# Lego Mindstorm with Linux Mini−HOWTO


## Luis Villa

luge@users.sourceforge.net



**Revision History**

| | |
|---|---|
| Revision 1.1 | October 29th, 2000 |



The Lego Group's Mindstorm Robotics Invention System (RIS) is probably the best reasonably cheap robotics kit available. However, the standard software is (unsurprisingly) MS Windows dependent. Don't despair− there are several options that allow Linux users to use their Mindstorms from within Linux. This Mini−HOWTO is intended to serve as a very brief introduction to the options available, and as a gathering point for more information.

# Table of Contents

# Table of Contents

# 1. Introduction

If you've seen those cool Lego Mindstorms kits, but despaired at the big "requires Windows" stickers on the side, don't despair: there is hope for you yet. Not only is there software that allows you to program your RCX from GNU/Linux, odds are you can do it in your favorite languages: C, Perl, Java and Forth are all available for your use. This document is not intended as documentation for all of these: rather, I hope to provide highlights and contact information for each of the various Linux options, so that this document can serve as a starting point for Linux users who are considering purchasing Mindstorms kits, or for Mindstorms owners who are considering switching to Linux and wonder whether or not they can continue to use their most expensive toys :)

It is extremely important to note that while I try to keep a pretty good finger on the pulse of the online Mindstorms community, I'm not omniscient: it's quite possible that this list is incomplete. If you know of other Mindstorms options that run on Linux, please drop me a note at luge@users.sourceforge.netso that I can add to the document and share the options with others who may be considering buying a Mindstorms to use with their GNU/Linux computers.

---

## 1.1. Acknowledgements

I'd like to thank the authors of the programs listed below, both for writing them in the first place for all of us to use and also for giving the earliest version of this document a once−over.

Additionally, I'd like to thank Dave Baum, who asked me to join with him in writing the book "Extreme Mindstorms". Writing that let me spend a lot of time with the Mindstorms that I wouldn't have had otherwise, including the time that led to this HOWTO. If you are interested in exploring "power programming" for the RCX, I'd strongly suggest this book. There are some other good books out there (notably the O'Reilly book) but this one is extremely up to date, and more importantly, just about every line of code in it (as far as I know) will work with Linux. I know this is a pretty shameless plug, but I'm pretty proud of it :) You can buy the book here.

Also, Steve Baker and Matthew Miller, and many others on Lugnet, provided many helpful suggestions to several early versions of this text. Their thoughts are present in several places throughout the text, and can be assumed to be responsible for all the good stuff. :)

Finally, thought not direcly related, Michael Littman, formerly of Duke University and now of AT&T actually paid me to play with LEGO last summer. Without the opportunity that he gave me, I would not be as involved or as knowledgeable with the online Mindstorms community as I now am.

---

## 1.2. Disclaimer

The LEGO Company (http://www.lego.com/) is very, very protective of their trademarks. This document is not in any way authorized by or associated with The LEGO Company, nor do I as the author claim to have any relationship with The LEGO Company. To see more about their policy on legal usage of their trademark, check out http://www.lego.com/info/fair.asp, particularly the part (about halfway down) titled "How LEGO(r) Enthusiasts May Refer to LEGO Products on The Internet." I have attempted to abide by those guidelines faithfully; like everything else in this document, though, if you spot a violation of the guidelines, please let me know by writing me.

## 1.3. Copyright

# 2. The Mindstorms Architecture

## 2.1. The Basic Hardware

In case you don't know, the Lego Mindstorms Kit is a robotics kit from The Lego Group that retails for about 200 US dollars. For that, you get a lot of Lego pieces, a large brick containing a CPU, an LCD, and some connectors (known as the RCX), a couple of motors, and some light and touch sensors that allow you to interact with the outside world.

The current release of the RCX kit is version 1.5, which will be replaced in spring 2001 by RCX 2.0. For the time being, any time I say "RCX" in this document, I mean "RCX 1.x." The differences between 1.0 and 1.5 are minimal, but the changes between 1.5 and 2.0 may be substantial− the exact extent of the changes is not yet known.

If you want to learn about the hardware in more detail (think: excruciating detail, from folks who have literally disassembled their RCX's in order to see what makes them tick) there are two important web sites to visit: Russell Nelson's Lego Mindstorms Internals and Kekoa Proudfoot's RCX Internals. Without these two sites, it is unlikely that much of the software below would exist.

