Санкт-Петербургский Государственный Университет

Математико-механический факультет

TRIK Mobile

Мобильная среда программирования робота

на платформе TRIK

Курсовая работа

студента 244 группы

Приходько Станислава Витальевича

Научный руководитель

ст. преп. каф. Системного программирования

Ю. В. Литвинов

Санкт-Петербург

2015

**Оглавление**

Введение ················································································· 3

Цель работы ············································································· 4

Обзор существующих решений ····················································· 4

TRIK Gamepad ·································································· 4

Постановка задачи ····································································· 5

Использованные инструменты ······················································ 5

Особенности реализации ····························································· 6

Общая структура ······························································· 6

Связь с роботом ································································· 6

Блоки ·············································································· 7

Результаты ··············································································· 7

Список литературы ···································································· 9

**Введение**

Сегодня, в период технологического прогресса человечества, все большую актуальность приобретают автоматизированные роботы, способные выполнять полезную работу. В связи с этим набирает обороты развитие робототехнического направления.

TRIK Studio - cреда обучения основам программирования и кибернетики. Среда позволяет создавать графические программы для роботов Lego® Mindstorms® NXT 2.0, Lego® EV3, 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.

TRIK Studio - программа, доступная для работы на операционных системах Windows и Linux. Однако в современном мире все большую популярность приобретают мобильные устройства [1]. Все больше и больше гаджетов окружают человека каждый день. Под рукой практически всегда есть то или иное устройство с возможностью соединения с устройством по Bluetooth или Wi-Fi. Поэтому возникает потребность в написании мобильных приложений для удовлетворения запросов пользователя.

Например, школьник младших классов хочет воспользоваться конструктором ТРИК, но у него может не быть под рукой ПК или ноутбука. Однако, смартфон, скорее всего, у него всегда есть с собой. В данной ситуации ему могло бы помочь мобильное приложение.

Мобильное приложения позволило бы расширить охват аудитории пользователей ТРИК, увеличить вариативность платформ для использования программного обеспечения и придать мобильность платформе ТРИК.

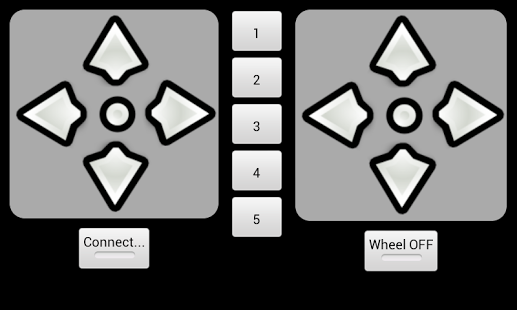
TRIKMobile является попыткой перенести часть функциональность программы TRIK Studio на мобильные устройства, а именно предоставить возможность создавать и редактировать модели программ и отправлять их на робота для исполнения.

**Цель работы**

Цель нашего проекта разработать приложение, позволяющее создавать программы для роботов, сопоставимые с программами, порожденными в TRIK Studio. Приложение должно иметь интуитивно понятный пользователю интерфейс. Должно быть адаптировано под небольшие, ограниченные по размеру экраны. Приложение должно быть кроссплатформенным.

**Обзор существующих решений**

TRIK Gamepad



В лаборатории ТРИК был разработан пульт для Android-устройств, с помощью которого можно управлять роботом ТРИК и выполнять простейшие действия с ним.

В отличие от этого решения, задача нашего проекта заключается в разработке среды программирования робота.

**Постановка задачи**

Передо мной была поставлена задача разработать способ соединения с роботом по сети Wi-Fi, реализовать возможность отправки скриптов на робота, реализовать содержание блоков и их структуру для написания программы для робота. Разработкой пользовательского интерфейса занимался коллега по команде Малютин Данила.

