

## LEGO Mindstorms RCX-Roboter mit Java programmieren



### Einen LEGO Mindstorms RCX-Roboter mit Java programmieren

Prinzipiell lassen sich Java-Programme für LEGO Mindstorms RCX-Roboter mit jedem beliebigen Editor erstellen und durch mit der LeJos-Installation mitgelieferte Skripte kompilieren und auf den RCX-Baustein hochladen.

Der von Gerhard Röhner auf dem Hessischen Bildungsserver bereitgestellte *Java-Editor* bietet eine komfortable und zugleich überschaubare Entwicklungsumgebung, die über seine graphische Oberfläche das Kompilieren und Übertragen erleichtert.

### Auf Motoren und Sensoren des Roboters zugreifen

Erstellen Sie mit dem Java-Editor zunächst eine neue Java-Klasse mit einer main-Methode. Um die LeJos-Klassen für Motoren und Sensoren nutzen zu können, sollte das entsprechende LeJos-Paket importiert werden: `import josx.platform.rcx.*;`

Das Paket ist unter <http://lejos.sourceforge.net/rcx/api/index.html> dokumentiert.

Die Klasse `Motor` stellt über die statischen Referenzen `A`, `B` und `C` eine Möglichkeit, über `Motor`-Objekte die Motoren des Roboters zu steuern. Wichtige Methoden sind

- `setPower(leistung)`      Stellt die Leistung des Motors ein.  
Zulässige Werte sind 0 bis 7.
- `forward()`              Veranlasst den Motor vorwärts zu drehen.
- `backward()`             Veranlasst den Motor rückwärts zu drehen.
- `stop()`                    Hält den Motor an.

Die Klasse `Sensor` stellt über die statischen Referenzen `S1`, `S2` und `S3` eine Möglichkeit, über die Methoden `readValue()` und `readBooleanValue()` der `Sensor`-Objekte Werte der angeschlossenen Sensoren zu ermitteln, z. B.

```
int aktuellerWert = Sensor.S1.readValue();  
boolean gedrueckt = Sensor.S2.readBooleanValue();
```

Vor dem ersten Auslesen sollten Sensoren durch den Aufruf ihrer Methode `activate()` eingeschaltet werden.

### Werte auf dem Display ausgeben

Die Klasse `LCD` stellt über die statischen Methoden `clear()` und `showNumber(int x)` eine Möglichkeit das Display des Roboter-Bausteins zu löschen bzw. einen Wert `x` auszugeben.

## Fahre bis zum Hindernis - ein Beispiel-Programm

Im folgenden Beispiel fährt der Roboter so lange gerade aus, bis der auf Port 1 angeschlossene Berührungssensor betätigt wird:

```
import josx.platform.rcx.*;

public class fahreBisTouch1 {

    public static void main(String[] args) throws Exception {
        Motor.B.setPower(5);
        Motor.C.setPower(5);
        Sensor.S1.activate();
        while (Sensor.S1.readBooleanValue() == false) {
            Motor.B.forward();
            Motor.C.forward();
            Thread.sleep(100);
            Motor.B.stop();
            Motor.C.stop();
        }
    }
}
```

## Das Programm auf den Roboter übertragen

Haben Sie Ihr Roboter-Programm fertig gestellt, so sollten Sie es mit dem Menüpunkt *Start >> Compilieren* kompilieren. Stellen Sie nun sicher, dass der Roboter vor dem mit dem USB-Kabel an den Rechner angeschlossenen Infrarot-Turm steht und



Abb. 1 Menü *Start* zum Kompilieren und Übertragen von Programmen

eingeschaltet ist und übertragen Sie das Programm mit dem Menüpunkt *Start >> Starte Applikation* auf den Roboter (siehe Abb. 1).

Auf dem RCX-Baustein können Sie das übertragene Programm nun mit dem grünen Knopf starten und ggf. wieder beenden (siehe Abb. 2).

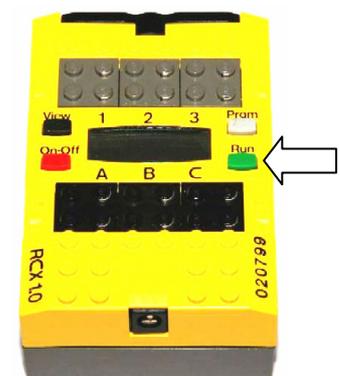


Abb. 2. Ansicht eines RCX-Bausteins

## Eine Klasse Roboter programmieren

Steigt die Komplexität der Programme, so wird es unabdingbar Teilschritte in Methoden auszugliedern. Hier bietet es sich an, Fähigkeiten eines Roboters in einer eigenen Klasse `Roboter` zusammenzufassen und ggf. durch geeignete Methoden zu ergänzen. In einer weiteren Klasse kann dann in einer `main`-Methode ein Objekt der Klasse erzeugt werden und durch eine geeignete Sequenz von Methodenaufrufen auf dem `Roboter`-Objekt das gewünschte Verhalten programmiert werden. Für das oben genannte Beispiel sollten in der Klasse `Roboter` Methoden zum vorwärts Fahren und zur Abfrage des Berührungssensors bereitgestellt werden:

```
import josx.platform.rcx.*;
public class Roboter {
    public void vor() throws Exception {
        Motor.B.setPower(5);
        Motor.C.setPower(5);
        Motor.B.forward();
        Motor.C.forward();
        Thread.sleep(100);
        Motor.B.stop();
        Motor.C.stop();
    }
    public boolean vorHindernis() {
        Sensor.S1.activate();
        return Sensor.S2.readBooleanValue();
    }
}
```

