



open
source
PRESS

user space

LEGO[®] Mindstorms[®] NXT

Einführung in die Roboter-Programmierung
mit NXC und Roberta[®]

Frank Knefel

Frank Knefel

LEGO® Mindstorms® NXT

Einführung in die Roboter-Programmierung mit NXC
und Roberta®

1. Auflage

Open Source Press

Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grunde sind die in dem vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor(en), Herausgeber, Übersetzer und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht, auch nicht für die Verletzung von Patentrechten, die daraus resultieren können. Ebenso wenig übernehmen Autor(en) und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. werden ohne Gewährleistung der freien Verwendbarkeit benutzt und können auch ohne besondere Kennzeichnung eingetragene Marken oder Warenzeichen sein und als solche den gesetzlichen Bestimmungen unterliegen.

Dieses Werk ist urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches – oder Teilen daraus – vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlags in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Bibliografische Information Der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Copyright © 2013 Open Source Press, München

Gesamtlektorat: Dr. Markus Wirtz

Satz: Open Source Press & Thomas Schraitle (AsciiDoc/DocBook/XSL-FO/XEP)

Roboterbilder und Soundfiles: Hanna und Karolin Kniefel

Umschlaggestaltung: Olga Saborov, Open Source Press

ISBN: 978-3-95539-024-2 (E-Book PDF)

<http://www.opensourcepress.de>

ISBN: 978-3-941841-85-7 (gedruckte Ausgabe)

Inhaltsverzeichnis

Vorwort	17
Einführung	21
I Roberta — ein Lebewesen	33
1 Roberta erwacht zum Leben	35
1.1 Hallo Welt.	36
1.1.1 Aufgaben.	45
1.1.2 Zusammenfassung.	45
1.2 Erste Bewegungen.	46
1.2.1 Aufgaben.	48
1.2.2 Zusammenfassung.	50
1.3 Roberta bewegt sich so exakt sie kann (Teil 1).	51

1.3.1	Aufgaben.	53
1.3.2	Zusammenfassung.	53
1.3.3	Exkurs: Etwas Mathematik.	54
1.3.4	Exkurs: Zeichnen mit dem nxcSimulator.	58
1.4	Roberta wiederholt Aktionen.	60
1.4.1	Aufgaben.	64
1.4.2	Zusammenfassung.	65
1.4.3	Exkurs: Vergleich von Zahlen und Werten.	65
1.4.4	Exkurs: Schleifen.	66
1.5	Roberta bewegt sich wirr.	70
1.5.1	Aufgaben.	72
1.5.2	Zusammenfassung.	72
1.5.3	Exkurs: Variablen.	73
1.5.4	Exkurs: Zuweisungen.	75
1.6	Roberta macht Musik.	76
1.6.1	Aufgaben.	79
1.6.2	Zusammenfassung.	79
1.7	Roberta bewegt sich so exakt sie kann (Teil 2).	80
1.7.1	Aufgaben.	82
1.7.2	Zusammenfassung.	84

2 Roberta bekommt Sinne 85

2.1	Roberta kann tasten.	85
2.1.1	Aufgaben.	89
2.1.2	Zusammenfassung.	90
2.2	Roberta kann sehen.	90
2.2.1	Aufgaben.	93
2.2.2	Zusammenfassung.	95
2.3	Roberta kann hören.	96
2.3.1	Aufgaben.	98
2.3.2	Zusammenfassung.	99
2.4	Roberta mit Fledermaussinnen.	99
2.4.1	Aufgabe.	102
2.4.2	Zusammenfassung.	104
2.4.3	Exkurs: Ultraschallsensoren im nxcSimulator.	104
2.5	Roberta ist multitaskingfähig.	106
2.5.1	Aufgaben.	108
2.5.2	Zusammenfassung.	108
2.6	Roberta erkennt Farben.	109
2.6.1	Aufgaben.	112
2.6.2	Zusammenfassung.	115
2.7	Projekt: Roberta zählt Radumdrehungen.	116
2.7.1	Aufgaben.	117

