

# QReal:Robots — an environment for teaching computer science and robotics in schools

Andrey Terekhov  
St. Petersburg State University  
ant@tercom.ru

Yurii Litvinov  
St. Petersburg State University  
yurii.litvinov@math.spbu.ru

Timofey Bryksin  
St. Petersburg State University  
timofey.bryksin@math.spbu.ru

## ABSTRACT

QReal:Robots tool is presented — a development environment for programming Lego Mindstorms NXT 2.0 and TRIK robotic constructors, that are used to teach computer science and robotics in schools. Differences between QReal:Robots and existing tools are discussed, TRIK robotic construction set is presented, QReal:Robots functionality for TRIK is described.

## Categories and Subject Descriptors

K.3.2 [Computers and education]: Computer science education

D.1.7 [Programming Techniques]: Computer and Information Science Education – Visual programming

## General Terms

Languages

## 1. ВВЕДЕНИЕ

Информатика – один из самых сложных школьных предметов. Связно это с тем, что детям приходится учиться создавать абстракции, принципиально незримые и неосознаваемые. Поэтому для упрощения понимания данного предмета в школах используется понятие исполнителя. Исполнитель – некая абстрактная сущность, способная исполнять команды и действовать в некоей среде. Наблюдая за поведением исполнителя, дети могут наблюдать, как работает их программа. Исполнитель был успешно использован в образовании американским педагогом Сеймуром Пейпертом в 1967 году [5], который применял механическую черепашку, управляемую внешними командами. Впоследствии механическая черепашка была заменена черепашкой на экране, вычерчивающей некоторые фигуры [9]. В таком виде идея Пейперта активно используется в школьном образовании до сих пор. В Советском Союзе со времен академика Ершова использовались похожие инструменты, например, Робик [1] и Рапира.

Однако, исполнитель, действующий на экране, все же остается недостаточно наглядным. Детям важно иметь возможность увидеть, как их исполнитель, управляемый программой, взаимодействует с окружающей средой, иметь возможность на него непосредственно воздействовать. Поэтому сейчас в школах активно внедряются робототехнические конструкторы [8]. Обычно такой конструктор позволяет собрать тележку с моторами и сенсорами, которая может управляться либо дистанционно, либо автономно программой, загруженной с компьютера. Примерами таких конструкторов могут служить Lego Mindstorms NXT [7] или робототехнический конструктор ТРИК [2, 12].

Программировать подобные устройства оказывается даже сложнее, чем черепашку Logo. Связано это с тем, что из конструктора можно собрать множество различных устройств с различным расположением моторов и сенсоров, выполняющих разные функции. Поэтому для того, чтобы конструкторы могли применяться детьми, которые программировать еще не умеют, для их программирования применяются специальные средства, существенно упрощающие этот процесс. Как правило, для этого используются визуальные языки и визуальные среды разработки. Для Lego Mindstorms NXT существуют такие среды, как Robolab [11], NXT-G, Microsoft Robotics Developer Studio [10]. Кроме них нашим коллективом была разработана среда QReal:Robots [3, 4], которая на данный момент уже внедрена в ряде школ и робототехнических кружков. В данной статье дается описание поддержки робототехнического конструктора ТРИК в данной среде.

## 2. СУЩЕСТВУЮЩИЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ РОБОТОВ

В данном разделе будут описаны существующие визуальные среды программирования роботов Lego Mindstorms NXT, их основная функциональность, достоинства и недостатки, из которых будут сформулированы требования к визуальным средствам поддержки конструктора ТРИК в QReal:Robots.

