

Санкт-Петербургский государственный университет

Кафедра Системного Программирования

Малютин Данила Павлович

Интерпретация JavaScript в TRIK Studio

Курсовая работа

Научный руководитель:
ст. преп. Литвинов Ю. В.

Санкт-Петербург
2016

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Выбор подхода	6
2.2. Обзор использованных продуктов	6
2.2.1. TRIK Studio	6
2.2.2. TrikRuntime	7
3. Реализация	9
3.1. Интеграция TrikRuntime в TrikStudio	9
3.2. Реализация интерпретатора скриптов	10
3.2.1. Интерпретация скриптов в TrikRuntime	10
3.2.2. Интерпретация скриптов в TRIK Studio	10
4. Апробация	13
Заключение	14
Список литературы	15

Введение

TRIK Studio — среда обучения основам программирования и кибернетики. Среда позволяет создавать графические программы для роботов Lego® Mindstorms® NXT 2.0, Lego® EV3 [3], TRIK и исполнять эти программы прямо на компьютере, посылая команды роботу через Bluetooth или USB-интерфейс, а также генерировать по диаграммам код на различных языках программирования и закачивать его для исполнения в робота. Но не всегда бывает удобно производить отладку программы прямо на роботе, или он попросту может быть недоступен, поэтому в TRIK Studio предусмотрена 2D-модель.

Двумерная модель позволяет симулировать поведение робота при исполнении программы в своём специальном окружении. Это позволяет её применять в обучении детей основам программирования [5]. Они могут почти в реальном времени наблюдать за изменениями в поведении робота при изменении программы.

На данный момент модель может работать только с программами, написанными на языке диаграмм, но для полноценного обучения программированию этого не достаточно, так как это не позволяет нарабатывать навык программирования на текстовых языках.

К тому же блочный язык пока не имеет аналогов для определённых конструкций текстовых языков, то есть не всякая программа на текстовом языке может быть реализована с помощью диаграмм. Например, робот TRIK [4] может исполнять программы, написанные на QtScript [1] — разновидности JavaScript. В нём имеется возможность подписываться на сигналы, что позволяет писать ”реактивные” программы, которые реагируют на изменение внешних условий. Такую программу нельзя написать с помощью блочных диаграмм, следовательно её нельзя исполнить на 2D-модели, хотя она будет работать на реальном роботе.

Соответственно возникает задача реализации поддержки различных текстовых языков двумерной моделью. Это бы позволило обеспечить плавный переход в обучении от программирования на графических языках к программированию на языках текстовых, а так же

обойти некоторые ограничения 2D-модели, связанные с её ориентированностью на интерпретацию блочных диаграмм. В данной курсовой работе рассматривается реализация поддержки языка QtScript (далее JavaScript) в TRIK Studio.

1. Постановка задачи

- Проанализировать архитектуру QReal/TRIK Studio [2] на предмет возможности встраивания интерпретации JavaScript.
- Добавить поддержку исполнения программ, написанных на языке QtScript, в TRIK Studio.
- Апробировать полученное решение, запустив несколько программ из примеров на QtScript на 2D-модели.
- Подготовить среду для реализации поддержки других текстовых языков 2D-моделью.

2. Обзор

2.1. Выбор подхода

Было рассмотрено два варианта решения поставленной задачи.

1. Реализация своего интерпретатора JavaScript, совместимого со средой выполнения робота, с нуля.
2. Переиспользовать интерпретатор, написанный для робота.

К минусам первого подхода можно отнести сложность обеспечения полного соответствия между средой выполнения на реальном роботе и в TRIK Studio. Заметно усложняется сопровождение за счёт необходимости синхронизировать интерпретатор с возможными изменениями в среде выполнения робота.

Минусом второго подхода является нетривиальность задачи интеграции библиотеки интерпретатора робота в TRIK Studio. Тем не менее, после анализа среды выполнения для роботов TRIK, было решено выбрать второй вариант.

