Using the LEGO Mindstorms RCX in 2017

John Holbrook | lego@johnholbrook.us | http://johnholbrook.us

May 2017

Contents

1 Introduction

Despite being the oldest of the three Mindstorms programmable bricks, the RCX remains my favorite. I love its simplicity, small size, and its relative reliance on traditional LEGO® studs, rather than Technic beams. However, using an RCX is in some ways harder and more complicated than when it was first released in 1998. This article will attempt to summarize some of the equipment, software, and techniques necessary to use an RCX in 2017.

2 Why use an RCX?

There are several reasons you might want to use an RCX instead of an NXT or EV3:

1. *You already have one.* Maybe you're a teacher, and your school never bought newer Mindstorms sets. Maybe you found your old RCX equipment in the attic, and want to play with it again. Maybe you're like me, and you never stopped playing with it in the first place.

2. *They're cheap.* If you have a limited budget to spend on Mindstorms parts, or you want to buy a kit for a child, but you aren't sure if they'll use it enough to justify the cost of a more-expensive NXT or EV3 set, the RCX could be a good choice.

3. *They're simple.* For small, uncomplicated, or permenant projects, the low cost and relative simplicity of the RCX make it a good choice.

4. *You have a thing for retro technology.* Admit it: There's something appealing about the 7-segment LCD display, the infrared communication, and the 8-bit SOC.

3 Hardware Guide

3.1 RCX Brick

The RCX brick comes in 3 hardware revisions:

- 1.0 - The original. The only revision to include a power input jack, so that in addition to 6 AA batteries, it can also be powered by a 9-12V **AC** power supply. (Yes, you read that right: It's an AC input).
- 1.5 (also called 1.1) - Included with the Robotics Invention System (RIS) 1.5 set. Identical to the 1.0 brick except that the power input jack has been removed. Officially, the 1.5 brick is also called an 1.0 brick, but with a serial number greater than approximately 300000 [1].
- 2.0 - Identical to the 1.5 brick except that the IR interface operates at twice the frequency[2] (76 kHz instead of 38 kHz). This has the effect of slightly reduced range, but enables the 2.0 brick to talk to Spybotics products [3].

Aside from those differences, all hardware revisions are functionally identical, and all firmware versions are compatible with all hardware revisions. I tend to favor the 1.0 brick due to the inclusion of the power input, and in any case the 1.0 seems to be the most commonly-available brick online.

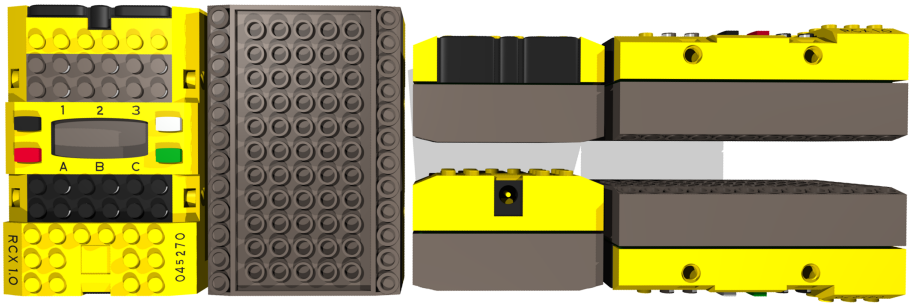

Figure 1: RCX 1.0 brick, from all sides. The 1.5 brick is identical except that there's no power jack at the bottom. The 2.0 brick is identical except that there's no power jack at the bottom and it says "RCX 2.0" instead of "RCX 1.0".

3.2 Infrared Tower

The infrared tower is used to send programs from a computer to the RCX, and data from the RCX back to the computer for logging. There are two models

[1] https://bricks.stackexchange.com/questions/1105
[2] https://bricks.stackexchange.com/questions/8049
[3] More info about Spybotics products at https://en.wikipedia.org/wiki/Lego_Spybotics

of infrared tower. One connects via a DE-9 [4] serial port and requires a 9-volt battery, while the other connects over USB, and can get enough power from that so as to not require a battery. The USB model is more convenient to use (due to the lack of additional battery), and seems to be more readily available online.