Первым и самым популярным на данный момент средством программирования роботов Lego Mindstorms NXT является среда Robolab. Robolab разработан на основе среды LabView [6], использует визуальный язык, описывающий элементарные команды роботу и управляющие конструкции. Robolab специально создавался для использования в школьном образовании, поэтому содержит в себе ряд возможностей, специфичных для этой цели. Интересным примером таких возможностей является наличие уровня использования программы. На начальных уровнях пользователю предлагается всего лишь заполнить места в шаблоне визуальной программы, состоящей из пяти блоков. При этом свободные места можно заполнять, используя лишь ограниченный набор блоков (порядка 20 штук). Таким способом можно описать лишь самые элементарные программы, которые заставляют робот двигаться и реагировать на одно-два события. Более продвинутые уровни дают возможность рисовать программы в свободной форме, помещая на диаграмму одновременно сколько угодно блоков, использовать арифметические выражения, переменные, подпрограммы, параллельно исполняемые части кода и так далее. Самые высокие уровни использования позволяют использовать существенно большее количество блоков, чем начальные. Всего Robolab имеет около 400 различных блоков. Robolab также позволяет по визуальной программе

сгенерировать программу для автономного исполнения и загрузить ее на робот. Также имеется возможность снимать показания с датчиков робота непосредственно на компьютер.

К недостаткам данной системы можно отнести устаревший пользовательский интерфейс (система в основном разрабатывалась в конце 1990-х и с тех пор существенно не обновлялась), отсутствие полного русского перевода и отсутствие возможности отладки созданной программы.

Следующая система, для конечных пользователей являющаяся даже более популярной, чем Robolab – это NXT-G, которая поставляется в стандартном комплекте конструктора NXT. Она также создана на основе среды LabView и также представляет программу в виде набора блоков, связанного потоком данных, однако предназначена для написания более простых программ. Среда довольно эргономична, но не позволяет, например, использовать математические выражения в текстовой форме. Если требуется записать какую-то формулу, нужно фактически составлять из блоков дерево разбора выражения этой формулы, что даже для простых и типичных робототехнических задач, как движение по линии, нетривиально, делает программу большой и практически нечитаемой. Поэтому среда NXT-G используется либо только на самых начальных этапах школьного образования, либо некоторыми школами не используется вовсе.

Среда Microsoft Robotics Developer Studio (MRDS) предназначена для программирования более сложных устройств, чем Lego Mindstorms NXT. В качестве типового примера используется так называемая стандартная модель робота, представляющая собой тележку с ноутбуком, сенсором Kinect, несколькими другими сенсорами и двумя силовыми моторами. Модель вычисления в данной среде – это набор взаимодействующих веб-сервисов, часть из которых может выполняться на компьютере, а часть на роботе, если компьютер и робот находятся в одной сети. Создание программы как и в средах, представленных выше, выполняется с помощью визуального языка, который визуализирует взаимосвязи между веб-сервисами. Таким образом, готовая программа состоит из набора независимых параллельно исполняемых компонент, общающихся друг с другом установленным пользователем способом. Данная среда имеет качественную систему симуляции, включающую в себя трехмерную модель робота с подробной эмуляцией физики.

В школьном образовании эта среда используется очень редко. Связано это, во-первых, с тем, что стандартная платформа, для которой прежде всего и создавалась эта среда, достаточно дорога. Один только ноутбук, расположенный на тележке, возможно, стоит дороже всего набора Lego Mindstorms NXT. Использование средств симуляции для целей образования оказывается недостаточным, поскольку любая даже самая точная модель – это все-таки модель и не воспроизводит ситуации, происходящие в реальном мире, а именно, обработку таких ситуаций и должны продемонстрировать школьные задачи по робототехнике. Поддержка Lego Mindstorms NXT в Microsoft Robotics Developer Studio заключается исключительно в дистанционном управлении роботом по каналу Bluetooth, что не позволяет создавать автономно исполняющиеся программы. Кроме того, вычислительная модель, используемая в MRDS,

ориентирована в основном на профессиональных программистов или студентов, занимающихся программной инженерией. Школьникам представлять себе взаимодействующие параллельно выполняющиеся веб-сервисы может быть довольно сложно. К тому же такая модель нетипична для программирования и не поможет школьникам в понимании ряда важных концепций императивного и структурного программирования.

Из приведенного выше следует, что уже существующие на рынке средства программирования роботов не могут в полной мере удовлетворить потребностей школьного образования. Даже наиболее подходящая для использования в школах среда Robolab имеет ряд недостатков (отраженных выше), и среди учителей имеется желание от нее отказаться. Поэтому существует потребность в разработке новой подобной среды, что и является основной мотивацией проекта QReal:Robots.