2.2. Обзор использованных продуктов

В рамках данной работы были рассмотрены среда для визуального программирования роботов TRIK Studio и среда выполнения Trik-Runtime, установленная на роботах TRIK.

2.2.1. TRIK Studio

TRIK Studio основана на проекте QReal¹, metaCASE системе, позволяющей пользователям создавать свои собственные визуальные языки и редакторы к ним. Основная часть TRIK Studio реализована как плагин к QReal.

¹CASE and metaCASE system, URL: <https://github.com/qreal/qreal> (дата обращения: 18.05.2015)

Основным является плагин `robotsPlugin` реализующий `ToolPlugin-Interface` — интерфейс плагинов `QReal`. Он же в свою очередь содержит `RobotsPluginFacade`, который отвечает за инициализацию и управление плагинами уже к самой `TRIK Studio`. Сами плагины делятся на несколько категорий, но для поставленной задачи интересна главным образом одна - `interpreters` (интерпретаторы). Именно эти плагины отвечают, как следует из названия, за интерпретацию программ роботами или 2D-моделью. За основную логику по управлению интерпретацией программы отвечает класс `interpreter` из `interpreterCore`, но его недостатком является то, что он спроектирован под работу с блоками на диаграмме и используют специфичные для них механизмы (например особый класс `Thread`).

2.2.2. `TrikRuntime`

`TrikRuntime`² является написанной на `Qt/C++` средой выполнения для контроллеров `TRIK`. Она разделяется на несколько частей (упрощённая диаграмма компонентов представлена на рис. 1):

- `trikCommunicator`
- `trikControl`
- `trikGui`
- `trikHal`
- `trikKernel`
- `trikNetwork`
- `trikRun`
- `trikScriptRunner`
- `trikServer`

²Runtime for `TRIK` controller, URL: <https://github.com/trikset/trikRuntime> (дата обращения: 18.05.2015)

