Teaching C/C++ Programming with Lego Mindstorms

David T. Butterworth
University of Queensland, Australia
d.butterworth@uq.edu.au

***Abstract*—Computer programming is a skill required in many professions, not just computer science. Lego Mindstorms NXT can be incorporated into a programming course to add hands-on interactivity that will better engage a broader range of students. Chosing the most suitable programming language is difficult, and this paper summarizes some experiences in teaching students using RoboLab and NXT-G for Mindstorms NXT. The text-based language RobotC is recommended for beginner and intermediate level courses, and various code examples are provided to assist teachers in building lesson plans. It is suggested that advanced programming should be taught in C++, and an example of using the NXT++ library to control a robot arm is presented. Teaching all levels of programming, using robotics, is more enticing and stimulating for students, and teachers can justify the purchase of expensive robot hardware by employing it in multiple areas of the school curriculum.**

I. Introduction

Lego is a popular tool for science and technology education, in particular the Technic and Mindstorms products [1]. The Mindstorms series began with the RCX, and the current NXT and NXT 2.0 kits (Fig. 1) have become a defacto standard for teaching beginner-level robotics to students [2][3].

It may be suggested that Lego Mindstorms is a robotics construction kit and thus only useful for schools that provide a specific robotics curriculum. It is true that the majority of Mindstorms projects are based around some type of mechatronic vehicle or device, but it is now widely accepted that robotics activities are an effective way to teach important core skills [4][5]. Furthermore, the flexibility of the Lego system and range of available parts means that its application in school activities is limited only by the creativity of teachers and students [6].

At present, the majority of students first encounter Lego Mindstorms during the secondary 6-12 education phase, once the average student has reached an appropriate level of development. However students relative competency with technology is always increasing, so just like primary K-6 level students are now familiar with the internet and may own a high-powered mobile phone, the young students of tomorrow will arrive at school having assembled and programmed their own robots at home.

This technology creep places a tremendous strain on teachers to update their own knowledge and lesson plans, and a tight budget means that a class-set of equipment cannot be updated at the same pace as technology change. If the purchasing responsibility is passed on to parents, such as with compulsory laptop policies, this reinforces the belief that technology education is only for those that can afford it. One alternative is that groups of students can share expensive technology resources by visiting a capstone school or education center. Another alternative is to make equipment purchases more cost-effective by utilising them to a greater extent in the curriculum.

Traditionally, technology resources like Lego Mindstorms will be utilized in a specific part of the school curriculum, such as a weekly science class or semester-long robotics program. The benefit of high-quality kits like Lego Mindstorms is their versatility and expandability, as opposed to the broad range of inferior robotics construction kits that can now be found in toy stores. Therefore curricula can be designed using Lego Mindstorms to more effectively engage students across a broader range of subject areas than simply robotics, including general mathematics and science classes.

Technology skills like programming should no longer be thought of as part of the information technology curriculum. The ability to write a computer program to solve a problem, or to streamline a processing task, is a skill required by tomorrow's young professionals in all lines of work. For this

Fig. 1. Lego Mindstorms RCX (top) and NXT (bottom)

reason, many people struggle to learn computer programming on-the-job, after leaving school, rather than through formal education.

It can be difficult to make the subject of programming interesting to students, and typical complaints are that "it is boring" or they "are not good at it". Therefore, by combining a programming language like C++ with Lego Mindstorms, the subject can become more engaging to a broader range of students, and the school gains extra utilization of an expensive resource. A systematic literature review of the effectiveness of robots as tools for teaching progamming can be found in [7].

II. Teaching experiences

The University of Queensland, Australia, conducts an outreach program [8] for secondary school students, teaching basic robotics and ICT (Information & Communications Technology) skills, and also holiday workshops for teachers who need to conduct their own classes. Robotics is taught using Lego Mindstorms in two 4-hour sessions, covering introductory topics and basic concepts in Artificial Intelligence.

The purpose of these sessions is to excite students about computer science and engineering, through the use of robotics, and potentially continue their studies at a university-level. In a typical session of 20 students, from a mix of schools, there will be 1 or 2 students who have never used Lego before. If we are to consider Lego the defacto standard for engineering education at the secondary and university levels, then these students have already been left behind. The majority of students have used Lego before, so they possess the skills necessary to read construction plans, spatially orientate the pieces, and efficiently construct a Lego model. As many as 30% of students have already had some exposure to the Robocup robotics competition [9], and it is these students who have skills that will be advantageous at the university-level.