### 3. СРЕДА QREAL:ROBOTS И ПОДДЕРЖКА LEGO MINDSTORMS NXT 2.0

Среда QReal:Robots позволяет создавать программы для робототехнического конструктора Lego Mindstorms NXT 2.0 путём комбинации блоков, представляющих элементарные команды роботу и управляющие конструкции. В отличие от рассмотренных выше сред язык QReal:Robots ориентирован на поток исполнения, а не поток данных --- связи показывают передачу управления от одного блока к другому. Такой подход более адекватно представляет императивные программы, поэтому более полезен в образовании. Более подробно про язык программирования для Lego Mindstorms NXT см. в [3, 4], , пример программы на этом языке представлен на рис. 1.

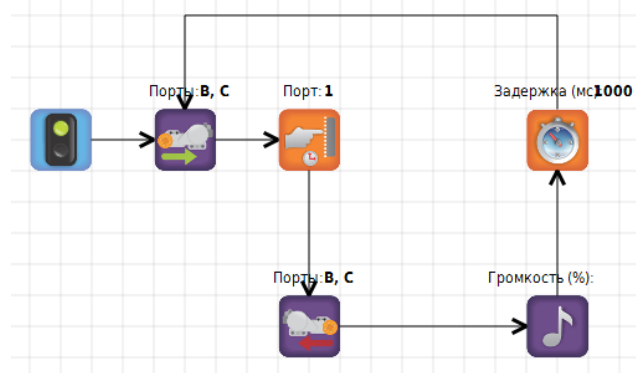


Рис. 1. Пример программы на языке QReal:Robots

Поддержка Lego Mindstorms NXT в QReal:Robots включает в себя возможность интерпретировать созданную программу на компьютере, управляя роботом прямыми командами по интерфейсу Bluetooth или USB. При этом текущий исполняемый блок подсвечивается на компьютере, давая возможность наблюдать за ходом выполнения программы. Имеется возможность создавать параллельно исполняемые фрагменты программ, в этом случае интерпретация на компьютере ведётся параллельно, несмотря на то, что на роботе команды всех фрагментов исполняются последовательно. Имеется возможность использовать в программах арифметико-логические выражения, которые помимо пользовательских переменных могут включать в себя

специальные переменные, содержащие текущие значения сенсоров (так называемые сенсорные переменные). При этом текущие значения всех переменных, включая сенсорные, отображаются в специальном окне рядом с диаграммой.

Кроме того, имеется возможность сгенерировать по диаграмме программу на языке Си, скомпилировать её и загрузить на робот по USB для автономного исполнения. После загрузки программа может быть запущена средствами операционной системы самого робота, связь с компьютером не требуется. QReal:Robots использует в качестве целевой платформы операционную систему nxtOSEK, на данный момент одну из самых быстрых операционных систем для контроллера NXT. Образ этой операционной системы и средства кросскомпиляции для неё входят в установочный пакет QReal:Robots, образ может быть загружен на робот прямо из среды, нажатием одной кнопки. При этом nxtOSEK совместим по формату исполняемых файлов и прямых команд со стандартной прошивкой, поэтому после загрузки образа существующие программы продолжают с роботом работать. Перед генерацией программа проверяется на корректность. Некоторые корректные с точки зрения интерпретатора программы не могут быть сгенерированы в структурный код на Си, в силу возможного наличия там фрагментов потока управления, не реализуемых без оператора goto. Этот оператор генератор не использует, поскольку порождаемый код должен быть понятен учащимся и представлять собой пример хорошей программы.

