

А.К.Карбозова¹, К.М.Маханов¹, А.М.Мустафина¹, М.А.Маукебаева²

¹Карагандинский государственный университет им. Е.А.Букетова;

²Жетысуский государственный университет им. И.Жансугурова

(E-mail: takanov@inbox.ru)

Применение NXT технологий в освоении основ робототехники в школе

В статье представлены результаты сравнения программных разработок, реализованных по технологии Lego Mindstorms NXT. Проведен сравнительный анализ скорости реализации программного кода, выполненного в двух разных вариантах. В первом случае программа написана в графической среде NXT-G, во втором — в среде RoboLab. Показано, что при использовании среды RoboLab робот выполняет маневры в два раза быстрее. Предложен пример, демонстрирующий отличия при реализации программы.

Ключевые слова: обучение программированию, робототехнические конструкторы, Lego Mindstorms NXT 2.0, Robolab, RobotC, NXT-G.

Введение

В 2015 г. в Казахстане впервые был проведен фестиваль робототехники «RoboLand – 2015», где команды из 6 школ города Караганды принимали участие в соревнованиях по пяти секциям. По правилам конкурса в соревнованиях могли участвовать только Lego конструкторы. В основном были представлены роботы, собранные на базе MINDSTORMS NXT 2.

При составлении программ для управления своими роботами школьники использовали наряду с графическими оболочками NXT-G и Robolab также и более гибкую среду RobotC [1, 2].

Результаты конкурса показали, что роботы команд, которые программировали в среде RobotC, оказались наиболее быстрыми и управляемыми: быстрее проходили дистанцию, совершали маневры и т.д. Менее «подвижными» оказались роботы, управление которых осуществлялось посредством программной среды Robolab. И наконец, следует отметить, что те роботы, которые были запрограммированы с использованием графической среды NXT-G, оказались самыми медлительными. Аналогичные результаты наблюдались и во время Международного конкурса «Робофинист – 2015», проходившего в сентябре 2015 г. в Санкт-Петербурге (Россия).

Таким образом, из сказанного выше следует, что для достижения наиболее высокой маневренности и скорости выполнения команд роботы-машинки, собранные на базе Lego-конструктора, предпочтительно программировать либо в среде Robolab, а еще лучше в RobotC. Однако использование последнего несколько затруднено. В первую очередь, нет достаточного количества литературных источников, плюс освоение языка занимает достаточно длительное время и требует знания большего количества команд, специального синтаксиса и лексики. По сравнению с ним освоение среды Robolab менее трудоемкое, к тому же имеется довольно большое количество учебников и пособий [3–5], в которых подробно рассматриваются те или иные комбинации при построении программного кода.

Нами была предпринята попытка на примере своих разработок продемонстрировать возможности построения программ для прохождения трассы в секции «Ралли» за максимально короткое время.

Цель настоящей работы — на примере построения простых программ показать различия в применении графической среды NXT-G и Robolab при решении задач, связанных с прохождением сложной трассы (ралли). В основу статьи был положен опыт работы со школьниками в кружке «Основы робототехники», который организован в 2013 г. во Дворце детей и юношества г. Караганды. Также использованы результаты и опыт ведения занятий курса «Робототехника» в ШОД «Дарын» и «Мурагер». Несмотря на сравнительно короткую историю (не более 1,5 года), учащиеся данных школ

сумели добиться высоких результатов и продемонстрировали свои способности как на республиканских, так и на международных конкурсах (в Санкт-Петербурге, 2015 г.).

Достижение поставленной цели осуществлялось путем решения следующих задач:

- разработка и сборка конструкции с учетом использования наименьшего количества деталей, что способствовало бы малому весу;
- освоение среды программирования NXT-G;
- освоение среды программирования Robolab;
- изготовление макета с траекторией трассы;
- набор программы, тестирование и оценка времени прохождения трассы.

При сборке необходимой конструкции робота-машинки, используемой в секции ралли, необходимо понимать, что мощность всех сервоприводов примерно одинакова (зависит от степени заряда аккумуляторов). Следовательно, машина, имеющая наименьший вес, имеет больше шансов для быстрого достижения финишной черты. Исходя из этих соображений была сконструирована простая конструкция робота-машинки, внешний вид которой представлен на рисунке 1.

