# LISP and LEGO MindStorms®: Perfect Together?

**Frank Klassner**
Department of Computing Sciences
Villanova University
Villanova, PA 19085
Frank.Klassner@villanova.edu

### Abstract

In this paper I explain the ongoing efforts at Villanova University to mould LEGO MindStorms into a suitable platform for college students to investigate a broad range of computer science and artificial intelligence topics within the Common Lisp programming language. This is done within a library called RCXLisp. The library has two features that distinguish it from other third-party packages and languages designed by hobbyists for programming the MindStorms platform. The first is that they were developed to support both remote control and on-board programming of MindStorms robots. The second is that they work with an extended firmware that supports targeted communication between multiple LEGO robots and command-center desktops rather than the broadcast protocol supported by standard LEGO MindStorms firmware.

### Introduction

Since the middle 1990's, several manufacturers have released standardized, low-cost robot platforms. Among the more recent models are ActivMedia's Pioneer robot, MIT's HandyBoard and Cricket controller cards, and LEGO's MindStorms kit. In this paper I explain the ongoing efforts at Villanova University to mould LEGO MindStorms into a suitable platform for college students to investigate a broad range of computer science and artificial intelligence topics within the Common LISP programming language. I selected LEGO MindStorms for three reasons:

**Cost:** A single MindStorms kit, with 750 construction pieces, sensors, and programmable hardware, costs approximately $200 and thus is one quarter the cost of a HandyBoard-based robot kit and one tenth the cost of a ActivMedia-based robot kit – two of the more commonly used platforms in colleges.

**Flexibility:** The MindStorms platform supports a suite of reusable snap-together sensors, effectors, and building blocks that can serve as the basis for a wide variety of programming projects.

**Student Interest:** Many students have played with LEGO building blocks as children, and therefore they are intrigued with working on LEGO-based projects.

This paper discusses the basic MindStorms kit and its strengths and weaknesses with respect to the needs of AI course projects. Based on this evaluation, the paper then presents RCXLisp, a Common Lisp library that I and my students have developed to fill the holes in MindStorms' capabilities. The library has two features that distinguish it from other third-party packages and languages designed by hobbyists for programming the MindStorms platform. The first is that it supports both remote control of and on-board programming of MindStorms robots. The second is that it works with an extended firmware developed at Villanova to support targeted communication between multiple LEGO robots and command-center desktops rather than the broadcast protocol used by standard LEGO MindStorms firmware. The paper concludes with a discussion of future development plans for RCXLisp.

### The LEGO MindStorms Platform

The basic LEGO MindStorms kit costs approximately $200 and contains 750 building block pieces. The kit's centerpiece is the programmable control unit, called an RCX. The RCX has a 16MHz CPU (Hitachi H8/3292 microcontroller), 32KB RAM, and houses an infrared (IR) transmitter/receiver for sending and receiving data and commands from a desktop PC or from other RCXs. The IR transceiver has a range of 15-25 feet, depending on lighting conditions and reflectivity of walls. The RCX unit has 3 input ports, 3 output ports, and a 5-"digit" LED display. Power is supplied by 6 AA batteries. MindStorms supports four kinds of sensor available from LEGO: touch, light, angle (rotation), and temperature. Other sensors available from third-party sources such as HiTechnic (www.hitechnic.com) include magnetic-field compasses, ultrasonic distance sensors, and infrared distance sensors. The platform currently supports three kinds of effector: motors, LEDs, and infrared emitters.

LEGO's primary development environment for programming the RCX has an interface that models programming as a process of dragging puzzle pieces (representing program steps) together to produce a chain (complete program). This GUI environment supports basic programming constructs such as loops, subroutines (though not true procedure calls), and concurrency.

The RCX's "operating system" is a replaceable firmware (currently version 2.0) that models a primitive virtual machine. The RCX's firmware can be used in autonomous mode (the robot's behavior depends only on the program loaded into its memory) or in direct-control mode (a control program on a desktop computer broadcasts a series of instructions to the robot). When used in autonomous mode, the firmware supports 32 16-bit global integer variables that can be shared by up to 10 threads ("tasks" in MindStorms parlance). Each thread can allocate up to 16 private variables. There is no firmware support for dynamic memory allocation. Only integer arithmetic is supported by the standard firmware. Since there is no firmware support for call stacks, nested function calls are not possible. With regard to direct-control mode, it should be noted that the RCX firmware supports only a broadcast protocol through its IR transceiver.

### MindStorms' Impediments As an AI Platform

MindStorms' lack of support for Common Lisp programming limits its usefulness for projects in the area of Intelligent Systems (IS). IS is a knowledge area of the Computing Curriculum 2001 standard that covers Artificial Intelligence and related fields, including topics such as search, knowledge representation, agents, machine learning, planning and robotics. Lisp has long been important in IS and the lack of ability to control the RCX using Lisp is an unnecessary obstacle.