- trikTelemetry
- trikWiFi

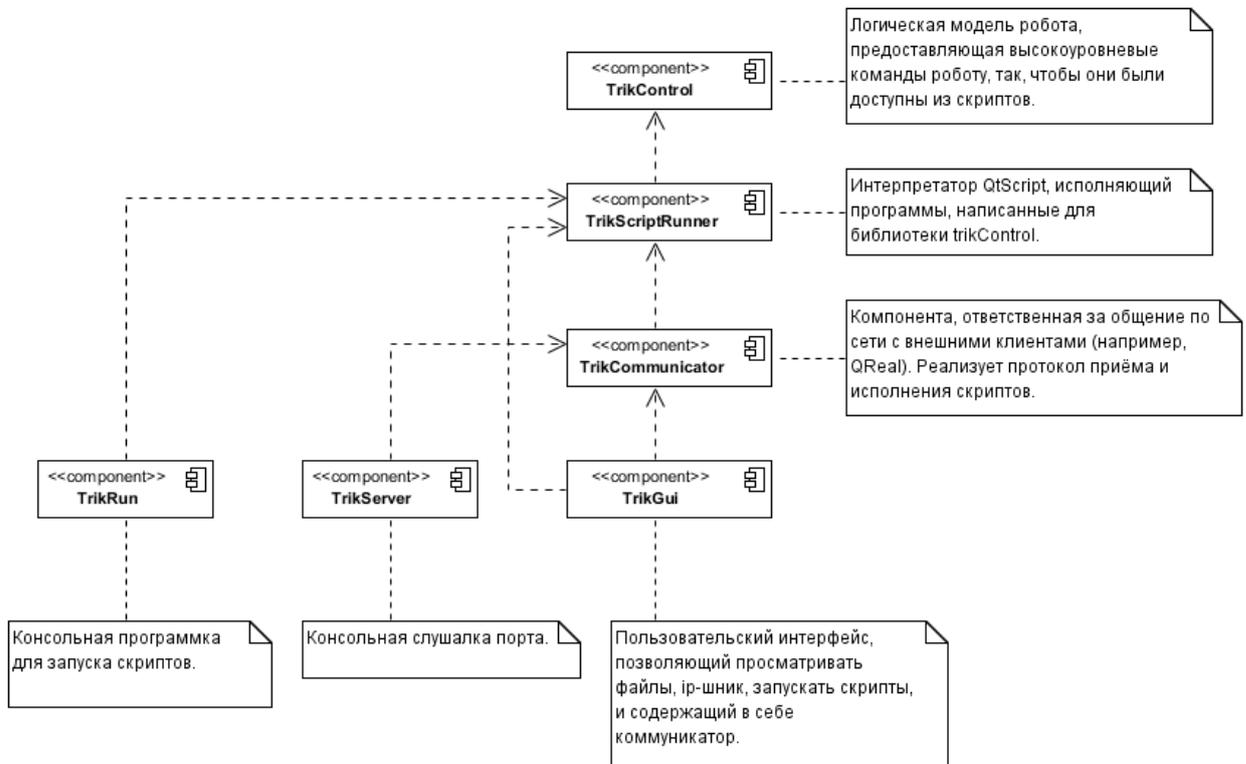


Рис. 1: Упрощённая UML-диаграмма общей структуры TrikRuntime

За интерпретацию скриптов QtS отвечают только trikControl, trikHal, trikKernel, trikNetwork и trikScriptRunner.

3. Реализация

3.1. Интеграция TrikRuntime в TrikStudio

Первой проблемой, которую надо было решить, являлась интеграция TrikRuntime в TRIK Studio, для того, чтобы можно было начать использовать предоставляемые ею классы для интерпретации скриптов. Так как среда времени выполнения TrikRuntime написана с использованием кроссплатформенной библиотеки Qt³, то её отдельная компиляция на ПК под управлением ОС Windows оказалась возможной без каких-то дополнительных изменений в коде. Таким образом проблема свелась к интеграции сборки нужных компонентов TrikRuntime в систему сборки TRIK Studio и выбору решения для обеспечения синхронизации между версиями TrikRuntime используемыми в TRIK Studio.

В качестве такого решения был выбран механизм git submodule⁴. Он позволяет подключить git репозиторий TrikRuntime к репозиторию TRIK Studio как папку (в данном случае расположенную в поддиректории thirdparty) и осуществлять синхронизацию за счёт обычных инструментов git, таких как fetch, pull и push.

Интеграция систем сборки оказалась затруднена ограничениями используемой системы qmake⁵. Например, много важной конфигурации для удобства вынесено в отдельный файл global.pri, который так же используется для сбора списка директорий с исходниками, но он входит в конфликт с аналогичным файлом в TRIK Studio. Поэтому пришлось отдельно добавлять исходники в список во всех библиотеках, где они используются. Это оказалось не слишком трудоёмкой задачей за счёт схожести в организации структуры проекта между TrikRuntime и TRIK Studio, но в будущем планируется выделить подобные вещи в отдельный файл, не зависящий от конфигурации.

³Домашняя страница библиотеки Qt, URL: <http://www.qt.io/> (дата обращения: 18.05.2015)

⁴Git Tools - Submodules, URL: <https://git-scm.com/book/en/v2/Git-Tools-Submodules> (дата обращения: 18.05.2015)

⁵qmake Manual, URL: <http://doc.qt.io/qt-5/qmake-manual.html> (дата обращения: 18.05.2015)

3.2. Реализация интерпретатора скриптов

3.2.1. Интерпретация скриптов в TrikRuntime

В TrikRuntime за интерпретацию скриптов отвечает класс TrikScriptRunner (рис. 2) из библиотеки trikScriptRunner. Он общается с различ-