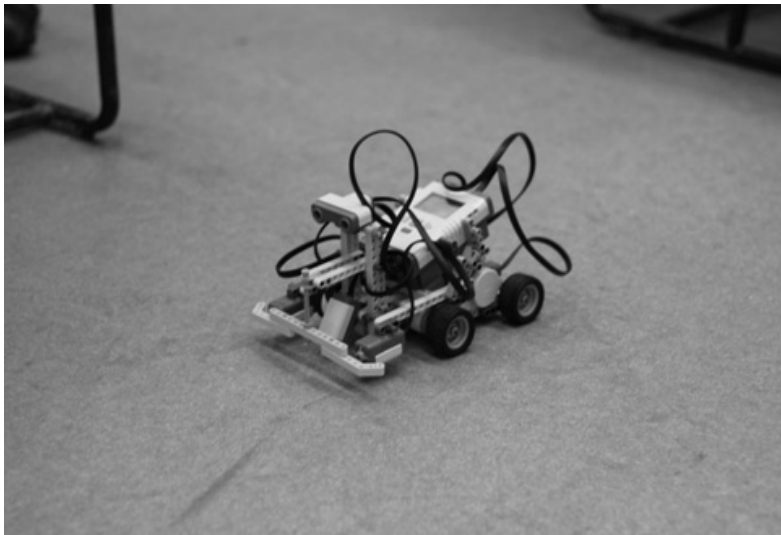


Рисунок 1. Конструкция простого робота-машинки

Следующий этап работы был посвящен программированию в графической среде NXT-G.

На данном этапе была поставлена цель — изучить способы организации повторяющихся действий в языке NXT-G; способы передачи данных между блоками.

Данная среда в прямом смысле графическая, так как при «написании» программ используются готовые блок-схемы, так называемые иконки.

Рассмотрим вкратце основы среды NXT-G. Графическая среда NXT-G используется для программирования модуля NXT Brick. Данный софт имеет интуитивно понятный интерфейс. Создание программ управления роботами напоминает создание блок-схем и осуществляется с помощью специальных блоков, размещаемых на LEGO-балках вдоль оси последовательности действий. Порядок выполнения программы определяется порядком следования блоков. Подключение новых компонентов выполняется путем их «перетаскивания» с палитры программирования на LEGO-балку, как показано на рисунке 2.

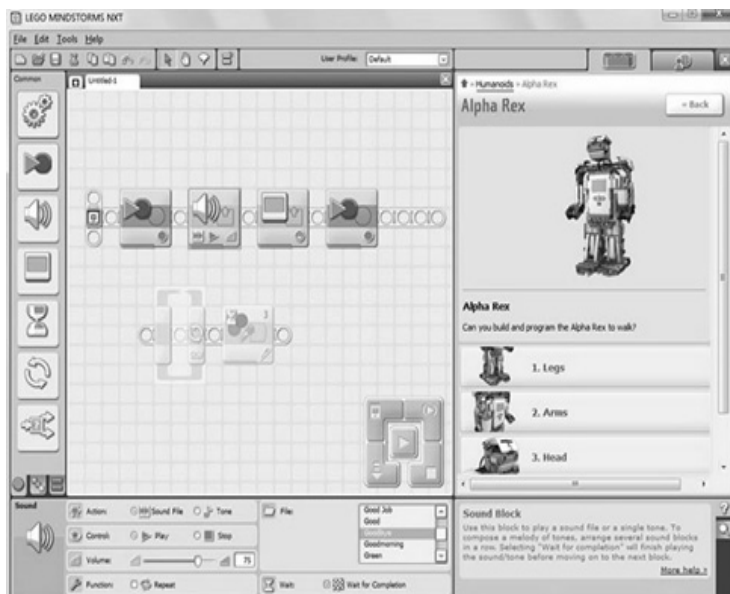


Рисунок 2. Внешний вид палитры программирования и Lego-балки

Каждый из иконок-блоков обладает набором уникальных характеристик, определяющих поведение робота. Например, блок «Движение», предназначенный для активации моторов, имеет параметры продолжительности хода и направления, а также мощность, передаваемую на двигатели. В NXT-G имеются блоки: для арифметических операций (сложения, вычитания, умножения и деления), для таймеров, для переменных, для сравнения числовых значений (меньше, равно, больше). Присутствуют блоки, отвечающие за звуковые эффекты роботов или за возможность бесконечного повторения установленных действий и за их завершение по определенным событиям. В программе возможно создание своих собственных компонентов, каждый из которых будет являться последовательностью стандартных блоков, объединенных вместе.