2.8	Projekt: Roberta hilft blinden Menschen.	119
2.8.1	Aufgaben.	121
2.8.2	Zusammenfassung.	122
2.9	Projekt: Sportlicher Wettkampf.	122
2.9.1	Aufgaben.	122
2.10	Projekt: Roberta kommuniziert mit Artgenossen.	124
2.10.1	Die Marionette.	126
2.10.2	Nachrichten austauschen.	130
2.10.3	Aufgaben.	136
2.10.4	Zusammenfassung.	136
3	Roberta bekommt ein Gedächtnis	139
3.1	Kurzzeitgedächtnis.	140
3.1.1	Aufgaben.	143
3.1.2	Exkurs: Arrays.	144
3.2	Langzeitgedächtnis.	146
3.2.1	Aufgaben.	149
3.2.2	Zusammenfassung.	154
II	Weitere Projekte mit Roberta	159
4	Roberta bei Roboterwettbewerben	161

4.1	Projekt: Roberta als Darstellerin.	162
4.1.1	Aufgaben.	166
4.1.2	Zusammenfassung.	166
4.1.3	Exkurs: Audacity.	167
4.2	Projekt: Roberta spielt Fußball.	170
4.2.1	Aufgaben.	175
4.2.2	Zusammenfassung.	176
4.3	Projekt: Roberta im Rettungseinsatz.	177
4.3.1	Aufgaben.	179
4.3.2	Zusammenfassung.	181
4.4	Projekt: Roberta beim Robot Game.	182
4.4.1	Aufgabe.	185
5	Sensoren — Funktionsweise und Eigenbau	187
5.1	Dein Körper als Widerstand.	190
5.1.1	Aufgabe.	191
5.1.2	Zusammenfassung.	192
5.1.3	Exkurs: Körperwiderstand.	192
5.1.4	Exkurs: Elektrischer Widerstand.	193
5.2	Mathematische Bestimmung des Widerstands.	193
5.2.1	Aufgabe.	196
5.2.2	Zusammenfassung.	196

5.3	Experimentelle Bestimmung des elektrischen Widerstands.	197
5.3.1	Material.	197
5.3.2	Durchführung.	199
5.3.3	Auswertung.	201
5.3.4	Aufgaben.	201
5.3.5	Exkurs: AD-Wandler.	202
5.3.6	Exkurs: Dualzahlen.	204
5.4	Sensoren selbst bauen: Temperatur- und Lichtsensor.	205
5.4.1	Aufgaben.	208
5.4.2	Exkurs: Halbleiterwiderstand.	209
5.5	Roberta mit Vogelsinnen.	211
5.5.1	Aufgaben.	213
5.5.2	Exkurs: Reedschalter.	213
5.6	Roberta bekommt eine Haut.	214
5.6.1	Aufgaben.	216
5.6.2	Exkurs: FSR-Sensor.	216
6	Roberta und die Wissenschaft	217
6.1	Experiment: Kraft und Geschwindigkeit.	217
6.1.1	Experimente.	220
6.1.2	Aufgaben.	221
6.1.3	Zusammenfassung.	222

6.1.4	Exkurs: Motor-Modi.	222
6.1.5	Zusammenfassung.	223
6.2	Projekt: Regeln und Steuern.	223
6.2.1	Aufgaben.	227
6.2.2	Zusammenfassung.	228
6.3	Projekt: Roberta im Forschungslabor (Messen).	229
6.3.1	Aufgaben.	230
6.3.2	Roberta misst Temperaturen – Eichung eines Temperatursensors.	232
6.3.3	Aufgaben.	234
6.3.4	Zusammenfassung.	237
6.4	Exkurs: SI-System.	238

III Programmierung und Programmierungsumgebung 241

7 BricxCC und nxcEditor 243

7.1	Grundlagen der Bedienung.	243
7.1.1	BricxCC.	244
7.1.2	nxcEditor.	246
7.1.3	Öffnen und Schließen von Programmen.	247
7.1.4	Speichern von Programmen.	249
7.1.5	Darstellung von Programmen.	250