## 2.2. Standard RCX Programming

The key to understanding the various Linux options is to first understand how the Mindstorms kit operates normally with MS Windows. In a nutshell, Lego provides a MS Windows software tool that lets you (or, more likely, a 12−14 year old) graphically assemble programs for the Mindstorm, using a building−block metaphor to create code. Once the program has been "assembled" in this way, the software compiles your program into a byte−code. This byte−code is then downloaded to the robot, where the firmware of the RCX processes the byte−code and controls the machine based on the instructions in the byte−code. Besides parsing byte−code, the RCX firmware has many OS−like functions: it controls the hardware, threading, and in particular, controls the IR port that is used to communicate with the robot. It also has the ability to accept specific commands (as opposed to complete programs) from the IR port or a special remote control, and move the robot based on those commands.

The standard firmware is currently in version 1.0 (even if you buy it with RIS version 1.5). It has pretty serious limitations− for example, since each variable is stored in a register and not in RAM, there can be only 32 variables. However, you can still do some pretty cool stuff with it, and version 2.0 of the firmware (which is available now for beta from LEGO) will significantly reduce these restrictions as well as be reverse compatible with 1.x hardware.

## 2.3. Where the Linux Tools Fit In

The different Linux Mindstorms programs function as replacments for different portions of the software chain discussed in the previous section. Some completely replace the default firmware with their own OS−like system or language interpreter. Others generate byte−code that matches the standard Lego byte−code, and use the standard firmware to interpret the byte−code once it has been generated. Finally, some merely generate the remote control codes that allow you to control the robot from the host PC, without giving you the option to run anything on the robot itself. These also use the standard Lego firmware.

## 2.4. Hardware Requirements for the Linux Host

Since most of these tools are command line based, hardware requirements are minimal− basically any Linux system should run them.

The one exception is the serial port, which must be present and may not be on some newer "legacy free" machines. All communication to the RCX is done via the IR tower, which is connected to the machine via a serial port. As a result, if you have no serial port connection, you will be unable to use the RCX unless you can buy an adapter. Furthermore, under certain circumstances, there may be problems with IRQs or serial port conflicts. This is particularly likely if your modem uses /dev/ttyS0. There are three fixes to this: first, attempt to use your second serial port for the IR tower. In most cases, this should work. If that doesn't help, just don't use your modem and your RCX at the same time. If that is unacceptable, then look in your kernel compile options (under "extended dumb serial driver options") for "support for sharing serial interrupts." Make sure it is on, and recompile.

Because Macs don't have standard serial ports, LinuxPPC users may have to get an adapter and make some modifications in order to use these tools. Dave Baum, NQC author and Mac user, has written instructions on how to do this.

## 2.5. The CyberMaster and Scout

Besides the RIS, Lego makes two other robotics systems− the CyberMaster (available only in Europe) and the Scout. Unfortunately, I believe that only one of these tools (NQC, discussed in Section 5) will work with these other tools. Generally speaking, if you are considering buying a Scout, unless you are severely financially restrained, go ahead and buy an RCX− the additional investment is worth it.

## 2.6. Mindstorms Vision Command

The Vision Command kit is a new addition to the Mindstorms line that uses a USB camera to do some cool stuff. Unfortunately, because USB camera support (in particular, USB Quickcam support) is still shaky under Linux, the product is not yet supported under Linux. If you want to look into hacking it yourself, you may want to look at this page which has a driver for a similar Quickcam.

## 2.7. Important Note about the MS Windows CD

Because many of the programs discussed below use the official Lego firmware, you may need your MS Windows CD. You won't need to ever boot MS Windows− your uptime is safe :) However, (if you run NQC or RCX.pm, among others), you may need to mount the CD to get the firmware when your batteries die. If you want to minimize this, find the file `firm0309.lgo` on the CD and copy it to a safe place on your Linux partition.

# 3. LegOS

## 3.1. Homepage

http://legOS.sourceforge.net

---

## 3.2. Author

Markus L. Noga

---

## 3.3. Type

Firmware replacement.

---

## 3.4. Language

C, C++.

---

## 3.5. Platforms

Developed on x86 GNU/Linux and tested on PPC Linux. It has also been ported to Cygwin and DJGPP on MS Windows. Solaris and Irix ports have been attempted but not all tools work.