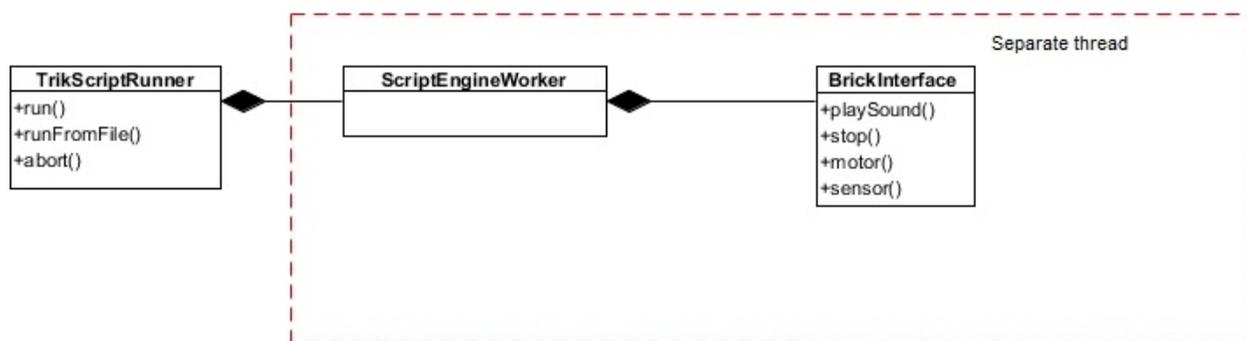


Рис. 2: Схема TrikScriptRunner

ными устройствами робота, такими как экран, всевозможные датчики, сенсоры и моторы, через интерфейс BrickInterface (рис. 3), предоставляемый библиотекой TrikControl вместе с интерфейсами, абстрагирующими устройства робота.

3.2.2. Интерпретация скриптов в TRIK Studio

TrikScriptRunner обеспечивает выполнение скриптов на языке JavaScript в том же окружении (то есть с тем же набором доступных методов, осуществляющих управление роботом), что и на реальном роботе, но он ничего не знает о двумерной модели.

Для того, чтобы обеспечить взаимодействие между интерпретатором скриптов и 2D-моделью, были написаны собственные реализации интерфейсов BrickInterface и всех нужных сенсоров, которые вместо общения с реальным роботом обращались бы к двумерной модели.

Предоставлением удобного интерфейса TRIK Studio для управления интерпретацией скриптов занимается класс TrikQtInterpreter (рис. 4). Он инициализирует класс TrikBrick, являющийся реализацией интерфейса BrickInterface, передавая ему указатель на двумерную модель,