The RCX has a hard limit of 32KB of memory, which presents a serious restriction for instructors who want to give students experience in working with the large real-time schedulers and planners that are often used in robotics in the study of Intelligent System design. One possible solution to this problem is to offload most of the computation to a desktop computer and to have the desktop application do the planning and scheduling and then send commands as necessary to the RCX to control its actions. To do this, the desktop application needs an API for sending action commands (as opposed to programs) to the RCX, and the RCX needs software that will listen for action commands and execute them.

Another shortcoming of MindStorms with respect to Intelligent System study is that the platform does not support point-to-point wireless protocols over its built-in IR port—all IR communication is broadcast. In other words, if you have three RCXs, two of them cannot exchange messages without the third receiving them and possibly altering its behavior. Furthermore, if you have a desktop application controlling the several robots of a team, it needs a way to address them individually. The standard Lego firmware, however, is limited to using a broadcast protocol that does not support such targeted message-passing, and therefore, so is software based on the Lego firmware, such as NQC. Since firmware is responsible for parsing IR messages for the MindStorms robots, any solution to this problem minimally requires development of a modified firmware that checks for "source" and "target" fields in a message before passing the message along to application software on the RCX.

A final MindStorms issue occurs with respect to machine learning (ML). Machine learning algorithms often require a state vector in which the elements must be measured more or less simultaneously. If one assumes a ML algorithm running on a desktop, it might build up a state vector by sequentially querying the RCX for the motor settings, sensor settings and values, button state, clock values and so forth. We have identified as many as 60 queries that need to be made to construct a state vector. Each IR message and response takes about 7ms, so it can take quite a while to build up the state vector, possibly violating the assumption of simultaneity. One possible solution is to develop firmware with an opcode to transmit all 60 state values in one long burst of IR.

**RCXLisp**

As part of a larger project aimed at improving the MindStorms platform's usefulness in collegiate computer science curricula [NSF CCLI grant #008884], I have developed the RCXLisp programming libraries. This software package allows one to work with the RCX unit from the LEGO MindStorms® kit using Common Lisp. Specifically, RCXLisp lets one

1. remotely control the RCX from a Common Lisp program running on a desktop computer,
2. write RCXLisp programs to run on the RCX,
3. create and compile RCXLisp programs for downloading to RCXs "on the fly,"
4. simultaneously control more than one RCX from a single MindStorms infrared transceiver tower
5. set up a network of RCX units that can communicate with each other in a targeted manner (as opposed to the "broadcast manner" supported by LEGO's kit).

The term `RCXLisp' actually refers to two related languages. The first is "Remote RCXLisp," which is a collection of macros, variables, and functions for remotely controlling RCX units from a desktop. The second language is "RCXLisp" proper, which is a subset of Common Lisp that can be cross-compiled to run on RCX firmware for autonomous control of the unit.

The RCXLisp packages are compatible with LEGO's firmware. They are, however, designed primarily to work with extended firmware my students and I designed that supports wireless networking and most of the opcodes from LEGO's firmware 1.0. This extended firmware is called "Mnet firmware," and supports directed IR communication by adding source and target fields to the basic LEGO protocol and by allowing each RCX to set a 1-byte ID value for itself.

The Mnet firmware also supports a bytecode for requesting all of the state information of an RCX, which is useful for monitoring and statistical analysis as well as for supplying state vectors for neural network training and for other machine learning applications.

Although the first version of this firmware does not support dynamic memory allocation, future versions are expected to include this capability (along with garbage collection) in order to support a larger subset of Common Lisp's functionality.

The RCXLisp libraries are supported on the Allegro (Franz), MCL (Digitool), and Xanalys Common Lisp environments. Currently the libraries only support the older serial-port infrared transceivers that early versions of MindStorms were supplied with. My future improvement plans for the libraries include work on supporting the new USB IR transceivers that LEGO is now producing.

I have used RCXLisp in my undergraduate and graduate Introductory Artificial Intelligence courses for the past two years (three offerings of undergraduate AI, one of graduate AI). In my talk I will discuss not only the design of the language but also my students' experiences with the language. I will also provide a demonstration of RCXLisp's remote-control capabilities from within a Common Lisp environment.

**Acknowledgments**

LEGO MindStorms and RCX are trademarks of the LEGO Group, which does not sponsor, authorize, or endorse any of the third-party work cited in this article. The author of this article has no financial relationship with the LEGO Group.

I am grateful to my graduate student Andrew Chang for his work on extending the MindStorms' firmware, and to my AI students for their patience and enthusiasm in vetting the RCXLisp libraries.

This material is based upon work supported by the National Science Foundation under Grant No. 0088884. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.