

# Installation von NQC

Philipp Bodewig, Christoph Weiler

7. August 2006

## Inhaltsverzeichnis

<b>1</b>	<b>Voraussetzungen</b>	<b>2</b>
<b>2</b>	<b>Erfüllen der Voraussetzungen</b>	<b>2</b>
2.1	Kernelmodule . . . . .	2
2.2	Deviceknoten . . . . .	2
2.3	Installation der benötigten Pakete . . . . .	3
2.4	FLEX . . . . .	3
2.5	GNU M4 . . . . .	3
2.6	GNU Bison . . . . .	4
<b>3</b>	<b>Installation von NQC</b>	<b>4</b>
<b>4</b>	<b>Installation der Firmware</b>	<b>5</b>

Dieser Artikel beschreibt die Installation der Lego Mindstorms Entwicklungsumgebung NQC unter Suse Linux 9.2 ohne Administratorzugang. Er sollte auch auf andere Betriebssysteme anwendbar sein.

Wir benutzen keine vorgefertigten Pakete sondern kompilieren und installieren alle Programme lokal. Wenn NQC für mehrere Benutzer verfügbar sein soll ist eine systemweite Installation die bessere Wahl.

Die ausführbaren Dateien werden im Ordner `/home/(user)/local/bin` installiert und der Ordner zur `PATH`-Umgebungsvariable hinzugefügt, sodass alle Programme direkt aus der Shell aufgerufen werden können.

Das Verzeichnis mit den Programmquellen (`/home/(user)/src`) kann nach der erfolgreichen Installation gelöscht werden.

Der Ausdruck `"/home/(user)"` bzw. `/home/<user>` muss natürlich durch das eigene Heimatverzeichnis ersetzt werden.

Fragen und Anmerkungen bitte per Mail an `bodewig @ stud.uni-heidelberg.de`.

## 1 Voraussetzungen

Voraussetzungen für die Installation von NQC sind:

- Die Kernelmodule für die USB-Schnittstelle (*usbcore*) und darauf aufbauend das Modul für den Mindstorms USB-Tower (*legousbtower*)
- Der Deviceknoten `/dev/usb/legousbtower0` (standardmäßig nicht von SuSE angelegt)
- Zum kompilieren von NQC:
  - FLEX[4]
  - GNU Bison[5]
  - GNU M4[6] (Voraussetzung zur Kompilation von Bison)

## 2 Erfüllen der Voraussetzungen

### 2.1 Kernelmodule

In diesem Abschnitt muss der USB-Tower mit dem Rechner verbunden sein!

Wenn der Befehl

```
lsmod | grep legousbtower
```

ohne Ausgabe terminiert ist das Modul nicht geladen und es sind weitere Maßnahmen am Kernel nötig. In unserem Falle ist das Modul im Kernel vorhanden und wurde mit dem Einstecken den USB-Towers auch korrekt eingebunden.

### 2.2 Deviceknoten

Der Befehl

```
ls /dev/usb/
```

listet alle Knoten im Verzeichnis `/dev/usb` auf. Wenn der Knoten `legousbtower0` nicht vorhanden ist muss er vom Systemadministrator mit dem Befehl

```
mknod -m 666 /dev/usb/legousbtower0 c 180 160
```

erstellt werden.

## 2.3 Installation der benötigten Pakete

Da wir keinen Administratorzugang haben legen wir das Verzeichnis `local` im Heimatverzeichnis mit dem Befehl

```
mkdir ~/local
```

an, um dort die benötigten Pakete zu installieren. Um einen komfortableren Zugang zu den ausführbaren Dateien zu haben, erweitern wir die `PATH`-Variable um `/home/(user)/local/bin`.

```
export PATH=/home/<user>/local/bin:$PATH
```

Da nach einem erneuten Einloggen `PATH` zurückgesetzt wird, tragen wir diesen Befehl auch in die Datei `/home/(user)/.bashrc` ein, die bei jedem Login ausgeführt wird.

```
echo "export PATH=/home/<user>/local/bin:\$PATH" >> ~/.bashrc
```

Des weiteren legen wir das Verzeichnis `/home/(user)/src` an, um dort die Programmquellen zu kompilieren.

## 2.4 FLEX

Zur Installation von Flex wechseln wir zunächst ins unser gerade erstelltes `src`-Verzeichnis

```
cd ~/src
```

um dort die Quellen für Flex herunterzuladen und zu entpacken,

```
wget http://ftp.gnu.org/non-gnu/flex/flex-2.5.4a.tar.gz
tar xvfz flex-2.5.4a.tar.gz
```

und anschließend zu installieren

```
cd flex-2.5.4
./configure --prefix=/home/<user>/local && make && make install
```

## 2.5 GNU M4

Die Installation von M4 läuft ähnlich ab wie die gerade durchgeführte.

```
cd ~/src
wget http://ftp.gnu.org/gnu/m4/m4-1.4.tar.gz
tar xvfz m4-1.4.tar.gz
cd m4-1.4
./configure --prefix=/home/<user>/local && make && make install
```

