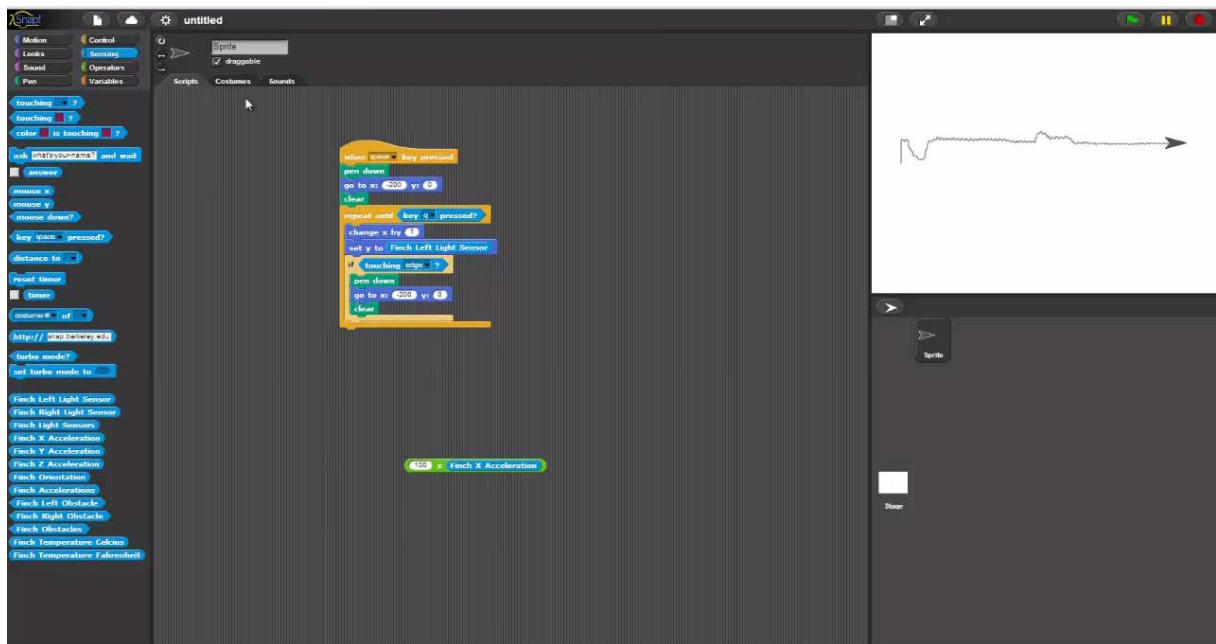


Abbildung 18 – Snap! Oberfläche mit Beispielprogramm¹⁰⁶

Snap! ist unter den grafischen Sprachen eine sehr übersichtliche Entwicklungsumgebung. Durch die weite Verbreitung ihrer Vorgängersprachen hat sich auch schnell eine große Community um Snap! aufgebaut, weshalb es ausreichend Support bei Problemen gibt. Eine große Anzahl an Video-Tutorials sowie Lehr- und Lernunterlagen werden im Internet kostenlos bereitgestellt, auch eine Vielzahl an Beispielprogrammen ist kostenlos herunterladbar.

8.1.4 Scratch 2.0 (offline Editor)

Alter: 8+¹⁰⁷

Systemsprachen: Deutsch, Englisch, Französisch, weitere¹⁰⁸

¹⁰⁶ <https://dl.dropboxusercontent.com/u/9303915/Videos/FinchAndSnapTutorial.mp4>, (03.02.2014)

¹⁰⁷ vgl. Scratch, <http://scratch.mit.edu/about/>, (04.02.2014)

¹⁰⁸ vgl. Snap!, <http://snap.berkeley.edu/snapsource/snap.html>, (03.02.2014)

Betriebssystem(e): Windows, Mac, Linux (keine Versionsinformation); Adobe Air V2.6+¹⁰⁹; Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹¹⁰

Basiert auf: nicht bekannt/unabhängig entwickelt

Entwickler/Lizenz: Lifelong-Kindergarden-Group am Media-Lab des Massachusetts Institute of Technology (MIT)¹¹¹

Kosten: kostenlose Downloads

Die Programmierumgebung Scratch, die von der Lifelong-Kindergarden-Group am Media-Lab des MIT entwickelt wurde, stellt die Grundlage für viele grafische Programmiersprachen, wie zum Beispiel Snap! (siehe Kapitel 8.1.3: „Snap!“), dar. Mit fast fünf Millionen veröffentlichter Projekte ist Scratch eine der am weitest verbreiteten grafischen Programmiersprachen geworden, entsprechend groß ist die Community. Durch die Vielzahl an zugänglichen Projekten ist es möglich, für viele Anwendungsbeispiele exemplarische Programme sowie Problemlösungen bei komplexen Problemen zu finden. Entwickelt wurde die Umgebung, um Kindern und Jugendlichen Grundkonzepte der Programmierung zu vermitteln. Als Zielgruppe werden auf der Herstellerseite Acht- bis Sechzehnjährige genannt. Durch die gegebene Einfachheit der Umgebung einerseits und den großen, weiter wachsenden Funktionsumfang andererseits wird Scratch aber von Jüngeren wie Älteren benutzt.

Scratch wurde ursprünglich als reine Browserumgebung entwickelt. In der Version 2.0 wurde ebenfalls ein Offline-Editor (siehe Abbildung 19) veröffentlicht, welcher für die Erstellung von Programmen für den Finch benötigt wird. Abgesehen davon, dass die Fenster, die im Snap!-Editor rechts außen und bei Scratch 2.0 links außen sind (virtuelle Umsetzung, Objektübersicht), ist die Umgebung beinahe ident, weshalb die Vorgehensweise bei der Programmerstellung hier nicht erneut geschildert wird (siehe Kapitel 8.1.3: „Snap!“)¹¹².

¹⁰⁹ vgl. Scratch, http://wiki.scratch.mit.edu/wiki/Offline_Editor, (04.02.2014)

¹¹⁰ vgl. Finch, <http://www.finchrobot.com/software/birdbrain-snap-v02-alpha>, (03.02.2014)

¹¹¹ vgl. Scratch, <http://scratch.mit.edu/about/>, (04.02.2014)

¹¹² vgl. Scratch, <http://scratch.mit.edu/about/>, (04.02.2014)

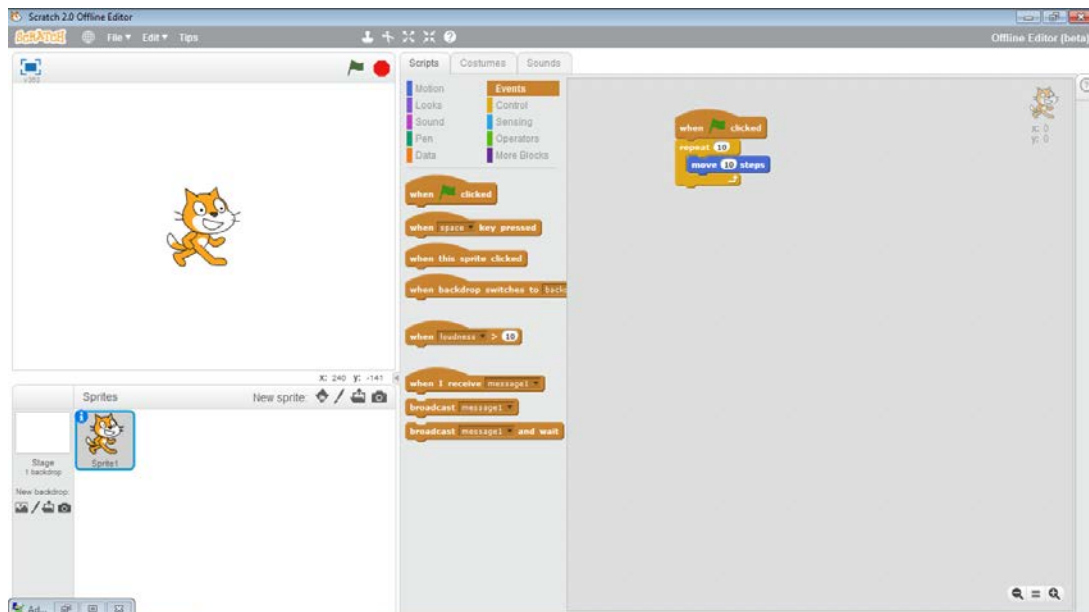


Abbildung 19 – Scratch Beispielprogramm im offline Editor¹¹³

Der Funktionsumfang von Scratch 2.0 und Snap! ist nicht völlig ident, da es aber nur geringe Unterschiede gibt, sind diese im Rahmen dieser Arbeit nicht relevant. Der Hauptunterschied für die Verwendung in einer Schule ist, dass Scratch 2.0 als Offline-Version verfügbar ist und auch nur mit dieser Version in Kombination mit dem Finch verwendet werden kann. Wird der Finch als Robotermodell mit Scratch 2.0 programmiert, so kann jederzeit auch zum Snap!-System gewechselt werden, da der Roboter beide Systeme unterstützt und diese durch die Vielzahl an Gemeinsamkeiten (meist) ohne Umschulung verwendet werden können.

8.1.5 LabVIEW

Alter: keine Angaben

Systemsprachen: Deutsch, Englisch, Französisch, weitere

Betriebssystem(e): Windows Vista / 7 / XP; Finch benötigt Schnittstellensoftware: Windows (keine Versionsinformation)¹¹⁴

Basiert auf: nicht bekannt/unabhängig entwickelt

Entwickler/Lizenz: National Instruments

Kosten: 995 Euro bis 5160 Euro¹¹⁵

Die grafische Programmiersprache LabVIEW von National Instruments wird durch das Verschieben von Blöcken mittels Drag & Drop verwendet. Entgegen den meisten

¹¹³ <http://robotikas.blogspot.co.at/2013/08/disponible-la-version-beta-de-scratch.html>, (04.02.2014)

¹¹⁴ vgl. Finch, <http://www.finchrobot.com/software/labview>, (05.02.2014)

¹¹⁵ vgl. National Instruments, <http://www.ni.com/labview/buy/d/>, (04.02.2014)

anderen grafischen Sprachen ist LabVIEW für den Einsatz durch Ingenieure und Wissenschaftler gedacht und richtet sich nicht (direkt) an die Bezugsgruppe dieser Arbeit.

