

Санкт-Петербургский государственный университет

Кафедра системного программирования

Белков Роман Владимирович

Программирование роботов ТРИК с  
помощью JVM

Курсовая работа

Научный руководитель:  
ст. преп. Кириленко Я. А.

Санкт-Петербург  
2016

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Постановка задачи</b>	<b>4</b>
<b>2. Существующие решения</b>	<b>5</b>
2.1. LeJOS . . . . .	5
2.2. Pi4J . . . . .	5
<b>3. Выбор реализации JVM</b>	<b>6</b>
<b>4. Реализация библиотеки</b>	<b>7</b>
4.1. Выбор языка для библиотеки . . . . .	7
4.2. Обзор устройств ТРИК . . . . .	7
4.3. Событийно-ориентированное программирование . . . . .	7
4.4. Документация и примеры программ . . . . .	8
<b>5. Применение</b>	<b>10</b>
<b>Результаты</b>	<b>11</b>
<b>Список литературы</b>	<b>12</b>

# Введение

По своим вычислительным ресурсам доступные робототехнические контроллеры приближаются к показателям персональных компьютеров пяти-/шестилетней давности. Это позволяет постепенно применять в разработке современные методологии и технологии, уходя от классических для микроконтроллеров низкоуровневых технологий к более богатому и инструментарию разработки с высоким уровнем абстракции для более эффективного использования комплексных возможностей робототехнических контроллеров.

Поскольку писать и поддерживать код на С тяжело и дорого относительно, скажем, Java или С# [3], то иметь одну гибко конфигурируемую библиотеку гораздо проще и дешевле, чем поддерживать несколько нативных библиотек и пользоваться различными компиляторами при использовании различного типа аппаратного обеспечения. Это может не быть проблемой в небольших масштабах, но при том гигантском росте<sup>1</sup>, который демонстрирует ”интернет вещей”, масштабируемость разработки — невероятно сложная задача.

Однако не только промышленные языки программирования и инструменты оказывают положительное влияние на робототехнику. Из-за большого образовательного потенциала робототехники возможно обучение использованию профессиональных инструментов, языков программирования, а также знакомству с идеями и приёмами, используемых в реальной разработке.

Идейно данная работа перекликается с трудами[2], в которых была представлена библиотека для программирования роботов, написанных на языках платформы .NET

---

<sup>1</sup><http://www.gartner.com/newsroom/id/3165317>

# 1. Постановка задачи

В рамках данной работы были поставлены следующие задачи.

- Выбрать реализацию среды исполнения JVM и после развернуть выбранную среду на контроллере ТРИК.
- Воздать библиотеку времени исполнения для программирования роботов, отвечающую следующим параметрам:
  - библиотека должна поддерживать все популярные сенсоры, включённые в наборы поставки робототехнического контроллера ТРИК;
  - предоставляемый API библиотеки должен быть высокого уровня, то есть пользователь должен знать минимум технических деталей для создания программы поведения робота и работы с сенсорами или актуаторами;
  - библиотека должна поддерживать возможность программирования в событийно-ориентированном стиле.

## 2. Существующие решения

### 2.1. LeJOS

Операционная система и набор библиотек<sup>2</sup>, позволяющие писать на языках платформы JVM для контроллеров Lego. Из-за проприетарности контроллера Lego решение представляет лишь ограниченный интерес, поскольку, несмотря на хорошую архитектуру, использует множество специфичных для контроллера Lego решений и обходных путей. Например, LeJOS использует свою ОС для работы программ.

### 2.2. Pi4J

Библиотека<sup>3</sup>, предоставляющая разработчикам удобную функциональность для работы с системой ввода-вывода при написании программ на языках платформы JVM для контроллера Raspberry Pi. Библиотека предоставляет пользователю для работы с шинами SPI, I2C и может брать на себя работу с прерываниями при работе с контактами GPIO. Однако такой уровень абстракции недостаточно высок для наших задач, поскольку требуется работать непосредственно с датчиками, при этом избегая конфигурирования контактов GPIO и необходимости писать код, отвечающий за отправку сигналов на нужные адреса шин или контактов.

---

<sup>2</sup><http://www.lejos.org/>

<sup>3</sup><http://pi4j.com/>

### 3. Выбор реализации JVM

Мы рассматривали две наиболее широко известные доступные для свободного использования реализации JVM: OpenJDK и Oracle JDK. Поскольку сравнение реализаций JDK — масштабная работа, то было решено сделать выбор, опираясь на результаты из сторонних источников. Поскольку тесты показали, что Oracle JDK в среднем в 10 раз быстрее OpenJDK [4], была выбрана именно Oracle JDK.

В процессе разворачивания JDK сложностей не возникло, поскольку Oracle предоставляет гибко конфигурируемый пакет для создания JDK и всё, что оставалось сделать нам — аккуратно следовать инструкциям и получить рабочую JDK на работе.

## 4. Реализация библиотеки

### 4.1. Выбор языка для библиотеки

При выборе языка для библиотеки мы рассматривали следующие варианты: Java, Scala, Groovy, Clojure, Kotlin. Основными требованиями, выдвинутыми к языку библиотеки, являются поддержка языком функциональной парадигмы и доступность API библиотеки из программ на Java в случае, если библиотека написана не на Java.