Кроме того, платформа NXT-G включает в себя: подробные инструкции по сборке огромного ряда моделей роботов LEGO; дополнительные инструменты для создания изображений, редактирования звуков и калибровки сенсоров; контроллер для передачи, запуска и остановки созданных программ и просмотра справочной информации о микрокомпьютере (ресурсах памяти, параметрах связи и т.д.). Имеется возможность дистанционного управления подключенного к компьютеру робота.

Среди основных достоинств среды графического программирования — наглядность и простота в использовании, позволяющие быстро освоить программное обеспечение, без особых знаний и усилий. Однако диапазон функциональных возможностей NXT-G весьма ограничен и требует для работы значительных ресурсов персонального компьютера. Последнее, в свою очередь, является существенной преградой при разработке сложных проектов.

Отметим также, что платформа NXT-G была разработана специально для LEGO специалистами компании National Instruments. Данная среда программирования создана на основе собственного продукта компании — программного обеспечения LabVIEW.

В целом опыт работы показал, что использование данной среды наиболее эффективно применительно к детям младшего возраста, т.е. к учащимся 5, 6^х классов.

Дальше мы рассмотрим итоги реализации следующей задачи — освоение среды программирования RoboLab. Необходимо сразу отметить, что на уроках по робототехнике наибольшую популярность среди среднего звена (7–9 классы) приобрела именно среда RoboLab. По сути, она является также графической, но предоставляет больше возможностей и маневренности при составлении программ [1].

RoboLab — это многофункциональная графическая среда программирования, которая также создана на основе LabView и ориентирована на самые разные возрасты — от дошкольников до студентов. RoboLab позволяет программировать несколько типов микроконтроллеров — Control Lab, RCX, NXT, а также проводить независимые расчеты на компьютере.

При запуске RoboLab предлагает три уровня работы: Администратор, Программист и Исследователь. Нажатие на кнопку Выход завершит работу RoboLab.

Нажатием на кнопку Помощь вызывается контекстная справка об объектах, расположенных под указателем мыши.

Кнопка Авторы открывает список разработчиков.

При запуске среды на экране появляются два основных окна, относящихся к одному проекту: Front Panel и Block Diagram. Первое (Передняя панель) для программирования не пригодится, но его можно использовать в режиме Исследователя. Второе, в котором уже расположены две пиктограммы светофоров (рабочее поле программы), предназначено для составления программы. Его можно развернуть на весь экран и приступить к работе. Два вспомогательных окна — Tools Palette и Functions Palette содержат все необходимое для составления программы. В случае закрытия их снова можно вывести на экран через пункт Windows верхнего меню.

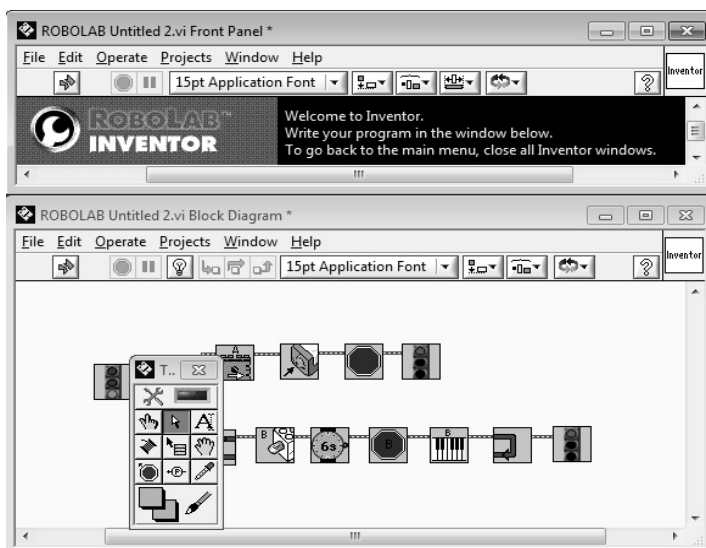


Рисунок 3. Пример программирования в среде RoboLab

Программа похожа на блок-схему, положенную на левый бок. Она читается слева направо, хотя блоки располагать можно как угодно. Блоки команд находятся в окне Functions Palette (Палитра команд). Они связываются между собой проводами, а также управляются инструментами, находящимися в меню Tools Palette (Палитра инструментов).