---

## 3.6. Description

LegOS is a pre−emptive multitasking POSIXish OS for the RIS. Programs are written in standard C or C++, compiled on the PC using gcc (built as a cross−compiler), and then downloaded to the RCX where they are executed. Basically, anything you can write in C or C++ (and 32K of RAM, of course ;) you can write in legOS. Interesting features include random(), floating point emulation, threading with POSIX semaphores, and the ability to store multiple programs. It also includes functionality for sending and receiving data from Linux and MS Windows PCs. This power (legOS is almost definitely the most powerful of the RCX alternative software systems) comes at a slight cost: because it uses gcc, legOS is probably the most complex system to set up of the various Linux alternatives, and requires the largest download of tools.

---

# 4. Lego::RCX.pm

## 4.1. Homepage

http://members.home.com/quillan/lego/rcx.pm.html

## 4.2. Author

John C. Quillan

## 4.3. Type

Remote control library.

## 4.4. Language

Perl.

## 4.5. Platforms

GNU/Linux, MS Windows, and Solaris.

## 4.6. Description

Lego::RCX.pm is basically a perl library for remote control of the RCX via the IR tower. It takes over the IR tower and sends commands that the standard firmware can interpret and act upon. If you already have perl installed (and who doesn't?) this is a very quick and simple way to control your robot. No installation is necessary: merely copy the files into the correct library directories, and add "use RCX.pm" to the top of your perl script. I'm not sure if this has been done yet, but it would make a ridiculously easy way to interface a robot with a CGI script.

# 5. Not Quite C (NQC)

## 5.1. Homepage

http://www.enteract.com/~dbaum/nqc/index.html

NQC Linux page at mattdm.org− host for NQC rpms

## 5.2. Author

Dave Baum

## 5.3. Type

Native byte−code compiler.

## 5.4. Language

A C−like language, called (of course) Not Quite C. Should be pretty easy to learn for anyone with even minimal coding experience.

## 5.5. Platforms

GNU/Linux, MS Windows, and Macintosh.

## 5.6. Description

NQC is a byte−code compiler that takes programs written in a C−like syntax and compiles it (on the PC) into byte−code that can be understood by the standard Lego firmware. This approach has strengths and drawbacks: for example, the standard firmware can handle only 32 variables and so NQC is similarly limited. However, you can do a surprising amount within these limitations. Setup is pretty simple and the project as a whole is very well documented. This is also probably the most popular alternative programming system, so there are a lot of people willing and ready to help out if you start using NQC.

As already mentioned, NQC is the only option (right now) that supports the Cybermaster and Scout products under Linux. Furthermore, a beta version of it works with version 2.0 of the firmware, making it the first alternative programming system to support the added functionality of the new firmware.

# 6. pbForth

## 6.1. Homepage

http://www.hempeldesigngroup.com/lego/pbFORTH/

## 6.2. Author

Ralph Hempel

## 6.3. Type

Firmware replacement.

## 6.4. Language

Forth, a common script−like language usually used for embedded systems.

## 6.5. Platforms

GNU/Linux, MS Windows.

## 6.6. Description

pbForth is basically a complete Forth interpreter which replaces the standard firmware. Once it is there, you download Forth scripts to the robot, and the interpreter then interprets and runs the scripts. There are no limitations on the number of variables, and there are a number of "libraries" that are provided for functionality like interactive debugging. This is about the ultimate in simplicity for tool setup: all you have to do is download a binary, and then write code and download it. No other tools, compilers, interpreters reside on the PC. That said, there is a cross−platform TCL GUI available that simplifies downloading of scripts and interaction with the PC.

# 7. TinyVM and leJOS

## 7.1. Homepages

http://tinyvm.sourceforge.net

http://sourceforge.net/projects/leJOS/

---

## 7.2. Author

Jose Solorzano

---

## 7.3. Type

Firmware replacement.

---

## 7.4. Language

Java.

---

## 7.5. Platforms

GNU/Linux, Win32

---

## 7.6. Description

As the name implies, TinyVM is indeed a small Java Virtual Machine that is downloaded to the RCX to replace the standard firmware. Java programs are then written and byte−compiled on the PC and downloaded to the RCX. A TinyVM program can use some standard Java libraries, and a library for control of sensors, motors, and such on the Mindstorm is provided. TinyVM requires an already functional java compiler.

leJOS is a similar project (in fact, it is a code fork) by the same author. It is much higher footprint (currently about 5K additional size) but also includes substantial additional functionality, including support for floating point and String constants. Other planned features include garbage collection and multiple program loading.

