Chapter 6

Programming the RCX

Solutions in this chapter:

- What Is the RCX?
- Using LEGO RCX Code
- Using the NQC Language
- Using Other Programming Languages
- Divide and Conquer: Keeping Your Code Organized
- Running Independent Tasks

Introduction

As we explained in the Introduction, this book is not about programming—there are already many good resources about programming languages and techniques, and about programming the RCX in particular. However, the nature of robotics (often called *mechatronics*) is such that it combines the disciplines of mechanics, electronics, and software, meaning you cannot discuss a robot's mechanics without getting into the software that controls the electronics that drives the machine. Similarly, you cannot write the program without having a general blueprint of the robot itself in your mind. This applies to the robots of this book as well. Even though we are going to talk mainly about building techniques, some projects have such a strong relationship between hardware and software that explaining the first while ignoring the latter will result in a relatively poor description. For these reasons, we cannot simply skip the topic, we need to lay the foundations that allow you to understand the few code examples contained in the book.

In the previous chapters, we mentioned the RCX many times, having assumed that you are familiar with the documentation included in the MINDSTORMS kit and know what the RCX is. The time has come to have a closer look at its features and discover how to get the most from it. We will describe its architecture and then give you a taste of the broad range of languages and programming environments available, from which you can choose your favorite. Our focus will be on two of them in particular: RCX Code, the graphic programming system supplied with the kit, and NQC, the most widespread independent language for the RCX.

The last sections of the chapter provide a complete code example, which is meant to help explain how to write well-organized code that is easy to understand and maintain, and is designed to familiarize you with the programming structures you'll find later in the book.

What Is the RCX?

The RCX is a computer. You are used to seeing computers that have a keyboard, a mouse, and a monitor—devices created to allow human users to interface with their computers—but the RCX hasn't got any of those features. Its only gates to the external world are a small display, three input ports, three output ports, four push-buttons, and an infrared (IR) serial communication interface. The RCX is actually more similar to industrial computers created to control machinery than it is to your normal desktop computer. So, how can you program it if it hasn't any

user interface? You write a program on your PC, then transfer it to the RCX with the help of the IR tower (a device designed to work as a link between the PC and the RCX), and, finally, the RCX executes it.

To understand how the RCX works, imagine a structure made of multiple layers. At the very bottom is the processor, an Hitachi H8300, which executes the machine code instructions. The processor cooperates with additional components that convert signals from the ports into digital data, using chips that provide memory for data and program storage. Just as with most computers, the memory of the RCX is made up of two types: read-only memory (ROM) and random access memory (RAM). The content of the ROM cannot be altered or cancelled in any way, since it is permanently written on the chips, while the data in the RAM can be replaced or modified. The RAM requires a continuous power supply in order to retain its content. When the supply breaks, everything gets erased.

Above the processor and circuit layer you find the ROM code. When you unpack your brand new RCX, there's already some code stored in its internal ROM that's aimed at providing some basic functionality to the RCX: input ports signal conversion, display and output ports control, and IR communication. If you are familiar with the architecture of a personal computer, you can compare this ROM code to the basic input/output system (BIOS), the low-level machine code which is in charge of booting the computer at startup and interfacing with the peripherals.

An RCX with just the ROM code is as useless as a personal computer with just the BIOS. On top of the ROM code layer the RCX runs the *firmware*, which, to continue with our comparison to computers, is its *operating system*. The term *firmware* denotes a kind of software the user normally doesn't alter or change in any way; it's part of the system and provides standard functionality, as operating systems do. In RCX, the firmware is not burned into the system like the ROM code, rather it is stored in the internal RAM, and you download it from your PC using the infrared interface. The LEGO firmware was copied to your PC during the installation of the MINDSTORMS CD-ROM, and transferred to your RCX by the setup process.

The firmware is not the final layer of the system: on top of it there's your own code and data. They will be stored in the same RAM where the firmware is, but from a logical standpoint they are considered to be placed at a higher level. As we explained earlier, you write your code on the PC, then send it to the RCX through the infrared interface. The MINDSTORMS software on the PC side, called *RCX Code*, translates your program (made of graphical code blocks) into a compact form called *bytecode*. The RCX receives this bytecode via the IR

interface and stores it in its RAM. When you press the **Run** button, the firmware starts *interpreting* the bytecode and converting its instructions into actions.

WARNING

Because the firmware is stored in RAM, it will vanish if your RCX remains without power for more than a few seconds, and you will have to reload it before using your RCX again. When you power off your RCX, the RAM remains supplied just to keep the firmware in existence, and this is the reason why the RCX will slowly drain the batteries even when switched off. If you plan not to use it for more than a few days, we suggest you remove the batteries to preserve them. Remember that when you need your RCX again, you will have to reload the firmware.

Let's summarize the process from the top to the bottom level:

- You write your program using RCX Code, the MINDSTORMS software on the PC side.
- RCX Code automatically translates your program into a compact format called *bytecode*.
- Using the IR link between the PC—via the IR tower—to the RCX, you transfer the bytecode version of your program to the RAM of the RCX.
- The firmware interprets your bytecode and converts it into machine code instructions, calling the ROM code routines to perform standard system operations.
- The RCX processor executes the machine code.

Most of these steps are hidden to the user, who simply prepares the program on the PC, downloads it to the RCX, presses the **Run** button, and watches the program execute.

A Small Family of Programmable Bricks

The RCX belongs to a small LEGO family of *programmable bricks*. The first to appear on the scene was the Cybermaster, a unit that incorporates two motors, three input ports, and one output port. It shares with the MINDSTORMS

devices the ability to be programmed from a PC, with which it communicates through the "tower," which in this case is based on radio frequency instead of infrared transmission. But the similarities end here, and the Cybermaster has more limitations than the RCX:

- Its three input ports work with passive sensors only.
- The firmware is in ROM instead of RAM. This means that it's not possible to upgrade it to a newer version.
- The RAM is much smaller than the one in the RCX and can host only very short programs.

The *Scout*, contained in the Robotics Discovery Set, is programmable from the PC with the same IR tower of the RCX (not included in the set), but features a larger display that allows some limited programming, or better said, it allows you to choose from among various predefined behaviors. It features two output ports, two input ports (passive sensors only), and one embedded light sensor. Like for the Cybermaster, the firmware is in ROM and cannot be upgraded or modified.

Using LEGO RCX Code

RCX Code is the graphical programming tool that LEGO supplies to program the RCX. If you have installed the MINDSTORMS CD-ROM, followed the lessons, and tried some projects, you're probably already familiar with it.

RCX Code has been targeted to kids and adults with no programming experience, and for this reason it is very easy to use. You write a program simply by dragging and connecting *code blocks* into a sequence of instructions, more or less like using actual LEGO bricks.

There are different kinds of code blocks that correspond to different functions: You can control motors, watch sensors, introduce delays, play sounds, and direct the flow of your code according to the state of sensors, timers, and counters. RCX Code also provides a simple way to organize your code into *subroutines*, groups of instructions that you can call from your main program as if they were a single code block.

When you think your code is ready to be tested, you download it to the RCX through the IR tower. The RCX has five *program slots* that can host five independent programs. When downloading the code, you choose which slot to download to, and with the **Prgm** push-button on your RCX, you select which program to execute.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.