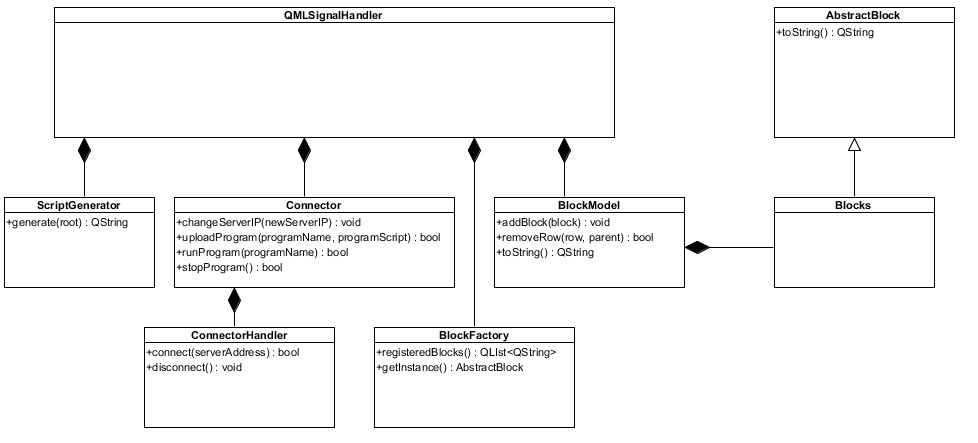
**Использованные инструменты**

Для реализации проекта был выбран кроссплатформенный фреймворк Qt 5. Основная логика программы написана на языке С++.

Для тестирования приложения на мобильном устройстве, работающем под управлением операционной системы Android, были использованы Android SDK, набор инструментов Android NDK, утилита для автоматизации процесса сборки программы Apache Ant, AVD Manager для установки эмулятора мобильного устройства, установлен AVD for Nexus 7 с процессором Intel Atom (x86) на платформе Android 5.0.1.

**Особенности реализации**

**Общая структура**

****

Основными компонентами проекта являются QMLSignalHandler, ScriptGenerator, Connector, ConnectorHandler, BlockFactory и BlockModel.

QMLSignalHandler отвечает за взаимодействие с пользовательским интерфейсом на qml.

ScriptGenerator генерирует по модели qts скрипты, которые впоследствии должны быть отправлены на робота.

**Связь с роботом**

Для осуществления взаимодействия с роботом разработан класс Connector, в котором поддержаны методы загрузки программного скрипта, построенного с помощью ScriptGenerator, на робота, запуска отправленной программы для исполнения на роботе с помощью команды run, остановки выполнения программы на роботе в результате выполнения команды stop, обновления IP-адреса робота, к которому осуществляется подключение. Сами подключение и отключение соединения с роботом вынесены в отдельный обработчик ConnectorHandler, через который и происходит установление связи с роботом и прекращение связи с ним.

**Блоки**

Для хранения блоков создана модель, обладающая возможностью добавления и удаления элементов. Все добавляемые на палитру блоки хранятся в корневой модели приложения.

Каждый отдельный блок программы наследуется от абстрактного блока и позволяет вносить изменения в атрибуты, описанные для каждого блока в отдельности. Некоторые структурные блоки, реализовывающие такие операторы, как условный переход if, циклы for, while и некоторые другие, подразумевают наличие тела, представляющего собой набор блоков, хранимых в модели-потомке, который поддерживает возможность добавления новых блоков. Отдельно выделяется блок условного оператора if, подразумевающий наличие ветвления по else. Он требует отдельной обработки и особого способа отрисовки на экране. Его отличительная особенность заключается в наличии двух моделей-потомков, отвечающих за ветви then и else.