Figure 2: Two IR towers: USB on the left; Serial on the right. Photo: Brickipedia

3.2.1 USB Infrared Tower on 64-bit Windows

For many years, the common wisdom was that the USB infrared tower was not supported on 64-bit versions of windows, the reason being that LEGO never wrote a 64-bit driver for the USB tower. Recently, however, a 64-bit compatible USB tower driver [5] has become available, courtesy of the people at LegoEngineering.com, a joint project of Tufts University's Center for Engineering Education and Outreach, LEGO Education, and various other educators. Note that this driver is only compatible with ROBOLAB software (more info on RCX languages below); thus, the USB IR tower can only be used with ROBOLAB on 64-bit versions of Windows. The serial IR tower can be used on 64-bit versions of windows in conjunction with a USB-to-DE-9 adapter.

3.3 Motors

RCX-compatible motors come in 2 shapes, which I'll call "rectangular" and "cubic". The cubic motors appear in more sets, and are in my opinion easier to work with.
Cubic motors come in 2 hardware revisions. However, unlike the RCX brick, there are no visual cues on the outside of the motors to tell the hardware revisions apart. The primary difference between the 2 revisions is that the older

[4] Often incorrectly called a "DB-9" port; more info at http://www.nullmodem.com/DB-9.htm

[5] http://www.legoengineering.com/rcx-usb-tower-support/

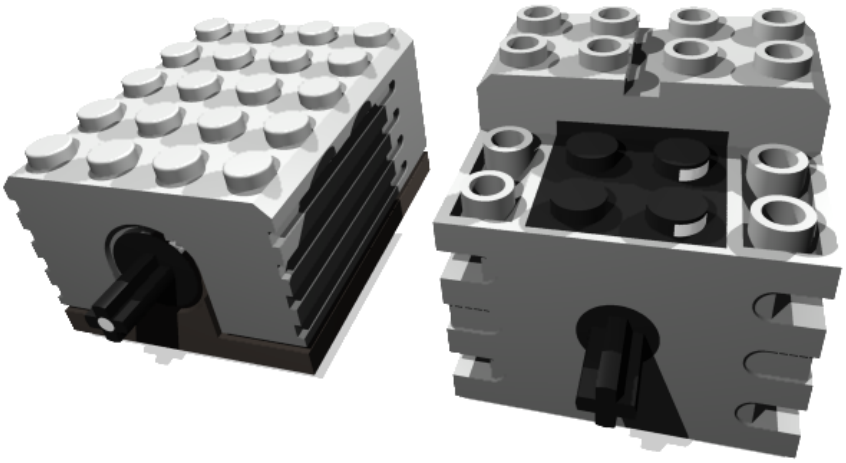

Figure 3: Motors. Left: rectangular motor (2838c01). Right: cubic motor (71427c01 /43362c01).

motors use internal gears made of metal, whereas in the newer motors the internal gears are made of plastic. As a result, the older hardware revision is more durable. It's also heavier, which is the best way to tell the difference between the two hardware revisions. You'll see the older revision described on Bricklink as "Older, heavier weight" and the newer revision described as "Newer, lighter weight".

3.4 Sensors

The first-party sensors available for the RCX are Light, Rotation, Touch, and Temperature. In addition, a number of third-party sensors are available, and LEGO® also produced a "DCP Sensor Adapter" which converted the RCX connector to a 7-Pin DIN socket compatible with LogIT Microsense scientific sensors made by DCP Microdevelopments [6].

[6] More info on using such sensors with the RCX at http://ccgi.dcpmicro.plus.com/DCPMICRO/lego.htm