Obwohl es beim Lizenzerwerb für Hochschulen oder im Rahmen einer Bestellung höherer Stückzahlen Preisnachlässe gibt, so richtet sich auch der Preis einer Einzellizenz mit etwa 1000 Euro (ein Jahr Support inklusive) nicht direkt an Schulen, deren Budget dadurch vermutlich überstiegen werden würde, zumindest bei der Ausstattung einer ganzen Klasse. Aus diesen Gründen wird LabVIEW im Rahmen dieser Arbeit auch nicht weiter behandelt.

8.1.6 Greenfoot

Alter: 15+¹¹⁶

Systemsprachen: Deutsch, Englisch, Französisch, weitere¹¹⁷

Betriebssystem(e): Windows Vista / 7 / 8 / XP, Mac OSX 10.5+, Linux; Finch benötigt

Schnittstellensoftware: Windows (keine Versionsinformation)¹¹⁸

Basiert auf: Java

Entwickler/Lizenz: Programming Education Tools Group der Computing Education Research Group an der University of Kent in Canterbury

Kosten: kostenlose Downloads

Greenfoot ist eine Java-basierte Programmierumgebung, die entwickelt wurde, um Programmierungen zu lehren und zu lernen und um mithilfe der Sprache Java die Konzepte der objektorientierten Programmierung auf einfache und verständliche Weise zu vermitteln. Die Entwicklungsumgebung ermöglicht die Erstellung einer grafischen, interaktiven Welt, in welche durch die entsprechende Erweiterung der Roboter Finch integrierbar ist. Dem Anwender, der Anwenderin werden dafür drei Fensterbereiche zur Verfügung gestellt, die Welt (siehe Abbildung 20 vordergründiges Fenster links), in der die Programmausführung virtuell dargestellt wird, ein Klassendiagramm (siehe Abbildung 20 vordergründiges Fenster rechts), in dem die Welt mit ihren Objekten und darauf befindlichen Elementen dargestellt wird, und einen Steuerungsbereich (siehe Abbildung 20 vordergründiges Fenster unten),

¹¹⁶ vgl. PC Magazin, <http://www.pc-magazin.de/ratgeber/greenfoot-nicht-nur-fuer-den-informatik-unterricht-1006644.html>, (05.02.2014)

¹¹⁷ vgl. Greenfoot, <http://www.greenfoot.org/translations>, (05.02.2014)

¹¹⁸ vgl. Finch, <http://www.finchrobot.com/software/greenfoot>, (05.02.2014)

der den Programmablauf bestimmt. Zur Programmierung wird von einer vorher definierten Klasse ein neues Objekt erstellt, welches dann auf der Welt platziert werden kann. Wird dieses anschließend mit der rechten Maustaste angeklickt, so wird in einem Menü jede objektspezifische Operationsoption angezeigt. Werden Parameter benötigt, so erscheint ein entsprechendes Dialogfeld, das darauf hinweist. Durch Doppelklick auf ein Objekt kann der spezifische Java-Code in einem Editor geöffnet und bearbeitet werden (siehe Abbildung 20 hintergründiges Fenster – Eigenschaften des Objekts Crab) [vgl. 32 S 9 ff.].

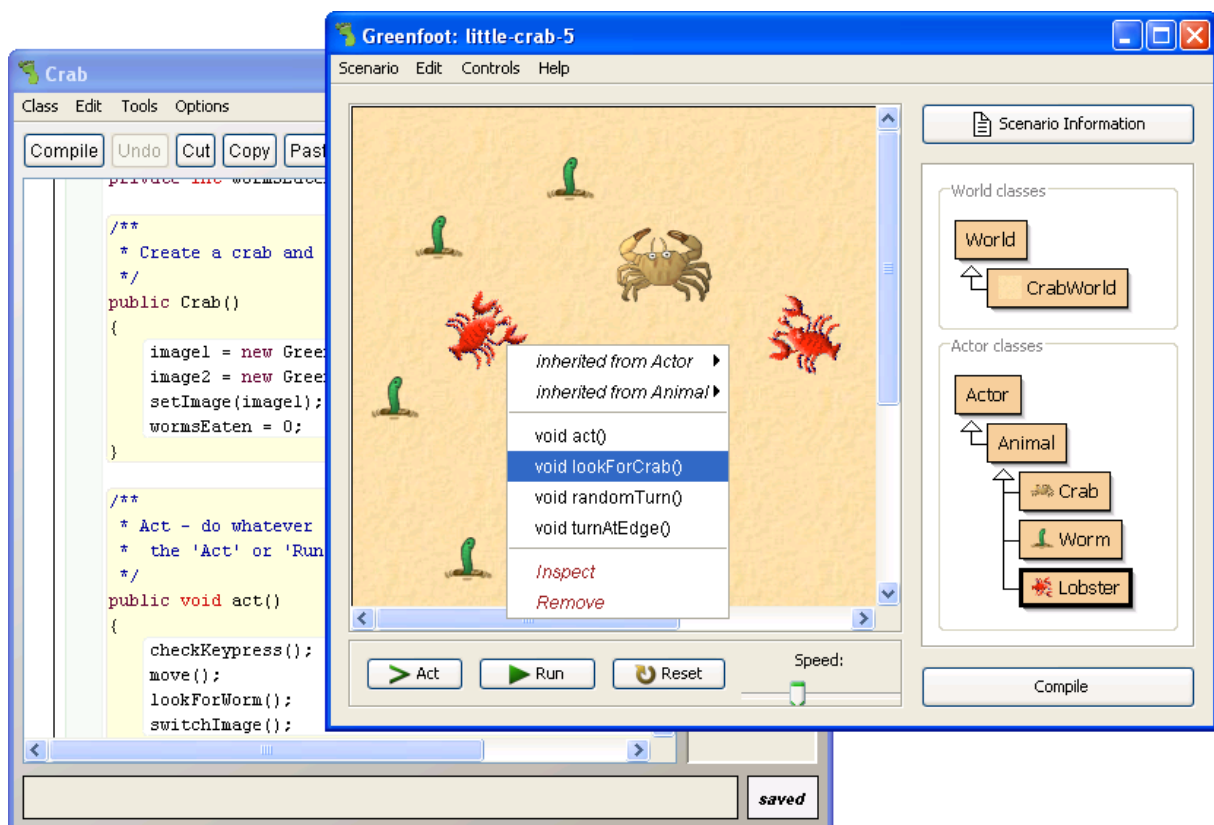


Abbildung 20 – Greenfoot Programmbeispiel / Java Editor¹¹⁹

Zusammenfassend kann festgehalten werden, dass es sich bei Greenfoot um eine Kombination aus einer grafischen und textuellen Programmiersprache handelt, die sowohl rein virtuelles Ausführen als auch die Interaktion mit dem Roboter Finch ermöglicht. Die Sprache ist speziell für den schulischen Einsatz entwickelt worden, eignet sich laut Hersteller jedoch erst ab einem Alter von 15 Jahren aufwärts, weshalb der Einsatz in der Unterstufe österreichischer Schulen nur erschwert möglich sein dürfte.

¹¹⁹ <http://www.roboticslearning.com/java/>, (05.02.2014)

8.1.7 RoboRealm

Alter: nicht bekannt

Systemsprachen: Englisch

Betriebssystem(e): Windows (keine Versionsinformation)¹²⁰

Basiert auf: nicht bekannt/unabhängig entwickelt

Entwickler/Lizenz: RoboRealm

Kosten: etwa 37 Euro, Bildungslizenz etwa 30 Euro¹²¹

RoboRealm ist eine spezielle Software, die eine Bild- und/oder Video-Auswertung ermöglicht und anhand der Daten zum Beispiel einen Roboter manövrieren kann. Die Grundvoraussetzung dafür ist vorhandenes Bildmaterial, das entweder durch eine Kamera im/am Roboter oder durch eine externe Quelle gewonnen wird. Da der Finch der einzige Roboter ist, der diese Software unterstützt und Relevanz für diese Arbeit hat, dieser jedoch keine Kamera integriert oder als Erweiterung integrierbar hat, kann dieser in Kombination mit RoboRealm (ohne größeren Aufwand) nur äußerst eingeschränkt verwendet werden. Die Umgebung bietet ein gutes Kontroll-Interface für den Finch an (siehe Abbildung 21), dieses bietet für den schulischen Einsatz jedoch kaum sinnvolle Anwendungsmöglichkeiten, weshalb diese Sprache in der vorliegenden Arbeit nicht näher behandelt wird.

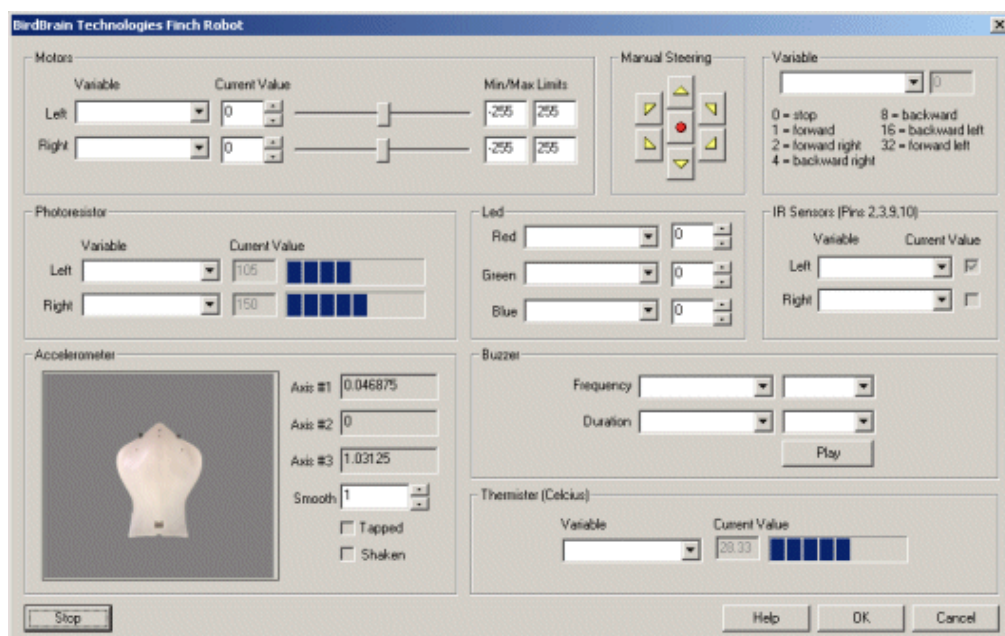


Abbildung 21 – RoboRealm Finch Interface¹²²

¹²⁰ vgl. RoboRealm, http://www.roborealm.com/forum/index.php?thread_id=171, (06.02.2014)

¹²¹ vgl. RoboRealm, <http://www.roborealm.com/registration/purchase.php>, (06.02.2014)

8.1.8 S2 GUI

Alter: 14+¹²³

Systemsprachen: Deutsch, Englisch, Französisch, weitere¹²⁴

Betriebssystem(e): Windows XP / Vista / 7¹²⁵

Basiert auf: nicht bekannt/unabhängig entwickelt

Entwickler/Lizenz: Parallax

Kosten: kostenloser Download

Das S2 GUI (Scribbler 2 Graphical User Interface) ist eine grafische Entwicklungsumgebung (siehe Abbildung 22) zur Programmierung der Scribbler 2 Roboter, die von der Herstellerfirma kostenlos zum Download angeboten wird. Um ein Programm zu erstellen, werden mittels Drag & Drop Aktionsblöcke (siehe Abbildung 22 links) zwischen einen Start- und End-Block gezogen. Diese Blöcke finden sich bereits bei Programmstart in dem Bereich zur Programmerstellung (siehe Abbildung 22 weiß kariert). Zusätzlich bietet die überschaubare Oberfläche ein Icon-basiertes Verwaltungsmenü (siehe Abbildung 22 oben).