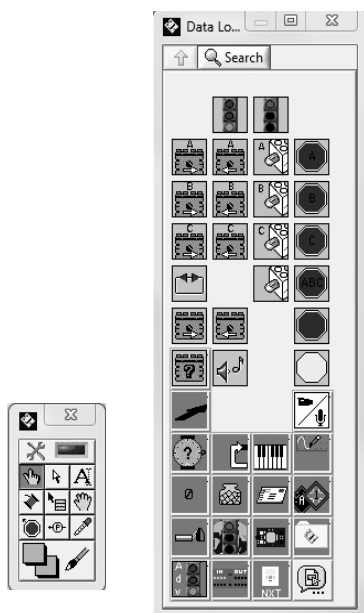


Рисунок 4. Палитра команд и палитра инструментов

Дальше мы самостоятельно приготовили каркас будущей трассы. Расчет траектории движения и уровень сложности основаны на реальных трассах, используемых в соревнованиях. Вид используемой нами трассы представлен на рисунке 5.

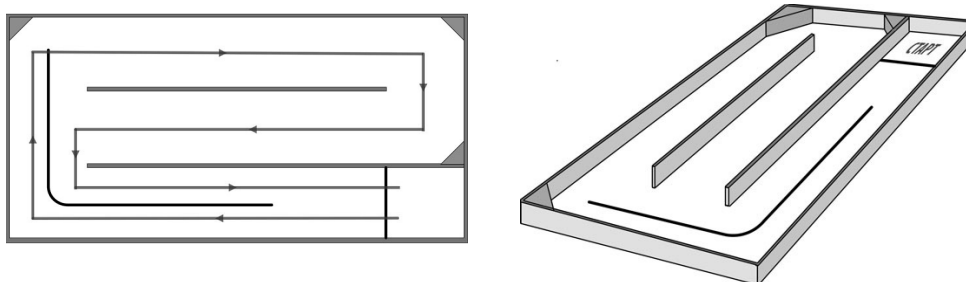


Рисунок 5. Вид трассы, которую должен проехать робот за минимальное время

Последняя задача по «написанию» программы, тестированию и отладке нами была осуществлена с участием учеников 7 класса.

Алгоритм разработанной программы работает по следующему порядку. При включении питания робот находится в режиме ожидания и ждет развития событий, запрашивая данные с датчиков, в частности, с ультразвукового датчика, предназначенного для определения расстояния до ближайшего препятствия. Реализуется следующий алгоритм: «если расстояние до препятствия менее 20 сантиметров, то робот «разворачивается» на 30° (либо вправо, либо влево). Если ультразвуковой датчик «сигнализирует» о наличии препятствия, то робот «разворачивается» в обратном направлении на 60° и делает 5 оборотов двигателя вперед». Цикл повторяется до тех пор, пока ультразвуковой датчик не покажет отсутствие препятствия. Следует отметить, что во время реальных соревнований участникам дается дополнительное время для изменения программы, и тогда участники могут подогнать код программы под конкретную трассу. В нашем случае мы за основу брали показания ультразвукового датчика. Предполагалось, что траектория неизвестна, т.е. робот должен преодолеть любую трассу.

Второй датчик, «датчик касания», срабатывает при наличии препятствий (расположенные очень низко либо очень узкие), которые не были обнаружены ультразвуковым датчиком. При активации данного датчика реализуется алгоритм программы, схожий с приведенным выше.

Таким образом, задача робота — за минимальное время преодолеть препятствие. Ниже на рисунке 6 представлен вид программы, разработанной в среде NXT-G (a) и RoboLab (b).