Sessions were originally conducted using the Lego Mindstorms RCX brick. These units have an excellent lifespan, however some wiring cables and general Lego pieces require occasional replacement. However, many students reported that they were already familiar with the newer NXT brick and its programming system, and so because of this technology creep, a new class set of Mindstorms NXT was purchased. At present, Lego Mindstorms is not widely used at the primary school level, so the RCX brick is a good choice for younger students. Secondary schools might consider donating old RCX sets to the junior students, or they may be used for multi-brick Lego projects like the Robocup Dance competition.

The graphical programming software Lego RoboLab v2.9 was used to program both the RCX, and the newer NXT brick, which is actually supplied with different software. RoboLab is a simple and efficient way for students to start learning programming and hard-working students will design a program that fills an entire screen, indicating they are ready for a more advanced assignment. The NXT-G software supplied with Mindstorms NXT does offer some additional programming functionality, however its more cluttered appearance makes it harder to use for students or teachers who are new to programming. Unfortunately the RoboLab software environment now looks quite dated, and many students are using NXT-G at their own school, so the session materials were recently re-written to reflect this.

More experienced students can also program the Mindstorms NXT using a text-based programming language, such as RobotC, which is covered in section IV. Teaching text-based programming is beyond the scope of an 8-hour introductory session, however many students are using RobotC in the robotics curriculum at their secondary school. The University of Queensland also hosts the Robocup Junior competition for the State of Queensland, and it is worth noting that the leading teams in the Robocup Rescue and Robocup Soccer categories are using RobotC to program their robots.

III. Chosing a programming language

It is possible to program the Mindstorms NXT using all the typical programming languages. Table I shows some of the language implementations that are available.

Some of these languages can be used to write a program that runs on the actual NXT brick, which requires firstly downloading a new operating system called firmware to the NXT. Other languages can be used to write a program for your desktop PC that remotely controls the NXT. The first option is most suited to building an autonomous mobile robot, however remote control is sometimes used where more computer processing power is required. A PC-based program is more suited to a robot arm or experiment that sits on the desk, and may communicate via USB cable or wireless Bluetooth methods.

Deciding which programming languages should be taught is a topic of much debate, and is best discussed in consultation with industry professionals and university staff. The basic skills of programming are transferrable and can be taught with any language, however it is the author's opinion that students will have an advantage if they finish secondary school with some exposure to Visual Basic, C and C++. Visual Basic can be used to automate tasks in the Microsoft Office application suite, so this will benefit students who go on to work in areas of business or management. The C language can be used to write programs for a massive range of platforms, including Lego Mindstorms for secondary students, or embedded Microprocessors for engineering students. Experience with programming in C++ will be mostly beneficial to potential computer science or engineering students, who will have an excellent head-start if they have experience with this language.

The use of Lego Mindstorms for teaching C programming is discussed in [10] and the C implementation RobotC has been compared with other languages in [11][12]. Delman et al. discuss using a variant of C++ for programming RCX and NXT robots in [13] and [14].

When chosing a language to program the Mindstorms NXT, it is important to realize that many of the language implementations are not complete. This means, for example, that if you program the NXT using a variant of the C language called NXC (Not eXactly C) [15], that the complete range of

TABLE I
LIST OF SOME TYPICAL PROGRAMMING LANGUAGES FOR THE MINDSTORMS NXT

Language	Implementation	On-board program	Remote control
C	RobotC	√	
C	NXC	√	
C/C++	nxtOSEK	√	
C++	NXT++		√
C++	Ander's C++ Library		√
C++	NXTface		√
C++	Lestat		√
C#	MSRDS		√
.NET	Mindsqualls		√
Python	NXT-Python		√
Java	leJOS	√	
Matlab	RWTH NXT Toolbox		√

C commands is not available. Therefore teachers should chose a language that has a large user support base, which is a good indicator of high-quality software. The author recommends a C implementation called RobotC [16], and using C++ with the NXT++ library [17].