Die Klasse `fahreBisTouch2` sieht nun wesentlich übersichtlicher aus:

```
public class fahreBisTouch2 {
    public static void main(String[] args) throws Exception {
        Roboter meinRoboter = new Roboter();
        while (meinRoboter.vorHindernis() == false)
        {
            meinRoboter.vor();
        }
    }
}
```

Dieses Vorgehen ermöglicht es ähnlich wie bei so genannten „Mini-Sprachen“ wie *Kara* dem Marienkäfer oder *Niko* dem Roboter vorzugehen, nur dass hier nicht virtuelles Verhalten sondern das tatsächliche Verhalten eines Roboters gesteuert wird!

## Motivierende Aufgaben

Motivierende Aufgaben sollten Vorgänge simulieren, für die in der realen Welt Roboter eingesetzt werden. Dies wären z. B. Roboter, die mittels eines Lichtsensors eine Linie auf dem Boden verfolgen

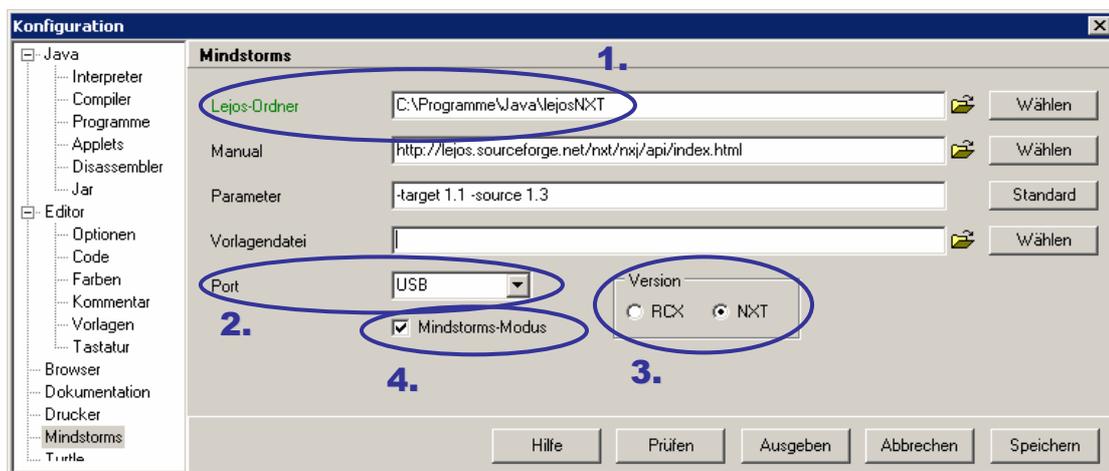


(Vorbild autonome Transportroboter in Produktionshallen) oder Roboter, die sich selbständig im Raum bewegen und dabei mittels eines Ultraschallsensors Hindernissen ausweichen (Vorbild „intelligenter“ Staubsauger/ Rasenmäher oder Rescue-Roboter zur Bergung von Verletzten).



## Einen Computer für die Arbeit mit LeJos einrichten

- Stellen Sie sicher, dass das *Java SE Development Kit* (JDK) installiert ist (ggf. unter <http://java.sun.com/javase/downloads/> herunterladen und installieren).
- Das freie *LeJos*-Paket von Sourceforge herunterladen und installieren:
  - für den NXT <http://lejos.sourceforge.net/nxj-downloads.php>
  - für den RCX <http://lejos.sourceforge.net/rcx-downloads.php>
- Den *Java-Editor* für das Programmieren von Mindstorms Robotern einrichten:
  - Stellen Sie sicher, dass der Java-Editor installiert ist (ggf. unter (<http://lernen.bildung.hessen.de/informatik/javaeditor/index.htm>) herunterladen und installieren.
  - Den Java-Editor starten und im Menü *Fenster >> Konfiguration* aufrufen, im linksseitigen Menü den Eintrag *Mindstorms* auswählen und
    1. im ersten Textfeld als *Lejos-Ordner* das Verzeichnis der LeJos-Installation eintragen,
    2. als Port *USB* auswählen,
    3. die eingesetzte Minstorms-Version *NXT* oder *RCX* auswählen
    4. und die Checkbox *Mindstorms-Modus* aktivieren:



- Nun kann über das Menü *Start >> Firmware Download* die LeJos-Firmware auf den Roboter-Baustein übertragen werden:



- Voilà - Sie können nun Ihre Programme erstellen, kompilieren, auf den Roboter-Baustein übertragen und dort ausführen!