---

# 8. Remote Java APIs

## 8.1. Homepages

RCX Java API

RCXPort Java Interface

## 8.2. Type

Remote Control Libraries

## 8.3. Language

Java.

## 8.4. Platforms

GNU/Linux(?)

## 8.5. Description

Both of these libraries are (theoretically) cross–platform Java libraries that send signals that can be used to remotely control an RCX that is using the original firmware. Neither have been tested with Linux, and neither are even in active development, but they may be useful sources for someone, which is why I've included them here.

# 9. TCL RCX

## 9.1. Homepage

http://www.autobahn.org/~peterp/rcx/

http://www.demailly.com/tcl/rcx/

## 9.2. Authors

Laurent Demailly and Peter Pletcher

## 9.3. Type

Native byte−code compiler and remote control library.

## 9.4. Language

TCL.

## 9.5. Platforms

GNU/Linux, Win32

## 9.6. Description

The TCL RCX compiler has double−edged functionality: it can either compile a TCL script into RCX byte−code, or it can remotely control the robot via either a script or an interactive TCL shell. If TCL is your language of choice, this is the optimal solution. There are two versions available: Laurent's (at demailly.com) appears to be the original of the two. However, neither appear to have been updated since 1998.

# 10. Pylnp

## 10.1. Homepage

http://www.hare.demon.co.uk/lego/pylnp.html

---

## 10.2. Author

Les Smithson

---

## 10.3. Type

Remote control library.

---

## 10.4. Language

Python.

---

## 10.5. Platforms

GNU/Linux.

---

## 10.6. Description

Pylnp is a unique remote control library, not only because of it's use of Python, but also because instead of interfacing with the standard firmware, it interfaces with legOS. Because of this, it provides a great deal of customizability that the other remote control libraries may not have, since it can be used to invoke custom functions on the legOS side of the robot. The downside of this power, of course, is that you must learn legOS and write C programs to access this superior functionality.

---

# 11. Other Linux Tools

These programs aren't strictly Mindstorms related, but they may be of interest to Linux/Lego fans.

## 11.1. LeoCAD

LeoCAD is (as the name implies) a CAD program for constructing and renderging Lego models. There is a usable GTK port, which can be found at http://leocad.gerf.org/linux.htm. The author implies that it doesn't have full functionality, so this might make a fun project for someone.

## 11.2. POVRAY

Sets of Lego pieces have been created for use with the popular POVRAY 3−D rendering program. You can find the pieces at http://www.kawo1.rwth−aachen.de/~witte/projekte/lego/lego.html and POVRAY at http://www.povray.org.

# 12. Other sources of information

Now that you've read this, there are a number of other places on the web to further explore the Mindstorms. Most of these are not Linux specific, but most of them that have discussion forums probably have at least a few users who use Linux. Hopefully, I've stimulated you enough to explore some more. Enjoy!

## 12.1. LUGNET

The LEGO User Group NETwork site, http://www.lugnet.com/, has long been the center of the online Lego universe. It has great newsgroups with lots of people. In particular, the robotics forum has an avid group of very knowledgeable readers and posters. In addition, several of the options above have their own active newsgroups in the robotics/rcx/ hierarchy.

## 12.2. The Official Mindstorms Site

The official Mindstorms site at http://www.legomindstorms.com/ has lots of neat ideas, and even accepts NQC source for public distribution. However, don't expect gobs of info on Linux.

## 12.3. The Hardware Sites

I've already mentioned these, but it can't hurt. These are the most comprehensive and detailed looks at the RCX hardware that are available to folks who aren't LEGO engineers. The "original" site is http://www.crynwr.com/lego–robotics/ and the second is http://graphics.stanford.edu/~kekoa/rcx

## 12.4. Historic Lego Bots and Construction Guide

As you may know, the RIS was partially inspired by work done at MIT. The creator of the famous "Lego class" at MIT has a home page that includes not only links to the class (some seriously cool stuff in there!) but also to his Robot Builder's Guide, which has lots of great suggestions about actually building robots to execute all the great programs you've just learned how to write.