## 2.6 GNU Bison

Für diese Installation muss M4 korrekt installiert sein und die *PATH*-Variable richtig gesetzt sein.

```
cd ~/src
wget http://ftp.gnu.org/gnu/bison/bison-2.1.tar.gz
tar xvfz bison-2.1.tar.gz
cd bison-2.1
./configure --prefix=/home/<user>/local && make && make install
```

## 3 Installation von NQC

Zunächst wechseln wir wieder ins *src*-Verzeichnis und laden die Quellen für NQC herunter.

```
wget http://bricxcc.sourceforge.net/nqc/release/nqc-3.1.r4.tgz
tar xvfz nqc-3.1.r4.tgz
```

Anschließend öffnen wir das Makefile mit einem Editor, z.B.

```
kate nqc-3.1.r4/Makefile
```

und ändern die Zeile

```
PREFIX?=/usr/local
```

auf

```
PREFIX?=/home/<user>/local
```

damit NQC ins richtige Verzeichnis installiert wird, und entfernen das Kommentarzeichen (#) am Anfang der Zeile

```
# USBOBJ = rcxlib/RCX_USBTowerPipe_linux.o
```

um die Unterstützung für den USB-Tower zu aktivieren.

Jetzt öffnen wir die Datei *nqc-3.1-r4/rcxlib/RCX\_USBTowerPipe\_linux.cpp* und ändern Zeile 57 von

```
#define DEFAULT_TOWER_NAME "/dev/usb/lego0"
```

in

```
#define DEFAULT_TOWER_NAME "/dev/usb/legousbtower0"
```

Dieser Schritt ist wichtig damit NQC das richtige Device anspricht.

Anschließend laden wir noch eine alte Version des USB-Treibers für den Mindstorms-Tower herunter. Zwar ist der Treiber mittlerweile im Kernel integriert, jedoch lässt sich diese Version von NQC nicht ohne den Treiber kompilieren.

Den Treiber fanden wir unter [7] und [8], haben ihn jedoch noch einmal gespiegelt um ihn mit *wget* herunterladen zu können und damit diese Anleitung so einfach wie möglich zu halten.

```
mkdir LegoUSB
cd LegoUSB
wget http://www.rzuser.uni-heidelberg.de/~pbodewig/infoprak/legousbtower.c
wget http://www.rzuser.uni-heidelberg.de/~pbodewig/infoprak/legousbtower.h
cd ..
mv LegoUSB/ nqc-3.1-r4/rcxlib/
```

Jetzt können wir ins NQC-Verzeichnis wechseln und kompilieren

```
cd nqc-3.1-r4/
make && make install
```

Damit ist NQC installiert. Wir setzen noch 2 Umgebungsvariablen um NQC nicht jedes mal das Device mitteilen zu müssen

```
export RCX_PORT="usb"
export NQC_OPTIONS="-Trcx2"
```

und speichern diese auch in der *.bashrc*

```
echo "export RCX_PORT=usb" >> ~/.bashrc
echo "export NQC_OPTIONS="-Trcx2"" >> ~/.bashrc
```

Jetzt sollte die Kommunikation mit dem RCX funktionieren.

```
cweiler@max:~> nqc -near
No firmware installed on RCX
cweiler@max:~>
```

Im nächsten Abschnitt werden wir die Firmware auf dem RCX installieren.

## 4 Installation der Firmware

Die Firmware für den RCX ist im LEGO Mindstorms SDK2 [9] enthalten. Wir haben die Firmwaredatei noch einmal gespiegelt.

```
cd ~/src
mkdir firmware
cd firmware
wget http://www.rzuser.uni-heidelberg.de/~pbodewig/infoprak/firm0328.lgo
nqc -firmware firm0328.lgo
```

Wenn beim letzten Befehl keine Fehlermeldung erscheint, ist die Firmware korrekt installiert und NQC-Programme können mit

```
nqc -d programm.nqc
```

auf den RCX geladen werden.

## Literatur

- [1] *NQC - Not Quite C*,  
<http://bricxcc.sourceforge.net/nqc/>.
- [2] *legOS/BrickOS*,  
<http://brickos.sourceforge.net/>.
- [3] *leJOS - Java for the RCX*,  
<http://lejos.sourceforge.net/>.
- [4] *flex - The Fast Lexical Analyzer*,  
<http://flex.sourceforge.net/>.
- [5] *GNU Bison*,  
<http://www.gnu.org/software/bison/>.
- [6] *GNU M4*,  
<http://www.gnu.org/software/m4/>.
- [7] Koders.com: *legousbtower.h*,  
<http://www.koders.com/c/fid59ABC11B7FC3580A148149BAC1C97CEE3098320D.aspx>.
- [8] Koders.com: *legousbtower.c*,  
<http://www.koders.com/c/fid689CD5450EC72BC7F355FA024CF70A1A181C531E.aspx>.
- [9] *LEGO Mindstorms RCX 2 Beta SDK*,  
<http://mindstorms.lego.com/sdk2beta>.