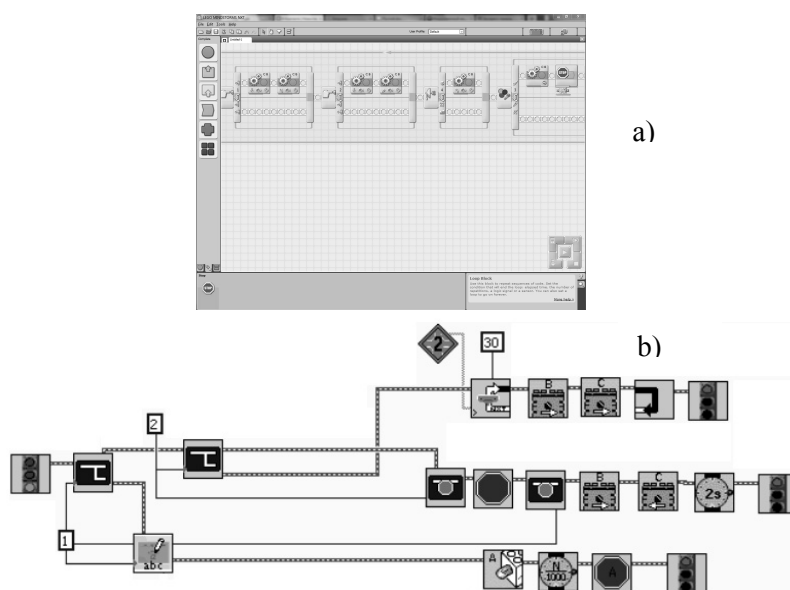


Рисунок 6. Коды программ для прохождения трассы с препятствиями, написанные в средах NXT-G (a) и RoboLab (b)

По результатам прохождения трассы нами было установлено, что в обоих случаях робот проходит трассу. Однако в случае использования программы, разработанной в среде NXT-G (рис. 6 а), он тратит больше времени и преодолевает трассу за 4 минуты, а в случае использования среды RoboLab (рис. 6 б) на преодоление той же трассы уходит около 2 минут.

Большое различие во времени прохождения трассы объясняется тем, что возможности оболочек для программирования сильно различаются. Особенно различие проявляется при преодолении угловых разворотов. В случае использования среды NXT-G программист может осуществить разворот следующим образом: он «останавливает» двигатель одного колеса и продолжает вращение второго (в зависимости от того, в какую сторону надо повернуть). В случае применения среды RoboLab оба двигателя не прекращают вращение, только вращаются в разные стороны, что способствует более быстрому завершению маневра. Это, конечно же, далеко не единственный пример имеющихся различий, но он дает возможность явно продемонстрировать отличия. Также управление при помощи RoboLab будет более точным, так как этот язык позволяет вести расчеты в формате с плавающей точкой. Такие расчеты особо актуальны при программировании робота на олимпиаде, конкурсе и т.д.

Выводы

В заключение еще раз проведем небольшое сравнение языков RoboLab и NXT-G. NXT-G поставляется вместе с конструктором Lego. RoboLab покупается отдельно. Тем не менее RoboLab представляет собой более мощное средство программирования, поскольку поддерживает некоторые команды языка СИ.

Для занятий в робототехническом кружке RoboLab слишком сложен, программы громоздкие и не полностью интуитивно понятные. NXT-G прост и понятен, программы легко читаются. Программирование простейших действий (движения) может укладываться в одну–две команды. Все параметры команды настраиваются в отдельном окне и хранятся внутри блока команды, не загромождая программный код. В RoboLab параметры в виде разноцветных модификаторов сетью опутывают код, что снижает его читаемость и понятность. Но плюс такого подхода в том, что несколько команд могут быть подключены к одному модификатору. Еще одна проблема — вывод данных на дисплей. В NXT-G все элементарно — для вывода информации выделена всего одна команда, которая может выводить текст, изображение или чертеж. Для вывода числовых данных предусмотрена команда приведения типа в RoboLab. Для работы с дисплеем представлена целая серия команд, что не совсем удобно. Более того, не ясно, как выводить числовые данные (команды приведения типов отсутствуют). Огромный минус RoboLab состоит в том, что в нем перемешаны команды для RCX и для NXT, причем не все старые команды можно использовать с новым контроллером. А иногда, напротив, происходит дублирование команд, что вынуждает нас изучать аналогичные конструкции. Хотя разработчики утверждают, что RoboLab поддерживает программирование на СИ, далеко не все возможности языка могут быть реализованы в текстовом виде. В основном тексте можно записать арифметические операции и функции.