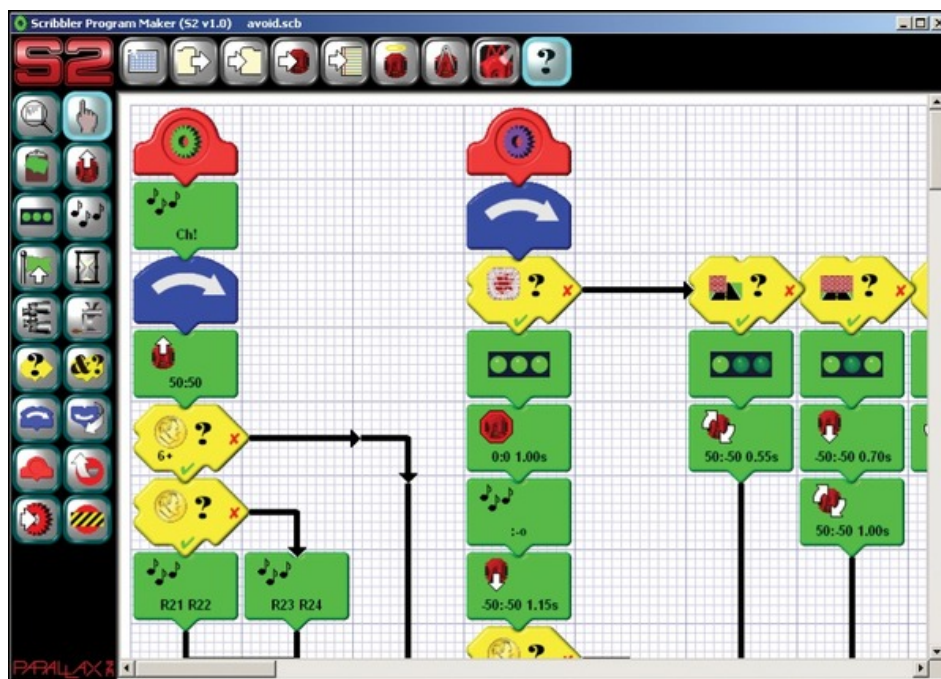


Abbildung 22 – S2 GUI Beispielprogramm¹²⁶

¹²² [http://www.roborealm.com/help/BirdBrain Technologies Finch Robot.php](http://www.roborealm.com/help/BirdBrain_Technologies_Finch_Robot.php), (06.02.2014)

¹²³ vgl. Parallax, <http://www.parallax.com/product/28136>, (06.02.2014)

¹²⁴ siehe Installationsoptionen des S2 GUI

¹²⁵ vgl. Pololu, <http://www.pololu.com/product/1610>, (06.02.2014)

Das S2 GUI stellt eine einfache und überschaubare Oberfläche bereit, die Kindern und Jugendlichen einen schnellen Einstieg in die Welt der ikonischen Programmierung bieten soll. Da die Oberfläche speziell auf den Scribbler 2 zugeschnitten, ist gibt es keine überflüssigen Optionen, die den User verwirren könnten. Trotz der entgegenkommenden Usability empfiehlt der Hersteller die Verwendung des S2 GUI erst ab einem Alter von 14 Jahren. Aufgrund der einfachen Handhabung ist es aber denkbar, schon deutlich früher in die Entwicklung mit dieser Umgebung einzusteigen.

8.1.9 ROBO PRO

Alter: 10+¹²⁷

Systemsprachen: Deutsch, Englisch, Französisch, weitere¹²⁸

Betriebssystem(e): Windows XP / Vista / 7 / 8¹²⁹

Basiert auf: nicht bekannt/unabhängig entwickelt

Entwickler/Lizenz: Fischertechnik

Kosten: etwa 30 Euro¹³⁰

ROBO PRO ist die grafische Variante zur Programmierung der Fischertechnik Computing-Produkte, die vom Hersteller selbst entwickelt wurde. Die Entwicklungsoberfläche (siehe Abbildung 23), die von Kindern ab zehn Jahren verwendet werden kann (so der Hersteller), unterstützt Drag & Drop Programmierung. Allerdings ist es notwendig, weitere Zuweisungen wie entsprechende Hardwareeingänge und Ausgänge in einem separaten Fenster einzustellen. Soll ein Programm erstellt werden, wird zuerst in der Elementgruppenauswahl (siehe Abbildung 23 links oben) die Gruppe der Programmelemente gewählt, die dann im gleichnamigen Bereich (siehe Abbildung 23 links unten) auswählbar sind. Diese werden in den Bereich zur Programmerstellung gezogen (siehe Abbildung 23 Hauptbereich). Durch einen Doppelklick darauf öffnet sich das entsprechende Fenster um benötigte/mögliche Parameter festzulegen. Zur

¹²⁶ <http://www.pololu.com/picture/view/0J3132>, (06.02.2014)

¹²⁷ vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-35/usetemplate-2_column_pano/, (07.02.2014)

¹²⁸ Sprachauswahl in der ROBO PRO Oberfläche

¹²⁹ vgl. Fischertechnik, http://www.fischertechnik.de/desktopdefault.aspx/tabid-21/39_read-35/usetemplate-2_column_pano/, (07.02.2014)

¹³⁰ siehe ebenda

Steuerung des Programmablaufs wird dem Nutzer sowie der Nutzerin ein ikonisches Menü (siehe Abbildung 23 Leiste oben) angeboten.

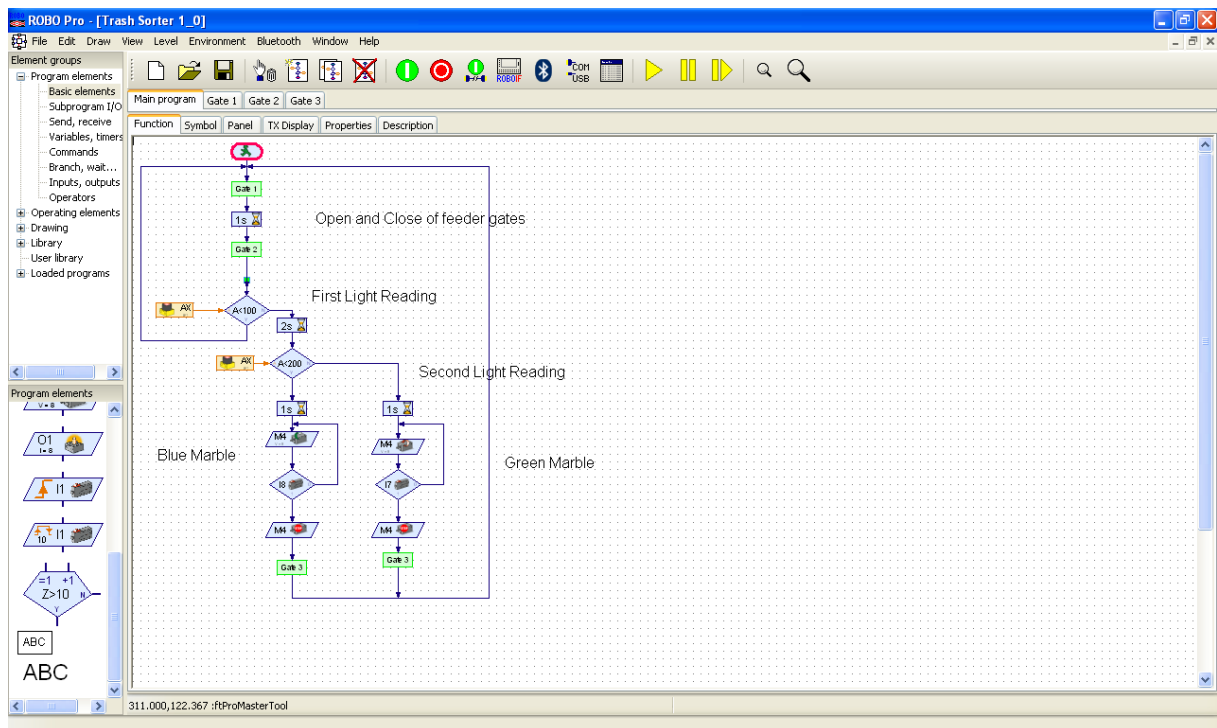


Abbildung 23 – ROBO PRO Programmbeispiel¹³¹

ROBO PRO bietet ein überschaubares Fenster zur Roboterprogrammierung an, das auch die Darstellung größerer, komplexerer Programme erlaubt. Durch die Notwendigkeit, bestimmte Parameter der Komponenten nachzutragen, kann die Software in der Anwendung umständlicher wirken als Produkte anderer Hersteller. Für junge Schüler und Schülerinnen eignen sich Konkurrenzprodukte eventuell besser.