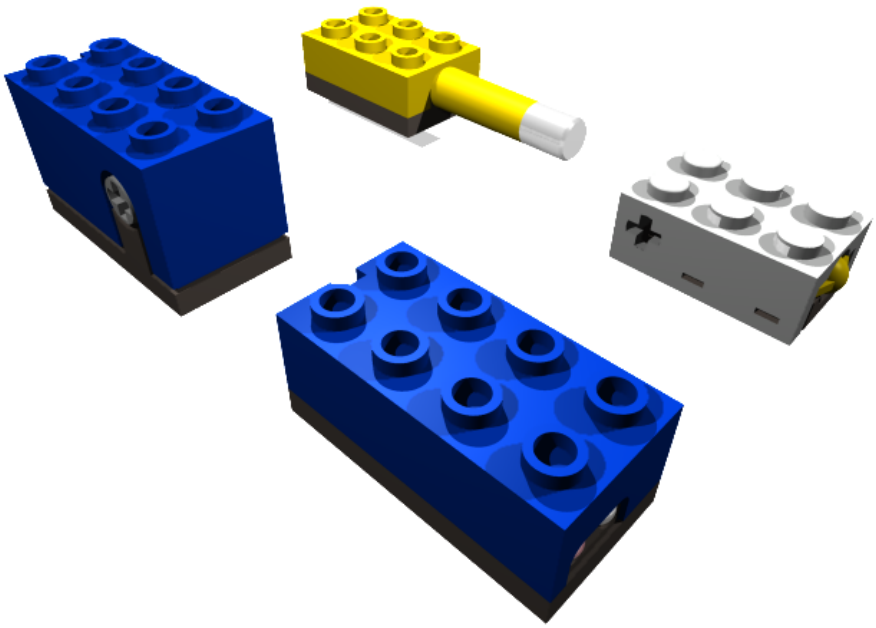

Figure 4: First-Party sensors. Clockwise from back left: Light sensor, rotation sensor, temperature sensor, touch sensor

3.5 Wires

The wires used for connecting motors and sensors to the RCX brick are terminated at either end with a 2x2x2/3 brick with electrical contacts on the top and bottom. Wires are available in lengths of 15, 17, 20, 21, 26, 36, 46, 69, 162, and 378 studs (1 stud = 8 mm = 0.315 in) [7]. Every first-party sensor except for the touch sensor comes with a short wire permenantly affixed to the sensor.

[7] Computed using the excellent LEGO unit converter available at http://studs.sariel.pl/

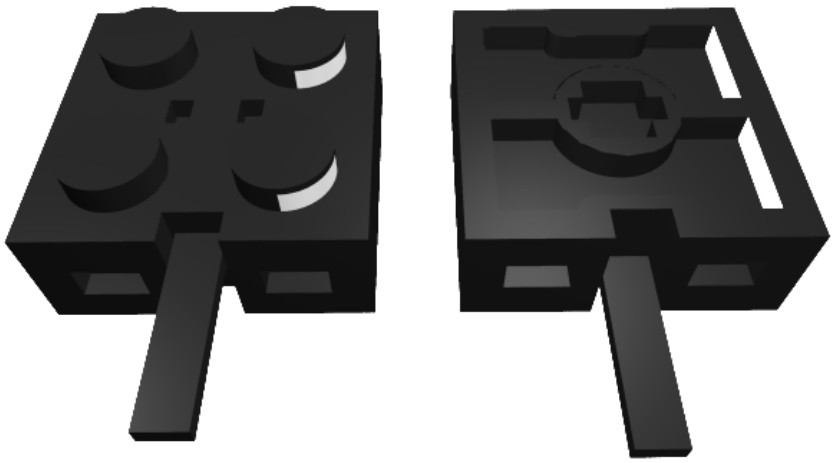

Figure 5: Wire connector for RCX motors and sensors, seen from the top (at left) and bottom (at right)

3.6 Remote

There also exists an infrared remote control (part number 9738), which can control the motor outputs of the RCX brick in real time without the need for programming.

Figure 6: RCX remote. Photo: Brickipedia

4 Buying an RCX

4.1 Buying a Set

If you're new to Mindstorms, you'll probably want to buy a set containing a bunch of parts. eBay (https://www.ebay.com/) and Bricklink (https://www.bricklink.com/) are both good places to buy RCX sets online.

The following sets contain RCX bricks:

Set #	Set Name	RCX Brick Version	Motors Included	Sensors Included	IR Tower Included
9790	ROBOLAB Team Challenge Set	1.0	2 old cubic	2 touch, 1 light	Serial
9719	Robotics Invention System 1.0	1.0	2 old cubic	2 touch, 1 light	Serial
9785	ROBO Technology Set (Serial)	2.0	2 old cubic	2 touch, 1 light	Serial
9786	ROBO Technology Set (USB)	2.0	2 old cubic	2 touch, 1 light	USB
9793	ROBOLAB Team Challenge Set (Serial)	2.0	2 rectangular	2 touch, 1 light	Serial
9794	ROBOLAB Team Challenge Set (USB)	2.0	2 rectangular	2 touch, 1 light	USB
3804	Robotics Invention System 2.0	2.0	2 old cubic	2 touch, 1 light	USB

Table 1: Sets containing RCX bricks.

The ideal set would be one that contains an RCX 1.0 brick, older cubic motors, and a USB IR tower. As you can see, such a set doesn't exist. So, you'll have to decide which of those features you care about most and which you could do without. Alternatively, you could purchase either set 9790 or 9719 and supplement that with a USB IR tower, which can be indiviually purchased for relatively little online.

