

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра Системного Программирования

Агеев Денис Витальевич

Создание онлайн инфраструктуры для конфигурирования и управления роботами

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:
ст. преп. Брыксин Т. А.

Рецензент:
ст. преп. Литвинов Ю. В.

Санкт-Петербург
2015

SAINT-PETERSBURG STATE UNIVERSITY

Chair of Software Engineering

Ageev Denis

Creating online infrastructure for robots
configuration and control

Bachelor's Thesis

Admitted for defence.
Head of the chair:
professor A.N. Terekhov

Scientific supervisor:
assistant T.A. Bryksin

Reviewer:
assistant Y.V. Litvinov

Saint-Petersburg
2015

Оглавление

Введение	4
1. Постановка задачи	5
2. Предметная область	6
2.1. Обзор существующих решений	6
2.1.1. Google Play	6
2.2. Контроллер ТРИК	6
3. Разработанное решение	9
3.1. Требования и общая архитектура	9
3.2. Компоненты	10
3.2.1. Веб-приложение	10
3.2.2. Сервер маршрутизации роботов	15
3.2.3. Контроллер ТРИК	17
3.3. Проверка работоспособности системы	18
4. Результаты	20
5. Дальнейшее развитие	21
Список литературы	22

Введение

В настоящее время особое место в развитии технологий занимает робототехника. Эта отрасль привлекает не только опытных программистов, но и обычных школьников. Робототехника внедряется в образовательный процесс, появляются кружки робототехники, использующие специальные робототехнические конструкторы [12].

Одним из примеров робототехнического конструктора является ТРИК, созданный на основе многолетних разработок преподавателей кафедры системного программирования и теоретической кибернетики СПбГУ [10, 1].

Для внедрения робототехники в образовательный процесс недостаточно иметь только конструктор и контроллер робота. Необходимо наличие специального программного обеспечения, которое позволит этих роботов программировать.

ТРИК имеет мощную инфраструктуру для создания программ для роботов. Для школьников сделана специальная визуальная среда программирования TRIK Studio¹.

Данная работа ведется в рамках переноса визуальной среды программирования TRIK Studio в веб. Владимир Захаров в своей работе разрабатывает онлайн среду графического программирования роботов. Но несмотря на наличие данной среды, до сих пор остается проблема доставки программ на робота. В классическом варианте пользователь должен настраивать подключение к роботу, устанавливать специальные драйвера, с чем могут возникнуть трудности. Также сейчас нет удобной функциональности, которая позволяла бы конфигурировать подключаемые периферийные устройства к роботу. Поэтому в рамках данного проекта было решено создать онлайн инфраструктуру для управления и конфигурирования роботов.

¹Среда TRIK Studio, URL: <http://blog.trikset.com/p/trik-studio.html> (дата обращения 17.05.2015)

1. Постановка задачи

Целью данной бакалаврской работы является создание инфраструктуры для роботов, позволяющей удаленно конфигурировать периферийные устройства роботов, собранных на базе платформы ТРИК, загружать и запускать на них программы.

Для достижения этой цели был сформулирован следующий набор задач.

- Разработка общей архитектуры системы для управления и конфигурирования роботов.
- Реализация системы для управления и конфигурирования роботов.
- Интеграция с онлайн средой графического программирования роботов, разрабатываемой в работе Захарова Владимира.
- Проверка работоспособности разработанного решения.

2. Предметная область

2.1. Обзор существующих решений

Перед решением поставленной задачи был произведен анализ предметной области и проанализированы существующие решения. В ходе анализа не было найдено аналогичных решений в области робототехники. Однако были найдены схожие решения в смежных областях, например, в мобильных технологиях.

2.1.1. Google Play

Google Play — магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения.

Пользователь может подключаться к данной системе, регистрировать свои устройства и с помощью веб-приложения или специального мобильного приложения загружать на мобильное устройство программы. Данная система позволяет регистрировать несколько мобильных устройств — это дает возможность централизованного управления мобильными устройствами.

Данная система обладает схожим рабочим процессом с нашей инфраструктурой для управления и конфигурирования роботов. В нашей системе пользователь также регистрирует робота. Далее пользователь имеет возможность подключиться к системе и загружать на него программы, созданные в онлайн среде графического программирования.

2.2. Контроллер ТРИК

Контроллер ТРИК совместим с широким спектром периферийных устройств, имеет в своем составе все необходимое оборудование для управления двигателями постоянного тока и сервоприводами, а также для приема и обработки информации от цифровых и аналоговых дат-

чиков, микрофонов, видеомодулей. Контроллер снабжён цветным сенсорным дисплеем, программируемыми кнопками, есть поддержка WiFi, Bluetooth 4.0 (включая LE). В контроллере установлены встроенные защиты от перегрузки по току и от глубокой разрядки аккумулятора. Таким образом кибернетический конструктор ТРИК позволяет собирать множество различных робототехнических моделей: от базовых образовательных до самых современных, способных ”видеть”, ”слышать”, взаимодействовать друг с другом.

Контроллер ТРИК работает на основе трех процессоров: DSP, ARM, MSP. На центральном ARM-процессоре контроллера работает ОС Linux. Поэтому автономные модели можно программировать практически на любом языке программирования. Но большая часть разработок ведется на компьютерах, процессоры которых отличны от процессора ARM, поэтому специально для этого был разработан trikSDK², выполняющий кросс-компиляцию для платформы ARM. Пакет содержит стандартный набор GCC инструментов, адаптированный под проект ТРИК, позволяющий разрабатывать приложения для ARM-систем.