7.1.6	Einzug — Tabulator.	252
7.1.7	Kompilieren von Programmen.	252
7.1.8	Öffnen von Beispielprogrammen.	253
7.1.9	Übertragen und Ausführen von Programmen.	254
7.1.10	Änderungen rückgängig machen und wiederherstellen.	256
7.1.11	Liste mit NXC-Befehlen.	256
7.1.12	Ladezustand und Speicherauslastung anzeigen.	257
7.2	Dateien verwalten.	258
7.2.1	Dateimanager.	258
7.2.2	Darstellen von Dateien des NXT-Roboters.	260
7.2.3	Darstellen von Dateien des Computers.	261
7.2.4	Herunterladen von Dateien auf den Roboter.	262
7.2.5	Heraufladen von Dateien auf den Computer.	263
7.2.6	Löschen von Dateien.	264
7.2.7	Aktualisierung von Dateien des Computers.	264
7.2.8	Starten von Programmen.	265
7.2.9	Stoppen von Programmen.	265
7.2.10	NXT-Namen ändern.	265
7.3	Wichtige Tastenkombinationen.	266
8	Der nxcSimulator	269

8.1	Bedienung.	271
8.2	Roboter bearbeiten.	275
8.3	Arbeitsfläche.	280
8.4	Projekt anlegen.	282
8.5	Darstellungsfehler beheben.	282
IV Anhang		285
A Installation der Programmiersoftware		287
A.1	BricxCC.	288
A.2	nxcEditor.	290
A.2.1	nxcEditor als Live-CD.	291
A.2.2	nxcEditor als USB-Stick.	293
A.2.3	openSUSE mit nxcEditor installieren.	297
A.2.4	nxcEditor in einem beliebigen Linux installieren.	297
A.2.5	Exkurs: Die Kommandozeile – Linux und Mac OS X. ...	301
A.2.6	nxcEditor mit VMware Player oder VirtualBox.	304
A.2.7	Test des nxcSimulators.	311
A.3	Exkurs: Firmware.	312
A.4	NXT und Mac OS X.	313
A.4.1	nexttool.	317

A.4.2	nbc.	318
B	Bauanleitungen	319
B.1	Bauanleitung: Roberta.	321
B.2	Tastensor.	322
B.3	Lichtsensord oder Farbsensord.	324
B.4	Geräuschsensord.	324
B.5	Ultraschallsensord.	325
B.6	Ultraschallsensord mit Motor.	325
B.7	Hand (mit selbst gebautem Drucksensord).	327
B.8	Bauteil zum Greifen.	328
B.9	Bauanleitung für Robertas Grundstück.	328
B.10	Materialliste für Experiment zum Widerstand.	329
B.11	Bauanleitungen für eigene Sensoren.	330
B.11.1	SP-Kabel mit original NXT-Adapterkabel.	331
B.11.2	SP-Kabel mit Lüsterklemme.	334
B.11.3	SP-Kabel mit 2-mm-Steckern.	335
B.11.4	Bauanleitung für den Lichtsensord.	339
B.11.5	Bauanleitung für den Temperatursensord.	339
B.11.6	Bauanleitung für den Drucksensord.	340
B.11.7	Bauanleitung für den Magnetsensord.	342

B.11.8	Bauanleitung für die Messfühler.....	343
--------	--------------------------------------	-----

Index		345
-------	--	-----

Vorwort

Vor einigen Jahren hatten wir an unserer Schule das große Glück, mit Hilfe von Sponsoren die Ausstattung für unseren Mathematik-Physik-Informatik-Kurs (kurz: Maphin) erneuern und Roboter anschaffen zu können.¹ Mit dem Wechsel der technischen Ausstattung musste allerdings auch der gesamte Unterricht für ein Halbjahr neu ausgearbeitet werden.

Der Unterricht sollte so angelegt sein, dass er Mädchen und Jungen gleichermaßen anspricht. Die Wahl fiel auf das vom Fraunhofer IAIS² entwickelte Roberta®-Konzept, denn hier geht es darum, bei beiden Geschlechtern Interesse an Informatik, Technik und Naturwissenschaften zu wecken und zu fördern. Zum Roberta®-Konzept gehört auch die Arbeit mit Robotern der Reihe LEGO® Mindstorms® NXT. Die Roberta®-Zentrale, an der ich meine Kursleiter-Schulung absolviert habe, befindet sich in Sankt Augustin. Über die Zentrale hinaus gibt es mittlerweile ein Roberta®-Netzwerk, das sich über ganz Deutschland erstreckt und auch

¹ Die Schule liegt in NRW – Maphin ist hier ein Wahlpflichtkurs für die Sekundarstufe I.