4.2 Buying Individual Parts

If you already have a bunch of LEGO® elements, you can save some money by buying only the electronic components you want. In my experience, Bricklink (https://www.bricklink.com/) is the best place to buy individual Mindstorms components (or individual LEGO® components of any kind). Here are part numbers for I've mentioned:

Part Name	Part #
RCX 1.0 brick	884a
RCX 1.5 brick	884b
RCX 2.0 brick	884c
IR Tower (USB)	x431c01
IR Tower (Serial)	9713
Rectangular motor	2838c01
Cubic motor, old	71427c01
Cubic motor, new	43362c01
Light sensor	9758
Rotation sensor	9756
Temperature sensor	9755
Touch sensor	9757
DCP Sensor Adapter	9917
Wire	Various; most begin with 5306bc or 5306ac
Remote control	9738

Table 2: Part numbers for RCX-era electronic components.

5 Programming an RCX

5.1 Windows

In my experience, programming an RCX brick is most easily done from a computer running Windows. Several languages are available. On 64-bit versions of Windows, the USB IR tower can only be used by ROBOLAB; see section 3.2.1 above for more info.

ROBOLAB is the best of a couple of pieces of software officially distributed by LEGO for programming RCX bricks. It's based on National Instruments LabView, and unlike later LabView-based Mindstorms software (NXT-G and EV3-G), the visual resemblence is striking. The latest version of ROBOLAB is 2.9.4c, though version 2.5.4 is similar; the primary difference is that 2.9.4c can also program NXT bricks. Unfortunately, it can be pretty difficult to get ahold of a copy of ROBOLAB these days if you don't already have one; relatively few copies are available on Bricklink, and can go for upwards of $50. Note that after installing v2.9.4 from a CD, you'll want to also install the 2.9.4c patch, which can be found at LegoEngineering.com [8].

These days, I do most of my RCX programming in **ROBOTC**. As the name implies, ROBOTC is a C-based language designed for robotics; versions of ROBOTC are also available for NXT/EV3 programmable bricks, VEX IQ "robot brain", VEX Cortex/Pic microcontrollers, and Arduino. Support for the RCX is not included with current versions of ROBOTC; however, a legacy RCX version [9] is available, and this has the added advantage of being free,

[8] http://www.legoengineering.com/robolab-2-9-4c-patch/
[9] http://robotc.net/download/rcx/

whereas other versions of ROBOTC are not. ROBOTC also comes packaged with a pretty good Integrated Development Environment, and lots of help and resources are available to learn how to use it, because of the support for many popular educational robotics platforms.

Another C-based language for the RCX is **NQC** [10] ("Not Quite C"). One of the oldest third-party languages for the RCX, NQC has the advantage of working on all three popular desktop operating systems, and developer Dave Baum has written several books about the RCX system, which include information about NQC and its use. Another advantage of NQC is that it uses the stock firmware on the RCX (i.e., the same as used by ROBOLAB), unlike ROBOTC or LeJOS.

For people who prefer programming in Java, there's **LeJOS** [11]. Short for "LEGO Java Operating System", versions of LeJOS are also available for the newer NXT and EV3 bricks, and LeJOS has proven to be one of the most popualr languages for third-party RCX programming.

Several other languages exist for the RCX, with which I am less familiar or which are hard to find nowadays. These include:

- **pbForth** [12] - an implementation of Forth [13] for the RCX. Deveoped by Ralph Hempel, who later worked on pbLua for the NXT and ev3dev for the EV3.
- **brickOS** [14] - allows you to program your RCX in C or C++. Never used it before. Latest release dates back to 2004, and I'm not sure if anybody still actively uses it.

Bricxcc [15] is a popular Mindstorms IDE for Windows which supports programming the RCX using NQC, LeJOS, brickOS, and other languages.

5.2 Linux

As far as I'm aware, there are no GUI Linux applications for controlling the RCX. However, if you're comfortable with the command line (and if you're a linux user, you probably are), there are several choices. These include:

- **NQC** [16] - the most popular choice on Linux. A cross-platform C-like language for the RCX. Also includes various command-line tools, including the ability to send arbitraty files and opcodes to the RCX, which can be useful for programming in some other languages and for general hacking purposes.