8.1.10 LEGO Mindstorms Education EV3 Software

Alter: 10+¹³²

Systemsprachen: Deutsch, Englisch [vgl. 33]

Betriebssystem(e): Windows XP / 8; Mac (keine Versionsinformation) [vgl. 33]

Basiert auf: LabVIEW

Entwickler/Lizenz: LEGO

Kosten: etwa 120 Euro Einzelplatz, etwa 420 Euro Schullizenz¹³³

¹³¹ <http://www.backintheclassroom.com/tag/robo-pro/>, (07.02.2014)

¹³² vgl. Lego, <http://education.lego.com/de-de/lego-education-product-database/mindstorms-ev3/2000045-lego-mindstorms-education-ev3-software-single-user/>, (07.02.2014)

¹³³ vgl. Lego, <http://shop.legoeducation.com/de/category/lego-mindstorms-education-ev3-22/>, (07.02.2014)

Die LEGO Mindstorms Education EV3 Software (siehe Abbildung 24), die auf LabVIEW von National Instruments basiert, wurde speziell für den Schulunterricht entwickelt und soll auch ohne Programmierkenntnisse von Kindern und Jugendlichen ab einem Alter von zehn Jahren verwendet werden können. Neben der Möglichkeit zur Roboterprogrammierung bietet die Software Zusatzfeatures wie zum Beispiel eine Messerwerterfassung einzelner Sensorwerte, die den Rahmen durchführbarer Experimente wesentlich erweitert. Es wird auch ein Inhaltseditor angeboten, der dem Lehrpersonal ermöglicht, Unterrichtsinhalte anzupassen und Unterrichtseinheiten zu erstellen. Schülern und Schülerinnen bietet er die Möglichkeit, eigene Programme zu dokumentieren, um so die Unterrichtsgestaltung und Bewertung zu erleichtern.

Um ein Programm zu schreiben, werden die gewünschten Programmblöcke, die in verschiedenfarbigen Kategorien zu Typen zusammengefasst sind (siehe Abbildung 24 unten links), mittels Drag & Drop in den Bereich zur Programmerstellung (siehe Abbildung 24 Hauptbereich) gezogen. Um dieses nach Fertigstellung auszuführen, wird den Usern eine Verbindungsübersicht neben den Steuerungsschaltflächen (siehe Abbildung 24 rechts unten) geboten.¹³⁴

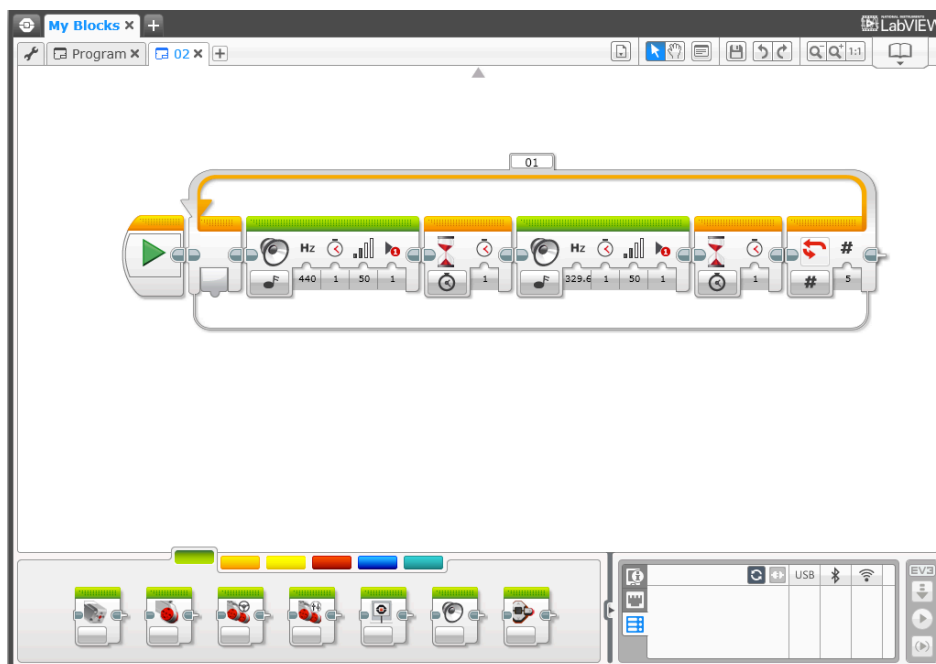


Abbildung 24 – Programmoberfläche – LEGO Mindstorms Education EV3 Software¹³⁵

¹³⁴ vgl. Lego, <http://education.lego.com/de-de/lego-education-product-database/mindstorms-ev3/2000045-lego-mindstorms-education-ev3-software-single-user>, (07.02.2014)

¹³⁵ <http://geekdad.com/2013/08/hands-on-ev3-mindstorms/>, (07.02.2014)

Die LEGO Mindstorms Education EV3 Software bietet dem Lehrpersonal sowie Schülern und Schülerinnen ein umfangreiches und gut durchdachtes Entwicklungssystem, das auf den Schuleinsatz zugeschnitten ist, und zusätzliche, nützliche Tools bereitstellt. Die Umgebung ist sehr intuitiv bedienbar und stellt eine übersichtliche Oberfläche mit hoher Usability zur Verfügung. Nachteilig ist lediglich der Kostenfaktor der Software. Allerdings ist diese in umfangreicheren Schulpaketen als Schullizenz enthalten.

8.1.11 Modkit

Alter: nicht bekannt

Systemsprachen: Englisch¹³⁶

Betriebssystem(e): unabhängig, browserbasiert; VEX IQ benötigt

Schnittstellensoftware: Windows; Mac (keine Versionsinformation)¹³⁷

Basiert auf: beeinflusst von Scratch

Entwickler/Lizenz: Modkit

Kosten: kostenlose Browseredition, kostenlose Schnittstellensoftware

Modkit ist eine Software zur grafischen Roboterprogrammierung, die von den verglichenen Robotermodellen derzeit nur für den VEX IQ verfügbar ist. Die Entwicklungsumgebung (siehe Abbildung 25) ist eine Browserapplikation, die zusätzlich für die Kommunikation mit dem Roboter eine Schnittstellensoftware benötigt.

Um ein Programm mit Modkit zu erstellen, müssen zuerst die verwendeten Hardwarekomponenten des Roboters unter der Registerkarte „Hardware“ (siehe Abbildung 25 oben Mitte) per Drag & Drop in den Entwicklungsbereich (siehe Abbildung 25 Hauptbereich) gezogen werden. Der Anschluss der Komponente ist jeweils auszuwählen. Für die Programmerstellung werden die gewünschten Funktionsblöcke (siehe Abbildung 25 links außen) in den Entwicklungsbereich gezogen. Um den fertigen Code auszuführen, wird dem Anwender und der

¹³⁶ vgl. Modkit, <http://www.modkit.com/vex>, (07.02.2014)

¹³⁷ vgl. Modkit, <http://www.modkit.com/vex>, (07.02.2014)

Anwenderin ein sehr überschaubares Verwaltungsmenü (siehe Abbildung 25 rechts oben) zur Verfügung gestellt.¹³⁸

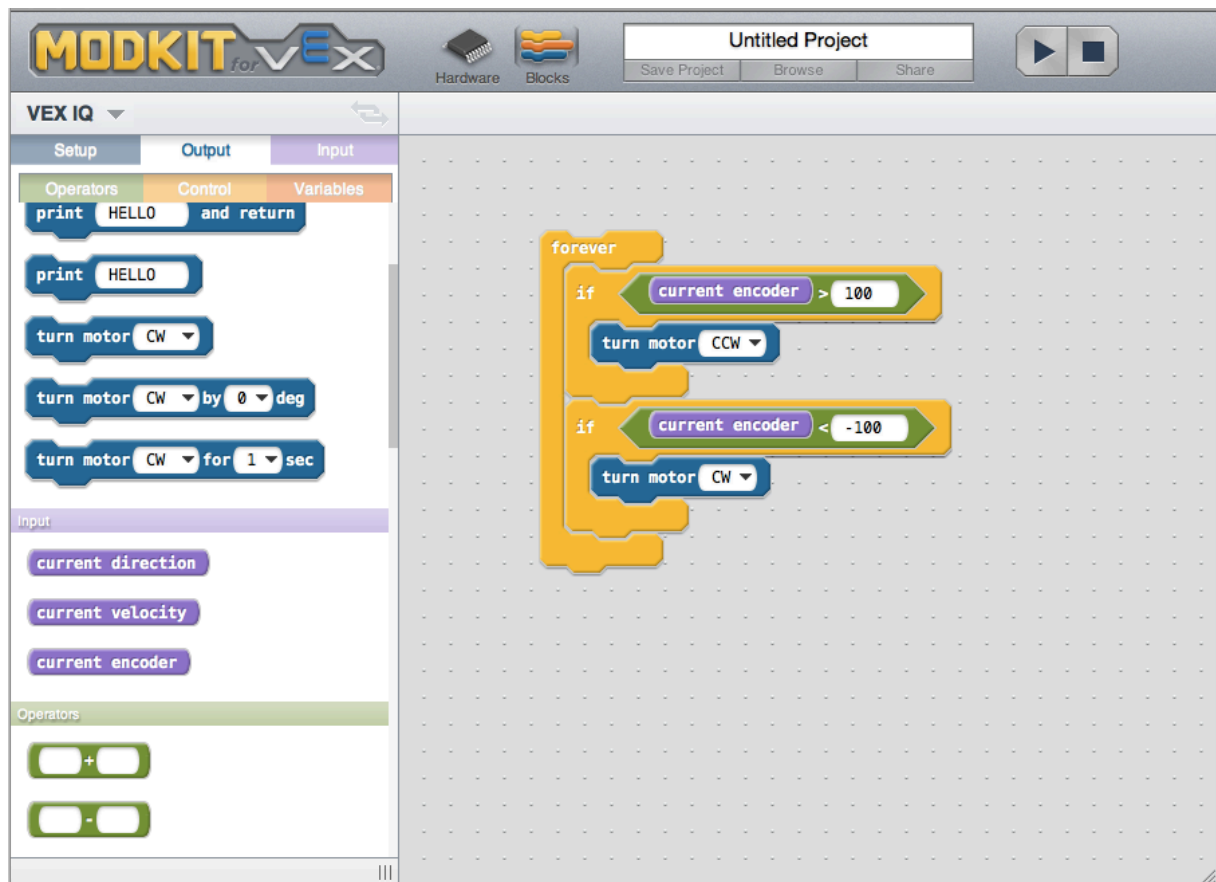


Abbildung 25 – MODKIT for VEX, Entwicklungsumgebung mit Beispielprogramm¹³⁹

Bei Modkit handelt es sich um ein übersichtliches, kostenloses Programm zur Roboterprogrammierung. Derzeit werden jedoch nur eingeschränkt Robotermodelle unterstützt. Die Entwicklungsumgebung ist nur in Englisch verfügbar, was entsprechende Grundvoraussetzungen an Anwender wie Anwenderinnen stellt. Als nachteilig kann angesehen werden, dass die Software browserbasiert ist und dadurch eine permanente Internetverbindung für die Programmerstellung notwendig ist.