Язык	Java	Scala	Groovy	Clojure	Kotlin
Объектно-ориентированный	+	+	+	-	+
Функциональный	-	+	-	+	+
Статическая типизация	+	+	- <sup>4</sup>	-	+
Хорошее взаимодействие с Java		-	+	-	+
Простой синтаксис	+	-	+	-	+

После проведения сравнения было решено выбрать Kotlin как язык для написания библиотеки. Хотя Kotlin на момент создания библиотеки был в предрелизной стадии и не имел большого сообщества разработчиков, авторы языка предложили поддержку и заверяли, что Kotlin будет верным выбором.

### 4.2. Обзор устройств ТРИК

Устройства контроллера ТРИК возможно разделить на два концептуальных для библиотеки типа: та, которая уже удобно поддерживается в прошивке с использованием event-файлов или FIFO-файлов и та, которая требует дополнительной доработки, то есть общение с устройством происходит через шину I2C.

### 4.3. Событийно-ориентированное программирование

Поскольку робот является системой, регистрирующей события при помощи сенсоров и реагирующих на поток данных, полученных из событий, при помощи актуаторов, то при программировании роботов удоб-

Оборудование	Тип
Периферия	
Силовой мотор с энкодером	шина I2C
Угловой сервопривод	event-файл
Аналоговые датчики	шина I2C
Цифровые датчики	event-файлы
Видеомодуль	FIFO-файлы
Плата	
Гироскоп	event-файл
Акселерометр	event-файл
Кнопки действий	event-файл

но использовать со бытийно-ориентированное программирование[1]. Современное развитие событийно-ориентированный подход получил в парадигме *реактивного* программирования. При таком подходе у пользователя есть возможность с лёгкостью описывать, как должны модифицироваться и обрабатываться данные. Именно поэтому одной из главных задач при разработке библиотеки была именно реализация этого пункта.

Проект Reactive Extensions<sup>5</sup> занимается созданием библиотек для создания асинхронных и событийно-ориентированных программ на различных языках программирования. Основная идея этих библиотек — расширение понятий Observer и Observable. Для Kotlin, несмотря на молодость языка, уже существует RxKotlin<sup>6</sup>. Однако было решено использовать именно RxJava, которая уже имела стабильные версии и признана как надёжная многими разработчиками, в то же время RxKotlin — библиотека в стадии ранней разработки и является обёрткой над RxJava<sup>7</sup>.

## 4.4. Документация и примеры программ

Поскольку библиотека ориентирована в том числе и на школьников средних и старших классов, одним из важным моментов при за-

<sup>5</sup><http://reactivex.io/>

<sup>6</sup><https://github.com/ReactiveX/RxKotlin>

<sup>7</sup><https://github.com/ReactiveX/RxJava>



вершении создания библиотеки было написание базовой документации и создание программ-примеров для наглядности. Так появились инструкция по разворачиванию JDK на контроллере, описание методов и их параметров, а так же три демо-модели: сегвей, "робот-охотник", следящий за целью, и пример группового взаимодействия роботов через TCP. Код сегвея написан на Kotlin в событийно-ориентированном стиле и на Java в императивном стиле. Две другие модели написаны на Kotlin в событийно-ориентированном стиле.

## 5. Применение

Разработанная библиотека применялась при создании демонстрационных моделей на базе контроллера-конструктора ТРИК. Такие модели, как сегвей и робот-охотник, следящий за объектом, демонстрировались на многочисленных выставках.

# Результаты

В ходе выполнения работы были получены следующие результаты.

- Развёрнута Java SE в виде Oracle JDK на ТРИК.
- Реализована библиотека времени исполнения для работы с периферией и датчиками контроллера:
  - реализована поддержка парадигмы реактивного программирования;
  - созданы примеры программ и написана документация.
- Материалы работы были представлены на конференции SECR-2015.
- Исходные коды проекта и документация выложены в свободный доступ.<sup>89</sup>

---

<sup>8</sup><https://github.com/RomanBelkov/TrikKotlin>

<sup>9</sup><https://github.com/RomanBelkov/TrikKotlinDemos>

## Список литературы

- [1] Arrows, Robots, and Functional Reactive Programming / Paul Hudak, Antony Courtney, Henrik Nilsson, John Peterson // Summer School on Advanced Functional Programming 2002, Oxford University. — Vol. 2638 of Lecture Notes in Computer Science. — Springer-Verlag, 2003. — P. 159–187.
- [2] Kirsanov Alexander, Kirilenko Iakov, Melentyev Kirill. Robotics Reactive Programming with F#/Mono // Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia. — CEE-SECR '14. — New York, NY, USA : ACM, 2014. — P. 16:1–16:5. — URL: <http://doi.acm.org/10.1145/2687233.2687249>.
- [3] McConnell Steve. Code Complete, Second Edition. — Redmond, WA, USA : Microsoft Press, 2004. — P. 61–63. — ISBN: 0735619670, 9780735619678.
- [4] Дидух А.И., Тищенко В.В. Сравнение быстродействия Java на микрокомпьютере Raspberry Pi // Вісник Національного технічного університету України Київський політехнічний інститут. Серія: Радіотехніка. Радіоапаратобудування. — 2015. — Vol. 60.