² <http://www.iais.fraunhofer.de/>

schon Nachbarländer in Europa erreicht hat. Für weitere Informationen zum Roberta®-Konzept empfehle ich die Homepage des Projekts.³

Es sollte allerdings keine Schülerin und kein Schüler Geld für einen Roboter oder Software ausgeben müssen, um die Inhalte des Unterrichts zu Hause nachzuarbeiten. Da unsere Schule seit Jahren Linux als Betriebssystem für die Computer nutzt, fiel die Wahl auf die Programmiersprache NXC. Leider gab es für NXC keine Linux-Entwicklungsumgebung, aus der heraus beispielsweise die selbst erstellten Programme auf den Roboter übertragen werden konnten, und erst recht keinen Robotersimulator, mit dem das Erlernte zu Hause wiederholt werden konnte. Aus der Not heraus habe ich dann für meinen Unterricht den *nxcEditor* (inkl. *nxcSimulator*) programmiert.⁴ Der *nxcEditor* verwendet den offiziellen Compiler für NXC – vielen Dank an John Hansen, der mir gestattet hat, den Compiler auch auf meiner Projektseite zum Download anzubieten.

Ursprünglich wollte ich für meine Schülerinnen und Schüler auch ein Skript schreiben – stattdessen gibt es nun dieses Buch.

Kurzum: Das Buch ist in Anlehnung an das Roberta®-Konzept entstanden, und ein großer Teil der Inhalte mit Schülerinnen und Schülern im Unterricht erprobt.

Einige Anmerkungen zum Inhalt: In den Kapiteln stehen nicht die einzelnen Elemente der Programmiersprache NXC oder die Bestandteile des NXT-Roboters im Vordergrund, sondern die Aufgaben, die der Roboter bewältigen muss. Die Befehle der Programmiersprache lernt man somit nicht erst einmal „trocken“, sondern nur in einem Zusammenhang, in dem sie benötigt werden. Neben der Programmiersprache NXC wird auch

³ <http://www.roberta-home.de>

⁴ Im Buch wird auch die Arbeit mit Windows und Mac OS X beschrieben.

die Funktionsweise einiger Sensoren erklärt, und für Bastler gibt es ein Kapitel zum Bau kostengünstiger Sensoren.

Die Kapitel bauen aufeinander auf. Einige erlauben es, frei zu entscheiden, ob man sie vertieft oder – wie in manchen Fällen möglich – zunächst überspringt. Wer das Buch einmal durchgelesen hat, wird feststellen, dass sich auch einige Kapitel oder Projekte untereinander tauschen lassen.

Am Ende jedes Kapitels befindet sich eine kurze Zusammenfassung, in der die neu vorgestellten NXC-Befehle noch einmal aufgelistet sind.

Auf die Bereitstellung von Lösungen für die Aufgaben wurde verzichtet, da es einerseits zu jeder Aufgabe nicht nur eine, sondern viele richtige Lösungen gibt, und man andererseits erst durch die Arbeit an der eigenen Lösung lernt zu programmieren. Es ist viel besser, bei auftretenden Schwierigkeiten mit jemandem über die Aufgabe und den Lösungsweg zu sprechen. Wer sich bei Problemen nur Musterlösungen anschaut, kann dadurch den Blick auf die eigenen Lösungen verlieren, die eventuell auch richtig wären.

Achtung: Wer als zukünftiger Kursleiter mit Robotern unter Verwendung der Aufgaben aus dem Buch arbeiten möchte, muss folgendes berücksichtigen: Roberta® ist eine eingetragene Marke der Fraunhofer Gesellschaft. Nur nach einer entsprechenden Schulung darf man seinen Roboter-Kurs auch Roberta®-Kurs nennen! Neben der richtigen Auswahl der Aufgaben unterliegt auch die Durchführung eines Roberta®-Kurses festgelegten Kriterien, die dieses Buch aber nicht vermittelt.

Für die Arbeit mit dem Buch sind Materialien auf der Homepage des Verlags hinterlegt.⁵ Es lohnt sich auch, hin und wieder einmal meine Projektseite zu besuchen,⁶ da ich die Software weiterentwickeln wer-

⁵ <http://www.opensourcepress.de/nxc>

⁶ <http://nxceditor.sourceforge.net>

de – wer Anmerkungen und Wünsche diesbezüglich hat, ist herzlich eingeladen, das dafür eingerichtete Forum zu nutzen.

Ich wünsche allen Leserinnen und Lesern viel Erfolg und Spaß beim Einstieg in die Roboter-Programmierung!