[10] http://bricxcc.sourceforge.net/nqc/
[11] http://www.lejos.org/rcx.php
[12] http://www.brickwiki.info/wiki/PbFORTH
[13] More info about the Forth programming language at https://en.wikipedia.org/wiki/Forth_(programming_language)
[14] http://brickos.sourceforge.net/
[15] http://bricxcc.sourceforge.net/
[16] http://bricxcc.sourceforge.net/nqc/

- **LeJOS** [17] - Java virtual machine for the RCX. Versions also available for the NXT and EV3.
- **pbForth** [18] - an implementation of Forth for the RCX. Deveoped by Ralph Hempel, who later worked on pbLua for the NXT and ev3dev for the EV3. An advantage of pbForth is that it compiles programs on the RCX brick itself rather than on the computer you're using, so all that's really required is the ability to transfer plaintext files from to and from the RCX, which can easily be accomplished on Linux with NQC.

It's worth noting that drivers for the USB IR tower are included with essentially all versions of the Linux kernel released in the last 10 years or so.

5.3 Mac

If you have a modern Mac, the easiest thing to do is to program your RCX in a Windows virtual machine (using a piece of software like VirtualBox, VMWare Fusion, or Parallels). I've run various versions of ROBOLAB and ROBOTC in a virtual machine on various Macs, with no apparent loss of functionality.

Mac software for programming RCX bricks does exist, but is generally either classic Mac OS software or OS X software compiled for PowerPC, meaning that it won't run on modern Macs. However, if you have an older Mac running Mac OS 9 or an older version of OS X (10.5 or older for running PowerPC software, 10.4 or older for running classic Mac OS software), it is often still possible to program an RCX natively on a Mac.

LEGO released versions of **ROBOLAB** for the Mac, and versions 2.5.4 and 2.9.4 are available as native OS X software compiled for PowerPC. As far as I'm aware, the functionality of Mac versions of ROBOLAB is identical to the corresponding version on Windows, though I've never used v2.9.4 for mac. ROBOLAB for Mac is even harder to find online than ROBOLAB for Windows, although it appears that later versions of the software were distributed on a single disc containing both Windows and Mac versions of the software.

NQC[19], **LeJOS**[20], and **brickOS**[21] all have varying degrees of compatibility with OS X. Sometimes, compiled versions of the software are available for the Mac; other times, it's necessary to compile the software yourself. I'm not aware of any Mac IDEs for any of those languages; the NQC website links to a Mac NQC IDE called MacNQC but the link is dead at the time of writing.

Since **pbForth**[22] compiles and runs code on the RCX itself, all that's necessary to use it on a mac is the ability to send plaintext files from the mac to the RCX, which can be accomplished with NQC.

[17] http://www.lejos.org/rcx.php
[18] http://www.brickwiki.info/wiki/PbFORTH
[19] http://bricxcc.sourceforge.net/nqc/
[20] http://www.lejos.org/rcx.php
[21] http://brickos.sourceforge.net/
[22] http://www.brickwiki.info/wiki/PbFORTH

6 Resources

If you know of any other resources that should be added to this list, feel free to let me know by dropping me an email at contact@johnholbrook.us.

6.1 Hardware/Building Guides

- **Linda Hamilton's Lesson Plans** (http://www.marshall.edu/LEGO/plans.html) - many project ideas suitable for home or classroom use, most of which use the RCX or can be built with parts from RCX kits.

6.2 Software/Hacking

- **RCX Firmware** (https://pbrick.info/rcx-firmware/) - info about RCX firmware, as well as downloads of various First- and Third- party firmware images.
- **Linux NQC Programming** (https://minordiscoveries.wordpress.com/2014/01/20/using-nqc-on-a-raspberry-pi-to-program-a-lego-mindstorms-rcx-brick/) - a detailed write-up covering programming an RCX brick using NQC on Linux. Written for use with a Raspberry Pi, but applicable to Linux systems in general.
- **RCX Internals** (http://www.mralligator.com/rcx/) - information about the internal workings of the RCX and Serial IR tower, including hardware teardown, discussion of IR communications protocol, and complete opcode reference.
- **RCX 2.5 SDK** (http://www.philohome.com/sdk25/sdk25.htm) - archives of various tools and documentation for use with the RCX, Spybotics, and Scout bricks which are no longer posted on LEGO's website.

6.3 Sharing

- **LDraw** (http://ldraw.org/) - an open standard for LEGO CAD programs. Various editors are available for many platforms. Unlike LDD (the official LEGO CAD program, which incidentally can output to LDraw format), LDraw's parts library is much larger, and includes RCX components.
- **r/mindstorms** (http://mindstorms.reddit.com/) - Good place to ask questions about the RCX and other Mindstorms products, as well as to see things other people have done.
- **LEGO Stack Exchange** (https://bricks.stackexchange.com/) - Not RCX-specific (or even Mindstorms-specific), but still a good place to ask questions about the RCX.