IV. ROBOTC

RobotC is a subset of the C programming language, and includes a program editor and debugging environment. A trial 30-day license is available free of charge, or a classroom license will cost USD$199 per year. There are other C language variants available with no licensing costs, including NXC (Not eXactly C) mentioned in Section III, so teachers may be tempted to employ these. However, from experience, the author would strongly recommend the use of RobotC due to the large number of resources and online support that is available, and the high-level of functionality available.

The level of support for RobotC surpasses that currently available for other C language variants for the NXT and the license cost is well justified. As one example, it was decided to use the NXC language for an undergraduate Computer Science course with NXT Mindstorms. While the software is free, the lack of support means you can be on your own if problems arise, and some fundamental programming features are not permitted, including floating point arithmetic and static variables. It is possible to install a user-created firmware update to add support for floating point arithmetic, however it does not annex significant other functionality.

It can be intimidating for teachers to incorporate RobotC into the curriculum if they are not personally trained in computer programming. It is best to firstly cover the fundamental elements of a computer program, and the RobotC documentation can help in this regard. A lesson plan can be constructed around specific problem solving tasks, and teachers can use the online resources for assistance in developing their own solution to the problem. There are usually multiple programming approaches to solve a problem, but if students are struggling the teacher can then provide assistance with reference to their own previously solved solution.

This paper provides some examples of RobotC code for solving typical programming problems, to be considered advanced-level for secondary students. The source code is available online [18] and teachers may use it to build their own solutions.

- State Machine template, for sequential problem solving.
- Drive forward in a straight line, using feedback from wheel sensors.
- Gradual start for motors, using non-linear speed profile.
- Testing the NXT 2.0 RGB color sensor.
- RGB color sensor as a light sensor (line follower and pre-processor definitions example).

V. C++ AND NXT++

C++ is a powerful object-orientated programming language but, in fact, RobotC is more than sufficient for directly programming the Mindstorms NXT. However if the goal is to teach C++ programming, which typically involves students creating purely screen-based GUI applications, then the NXT could be incorporated to add a level of physical interaction to the task. Examples include writing a stock control program linked to a robotic arm, or agitation and testing of samples in a scientific experiment.

The C++ program is written on a desktop PC, using a development environment like Microsoft Visual C++. The Mindstorms NXT is connected to the PC via USB cable, or paired using wireless Bluetooth. The NXT is controlled using an API (Application Programming Interface) in the NXT++ library, a free open-source C++ interface that communicates with the NXT using the Lego Fantom DLL (Dynamically Linked Library).

The NXT++ library was written in 2007, but development work has ceased in recent years. The result is that some advanced Lego NXT functionality is not well documented or been fully incorporated yet. However NXT++ does contain enough functionality to add a more physical dimension to a C++ project, and being open-source means that advanced C++ programmers can make any additions they require.

This paper introduces an update to the NXT++ library, culminating in the release of a new version: NXT++ 0.7 [19]. Most useful is the support for multiple NXT Bricks via USB, in addition to Bluetooth. New features added in v0.7 are:

- List all available NXT Devices connected via USB and Bluetooth, including device name and MAC address.
- Open connection to specific NXT Device, using device name or MAC address. Supports multiple simultaneous connections via USB and/or Bluetooth.
- Retrieve device name of NXT brick.
- Set new NXT device name.
- Retrieving NXT firmware version (fixed).

```
// gradual_motor_start_non_linear_ramp.c
// RobotC for Mindstorms NXT
//
// Gradual start of motors using non-linear speed ramp
// [David Butterworth, 2012]
//
// This program demonstrates how to gradually start a
// pair of NXT motors, drive at a steady speed,
// then gradually slow down again.
//
// This is useful for vehicles or robots, where
// suddenly applying maximum speed can cause the
// robot to "jerk", which may affect wheel encoder
// readings.
//
// The motor output speed is calculated using a
// non-linear (bell shaped) speed profile.
//
//    outputSpeed = finalSpeed
//       *( 0.5 - 0.5*cos( (1/stepsForRamp)*PI*step ) );
//
// finalSpeed = The desired output speed (amplitude).
// stepsForRamp = How fast to increase the motor speed.
//      You should experiment with this value, because it
//      depends on the loop execution time of your program.
// step = The ramp iterator.
//
// The outputSpeed will increase from 0 to finalSpeed, in
// stepsForRamp number of steps.
// However we can't calculate (1/stepsForRamp)*PI on the
// fly in RobotC, the equation output is zero, so we
// calculate a rampConstant during program compilation
// using a pre-processor definition.

// stepsForRamp
#define STEPS_FOR_RAMP 70 // 70 steps is good for demo.
                          // 25 is quite fast.
                         // 1 is minimum value, so you can
                        // see how much jerk the robot has.

#define MOTOR_PORT_LEFT  motorC
#define MOTOR_PORT_RIGHT motorA

//--------------------------------------------------------//

// Calculate the rampConstant during program compilation
#define RAMP_CONSTANT (PI/STEPS_FOR_RAMP)
// e.g.  (1/200)*PI = 0.015708

// Declare vars
int finalSpeed = 0;
float outputSpeed = 0;
int step = 0;

// Main
task main()
{
    // Set maximum final speed for motors
    finalSpeed = 70;

    // Loop forever
    while (true)
    {
        // Initial state is state1
        static int state = 1;

        switch (state)
        {
            case 1: // State 1
                // Ramp-up Motor Speed
                nxtDisplayString(3, "Faster...");
```

Fig. 2. Excerpt from author's code examples for RobotC [18]

An example of a C++ programming project from a University Computer Science course is presented in Fig. 3: A 4-DOF (Degree of Freedom) robot arm using Mindstorms NXT. The design utilized 5 NXT motors, thereby requiring 2 NXT Bricks. The NXT Bricks were connected to a laptop PC via USB and controlled with a C++ program and the NXT++ Library. A USB web camera was used with the OpenCV Computer Vision Library to implement a color-based Blob-tracking algorithm. The result was an advanced C++ programming project that incorporated existing Mindstorms NXT resources and more fully engaged the students involved, while they learnt a broader range of skills.

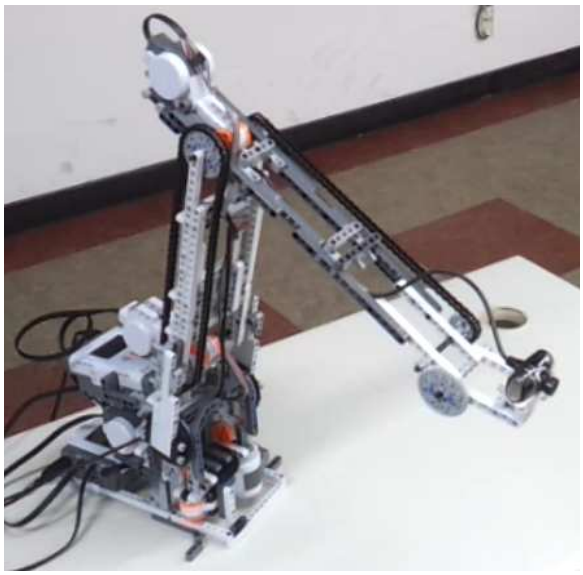

Fig. 3. 4-DOF Robot Arm using 2 Mindstorms NXT Bricks

VI. Conclusion

RobotC and Lego Mindstorms are highly recommended for teaching beginner-level programming. This implementation of the C language lacks many of the complexities that confuse students new to programming, while retaining enough of the C language syntax to be powerful enough for more advanced projects. It forms a good bridge towards teaching ANSI C to university-level students, and is more than sufficient for any Lego robotics challenge. Teachers should be wary of utilizing alternative, low-cost C language variants that lack functionality and support resources. Code examples in RobotC have been provided to assist teachers with lesson planning.

It is recommended that future engineering and science students should possess a competency in programming with C++. However, regardless of the language, the typical programming coursework can alienate students for whom programming is

a tool, rather than an interest in its own right. Using additional resources like Lego Mindstorms can make programming coursework more engaging to students, and the NXT++ library is presented as an effective way of connecting C++ with a hands-on problem. This paper presented a new version of NXT++, incorporating support for multiple NXT bricks.