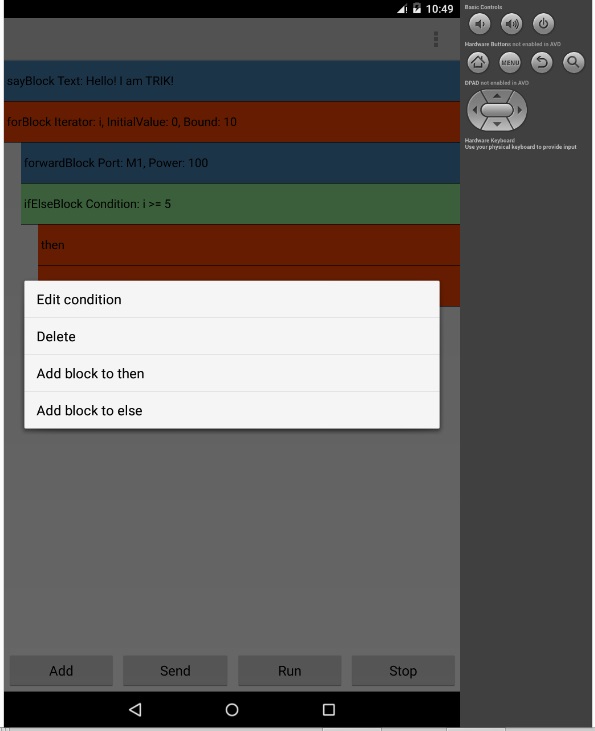
Добавление новых блоков на палитру осуществляется с помощью реализации паттерна “Абстрактная фабрика” [2]. Новые блоки создаются в зависимости от имени блока, указанного при выборе пользователя, и добавляются в соответствующую модель.

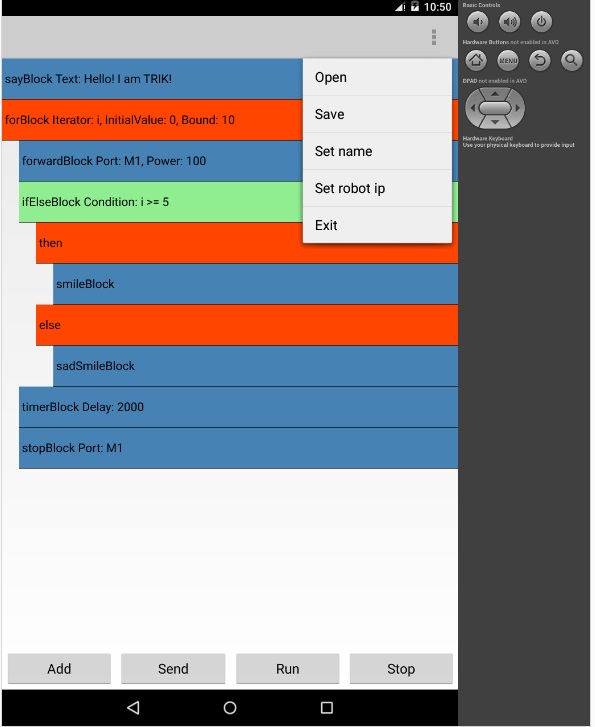
**Результаты**

В итоге реализовано соединение с роботом по сети Wi-Fi и поддержана возможность по нажатию соответствующей кнопки в приложении отправить программу на робота, выполнить программу на роботе или остановить работу программы.

Реализована модель для хранения блоков, описаны блоки для написания программ, разработана возможность добавления новых блоков, редактирования атрибутов и удаления блоков.

Протестирована работа приложения на устройстве AVD for Nexus 7 с процессором Intel Atom (x86) на платформе Android 5.0.1.





**Список литературы**

[1] Wikipedia. Usage share of operating systems // Википедия, свободная энциклопедия. –– (Дата обращения 12.05.2015). URL: <http://en.wikipedia.org/wiki/Usage_share_of_operating_systems>

[2] Design Patterns: Elements of Reusable Object-Oriented Software // Приемы объектно-ориентированного проектирования. Паттерны проектирования

–– Эрих Гамма [Erich Gamma], Ричард Хелм [Richard Helm], Ральф Джонсон [Ralph Johnson], Джон Влиссидес [John Vlissides], 2013