6.4 General/Miscellaneous

- **pbrick.info** (https://pbrick.info/) - Lots of great info about various discontinued Mindstorms bricks, including the RCX.
- **Bricklink** (http://www.bricklink.com) - The best place to identify parts, find the contents of sets, and buy individual parts.
- **LEGO Engineering** (http://www.legoengineering.com/) - Resources and info about various Mindstorms platforms, including the RCX. In particular, info about ROBOLAB patches and 64-bit Windows USB tower drivers.

6.5 Books

I haven't read these books, and they were published near the time of release of the RCX, so some of the information may now be obsolete or irrelevant.

- **Definitive Guide to LEGO MINDSTORMS, Second Edition** by Dave Baum (ISBN 1590590635)
- **Building Robots With Lego Mindstorms: The Ultimate Tool for Mindstorms Maniacs** by Mario Ferrari, Guilo Ferrari, and Ralph Hempel (ISBN 1928994679)
- **The Unofficial Guide to LEGO MINDSTORMS Robots** by Jonathan Knudsen (ISBN 1565926927)

7 Frequently Asked Questions

These questions are mostly compiled from searches of Quora, the Mindstorms subreddit, and the LEGO Stack Exchange.

I changed the batteries in my RCX and now all my programs are missing! What happened?

The RCX's programs and firmware are stored in RAM. This made the brick cheaper to manufacture, because RAM was much cheaper than flash storage in the 1990s. However, a downside of RAM is that it needs to be continuously powered to retain the data it's storing. Thus, when power is cut to the RCX, such as when batteries are removed, its memory is erased, and it reverts to the default firmware and programs.

Happily, inside the RCX there is a small capacitor, which maintains power to the RAM for approximately 20 seconds after the batteries are removed. Thus, if you change the batteries within that time, the brick's memory will be preserved. Changing the RCX's batteries within 20 seconds sounds difficult, but is actually not too hard with some practice.

I read on [Website] that I can't use the USB IR tower with 64-bit Windows! Is that true?

Yes, unless you want to program in ROBOLAB. See section 3.2.1 for more information.

Can I send data to the RCX from a [TV remote/IR port on an old laptop/IR LED and an Arduino]?

The short answer is **no**. The IR towers and remote for the RCX use a proprietary communications protocol, which no one has reverse-engineered and which is incompatible with the protocols used by television remotes or IR transmitters on older laptops.

I no longer want my RCX bricks/motors/sensors! What should I do with them?

Probably the best thing to do is to sell them on eBay or Bricklink. If for some reason you don't want to do that, send me an email (contact@johnholbrook.us) – maybe I'll take it off your hands.

I want my RCX to play music!

This is a surprisingly common thing to want to do. The easiest way I know of is to find or compose a piece of music in the Ring Tone Text Transfer Language [23] (RTTTL), and then use ROBOTC for RCX, which includes a tool (only visible if using menu-level "superuser") for transferring RTTTL tunes to ROBOTC commands which cause the RCX to play the described tune.

Can I use RCX motors or sensors with the NXT or EV3? What about with an Arduino or Raspberry Pi?

A converter cable[24] is available which converts the connector used by RCX motors and sensors to the RJ12 connector used by the NXT and EV3. The NXT was designed to be backwards-compatible with RCX motors and sensors, and it's easy to do this in ROBOLAB 2.9 or NXT-G (with the download of some additional sensor blocks for NXT-G).

Unfortunately, due to the EV3 using a slightly different pinout for the sensor ports, the LEGO converter cable will not connect RCX sensors to the EV3. Others have produced homemade converter cables which connect the sensors to the correct pins on the EV3, but in any case RCX sensors are unsupported in the official EV3 software, so use of such converters would have to be in conjunction with a third-party firmware such as LeJOS or ev3dev [25].

[23] https://en.wikipedia.org/wiki/Ring_Tone_Transfer_Language
[24] Part x1676
[25] http://ev3dev.org

I'm having troube downloading firmware or otherwise transferring large programs or files to the RCX!

In my experience, the most common cause of communications failures is interference from bright room or outside light. When downloading firmware or large files to the RCX, I find it helpful to place a box or other opaque structure over the RCX and IR tower in order to prevent visible light interference.