8.2 Textuelle Programmiersprachen

Nachdem oben ein Überblick über die wichtigsten grafischen Sprachen gegeben wurde, die kompatibel zu einem oder mehreren der behandelten Robotersystemen sind, werden die verwendbaren textuellen Sprachen vorgestellt. Da es für textuelle Programmiersprachen im Allgemeinen keine Altersempfehlungen gibt und textuelle

¹³⁸ vgl. Modkit Blog, <http://blog.modk.it/>, (07.02.2014)

¹³⁹ <http://blog.modk.it/2013/05/modkit-maker-faire-featuring-modkit-for.html> , (07.02.2014)

Sprachen keine bestimmte Entwicklungsumgebung und somit keine Systemsprache voraussetzen, wurde auf diese Punkte hier nicht eingegangen. Neben allgemeinen Eigenschaften wird jeweils ein einfaches kurzes Beispielprogramm (dieses soll jeweils „Hello, world!“ am Bildschirm anzeigen) angegeben, um einen kurzen Einblick in die Syntax zu ermöglichen.

8.2.1 Python

Betriebssystem(e): Windows, Mac, Linux, weitere¹⁴⁰; Finch benötigt

Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁴¹; LEGO

Mindstorms EV3 benötigt Schnittstellensoftware: Linux (funktioniert auch mit virtuellem Computer)¹⁴²; Fischertechnik ROBO TX Training Lab benötigt

Schnittstellensoftware: Windows (keine Versionsinformation) [vgl. 34]

Basiert auf: wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

Entwickler/Lizenz: Python Software Foundation (PSF)

Kosten: kostenlose Downloads

Python ist eine leistungsfähige, dynamische, objektorientierte Programmiersprache, die kostenlos verfügbar und auf beinahe jedem System einsetzbar ist. Besondere Vorzüge dieser Sprache sind, dass sie eine umfangreiche Standardbibliothek beinhaltet und vor allem, dass Python leicht zu erlernen ist und eine gute Programmlesbarkeit unterstützt. Dies liegt zum Teil an der kurz gehaltenen Syntax und an der guten Lesbarkeit des übersichtlichen Codes. Wird Python für umfangreiche Projekte eingesetzt, wird dadurch meist der Wartungsaufwand der Software erheblich reduziert. Diese Programmiersprache wird auch oft für die Erstellung von Programmschnittstellen verwendet, weil sie sich sehr gut eignet, um Verknüpfungen zu anderen Programmiersprachen und Programmen zu erstellen.¹⁴³

Python Programmbeispiel – hello world:

Python 2: `print "Hello, world!"`

Python 3: `print("Hello, world!")`

Python dürfte sich gut für die Verwendung an Schulen eignen, weil es einerseits eine dynamische, objektorientierte Sprache ist, mit der sich viele beispielhafte Probleme

¹⁴⁰ vgl. Python, <http://www.python.org/about/>, (04.02.2014)

¹⁴¹ vgl. Finch, <http://www.finchrobot.com/software/python>, (04.02.2014)

¹⁴² vgl. Topikachu, <https://github.com/topikachu/python-ev3>, (04.02.2014)

¹⁴³ vgl. Python, <http://www.python.org/about/>, (04.02.2014)

schnell lösen lassen, andererseits weil es sich um eine (verhältnismäßig) leicht zu erlernende Sprache handelt, die sich an natürlicher Sprache orientiert. Der europäische Python-Dachverband dazu auf seiner Homepage: „*Ferner ist der Verband der Ansicht, dass Python als leicht zu lernende Programmiersprache im Bildungsbereich eine sehr viel größere Rolle als bisher spielen sollte. Python sollte daher in die Lehrpläne von Schulen und Hochschulen fest integriert werden.*“¹⁴⁴

8.2.2 Jython

Betriebssystem(e): läuft auf jeder Java Virtual Machine; Finch benötigt

Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁴⁵;

Basiert auf: Python

Entwickler/Lizenz: Python Software Foundation license version 2

Kosten: kostenlose Downloads

Bei Jython handelt es sich um eine Möglichkeit, ein Programm für Java Virtual Mashines in Python zu schreiben. Da sich nur die Schnittstelle zum verwendeten Roboter unterscheidet, die Programmiersprache aber (beinahe) ident ist, wird die Jython-Umgebung im Rahmen dieser Arbeit nicht weiter behandelt (siehe Kapitel 8.2.1: „Python“).

8.2.3 Java

Betriebssystem(e): Windows, Mac, Linux, Solaris¹⁴⁶; Java Development Kit; Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁴⁷;

LEGO Mindstorms EV3 benötigt leJOS Firmware: Windows (keine

Versionsinformation)¹⁴⁸; Fischertechnik ROBO TX Training Lab benötigt

Schnittstellensoftware: Windows (keine Versionsinformation) [vgl. 35]

Basiert auf: wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

Entwickler/Lizenz: Oracle

Kosten: kostenlose Downloads

Java ist die Grundlage für die meisten Netzwerkanwendungen und dient in bestimmten Bereichen, wie der Entwicklung von mobilen Anwendungen, als

¹⁴⁴ vgl. Python Verband, <http://python-verband.org/>, (04.02.2014)

¹⁴⁵ vgl. Finch, <http://www.finchrobot.com/software/python>, (04.02.2014)

¹⁴⁶ vgl. Java, <http://www.java.com/de/download/manual.jsp>, (04.02.2014)

¹⁴⁷ vgl. Finch, <http://www.finchrobot.com/software/python>, (04.02.2014)

¹⁴⁸ vgl. Topikachu, <https://github.com/topikachu/python-ev3>, (04.02.2014)

Standard. Durch mehr als neun Millionen Entwickler weltweit gehört Java nicht nur zu den verbreitetsten Programmiersprachen, es gibt auch eine entsprechend große Community. Es handelt sich um eine objektorientierte, textuelle Sprache, die sich durch ihre Leistungsstärke und vor allem ihr Kompatibilität auszeichnet.¹⁴⁹

Java Programmbeispiel – hello world:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

8.2.4 JavaScript

Betriebssystem(e): unabhängig; Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁵⁰

Basiert auf: wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

Entwickler/Lizenz: Oracle

Kosten: kostenlose Downloads

JavaScript ist eine objektorientierte Skriptsprache, die ursprünglich entwickelt wurde, um den Interaktivitätsgrad von Internetseiten zu erhöhen, und ist kein Bestandteil der Java-Plattform. Die Syntax der Sprache wurde allerdings an Java und C++ angelehnt, um die Anzahl neuer Konzepte klein zu halten. Neben der Hauptfunktion von JavaScript - den Funktionsumfang von Webseiten zu erweitern - wird die Sprache inzwischen auch verwendet, um zum Beispiel Micro Controller zu programmieren. Das erweiterte Anwendungsgebiet resultiert daraus, dass es sich bei JavaScript um eine moderne und vergleichsweise leicht erlernbare Sprache handelt.^{151 152}

JavaScript Programmbeispiel – hello world (für Mozilla's Rhino¹⁵³):
print('Hello, world!');

¹⁴⁹ vgl. Java, <http://www.java.com/de/about/>, (04.02.2014)

¹⁵⁰ vgl. Finch, <http://www.finchrobot.com/software/javascript>, (05.02.2014)

¹⁵¹ vgl. Java, http://www.java.com/de/download/faq/java_javascript.xml, (05.02.2014)

¹⁵² vgl. Heise, <http://www.heise.de/newsticker/meldung/JavaScript-fuer-Mikrocontroller-1934018.html>, (05.02.2014)

¹⁵³ Mozillas Rhino ist eine Open-Source-Implementation von JavaScript um eingebettet in Java Applikationen dem Benutzer die Erstellung von Skripten zu ermöglichen, <https://developer.mozilla.org/de/docs/Rhino>, (08.02.2014)

Bei JavaScript handelt es sich um eine durchschnittlich komplexe Programmiersprache, die einen ausreichenden Funktionsumfang für die Programmierung eines Roboters bietet. Außerhalb von Browseranwendungen finden sich für die Sprache nur bedingt Anwendungsmöglichkeiten. Durch die Ähnlichkeiten zu Java und C++ kann das Erlernen von JavaScript weit verbreitete Grundkonzepte vermitteln, die einfach auf ähnliche Sprachen übertragen werden können.

8.2.5 Scala

Betriebssystem(e): unabhängig; Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁵⁴

Basiert auf: Java und weiteren Programmiersprachen, keine direkte Ableitung

Entwickler/Lizenz: École polytechnique fédérale de Lausanne¹⁵⁵

Kosten: kostenlose Downloads

Scala ist eine rein objektorientierte Sprache, deren Name für skalierbare Sprache steht. Damit ist gemeint, dass die Sprache in ihrem Umfang variabel ist und sich der kompakte Sprachkern von Benutzern und Benutzerinnen abändern und erweitern lässt. Scala-Programme haben nicht nur eine sehr ähnliche Syntax wie Java, sie lassen sich auch vice versa mit Java-Programmen kombinieren.¹⁵⁶

Scala Programmbeispiel – hello world:

```
object HelloWorld extends App {
    println("Hello, world!")
}
```

Die Programmiersprache Scala bietet für die Anwendung in Schulen keine ersichtlichen Vor- oder Nachteile zu vergleichbaren Sprachen wie Java, da die Unterschiede im Rahmen des Schulunterrichts (ohne spezielle Schwerpunktfom) kaum oder gar nicht zum Vorschein kommen.