Frank Knefel, Januar 2013

Einführung

Was erwartet dich in diesem Buch? Falls du dir diese Frage gestellt hast, hier zunächst einige kurze Antworten:

- Du erlernst die Programmiersprache NXC, wozu du keine Vorkenntnisse benötigst.
- Anhand einer ausführlichen Anleitung kannst du einen NXT-Roboter bauen und mit ihm alle NXC-Programme ausprobieren.
- Du lernst zwei praktische Anwendungen kennen, mit denen du die NXC-Programme programmierst und auf den NXT-Roboter überträgst, und zwar unter Windows wie unter Linux (die Arbeit mit Mac OS X wird auch erläutert).
- Du erfährst, wie du eigene Sensoren anfertigst und mit dem NXT-Roboter nutzt.

Darüber hinaus enthält das Buch viele Aufgaben und Projekte – hier kannst du also dein NXC-Wissen mit dem selbst gebauten Roboter erproben!

Wer ist Roberta?

Du brauchst einen Roboter — wir nennen ihn (bzw. sie) im Buch „Roberta“ in Anlehnung an das vom Fraunhofer IAIS entwickelte Roberta®-Konzept, denn hier geht es darum, bei beiden Geschlechtern Interesse an Informatik, Technik und Naturwissenschaften zu wecken und zu fördern.¹ Wenn es sich um deine erste Roberta handelt, du also noch keine oder wenig Erfahrung im Bau von Robotern hast, solltest du eine Bauanleitung verwenden. Es ist gar nicht so einfach, aus den Bauteilen etwas zu erschaffen, das stabil und zugleich flexibel genug ist, die notwendigen Bewegungen auszuführen und alle Sensoren aufzunehmen.

In Abschnitt B.1 findest du eine Bauanleitung, die ausschließlich Materialien aus den Basissets des LEGO Mindstorms NXT Systems nutzt (Educationset 9797, Spielwarensatz 8527 und 8547). Alle Aufgaben aus dem Buch kannst du mit dieser Roboterfrau lösen — nur für einige wenige Projekte benötigst du zusätzliche Erweiterungen.

Hast du Roberta nach dieser Anleitung erstellt, musst du sie unbedingt so gestalten, dass sie einem Lebewesen ähnlich wird. Deiner Phantasie sind keine Grenzen gesetzt: Sei kreativ und gib Roberta Augen, Nase (oder hat sie sogar zwei?), Arme, Ohren, Greifer, Tentakeln, ... — achte aber darauf, dass ausreichend Platz für Robertas Sensoren frei bleibt. Roberta benötigt sie für die Lösung einiger Aufgaben, so dass du wissen musst, wo sie anzubringen sind. Vielleicht gibt es dazu eine freie Bauanleitung oder eine Roberta, die schon mit allen Sensoren ausgestattet ist. Es wäre ja schade, wenn du die von dir angebrachten Bauteile, die Roberta etwas „Leben einhauchen“, wieder entfernen müsstest.

¹ Näheres dazu im Vorwort dieses Buches und natürlich auf der Webseite des Roberta-Projekts <http://www.iais.fraunhofer.de/>

Roberta und ihre Gefährten sind Amphibien, die sich im Wasser und auf dem Land fortbewegen. Die Idee ist dem Biorobotik-Projekt einer bekannten Schweizer Hochschule² nachempfunden. Deren „Salamandra robotica“ wurde für den Katastrophenschutz entwickelt; wir stecken uns hier aber weniger hohe Ziele: Roberta wird sich in den Aufgaben jedes Kapitels darauf beschränken, einzelne Wesenszüge und Handlungsweisen eines Lebewesens nachzuahmen. In kleinen Projekten werden u.a. auch Aspekte internationaler Roboterwettbewerbe erläutert, in denen sich Roberta beweisen muss.

Was brauchst du zum Programmieren?

Ist deine Roberta fertig, kommen wir zu unserem eigentlichen Anliegen: Wir wollen ihr sagen, was sie tun soll – und natürlich, *wie* sie es tun soll. Diese Anweisungen geben wir ihr in Form von Computer-Programmen, die wir in der Programmiersprache NXC schreiben.