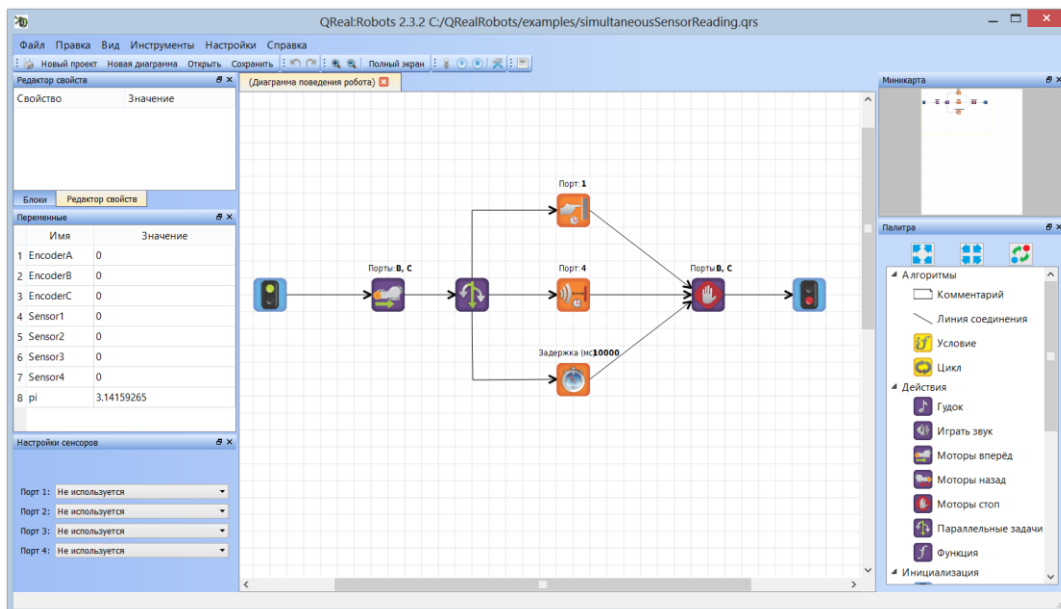
Созданную программу также можно исполнить без наличия настоящего робота, с помощью двухмерной модели. Двухмерная модель симулирует трехколесную тележку, используемую, например, в соревнованиях по робофутболу. Для модели робота возможно задать четыре разных сенсора, выбрав их тип и расположение относительно позиции тележки. Кроме самого робота происходит симуляция и окружающей его среды – возможно задание линий на полу, на которые реагируют сенсоры света и цвета, и стен, на которые реагируют сенсоры расстояния и касания (и сам робот,

останавливаясь при «врезании» в них).

При запуске программы на исполнение в режиме двухмерной модели управляющие команды передаются модели робота, и та начинает движение в соответствии с ними. Для более реалистичной симуляции возможно включить добавление шума, который будет вносить случайные возмущения в работу сенсоров и моторов, эмулируя проблемы, возникающие с устройствами реального мира. Конфигурацию робота и окружающего мира двухмерной модели можно сохранять и загружать по желанию пользователя.

В сравнении со средой Robolab QReal:Robots имеет следующие преимущества. Во-первых, это более современный пользовательский интерфейс, включающий в себя распознавание жестов мышью как инструмента быстрого создания элементов на диаграммах и связей между ними. Проведенные неоднократно юзабилити-тестирования с участием школьников и студентов показали достаточно высокое качество пользовательского интерфейса среды. Основным достоинством среды QReal:Robots относительно NXT-G является полноценная поддержка арифметических выражений, что позволяет решать гораздо более сложные задачи, создавая менее объемные диаграммы. В отличие от Robolab и NXT-G, QReal:Robots имеет средства отладки, позволяющие удобным образом контролировать процесс выполнения программы – визуальный интерпретатор программ и возможность использования двухмерной модели. В отличие от MRDS QReal:Robots позволяет создавать программы, доступные для автономного исполнения на роботе, и обладает существенно более простой моделью вычислений. Внешний вид пользовательского интерфейса среды представлен на рис. 2.

Также среда QReal:Robots является открытым и активно развивающимся проектом, что позволяет учителям высказывать свои замечания и, при желании и наличии соответствующих навыков, даже самим вносить необходимые изменения.