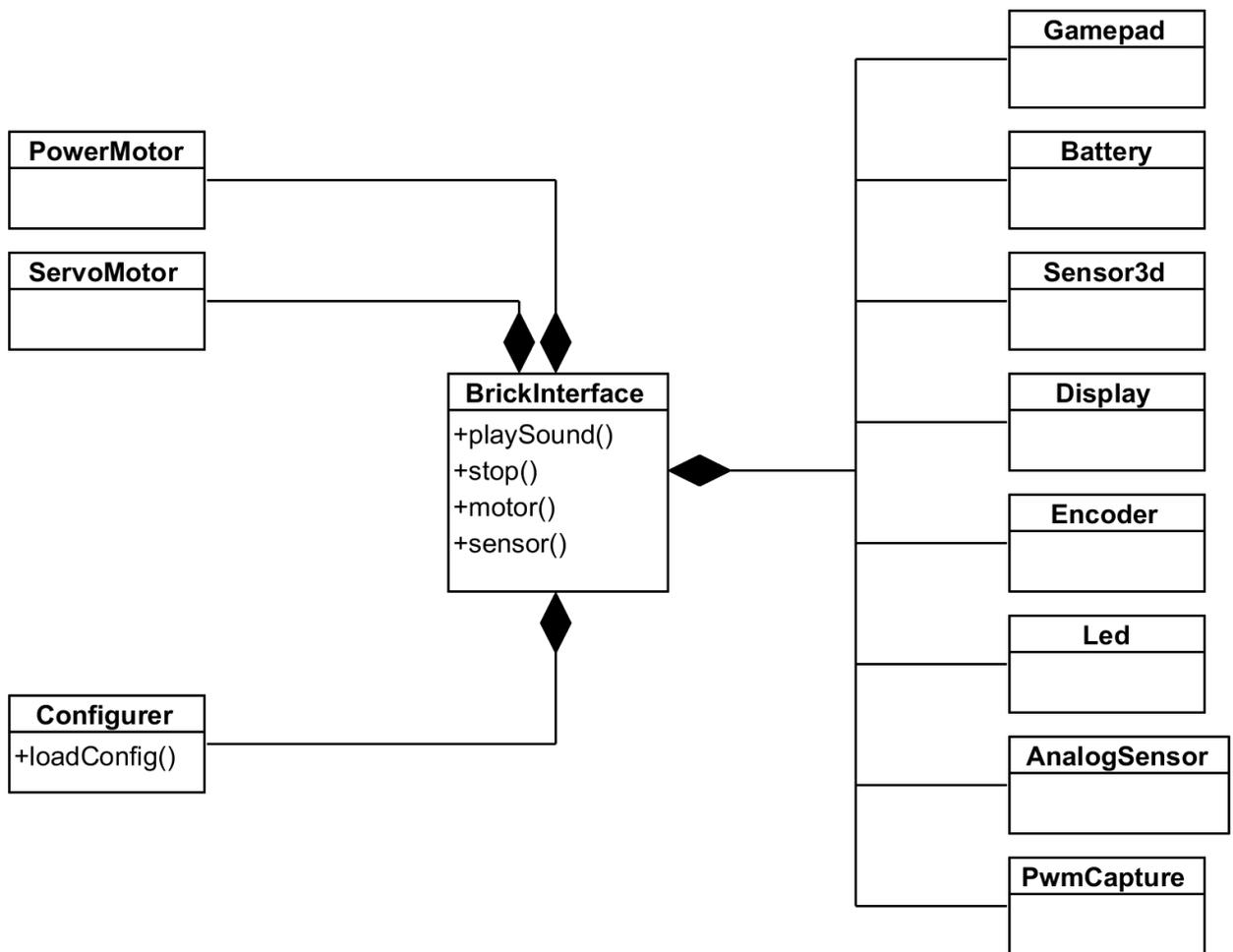


Рис. 3: Диаграмма классов BrickInterface

чтобы он мог потом, по требованию, инициализировать классы, эмулирующие устройства реального робота. Эти классы реализуют шаблон проектирования "Адаптер", транслируя запросы к контроллеру робота в запросы к частям двумерной модели.

За управление и инициализацию `TrikQtsInterpreter` отвечает `TrikKitInterpreter` - плагин, контролирующий интерпретацию для роботов TRIK. Он осуществляет интеграцию интерпретатора скриптов в графический интерфейс TRIK Studio. Например, он размещает на панели инструментов кнопки для старта и остановки интерпретации, следит за тем, чтобы не возникало конфликтов с кнопками блочного интерпретатора, и по нажатию на кнопку "Старт" считывает код скрипта из открытой вкладки текстового редактора на основе кроссплатформенной биб-