Einen NXT-Roboter muss man nicht zwingend mit NXC programmieren. Er „versteht“ verschiedene Programmiersprachen, mit denen man unter unterschiedlichen Betriebssystemen programmieren kann. Einige dieser Sprachen sind kostenlos, andere kosten Geld – einige gibt es nur für ein bestimmtes Betriebssystem, andere laufen unter Windows, Linux und Mac OS X. Die Wahl für dieses Buch fiel auf NXC, da es gratis ist und für viele Betriebssysteme zur Verfügung steht.

Die von John Hansen (er wohnt in Nashville, Tennessee) entwickelte und frei zur Verfügung gestellte Programmiersprache NXC kannst du unter Linux-, Windows- und Apple-Betriebssystemen verwenden. Die drei

² École polytechnique fédérale de Lausanne (EPFL), <http://www.epfl.ch/>

Buchstaben stehen für „Not eXactly C“, das bedeutet für dich, dass du zugleich einen Einblick in die Programmiersprache C bekommst, mit der sehr viel professionelle Software entwickelt wird; auch die Programmiersprache C gibt es für jedes Betriebssystem.

Grundsätzlich kannst du deine NXC-Programme, also den *Code*, mit jeder beliebigen Schreib-Software schreiben und speichern. Es gibt jedoch spezielle Anwendungen, sogenannte *Entwicklungsumgebungen*, die viele Arbeitsschritte erleichtern und dir einen guten Überblick über deine Programme liefern. Zwei dieser geeigneten Entwicklungsumgebungen sind kostenlos und kommen hier im Buch zum Einsatz: BricxCC und nxcEditor.

Das nur unter Windows einsetzbare Programm *BricxCC*³ gibt es lediglich in einer englischen Version. Es umfasst viele Zusatzprogramme, mit denen du z.B. ganz einfach Musik mit dem NXT-Roboter machen kannst. BricxCC bietet sehr einfachen Zugriff auf viele Funktionen des Roboters und ist für komplexere Programme zu empfehlen.

Den *nxcEditor*⁴ für Linux gibt es in deutscher und englischer Sprache. Im nxcEditor ist ein Robotersimulator integriert, mit dem du NXC-Programme auch ohne NXT-Roboter testen kannst. Der Simulator kennt allerdings nicht alle Befehle der Programmiersprache NXC – er unterstützt lediglich die ersten Programmiererfahrungen. Viele der im Buch erstellten Beispielprogramme kann der nxcSimulator allerdings ausführen. Im nxcEditor entscheidest du ganz einfach per Mausklick, ob dein Programm im Simulator oder auf dem Roboter ausgeführt wird. Den nxcEditor musst du aber nicht einmal unbedingt installieren, denn es gibt ihn für die meisten Computer auch als Live-System.

³ Das Programm bekommst du im Internet unter <http://bricxcc.sourceforge.net>

⁴ Das Programm *nxcEditor* erhältst du unter <http://nxceditor.sourceforge.net>.

Für die Programmierung des ersten Beispiels (Abschnitt 1.1) gibt es eine Anleitung sowohl für BricxCC als auch für den nxcEditor; in den folgenden Kapiteln findest du immer wieder Tipps und Aufgaben für den nxcSimulator.

Was ist der NXT-Stein?

Ohne den programmierbaren NXT-Stein (Abbildung 1) wäre Roberta nicht lebensfähig. Über ihn beeinflusst du ihr Verhalten, bekommt sie ihre Sinne und nimmt mit der Außenwelt Kontakt auf; er ist quasi ihr Herz und Gehirn zugleich.



Abbildung 1: NXT-Stein

An dem NXT-Stein befinden sich Anschlüsse (*Ports*), an die du Sensoren, Motoren und Lämpchen ansteckst. Die Sensoren werden mit den Ports 1, 2, 3 und 4 verbunden, die Motoren mit den Ports A, B und C; die Buchsta-

ben und Zahlen stehen oberhalb der Anschlüsse. Über den letzten Anschluss verbindest du den NXT-Stein per USB-Kabel mit einem Computer.

Den NXT-Stein schaltest du mit der orangefarbenen Taste in der Mitte ein. Ausgeschaltet wird er, indem man die rechteckige graue Taste so oft drückt, bis auf dem Display *Turn off?* zu lesen ist (Abbildung 2) – danach ist noch einmal die orangefarbene Taste zu drücken.