Using OOP (Object-Oriented Programming) languages like C++ will not provide any additional benefit beyond using RobotC, when creating a Lego robotics project. Furthermore, an advanced understanding of robotics or OOP will require specific study in these areas. However, Lego Mindstorms is one tool that can be used to initially engage a broader range of students in studying advanced programming, for whom the semantics of C++ might otherwise be considered dull and boring.

Unfortunately for teachers, the fast pace of technology change makes it difficult to decide which software, technology or specific skills should be addressed in the classroom. It can be said that knowledge in any programming language is sufficient for students to bootstrap an understanding of other languages. However, if we can observe a trend and recognize the technology skills of most importance, we can provide students with a competitive edge for their future. If today's students are learning with tablet PCs and building robots, how will you engage the next generation of young learners?

REFERENCES

[1] (2012, Jun.) LEGO Mindstorms. [Online]. Available: http://mindstorms.lego.com

[2] F. Klassner and S. D. Anderson, "LEGO Mindstorms: not just for K-12 anymore," *IEEE Robot. Autom. Mag.*, vol. 10, no. 2, pp. 12–18, Jun. 2003.

[3] T. Karp, R. Gale, L. A. Lowe, V. Medina, and E. Beutlich, "Generation NXT: Building Young Engineers with LEGOs," *IEEE Trans. Educ.*, vol. 53, no. 1, pp. 80–87, Feb. 2010.

[4] A. B. Williams, "The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering," *IEEE Trans. Educ.*, vol. 46, no. 1, Feb. 2003.

[5] Z. S. Roth, "The role of robotics in freshmen engineering curricula," in *Proc. 5th Biannual World Automation Congress*, vol. 14, Jun. 2002, pp. 389–394.

[6] S. Galvan, D. Botturi, A. Castellani, and P. Fiorini, "Innovative robotics teaching using LEGO sets," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'06)*, May 2006, pp. 721–726.

[7] L. Major, T. Kyriacou, and O. P. Brereton, "Systematic literature review: Teaching novices programming using robots," in *Proc. 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE'11)*, Apr. 2011, pp. 21–30.

[8] (2012, Jun.) Robotics Workshops at Univ. of Qld. [Online]. Available: http://www.uq.edu.au/ict/robotics-workshops

[9] (2012, Jun.) Robocup Junior Australia. [Online]. Available: http://www.robocupjunior.org.au

[10] S. H. Kim and J. W. Jeon, "Educating C Language using LEGO Mindstorms Robotic Invention System 2.0," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'06)*, May 2006, pp. 715–720.

[11] D. Swan, "Programming Solutions for the LEGO Mindstorms NXT: Which approach is best for you?" *Robot Magazine*, 2007. [Online]. Available: http://find.botmag.com/100701

[12] S. L. Salcedo and A. M. O. Idrobo, "New tools and methodologies for programming languages learning using the scribbler robot and Alice," in *Proc. Frontiers in Education Conference (FIE'11)*, Oct. 2011, pp. 1–6.

[13] A. Delman, L. Goetz, Y. Langsam, and T. Raphan, "Development of a System for Teaching CC++ Using Robots and Open Source Software in a CS1 Course," in *Proc. International Conference on Frontiers in Education: Computer Science & Computer Engineering (FECS'09)*, Jul. 2009, pp. 141–146.

[14] A. Delman, A. Ishak, L. Goetz, M. Kunin, Y. Langsam, and T. Raphan, "Development of a system for teaching CS1 in CC++ with Lego NXT Robots," in *Proc. International Conference on Frontiers in Education: Computer Science & Computer Engineering (FECS'09)*, Jul. 2010, pp. 396–400.

[15] (2012, Jun.) NXC (Not eXactly C). [Online]. Available: http://bricxcc.sourceforge.net/nxc

[16] (2012, Jun.) RobotC. [Online]. Available: http://www.robotc.net

[17] (2012, Jun.) NXT++ (original web site). [Online]. Available: http://nxtpp.clustur.com/

[18] D. Butterworth. (2012, Jun.) RobotC example programs. [Online]. Available: http://david.butterworth.org.au/rie2012

[19] (2012, Jun.) NXT++ 0.7 (new version). [Online]. Available: http://david.butterworth.org.au/rie2012