## 4. ПОДДЕРЖКА ПРОГРАММИРОВАНИЯ КОНСТРУКТОРА ТРИК В QREAL:ROBOTS

### 4.1 Робототехнический конструктор ТРИК

Основной мотивацией создания конструктора ТРИК стало отсутствие достаточно мощного и недорогого контроллера, пригодного для использования в образовательных целях. Контроллер конструктора Lego Mindstorms NXT 2.0 весьма ограничен в ресурсах – он имеет всего 256 Кб постоянной памяти и 64 Кб оперативной, что хоть и достаточно для большинства школьных задач, но сильно ограничивает свободу творчества. В новой версии конструктора от Lego под названием EV3 ситуация принципиальным образом не меняется — конструктор также позиционируется только для начальных стадий обучения. Контроллер ТРИК, несмотря на то, что создавался исходя из задач образования, может быть использован для решения широкого круга задач — например, «умный дом». Это возможно благодаря тому, что его ресурсы позволяют реализовать достаточно сложные алгоритмы на самом роботе, такие как распознавание видеобразов, звука, синтез речи и др. Это также расширяет его применимость в образовательной сфере, поскольку позволяет использовать эти алгоритмы в качестве базовых строительных блоков для пользовательских программ.

Контроллер имеет три отдельных вычислительных блока, включающих в себя процессор общего назначения ARM9, DSP для обработки видео, звука и т.д., и MSP430 для управления силовыми моторами. Контроллер обладает четырьмя выходами на силовые моторы, шестью выходами на серво-моторы и портами для датчиков. Это позволяет, используя один контроллер, создавать довольно сложные устройства, которые невозможно было бы собрать, используя только один блок конструктора Lego. Контроллер управляется операционной системой Linux, в образе ОС имеется набор таких распространенных библиотек, как Qt Embedded, к тому же при необходимости возможна установка дополнительных библиотек и утилит. Контроллер поддерживает связь с внешним миром по интерфейсам Wi-Fi и Bluetooth, имеет встроенный трехосный акселерометр и гироскоп.

### 4.2 Поддержка программирования конструктора ТРИК в QReal:Robots

Среда QReal:Robots изначально проектировалась так, чтобы иметь возможность быстро поддерживать новые конструкторы, поэтому, когда разработчики получили доступ к конструктору ТРИК, было принято решение поддержать его в среде. Этим предполагалось проверить лёгкость расширения системы и поддержки новых конструкторов, расширить возможности QReal:Robots, и предоставить средство визуального программирования для ТРИК.

Средства программирования конструктора ТРИК в QReal:Robots включают в себя, во-первых, среду времени выполнения, работающую на самом роботе, а во-вторых, средства отправки прямых команд и генератор программ для автономного исполнения в среде QReal:Robots. Для

пользователя QReal:Robots работа с конструктором ТРИК осуществляется точно так же, как и с конструктором Lego Mindstorms NXT 2.0, за исключением того, что необходимо указать, что диаграмма создается для конструктора ТРИК, и указать в настройках IP-адрес робота. После этого существующие диаграммы можно интерпретировать посылкой прямых команд на робот точно так же, как это было в случае платформы NXT. На данный момент поддерживается большая часть блоков, доступная для программирования роботов Lego.

Генерация программ для автономного исполнения на роботе устроена следующим образом. По диаграмме генератором, похожим на генератор программ для NXT, порождается код на языке QtScript. После этого данная программа может быть либо загружена на робот, либо загружена и немедленно исполнена. С точки зрения пользователя все это делается нажатием одной кнопки на панели инструментов. При этом загруженная программа сохраняется на роботе и впоследствии может быть исполнена без подключения к компьютеру средствами самого робота. Кроме того, программу, загруженную на робот, можно редактировать средствами самого робота (при удаленном подключении к роботу с компьютера). Использование скриптового языка позволяет избежать необходимости перекомпиляции программы после редактирования, так что внесённые правки могут быть применены немедленно. При этом достаточно мощное аппаратное обеспечение робота гарантирует достаточную для большинства задач скорость работы программы даже с учётом задержек, вносимых интерпретатором.