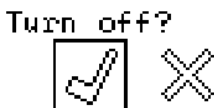


Abbildung 2: NXT-Stein ausschalten

Nach dem Anschalten befindet man sich im *Hauptmenü*, das nicht wie bei einem Computerprogramm vollständig, sondern nur teilweise zu sehen ist. Abbildung 3 zeigt alle Symbole des Hauptmenüs.



Abbildung 3: Alle Symbole des Hauptmenüs

Mit den dreieckigen Tasten bewegst du die einzelnen Menüpunkte in den Anzeigebereich. In ein Menü wechselt man mit der orangefarbenen Taste hinein und mit der rechteckigen grauen Taste wieder heraus. In jedem Untermenü ist die Bedienung gleich. Um eine im Untermenü ausgewählte Funktion auszuführen, wählt man diese mit der orangefarbenen Taste.

Sehen wir uns die Funktionen in den einzelnen Untermenüs genauer an:

My Files



Über das Menü My Files („meine Dateien“) findest du Dateien, die sich auf dem NXT-Stein befinden. Im Untermenü Software files befinden sich die vom Nutzer geschriebenen Programme. Wenn du nach dem Anschalten aus dem Hauptmenü immer wieder die orangefarbene Taste drückst, startet automatisch das zuletzt auf den NXT-Stein übertragene Programm.

NXT Program



Über die Untermenüs von NXT Program kannst du direkt auf dem NXT-Stein kurze Programme schreiben und ausführen. Dazu müssen z.B. die Sensoren so verbunden sein, wie es im ersten Untermenü beschrieben wird.

NXT Datalog



Hinter dem Menü NXT Datalog verstecken sich Untermenüs, über die man Messdaten über einen längeren Zeitraum aufzeichnet und anschließend auch speichert.

View



Die Untermenüs von View dienen dazu, sich die aktuellen Werte der angeschlossenen Sensoren anzeigen zu lassen. Zunächst wählt man

einen Sensor aus, dann den Port, an den er angeschlossen wurde – im nächsten Schritt erscheinen auf dem Display die aktuellen Werte des Sensors. Ist z. B. ein Lichtsensor angeschlossen, kann man Helligkeitsänderungen direkt verfolgen.

Bluetooth



Das Menü Bluetooth umfasst die Untermenüs für die Bluetooth-Kommunikation. Wenn Bluetooth aktiviert ist, erscheint das Bluetooth-Symbol in der oberen linken Ecke des Displays.

Settings



Über Settings erreichst du die Einstellungen des NXT-Steins. Hier änderst du die Lautstärke oder schaust unter NXT Version nach, welche *Firmware* (FW) sich auf dem NXT-Stein befindet (Abschnitt A.3).

Try Me



Im Menü Try Me finden sich kleine Demo-Programme, die man über die Untermenüs erreicht. Falls dieser Menüpunkt nicht funktioniert, wurden die Demo-Programme gelöscht (du benötigst sie hier auch nicht). Abhilfe schafft ein wiederholtes Aufspielen der Firmware.

Wie findest du Programmierfehler?

Falls du noch keinen Programmierfehler in einem Programm für Roberta gemacht hast, musst du diesen Abschnitt nicht bis zum Ende lesen. Empfehlenswert ist er, wenn du plötzlich auf der Suche nach einem Fehler bist und diesen nicht findest.

Überprüfe zunächst, ob der von dir gewählte Programmname tatsächlich die vorgeschriebene Endung `.nxc` hat. Fehlt diese Endung, gibt der Compiler Fehlermeldungen aus, die man keiner Zeile des Programms zuordnen kann (BricxCC und der nxcEditor machen dich auf diese Fehler aufmerksam).

Wenn ein Programm zwar erfolgreich kompiliert wird, dann aber auf dem Roboter nicht läuft, sind mehrere Fehlerquellen denkbar: Es ist möglich, dass der Speicher des Roboters so voll ist, dass das Programm nicht mehr erfolgreich übertragen werden kann – öffne den Dateimanager (siehe Abschnitt 7.2.1) und lösche überflüssige Dateien.

Ein weiteres Problem kann das USB-Kabel sein, das nicht am Stecker, sondern zu oft unsanft am Kabel aus dem PC oder dem NXT gezogen wurde – du musst es dann durch ein neues Kabel ersetzen, da die Drähte im Inneren beschädigt sind.