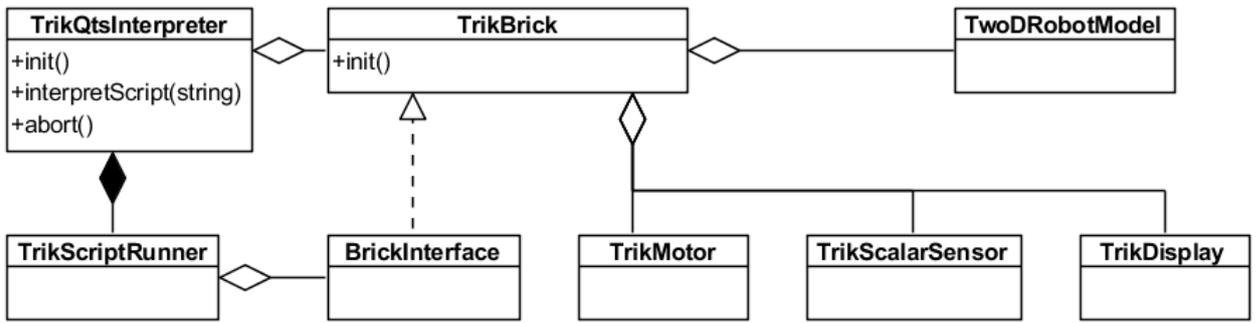


Рис. 4: Диаграмма классов TrikQtsInterpreter

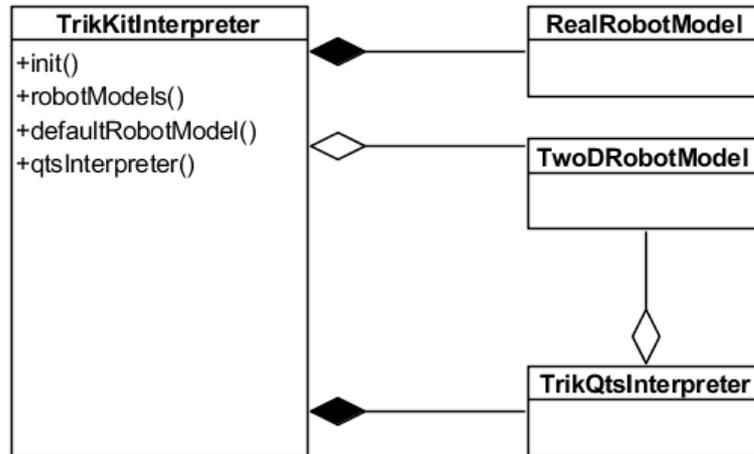


Рис. 5: Диаграмма классов TrikKitInterpreter

лиотеки QScintilla⁶ и передаёт его интерпретатору TrikQtsInterpreter.

⁶What is QScintilla, URL: <https://riverbankcomputing.com/software/qscintilla/intro> (дата обращения: 18.05.2015)

4. Апробация

Апробация производилась на примерах из стандартной поставки TRIK Studio, задействующих различные сенсоры и моторы. Сначала по диаграмме генерировался код на QtS, а затем запускалась интерпретация сгенерированного кода на двумерной модели (пример запущенной интерпретации на рис. 6). Недостающие сенсоры реализовывались по мере надобности и результат сравнивался с интерпретацией диаграмм.

Интерпретатор успешно справился с поставленной задачей, хоть в процессе тестирования и выявились некоторые проблемы со стороны пользовательского интерфейса, вызванные некорректными предположениями в коде, которые не учитывали возможность появления текстовых интерпретаторов в TRIK Studio. Например, в конфигурации окон TRIK Studio было жёстко прописана видимость окна с 2D-моделью только в режиме отладки диаграмм. При попытке перейти в режим отладки из текстового редактора, окно с двумерной моделью оставалось скрытым. Кроме того, пришлось переписать логику работы кнопок, отвечающих за запуск интерпретации, чтобы они учитывали возможность запуска интерпретатора из вкладки с кодом, а не диаграммой.