Делая общий вывод, можно сказать, что каждый из языков программирования имеет свои достоинства и недостатки. RoboLab неплохо подходит для программирования роботов на соревнованиях, но более сложен в изучении, нежели NXT-G. Последний, в свою очередь, менее функционален в плане точных расчетов, чем RoboLab.

Список литературы

- 1 Филиппов С.А. Робототехника для детей и родителей / Под ред. А.Л. Фрадкова. — СПб.: Наука, Сер. «Шаги в кибернетику», 2010. — 320 ст.
- 2 Lego Mindstorms. Статья в Wiki. — [ЭР]. Режим доступа: http://en.wikipedia.org/wiki/Lego_Mindstorms.
- 3 LEGO–конструирование — [ЭР]. Режим доступа: <http://www.homeedu.ru/user/uatml/00000011/oneurok/docum/vved.htm>.
- 4 Конструкторы LEGO DACTA в курсе информационных технологий, введение в робототехнику. — М.: Ин-т новых технологий, 2001.
- 5 Разовый набор Lego Dacta (книга учителя) ИНТ, 2009780RM.
- 6 Ананьевский М.С., Болтунов Г.И., Зайцев Ю.Е., Матвеев А.С., Фрадков А.Л., Шиегин В.В. Санкт-Петербургские олимпиады по кибернетике / Под ред. А.Л. Фрадкова, М.С. Ананьевского. — СПб.: Наука, 2006.
- 7 История робототехники. С древности до наших дней. — [ЭР]. Режим доступа: <http://www.myrobot.ru/articles/hist.php>.

А.К.Карбозова, К.М.Маханов, А.М.Мустафина, М.А.Маукебаева

NXT-технологиясын мектепте робототехника негіздерін меңгеру үшін қолдану

Мақалада Lego Mindstorms NXT-технологиясы бойынша өңделген программалық өнімдерді салыстыру нәтижелері келтірілген. Екі әр түрлі тәсілмен құрастырылған программалардың орындалу жылдамдықтарына салыстырмалы талдау жүргізілді. Бірінші тәсіл бойынша программа графикалық NXT-G ортасында өңделді, екінші жағдайда — RoboLab ортасында. RoboLab ортасын қолданған кезде робот барлық тапсырмаларды екі есе жылдам орындайтыны көрсетілген. Программаның орындалу ерекшеліктерін айқын байқауға мүмкіндік беретін мысал келтірілген.

A.K.Karbozova, K.M.Makhanov, A.M.Mustafina, M.A.Maukebaeva

NXT-technology in application mastered the basics of robotics school

This paper presents the results of a comparison of software development technology implemented by Lego Mindstorms NXT. There has been conducted a comparative analysis of implementation of the code realization rate implemented in two different ways. In the first case the program is written in a graphical field NXT-G, the second in the field of Robolab. It is shown that by using a medium Robolab robot performs maneuvers in half the time. We propose an example that demonstrates the differences in the implementation of the program.

References

- 1 Filippov S.A. *Robotics for kids and parents*, edit. A.L. Fradkov, Saint Petersburg: Science Series «Steps to cybernetics», 2010, 320 Art.
- 2 *Lego Mindstorms*. An article in the Wiki, [ER]. Access mode: http://en.wikipedia.org/wiki/Lego_Mindstorms.
- 3 LEGO-construction, [ER]. Access mode: <http://www.homeedu.ru/user/uatml/00000011/oneurok/docum/vved.htm>.
- 4 *Designers LEGO DACTA to date information technologies, introduction to robotics*, Moscow, INT, 2001.
- 5 *One-time set of Lego Dacta* (teacher's book) INT, 2009780RM.
- 6 Ananievski M.S., Boltunov G.I., Zaitsev Yu.E., Matveyev A.S., Fradkov A.L., Shiegin V.V. Saint Petersburg Olympiad on cybernetics, edit. A.L.Fradkov, M.S.Ananievski, Saint Petersburg: Nauka, 2006.
- 7 *The history of robotics. From ancient times to the present day*, [ER]. Access mode: <http://www.myrobot.ru/articles/hist.php>.