Manchmal hilft es, den Akku bzw. die Batterien des NXT-Steins kurz zu entfernen.

Wenn in einem Programm Tippfehler vorhanden sind, zeigt die Fehlerausgabe des Compilers bereits die fehlerhafte Zeile des Programms an. Dabei darfst du allerdings nur die erste Angabe beachten und solltest die darin enthaltene Zeilenangabe suchen. In dieser Zeile oder in der Nähe dieser Zeile liegt der Programmfehler. Manchmal entspricht die Beschreibung des Fehlers nicht ganz dem eigentlichen Tippfehler – dann musst du dein Programm sehr gewissenhaft überprüfen.

Wenn du den Fehler behoben hast, kompilierst du das Programm noch einmal. Manchmal schleichen sich aber auch logische Fehler in ein Programm ein; du hast zwar alles richtig geschrieben und der Compiler meldet keinen Fehler, aber der Roboter macht einfach nicht das, was er soll. Wenn ein professioneller Programmierer einen logischen Fehler nicht direkt findet, macht er manchmal einfach eine Pause, um an etwas anderes zu denken und den Kopf für eine neue Idee zur Fehlersuche frei zu bekommen. Es ist ganz normal, dass man manchmal sehr lange überlegen muss, um einen Fehler zu finden – den Frust, der dabei entsteht, musst du ertragen!⁵ Solltest du trotz längerer Überlegungen einen Fehler nicht finden, helfen dir vielleicht die folgenden Tipps:

- Vielleicht liefert ein Sensor gar nicht den Wert, den du in deinem Programm erwartest – gib den Wert des Sensors mit dem NumOut-Befehl aus. Statt eines Sensors kann es auch der Inhalt einer Variablen sein; lass dir einfach die Werte der Variablen oder mehrerer Variablen im NXT-Display anzeigen – wenn du den Fehler gefunden hast, löschst du die Zeilen für die Ausgabe der Variablen wieder.
- Wenn der Fehler erst am Ende eines lange laufenden Programms auftritt und es sehr mühsam ist, immer bis zu dieser Stelle im Programm zu warten, solltest du den vorderen Teil des Programms mit `/* . . . */` auskommentieren.
- Es kommt vor, dass ein Programm sich einfach selbst beendet oder nicht mehr weiter läuft. In diesem Fall musst du die Stelle finden, in der sich der Fehler versteckt. Der Trick mit den Kommentarzeilen

⁵ Einer meiner Schüler, der schon sehr lange eigene Programme schreibt, hat einmal zwei Stunden benötigt, um festzustellen, dass man den Befehl für den Sinus in NXC am Anfang groß schreibt: also `Sin` und nicht `sin`.

`/*...*/` funktioniert auch hier: Zunächst solltest du das ganze Programm bis auf `task main ()` und die zwei geschweiften Klammern `{...}` auskommentieren. Danach lässt du das Programm kompilieren und startest es auf dem Roboter. Wenn der Compiler keinen Fehler ausgibt und der Roboter wie erwartet funktioniert, nimmst du einen Teil des fehlerhaften Programms wieder dazu, indem du eine der von dir verwendeten Kommentarzeilen entfernst und an einer anderen Stelle im Programm wieder einfügst. Du kannst natürlich auch mehrere Teile im Programm auskommentieren. Das machst du so lange, bis du die fehlerhafte Stelle gefunden hast.

- Solltest du den Fehler nicht selbst finden, lasse jemand anderen dein Programm anschauen. Manchmal ist man für seine eigenen Fehler „blind“. Wenn du niemanden kennst, der dein Problem löst, kannst du auch ein Forum im Internet aufsuchen (falls du noch nicht volljährig bist, musst du dazu natürlich eine Erlaubnis deiner Eltern haben). Bei einer Programmiersprache wie NXC ist es einfach, seinen Fehler in einem Forum zu beschreiben – den Quellcode kann man im Forum direkt einfügen.

Exkurse und Projekte

Du wirst im Buch auf viele Exkurse und Projekte stoßen.

In den Exkursen findest du Zusatzinformationen zu den aktuellen Themen eines Kapitels. Manche Exkurse sind wichtig, um einen Sachverhalt überhaupt zu verstehen. Bei einigen Exkursen kannst du selbst entscheiden, ob und wann du sie liest.