¹⁵⁴ vgl. Finch, <http://www.finchrobot.com/software/scala>, (05.02.2014)

¹⁵⁵ ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, <http://www.epfl.ch/>, (05.02.2014)

¹⁵⁶ vgl. Scala, <http://www.scala-lang.org/what-is-scala.html>, (05.02.2014)

8.2.6 C

Betriebssystem(e): unabhängig; Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation)¹⁵⁷; Scribbler 2 benötigt Propeller GCC Compiler: Windows, Mac, Linux (keine Versionsinformation)¹⁵⁸; LEGO Mindstorms EV3 benötigt BricxCC IDE: Windows 7 / 8¹⁵⁹; Fischertechnik ROBO TX Training Lab benötigt Schnittstellensoftware: Windows (keine Versionsinformation) [vgl. 36 S 4]; VEX IQ benötigt ROBOTC: Windows XP / Vista / 7 / 8¹⁶⁰;

Basiert auf: wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

Entwickler/Lizenz: Dennis Ritchie und Ken Thompson

Kosten: kostenlose Downloads, außer ROBOTC (etwa 36 Euro aufwärts)

C zählt zu den am weitest verbreiteten Programmiersprachen und ist in (fast) allen Betriebssystemen verfügbar. Der Kern jedes gängigen Betriebssystems ist ebenfalls in C geschrieben. Es handelt sich um eine imperative Sprache, die eine sehr gute Performance aufweisen kann und auch für komplexe Programme mit einem überschaubaren und kurzen Code auskommt. Durch die kompakte Schreibweise ist C allerdings bei sehr umfangreichen Programmen unübersichtlicher als alternative Sprachen [vgl. 37 S 27 f.].

C Programmbeispiel – hello world:

```
#include <stdio.h>
int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

C ist eine sehr kompakte und schnelle Sprache, die für beinahe jedes Einsatzgebiet verwendbar ist. Durch die (grundsätzlich) fehlende Objektorientierung eignet sie sich nur bedingt für sehr umfassende Projekte. Bei längeren Programmen (wie es bei der Roboter-Programmierung oft der Fall ist) wird die Darstellung des Codes schnell unübersichtlich, weshalb sie sich auch nur begrenzt für den schulischen Einsatz eignet.

¹⁵⁷ vgl. Finch, <http://www.finchrobot.com/software/c-1>, (05.02.2014)

¹⁵⁸ vgl. Parallax, <http://www.parallax.com/product/28136>, (06.02.2014)

¹⁵⁹ vgl. Lego, <http://www.legomindstormsev3.com/ev3-programming/bricxcc.ev3>, (05.02.2014)

¹⁶⁰ vgl. ROBOTC, <http://www.robotc.net/download/vexrobotics/>, (05.02.2014)

8.2.7 C++

Betriebssystem(e): unabhängig; Finch benötigt Schnittstellensoftware: Windows, Mac, Linux (keine Versionsinformation); Scribbler 2 benötigt Propeller GCC Compiler: Windows, Mac, Linux (keine Versionsinformation)¹⁶¹; LEGO Mindstorms EV3 benötigt BricxCC IDE: Windows 7 / 8¹⁶²; Fischertechnik ROBO TX Training Lab benötigt Schnittstellensoftware: Windows (keine Versionsinformation) [vgl. 36 S 4]

Basiert auf: C

Entwickler/Lizenz: Bjarne Stroustrup

Kosten: kostenlose Downloads

Bei C++ handelt es sich um eine ISO¹⁶³-genormte Programmiersprache, die auf C basiert und eine Erweiterung dieser Sprache darstellt. Das Ziel war es, die schnell laufende und weit verbreitete, Sprache C für die Verwendung in größeren und komplexeren Projekten anzupassen und um Sprachelemente wie Klassen, Vererbung, Konstanten, usw. zu erweitern [vgl. 38 S 15 ff].

C++ Programmbeispiel – hello world:

```
#include <iostream>
int main()
{
    std::cout << "Hello, world!" << std::endl;
    return 0;
}
```

Da es sich bei C++ um eine der am weitesten verbreiteten Programmiersprachen handelt, ist sie vermutlich eine gute Grundlage für den Einstieg in einen technischen Beruf. Durch die Komplexität von C++ bietet diese allerdings nicht den einfachsten Einstieg in die (Roboter-)Programmierung.

8.2.8 C#

Betriebssystem(e): Windows; Finch benötigt Visual Studio Paket: Windows (keine Versionsinformation); Fischertechnik ROBO TX Training Lab benötigt Schnittstellensoftware: Windows (keine Versionsinformation) [vgl. 36 S 4]

Basiert auf: C, C++, wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

¹⁶¹ vgl. Parallax, <http://www.parallax.com/product/28136>, (06.02.2014)

¹⁶² vgl. Lego, <http://www.legomindstormsev3.com/ev3-programming/bricxcc.ev3>, (05.02.2014)

¹⁶³ Internationale Organisation für Normung

Entwickler/Lizenz: Microsoft

Kosten: kostenlose Downloads

C# ist eine, von Microsoft entwickelte, Programmiersprache, die für die Erstellung von Anwendungen für das .net Framework¹⁶⁴ gedacht ist. Es handelt sich dabei um eine leistungsstarke, objektorientierte Sprache, die auf eine schnelle Anwendungsentwicklung abzielt und auf Grundkonzepten von C und C++ basiert.¹⁶⁵

C# Programmbeispiel – hello world:

```
using System;
class Program
{
    public static void Main()
    {
        Console.WriteLine("Hello, world!");
    }
}
```

Bei C# handelt es sich um eine moderne, weit verbreitete Programmiersprache, die primär darauf abzielt, Programme auf Windows-Basis zu erstellen. Durch eine Syntax, die der natürlichen Sprache nahe ist, kann es Kindern und Jugendlichen leichter fallen, diese zu erlernen.

8.2.9 Visual Basic

Betriebssystem(e): Windows; Finch benötigt Visual Studio Paket: Windows (keine Versionsinformation); Fischertechnik ROBO TX Training Lab benötigt

Schnittstellensoftware: Windows (keine Versionsinformation) [vgl. 36 S 4]

Basiert auf: wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

Entwickler/Lizenz: Microsoft

Kosten: kostenlose Downloads

Visual Basic ist eine von Microsoft entworfene Sprache, deren Entwicklungsziel es war, eine Programmiersprache zur Verfügung stellen zu können, die Neulingen einen schnellen und einfachen Einstieg ermöglicht und gleichzeitig Experten mit leistungsstarken Funktionen Unterstützung anbietet. Die Syntax ist nahe der englischen Sprache, was Vorteile in Punkten wie Lesbarkeit des Codes und

¹⁶⁴ Softwareplattform von Microsoft zur Entwicklung und Ausführung von Anwendungsprogrammen

¹⁶⁵ vgl. Microsoft, <http://msdn.microsoft.com/de-de/library/kx37x362.aspx>, (05.02.2014)

Verständlichkeit bringt. Die Programmiersprache, sowie die dafür gängige Entwicklungsumgebung Visual Studio in der „Express Edition“, wird von Microsoft kostenlos angeboten.¹⁶⁶

Visual Basic Programmbeispiel – hello world:
MsgBox "Hello, world!"

Für den Einsatz im schulischen Sektor sollte sich Visual Basic einerseits durch dessen gute Lesbarkeit eignen, andererseits ist es von Vorteil, dass diese Sprache den nötigen Funktionsumfang bereitstellt, um sie auch über mehrere Schulstufen hinweg einsetzen zu können. Eine große Community, die hinter der Sprache steht, bietet den nötigen Support, sowie für den Unterricht geeignete Materialien und Programmbeispiele, an.

8.2.10 BASIC

Betriebssystem(e): unabhängig; BOE BOT benötigt PBASIC Compiler: Windows 2000 / XP / Vista / 7¹⁶⁷; Mac (keine Versionsinformation)¹⁶⁸

Basiert auf: wurde von mehreren Sprachen beeinflusst, keine direkte Ableitung

Entwickler/Lizenz: John G. Kemeny und Thomas E. Kurtz; Compiler: Parallax

Kosten: kostenlose Downloads

BASIC ist eine imperative, höhere Programmiersprache, die heute nur noch selten in der originalen Form verwendet wird. Viele gängige Programmiersprachen wie zum Beispiel Visual Basic leiten sich von BASIC ab. Da diese Programmiersprache nur noch selten Verwendung findet, wird hier nur kurz darauf eingegangen.

BASIC Programmbeispiel – hello world:
PRINT "Hello, world!"

Da BASIC mittlerweile nicht mehr State of the Art ist, sollte vor allem im schulischen Bereich eine Programmiersprache mit breiterem Anwendungsfeld im Fokus stehen. Zur Vermittlung grundlegender Programmierkonzepte kann BASIC aber auch heute noch verwendet werden.

¹⁶⁶ vgl. Microsoft, <http://msdn.microsoft.com/de-de/hh388573>, (06.02.2014)

¹⁶⁷ vgl. Parallax, <http://www.parallax.com/downloads/basic-stamp-editor-software>, (07.02.2014)

¹⁶⁸ vgl. Parallax, <http://www.parallax.com/product/28832>, (07.02.2014)

8.2.11 Processing

Betriebssystem(e): Windows XP / 7, Mac OSX 10.8, Linux Ubuntu 10.04, 12.04¹⁶⁹;

Finch benötigt Schnittstellensoftware: Windows, Mac (keine Versionsinformation)¹⁷⁰;

ROBO TX Training Lab benötigt Schnittstellensoftware: Windows (keine Versionsinformation)¹⁷¹

Basiert auf: Java

Entwickler/Lizenz: Ben Fry and Casey Reas¹⁷²

Kosten: kostenlose Downloads

Processing ist eine Java-basierte Programmiersprache, die auch (beinahe) dieselbe Syntax verwendet. Processing zielt auf das Einsatzfeld Grafik und Animation ab und soll dabei unterstützen, interaktive Medieninhalte zu erstellen. Als Zielgruppe können für die Processing-Programmierung eher Künstler und Designer gesehen werden, allerdings können auch nicht-grafische Programme erstellt werden.¹⁷³

Processing (Java) Programmbeispiel – hello world:

```
void setup(){
    println("Hello, world!");
}
```

Durch den vordergründigen Grafikbezug von Processing kann diese Sprache vor allem in Schulzweigen mit Fokus auf Design und Medien gewinnbringend eingesetzt werden und eine Grundlage für fächerübergreifenden Unterricht in entsprechenden Schwerpunktklassen bieten. In schulischen Zweigen ohne grafischen Schwerpunkt kann Processing grundsätzlich auch eingesetzt werden, um Grundfertigkeiten der Robotik und der Programmierung zu vertiefen, Vorteile gegenüber anderen Sprachen wie zum Beispiel Java sind in diesem Fall nicht zu erkennen.

¹⁶⁹ vgl. Processing, http://wiki.processing.org/w/Supported_Platforms, (07.02.2014)

¹⁷⁰ vgl. Finch, <http://www.finchrobot.com/downloads>, (07.02.2014)

¹⁷¹ vgl. Fischertechnik Community, <http://www.ftcommunity.de/ftComputingFinis/processingtx.htm>, (07.02.2014)

¹⁷² vgl. Processing, <http://processing.org/overview/>, (07.02.2014)

¹⁷³ vgl. ebenda

8.3 Robotersysteme und unterstützte Programmiersprachen im Überblick

Programmiersprache	Finch	Scribbler 2	Mindstorms EV3	ROBO TX Training Lab	VEX IQ	Arexx RP6 V2	Parallax Boe Bot
CREATE Lab Visual Programmer	X						
Finch Dreams	X						
Snap! ^{ooo}	X*	X*					
Scratch 2.0 (offline Editor) ^{ooo}	X*						
Greenfoot	X*						
RoboRealm	X						
S2 GUI		X					
ROBO Pro				X			
LEGO Mindstorms Education EV3 Software			X				
Modkit					X		
Python ^{oo}	X*		X*	X*			
Jython ^{oo}	X*						
LabVIEW	X**						
Java	X*		X*	X*			
Javascript	X*						
Scala	X*			X*			
C	X*	X*	X*	X*	X**	X	
C++	X*	X*	X*	X*			
C#	X*			X*			
Visual Basic	X			X*			
BASIC							X*
Processing	X*			X*			

*Gratis-Software benötigt **kostenpflichtige Software benötigt ***Hardware benötigt

Tabelle 6 - Robotersysteme und unterstützte Programmiersprachen im Überblick

9 Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde das Thema „Robotereinsatz im Schulunterricht“ behandelt, um die diesbezüglichen, fragmentarisch vorhandenen Informationen zu bündeln und einen Gesamtüberblick zu geben. Ebenfalls soll die vorliegende Arbeit als Hilfestellung dienen, um die Sinnhaftigkeit des Einsatzes der mechanischen Helfer an Schulen begründen zu können und die Auswahl erforderlicher Modelle und damit kompatibler und zweckmäßiger Programmiersprachen zu erleichtern. Dabei wurden Lehrer und Lehrerinnen als Adressaten in den Fokus gestellt.

Im Rahmen der Darstellung der aktuellen Ausgangssituation wurde eine Erhebung durchgeführt, die primär zum Ziel hatte, die Verbreitung von Robotern an österreichischen Schulen des allgemein- und berufsbildenden Segments zu erfassen. Es stellte sich heraus, dass 26 Prozent der 100 befragten Schulen aller Bundesländer aktuell Roboter im Schulunterricht einsetzen. Diese 26 Prozent verteilen sich sehr ungleichmäßig auf die Schultypen neue Mittelschulen, allgemeinbildende höhere Schulen und berufsbildende höhere Schulen. Robotersysteme finden an etwa 70 Prozent der BHS, 27 Prozent der AHS und nur an sechs Prozent der NMS Einsatz. Als Gründe gegen den Robotereinsatz wurden primär finanzielle Mängel sowie Personalmängel genannt. Zusammenfassend konnte festgehalten werden, dass Roboter an Schulen weiter verbreitet sein könnten.

Nach der Erhebung der aktuellen Situation wurden Gründe, die für eine weitere Verbreitung der programmierbaren Maschinen sprechen, dargelegt. Diese sind die Steigerung der Motivation von Schülern und Schülerinnen und ein damit einhergehender höherer Lernerfolg sowie höheres Engagement. Roboter unterstützen auch durch den praxisnahen Unterricht die Lernleistung und stellen reale Bezüge her. Vor allem dienen die Maschinen zur Interessenssteigerung im Bereich der MINT-Fächer und im Bereich der technischen Berufe. Dies trifft insbesondere auf Mädchen und Frauen zu.

Nach der Darlegung der gegenwärtigen Situation an österreichischen Schulen, bezogen auf den Robotereinsatz, und den Ausblicken infolge einer Erhöhung ihrer Einsatzquote wurden die möglichen Einsatzgebiete vordergründig behandelt.

Zusammenfassend kann festgehalten werden, dass sich die Automaten besonders für die Verwendung im Rahmen des MINT-Unterrichts eignen, weshalb die Lehrpläne entsprechender Schulfächer hinsichtlich sinnvoller Einsatzmöglichkeiten von Robotern analysiert wurden. Solche Möglichkeiten konnten in jeder Schulstufe der Gegenstände Mathematik, Physik und Informatik ausgemacht werden. Abschließend konnten auch Gründe für den fächerübergreifenden Unterricht dargelegt werden, für den sich durch das breite Einsatzfeld laut Lehrplänen Robotersysteme besonders gut eignen würden.

Nach der Erhebung der gegenwärtigen Situation an österreichischen Schulen, der ausführlichen Befassung mit der Sinnhaftigkeit des Einsatzes von Robotermodellen im Schulunterricht und der Darlegung möglicher Einsatzgebiete basierend auf den Lehrplänen in Österreich wurde eine Hilfestellung verfasst, deren Ziel es ist die Unterrichtsintegration zu erleichtern. Dafür wurden die Vor- wie Nachteile von theoretischen und praktischen Systemen besprochen, und es wurde ersichtlich, dass rein theoretische Systeme weniger erfolgversprechend sind als praktische oder kombinierte (theoretische und praktische) Systeme.

Nach der Kategorisierung von Robotersystemen in fertige Modelle, modulare Modelle und Bausatzmodelle wurden jeweils entsprechende Vorteile und Einsatzfelder behandelt. Aus jeder der festgelegten Kategorien wurden mögliche Roboter vorgestellt, die sich aufgrund ihrer Spezifikationen für bestimmte Anwendungsgebiete bezogen auf den Schuleinsatz besonders eignen.

Die Unterschiede von grafischen und textuellen Programmiersprachen, sowie damit einhergehende Vor- und Nachteile für den Schulalltag wurden dargelegt. Potenzielle kompatible Programmiersprachen, die zur Programmierung eines oder mehrerer der genannten Robotermodelle herangezogen werden können, fanden im Detail Erwähnung. Dabei wurde versucht, einen möglichst umfassenden Überblick zu präsentieren. Eine Gesamtübersicht über die Kombinationsmöglichkeiten von Robotermodellen und Programmiersprachen aller genannten Roboter und Sprachen wurde abschließend zur einfachen Systemwahl zusammengestellt.

10 Schlussbemerkung und Ausblick

Im Rahmen dieser Arbeit wurde das Thema Robotereinsatz im Schulunterricht näher betrachtet. Anhand der Untersuchung und der aufgezeigten Vorteile durch den Robotereinsatz im Unterricht konnte gezeigt werden, dass die programmierbaren Maschinen noch häufiger zur Wissensvermittlung an Schulen eingesetzt werden sollten. Der Lehrplan sowie die Theorie vom projektorientierten Lernen offerieren zahlreiche Anknüpfungspunkte für vielfältigste Robotik-Projekte.

Den vorbereitenden Überlegungen im Vorfeld eines derartigen Projektes kommt essenzielle Bedeutung zu. Sowohl die zu adressierenden Schüler und Schülerinnen als auch Inhalt und Umfang eines Robotik-Projektes müssen zu diesem Zeitpunkt aufgrund reiflicher Überlegungen festgelegt werden. In diesem Stadium findet auch die Wahl eines Systems statt. Relevante Informationen über Robotermodelle und potenziell geeignete Programmiersprachen können dieser Arbeit entnommen werden.

Aus der Untersuchung geht hervor, dass in Erwägung gezogene Robotik-Projekte mangels finanzieller oder personeller Ressourcen zum Teil jedoch nicht realisiert werden können.

Gelingt aber die Umsetzung eines Robotik-Projektes, profitieren Schülerinnen und Schüler nicht nur durch größere Lernerfolge infolge höherer Motivation. Auch kann das Interesse für eine technische Karriere geweckt werden, was aufgrund des derzeitigen Ingenieurmangels auch volkswirtschaftlich begrüßenswert wäre.

11 Quellenverzeichnis

- [01] Regenstein K.: „Modulare, verteilte Hardware-Software-Architektur für humanoide Roboter“, Verlag: KIT Scientific Publishing, 2010
- [02] Zimmermann H. H.: „Die Rolle der Schule in der Informationsgesellschaft“, online unter <http://is.uni-sb.de/zimmermann/pdf/1994a.pdf>, (27.01.2014)
- [03] Giesecke H.: „Wozu ist die Schule da?“, In: Neue Sammlung 35, Seite 93-104, 1995, online unter: http://www.uni-potsdam.de/fileadmin/projects/erziehungswissenschaft/documents/studium/Textboerse/pdf-Dateien/giesecke_wozuschule.pdf, (27.01.2014)
- [04] Kahlert H., Mansel J.: „Bildung und Berufsorientierung: Der Einfluss von Schule und informellen Kontexten auf die berufliche Identitätsentwicklung“, 1. Auflage, Verlag: Beltz Juventa, 2007
- [05] Vandeveld C., Saldien J., Ciocci M. C., Vanderborcht B.: „Overview of Technologies for Building Robots in the Classroom“, In: Proceedings of the 4th International Conference on Robotics in Education Lodz University of Technology, Polen, 2013
- [06] Wiesner B.: „Lernprozesse mit Lernumgebungen unterstützen: Roboter im Informatikunterricht der Realschule“, In: Lecture Notes in Informatics, Didaktik der Informatik - Aktuelle Forschungsergebnisse, Seiten 23-32, Gesellschaft für Informatik, Bonn, 2008