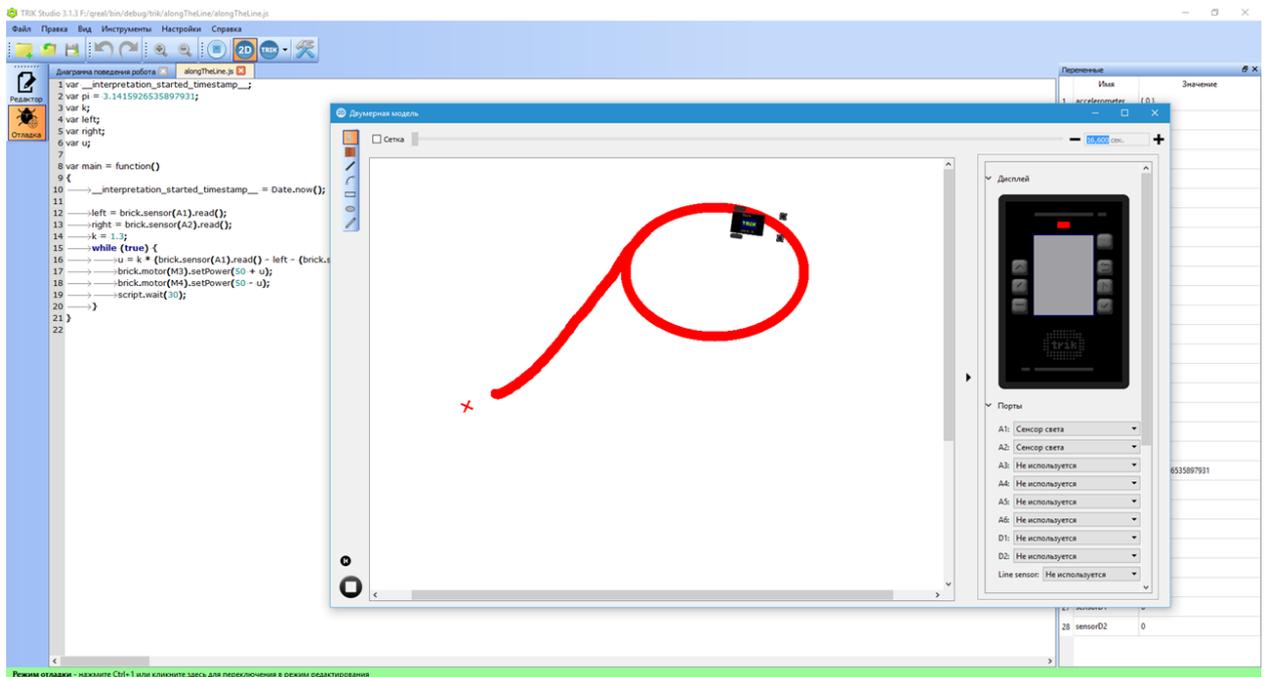


Рис. 6: Запущенная интерпретация на примере alongTheLine

Заключение

Полученные результаты:

- В TRIK Studio добавлена поддержка исполнения 2D-моделью программ на языке QtScript.
- Решение апробировано на стандартных примерах программ для робота из поставки TRIK Studio.
- Интерфейс TRIK Studio подготовлен к добавлению новых текстовых интерпретаторов.

Дальнейшие перспективы:

- Планируется разработать более универсальный способ по добавлению новых интерпретаторов в TRIK Studio.
- Переработать интерфейс TRIK Studio с целью более удобного управления и его состоянием.
- Добавить более глубокие возможности по отладке кода, такие как, например, подсветка выполняющейся в данный момент строки кода.

Список литературы

- [1] The Qt Company Ltd. Qt Script // Qt Documentation. — 2015. — URL: <http://doc.qt.io/qt-5/qtscript-index.html> (дата обращения: 08.10.2015).
- [2] Архитектура среды визуального моделирования QReal / Терехов А.Н., Брыксин Т.А., Литвинов Ю.В. и др. Системное программирование. Т. 4. — СПбГУ, 2000.
- [3] Когутич. Д.А., Смирнов М.А., Литвинов Ю.В. Поддержка конструктора EV3 в TRIK Studio // Материалы научно-практической конференции студентов, аспирантов и молодых учёных "Современные технологии в теории и практике программирования". — Изд-во Политехн. ун-та, 2015. — С. 40–41.
- [4] ООО КИБЕРТЕХ. Кибернетический конструктор ТРИК. — 2015. — URL: <http://www.trikset.com> (дата обращения: 01/10/2015).
- [5] Терехов А.Н., Брыксин Т.А., Литвинов Ю.В. Среда визуального программирования роботов QReal:Robots // III Всероссийская конференция «Современное технологическое обучение: от компьютера к роботу» (сборник тезисов), СПб., 2013, С. 1-4 (электронная публикация). — 2015. — URL: http://www.239.ru/userfiles/file/Abstract_CompRobot2013.pdf (дата обращения: 29.04.2015).