
Lejos

LEGO-Mindstorms-Roboter in Java programmieren

Jürgen Stuber

`juergen@jstuber.net`



Die Mindstorms-Großfamilie

					
RCX	Spybot	Cyber- master	Scout	Micro- Scout	Code- Pilot
NQC Lejos BrickOS pbForth ⋮	NQC	NQC	NQC Tasten	VLL	VLL Barco- des

Der RCX

- 3 Sensoreingänge (10 Bit A/D-Wandler)
- 3 Motorausgänge A/B/C (9V, PWM)
- Infrarotschnittstelle (2400 Bit/s, kein IrDA)
- 4-stelliges LCD
- 4 Tasten
- Stromversorgung über 6 AA-Batterien
(Schulversion mit Trafoanschluß)
- 8 Bit Hitachi H8/300 Mikrokontroller 16MHz
64KB Adressraum: 16KB ROM, 32KB RAM



Infrarot-Kommunikation mit dem PC



Serieller Tower



USB Tower

Beide laufen unter Linux

Freie Programmierumgebungen

NQC (Not Quite C)

setzt auf die original LEGO-Firmware auf
auch für Bausteine mit fester Firmware
Ähnlichkeit mit C bzgl. Syntax

Lejos

Java-Umgebung → gleich mehr

pbForth

Forth (Postfix, Stacksprache), interaktiv

BrickOS

Betriebssystem für echte C-Programme

und viele weniger bekannte ...

Fast unbegrenzte Möglichkeiten

- Rover, Greifarm, Tisch abräumen, Seilbahn
- Tresor, Plotter
- Balancieren
- Datenerfassung (Temperatur)
- LEGO-Eisenbahn analog oder digital steuern
- Wettkämpfe: Sammeln, Dragster-Rennen, Linie folgen, Sumo, ...

Ein RCX-Wimmelfilm

The
Cooperative Cybernetic Crate
Contraption

by

Seattle Mindstorms and Robotic Techies
(S.M.A.R.T.)



Lejos

Java-Umgebung für den RCX:

- Compiler (dünner Wrapper um `javac`)
- Linker
- Virtuelle Maschine für den RCX (ca 16kB)
- Spezielle Bibliothek für die RCX-Hardware

Schneller und mächtiger als NQC
Einfacher als BrickOS

Programmieren

1. Programm schreiben

Emacs, vi, Eclipse, ...

2. Programm übersetzen

```
lejosc Programm.java
```

3. Programm linken

```
lejoslink Programm.java -o Programm.bin
```

4. Programm auf den RCX laden

```
lejosdl Programm.bin
```

5. Program starten

Am RCX oder per Fernbedienung

MotorDemo

SensorDemo



Sensortypen

SENSOR_TYPE_RAW	Rohwerte, kein bestimmter Typ
SENSOR_TYPE_TOUCH	Berührungssensor
SENSOR_TYPE_TEMP	Temperatursensor
SENSOR_TYPE_LIGHT	Lichtsensor
SENSOR_TYPE_ROT	Rotationssensor

Sensorbetriebsarten

SENSOR_MODE_RAW	Rohwerte, 0–1023
SENSOR_MODE_BOOL	0 oder 1
SENSOR_MODE_EDGE	Zählt Flanken
SENSOR_MODE_PULSE	Zählt steigende Flanken
SENSOR_MODE_PCT	Prozentwerte, 0–100
SENSOR_MODE_DEGC	Temperatur in °C
SENSOR_MODE_DEGF	Temperatur in °F
SENSOR_MODE_ANGLE	Winkel in 1/16 Umdrehungen

LejWay



View



Sprachumfang

Java als Sprache: fast komplett
Die wichtigsten Einschränkungen

- keine Garbage Collection
- keine Arithmetik für double, long
- kein switch
- Arrays nur bis 511 Elemente
- keine Klassenobjekte

Bibliotheken

Java

String, StringBuffer, Math, Thread, System, Random, ...

IO-Streams für die Kommunikation

(alles notwendigerweise mehr oder weniger eingeschränkt)

RCX

Motoren, Sensoren, Kommunikation, LCD, Tasten, Timer,
Töne, Batteriespannung

Callbacks bei Ereignissen (Sensoren, Timer, Knöpfe, IR)

Robotik

Behavior (Verhalten), Navigation

Kommunikation

- Standardprotokoll (F7):
einzelnes Bytes an alle
Interaktion mit NQC
- LNP:
Adressierung an einzelne RCX
Sicherung gegen Übertragungsfehler
- Empfänger für Fernbedienung
- ... und weitere Protokolle ...
- Darauf aufbauend Datenströme gemäß JAVA API
- Mini-Webserver auf RCX mit Proxy auf PC

Verhalten

Behavior API

- komplexes Verhalten aus einfachen Einzelaktionen
- Prioritäten: wichtige Teilverhalten unterdrücken unwichtige
- jedes Teilverhalten ist ein Objekt

`b.takeControl()` : Will das Verhalten aktiv werden?

`b.action()` : das Verhalten ausführen

`b.suppress()` : das Verhalten unterdrücken

BehaviorDemo



Sumo



Ende

<http://www.lejos.org>

<http://www.jstuber.net>