- [07] Zander S.: „Motivationale Lernvoraussetzungen in der Cognitive Load Theory: Zwei Studien zum Einfluss motivationaler Lernvoraussetzungen auf die kognitive Belastung beim Lernen mit unterschiedlichen Instruktionsdesigns“, Verlag: Logos Verlag Berlin, 2010
- [08] Papert S. A.: „Mindstorms: Children, Computers, And Powerful Ideas“, 2. Auflage, Verlag: Basic Books, 1993
- [09] Peter Protzel, Interview mit FOCUS Online über den Einsatz von Robotern im Schulunterricht, online unter: http://www.focus.de/wissen/mensch/hightech_unterricht/experte_aid_25356.html, (09.02.2014)
- [10] Roberta Abschlussbericht, online unter: http://www.iais.fraunhofer.de/uploads/media/Abschlussbericht_Roberta_2007-11-21.pdf, (13.01.2014)
- [11] Blumenfeld P. C., Soloway E., Marx R. W., Krajcik J. S., Guzdial M., Palincsar A.: „Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning“, In: Educational Psychologist, Ausgabe 26, Seiten 369-398, Lawrence Erlbaum Associates Inc., 1991
- [12] Alimisis D.: „Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods“, Verlag: School of Pedagogical and Technological Education (ASPETE), 2009
- [13] Dickmanns E. D.: „Vision: Von Assistenz zum Autonomen Fahren“, In: Fahrerassistenzsysteme mit maschineller Wahrnehmung, Seiten 203-237, Verlag: Springer-Verlag Berlin Heidelberg, 2005
- [14] Bundesgesetzblatt für die Republik Österreich, Jahrgang 2004, ausgegeben am 8.Juli.2004, Teil II, 277. Verordnung: Änderung der Verordnung über die Lehrpläne der allgemein bildenden höheren Schulen; Bekanntmachung der Lehrpläne für den Religionsunterricht
- [15] Bundesgesetzblatt für die Republik Österreich, Jahrgang 2003, ausgegeben am 23.Dezember.2003, Teil II, 571. Bekanntmachung: Lehrpläne für den katholischen Religionsunterricht an Hauptschulen, an der Unterstufe allgemein bildender höherer Schulen, an berufsbildenden höheren Schulen sowie an Berufsschulen
- [16] Bundesgesetzblatt für die Republik Österreich, Jahrgang 2000, ausgegeben am 11.Mai.2000, Teil II, 134. Verordnung: Lehrpläne der Hauptschulen; Bekanntmachung der Lehrpläne für den Religionsunterricht an diesen Schulen
- [17] Hoffmann Jörg: „Analog/Digital-Wandler“, In: Messen nichtelektrischer Größen, Seiten 190-210, Verlag: Springer-Verlag Berlin Heidelberg, 1996
- [18] Hering E., Schönfelder G.: „Temperaturmesstechnik“, In: Sensoren in Wissenschaft und Technik, Seiten 378-404, Verlag: Vieweg+Teubner, 2012
- [19] Dijkstra E. W.: „Was ist Informatik? – Eine Begriffsklärung“, online unter: http://www.informatikjahr.de/fileadmin/content/documents/Thementexte_INF/Was_ist_Informatik_0605_09.pdf, (18.01.2014)
- [20] Vorhölter K.: „Sinn im Mathematikunterricht: Zur Rolle von mathematischen Modellierungsaufgaben bei der Sinnkonstruktion von Schülerinnen und Schülern.“, Verlag: Barbara Budrich, 2009
- [21] Reiter A.: „20 Jahre Schulinformatik in Österreich und IKT-Einsatz im Unterricht“, 1. Auflage, Verlag: CDA, 2005
- [22] Rich E.: „Artificial Intelligence“, Verlag: McGraw-Hill, New York, 1983
- [23] Arnold K. H., Sandfuchs U. Wiechmann J.: „Handbuch Unterricht“, 2. Auflage, Verlag: UTB, Stuttgart, 2009

- [24] ScoutWalker III Manual - May 8, 2010, online unter: https://solarbotics.com/product/k_w3/, (07.01.2014)
- [25] E-Mail: danielbw@betterbots.com vom 07.01.2014
- [26] Rabeder K.: „Programmieraufgaben für Pidgets und den Raspberry Pi im Informatikunterricht“, Diplomarbeit am Institut für Systemsoftware am Johannes Kepler Universität in Linz
- [27] Müller U.: Benutzerhandbuch - „fischertechnik ROBO TX Controller - umFish50.DLL - Übersicht von Programmiermöglichkeiten auf Basis der ftMscLib.DLL - umFish50.DLL - FishFaceTX.DLL“, online unter: <http://www.ftcommunity.de/ftComputingFinis/pdf/umFish50.pdf>, (22.01.2014)
- [28] Parallax – Board of Education – USB – Benutzerhandbuch: online unter: <http://www.parallax.com/sites/default/files/downloads/28850-BOE-USB-Manual-v2.0.pdf>, (25.01.2014)
- [29] Sasidaran D.: „PBASIC Language Basics“, online unter: <http://peabody.sapp.org/class/dmp2/read/Pbasic.pdf>, (25.01.2014)
- [30] Klauer K. J., Leutner D.: „Lehren und Lernen: Einführung in die Instruktionspsychologie“, 1. Auflage, Verlag: Beltz, 2007
- [31] Snap - Benutzerhandbuch, <http://snap.berkeley.edu/SnapManual.pdf>, (03.02.2014)
- [32] Kölling M.: „Einführung in Java mit Greenfoot“, 1. Auflage, Verlag: Pearson Studium, 2010
- [33] E-Mail: Roger.Wolf@legoeducation.eu vom 07.02.2014
- [34] Müller U.: Benutzerhandbuch - „FishFaTX für Python“, <http://www.ftcommunity.de/ftComputingFinis/pdf/FishFaTXPy311.pdf>, (04.02.2014)
- [35] Müller U.: Benutzerhandbuch - „ftComputing für Java“, <http://www.ftcommunity.de/ftComputingFinis/pdf/Eclipse34FishTX.pdf>, (04.02.2014)
- [36] Müller U.: Benutzerhandbuch - „umFish50.DLL“, <http://www.ftcommunity.de/ftComputingFinis/pdf/umFish50.pdf>, (05.02.2014)
- [37] Wolf J.: „C von A bis Z: Das umfassende Handbuch“, 3. Auflage, Verlag: Galileo Computing, 2009
- [38] Wolf J.: „Grundkurs C++“, 2. Auflage, Verlag: Galileo Computing, 2013

12 Anhang

12.1 Fragebogen der Schulumfrage

Schulname

Schuladresse

Schulkennzahl

Sehr geehrte Schulleitung / Administration!

Ich, Martin Neppel, mache im Rahmen der Diplomarbeit meines Studiums der Unterrichtsfächer Chemie und Informatik eine Umfrage zum Thema Robotik bezogenem Schulunterricht. Ihre Schule entspricht dem Umfragebezugsfeld und wurde als Stichprobenteil zufällig ausgewählt. Da der Umfragerahmen sehr eng gezogen wurde wäre es sehr wichtig eine Antwort von Ihnen zu erhalten. Ich würde mich sehr freuen wenn Sie sich die kurze Zeit nehmen. Sollten Sie eine Frage nicht beantworten können, so lassen Sie diese bitte aus, versuchen Sie aber die Hauptfrage (fett, unterstrichen) zu beantworten. Danach sind nur noch die entsprechenden Felder der Hauptfrage zu beantworten (entweder grün [ja] **oder** gelb [nein]). Jede zusätzliche Anmerkung ist willkommen! Schreiben sie diese einfach zum jeweiligen Punkt.

Vorgehensweise: Klicken Sie im E-Mail Programm einfach auf Antworten, wählen Sie die zutreffenden Antworten bzw. füllen Sie die entsprechenden Bereiche aus und returnieren Sie die Nachricht. Für Rückfragen stehe ich Ihnen per E-Mail oder unter der Telefonnummer 0664/5135161 jederzeit zur Verfügung. DANKE!

Werden in irgendeiner Form Roboter im Unterricht verwendet? Ja Nein

Wurde jemals darüber nachgedacht? Ja Nein

Was sprach gegen eine Umsetzung - gibt es dahingehende Vorhaben?

In welchen Schulfächern?

Informatik Mathematik Physik Chemie Biologie

Weitere (z.B.: Maschinenbau, ...):

In welcher Altersstufe (Klassen)?

Schulstufen: 1. 2. 3. 4. 5. 6. 7. 8.

Institut für Softwaretechnologie

Weitere (z.b.: HTL 2., 3.):

Welches Modell und welche Programmiersprachen werden verwendet (z.B.: Lego Mindstorms NXT, Industrieroboter,...)?

Ich danke Ihnen schon jetzt für Ihre Antwort!

Beste Grüße,
Martin Neppel

12.2 [25] E-Mail: danielbw@betterbots.com vom 07.01.2014

Hi Martin

I developed the fluke at Georgia Tech alongside others working on the IPRE project. My company Betterbots has licensed the technology for manufacture and sale. You can find more information about the IPRE project at www.roboteducation.org.

Let me know if you have any additional questions.

On Tue, 7 Jan 2014, NEPPEL Martin wrote:

> Hi!

> I?m an Austrian student and I?m writing on a paper about the use of

> robots for education. I would like to know if the IPRE Fluke2 Add-On

> Board is a product of your company betterbots, or if you are a

> distributor. In the second case, can you tell me who invented the IPRE Fluke2 Add-On Board?

> Thank you for your answer!

> Regards, Martin Neppel

12.3 [33] E-Mail: Roger.Wolf@legoeducation.eu vom 07.02.2014

Sehr geehrter Herr Neppel, vielen Dank für Ihre Nachricht.

Die LEGO® MINDSTORMS Education EV3 Software erhalten Sie von uns in einer zweisprachigen Version, deutsch und englisch für die beiden Betriebssysteme Windows und Mac.

Die Systemvoraussetzungen sind recht niedrig, sodass Sie die Software problemlos auf einem Windows XP System nutzen können. Selbstverständlich ist die Software auch Windows 8 kompatibel.

Mit freundlichen Grüßen

Roger Wolf
Internal Sales Advisor
Germany Sales Direct
LEGO Education Europe

Phone

Direct 0049 89 45346350

0049 89 45346352

Fax 0049 89 45346359

Mobile 0049 152 22534060

E-mail roger.wolf@legoeducation.eu

LEGO Education Europe
Technopark II
Werner-von-Siemens-Ring 14
85630 Grasbrunn

www.LEGOeducation.de

Von: NEPPEL Martin [<mailto:martin.neppel@edu.uni-graz.at>]

Gesendet: Freitag, 7. Februar 2014 12:08

An: Verkauf LED

Betreff: LEGO MINDSTORMS Education EV3-Software

Sehr geehrtes Lego Team!

Ich benötigte ein Paar Informationen zu LEGO MINDSTORMS Education EV3-Software, die ich auf Ihrer Homepage nicht auffinden konnte. Ich wüsste gerne in

welche Sprachen die Software unterstützt, auf welchen Betriebssystemen diese läuft und welche Systemvoraussetzungen dies hat. Vielen Dank vorab für Ihre Antwort!

Beste Grüße,
Martin Neppel