С технической точки зрения среда времени выполнения, работающая на роботе, состоит из интерпретатора языка QtScript, коммуникационного модуля, обеспечивающего прием команд от QReal:Robots, и библиотеки, реализующей базовые команды робота. Поддержка языка QtScript для библиотеки Qt Embedded позволяет получить доступ к классам этой библиотеки непосредственно из программы на QtScript. Это делает процесс программирования весьма удобным и гибким. Точно так же реализованы и прямые команды, это не более чем динамически генерируемые программы на QtScript, исполняющие простые инструкции.

Указанный инструментарий уже сейчас позволяет реализовывать типовые для конструктора Lego задачи, такие как движение робота по линии. Решается такая задача с использованием кибернетического понятия “регулятор” – математического выражения, рассчитывающего управляющие воздействия на моторы по данным с датчиков и накопленной истории поведения робота. Программа на QReal:Robots состоит из порядка десятка блоков, и транслируется в программу на QtScript (тоже довольно небольшую), которая раз в 20 миллисекунд опрашивает датчики света и вычисляет управляющее воздействие.

## 5. ЗАКЛЮЧЕНИЕ

Представленный в статье набор инструментов сейчас активно внедряется в школьном образовании. Среда QReal:Robots в версии для Lego Mindstorms NXT уже используется в некоторых российских школах и кружках по робототехнике, представлялась на ряде конференций и соревнований, в

результате чего был выявлен большой интерес к проекту со стороны российских учителей. Реализация поддержки конструктора ТРИК в QReal:Robots показала, что среда легко расширяема, новые конструкторы, отдельные контроллеры или даже конкретные модели роботов могут быть интегрированы со средой в достаточно короткие сроки. Это особенно важно с учётом того, что среда имеет открытый исходный код, так что любой энтузиаст может самостоятельно реализовать поддержку нужного ему аппаратного обеспечения.

## 6. ЛИТЕРАТУРА

- [1] Звенигородский Г.А. Описание языка Робик, URL: <http://ershov.iis.nsk.su/archive/eaindex.asp?lang=1&did=7639>, дата обращения: 13.07.2013
- [2] Конструктор ТРИК, URL: <http://www.trikset.com/>, дата обращения: 13.07.2013
- [3] Литвинов Ю.В., Применение DSM-платформы QReal при разработке среды программирования роботов QReal:Robots // Системное программирование, Т. 7, № 1. СПб., Изд-во СПбГУ. 2012. С. 144 - 155. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
- [4] Литвинов Ю.В., Визуальные средства программирования роботов и их использование в школах // Современные информационные технологии и ИТ-образование, сборник избранных трудов VII Международной научно - практической конференции. Под ред. проф. В.А. Сухомлина. М.: ИНТУИТ.РУ. 2012. С. 858 - 868.
- [5] cyberneticzoo.com, 1969 – The Logo Turtle – Seymour Papert et al, URL: <http://cyberneticzoo.com/?p=1711>, дата обращения: 13.07.2013
- [6] LabVIEW System Design Software, URL: <http://sine.ni.com/np/app/main/p/docid/nav-104/lang/ru/>, дата обращения: 13.07.2013
- [7] LEGO Mindstorms homepage, URL: <http://mindstorms.lego.com/en-us/Default.aspx>, дата обращения: 13.07.2013
- [8] Luchin RM. Control engineering education with help of model-based program tools for robotic construction sets. *Advances in Control Education*, Vol. 9, 2012. pp. 425 – 429.
- [9] MyRobot, Язык программирования Лого, URL: <http://myrobot.ru/logo/aboutlogo.php>, дата обращения: 13.07.2013.
- [10] Microsoft Robotics Developer Studio homepage, URL: <http://www.microsoft.com/robotics/>, дата обращения: 13.07.2013
- [11] Robolab home page, URL: <http://www.ceeo.tufts.edu/robolabatceeo/>, дата обращения: 13.07.2013
- [12] Terekhov AN, Luchin RM, Filippov SA. Educational cybernetical construction set for schools and universities. *Advances in Control Education*, Vol. 9, 2012. pp. 430 – 435.