Пример робототехнической модели, собранной на базе контроллера ТРИК показан на рис. 1.

Конфигурирование работа

В рамках проекта ТРИК для контроллера уже было реализовано программное обеспечение — trikRuntime³. Данное программное обеспечение позволяет задавать конфигурацию робота с помощью специальных файлов.

Под конфигурированием робота мы понимаем возможность указывать в его настройках, какие устройства на каких портах мы будем использовать, а также с какими параметрами их запускать.

Конфигурация роботов на контроллере задается с помощью двух файлов: системного конфигурационного файла и текущей модели.

²Пакет инструментов trikSDK, URL: <http://downloads.trikset.com/updates/sdk/latest-trik-sdk.sh> (дата обращения 17.05.2015)

³Рабочее окружение trikRuntime, URL: <https://github.com/trikset/trikRuntime> (дата обращения 17.05.2015)

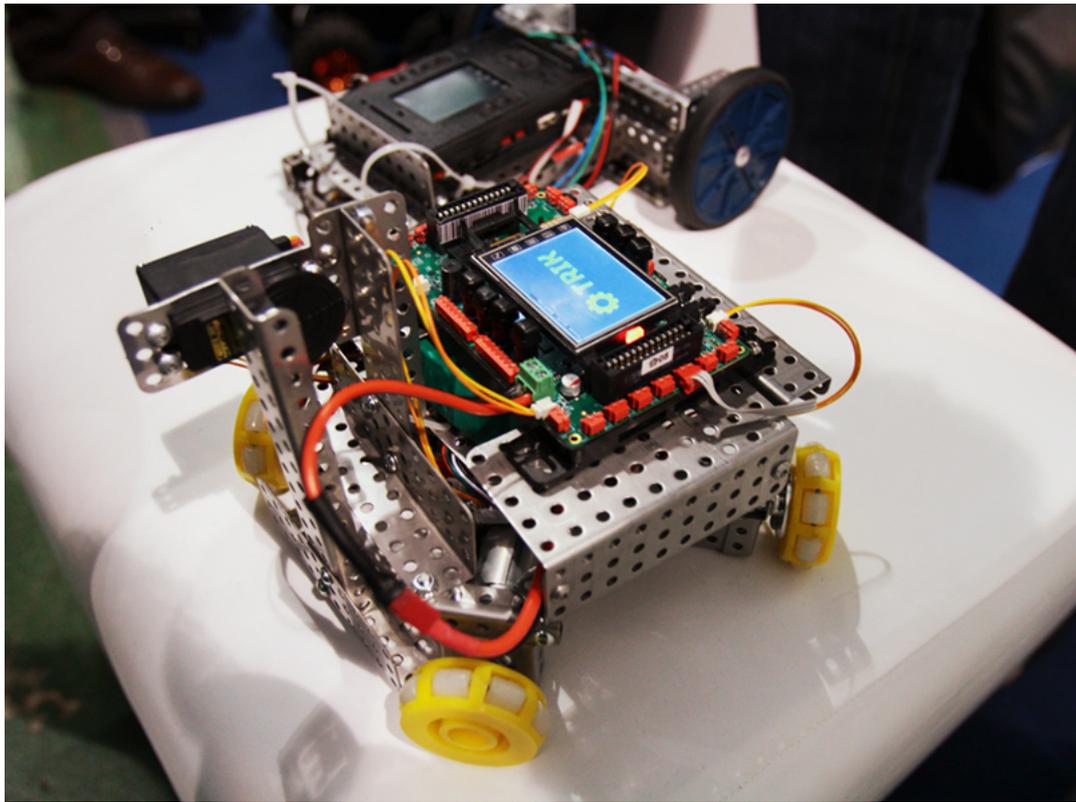


Рис. 1: Робототехническая модель на базе контроллера ТРИК

- Системная конфигурация описывает, какие порты доступны для конфигурирования, и какие устройства можно к ним подключать. Кроме того, содержит общую информацию о всех устройствах и их параметрах работы на робототехнической модели.
- Модель содержит текущую конфигурацию подключенных устройств с их параметрами работы.

3. Разработанное решение

3.1. Требования и общая архитектура

В процессе проектирования были составлены следующие требования к системе управления и конфигурирования роботов.

- Отображение текущей модели роботов и возможность ее изменять с помощью пользовательского интерфейса.
- Возможность удаленного запуска на роботах программ, созданных с помощью онлайн среды графического программирования.
- Возможность подключения робота к онлайн-сервису и ожидания приема программ или новой конфигурационной модели от пользователя.

Такая система может быть реализована с помощью набора обособленных компонент. Часть из них отвечает за взаимодействие роботов с пользователем, другая же часть отвечает за отображение и конфигурирование роботов. Для обеспечения взаимодействия компонент сервиса используется специально разработанный для этого протокол, основанный на представлении данных в формате JSON. Схема взаимодействия компонент системы показана на рис. 2.

Кратко опишем каждую из компонент системы.

- Веб-приложение

С помощью данного приложения пользователь может зарегистрировать робота и подключить его к приложению. После этого у него появится возможность конфигурирования робота и отправки на него созданных им в графическом редакторе программ.

- Сервер маршрутизации роботов

Сервер обеспечивает взаимодействие между пользователем и роботом. Сервер хранит список всех подключенных роботов и их текущие конфигурации.

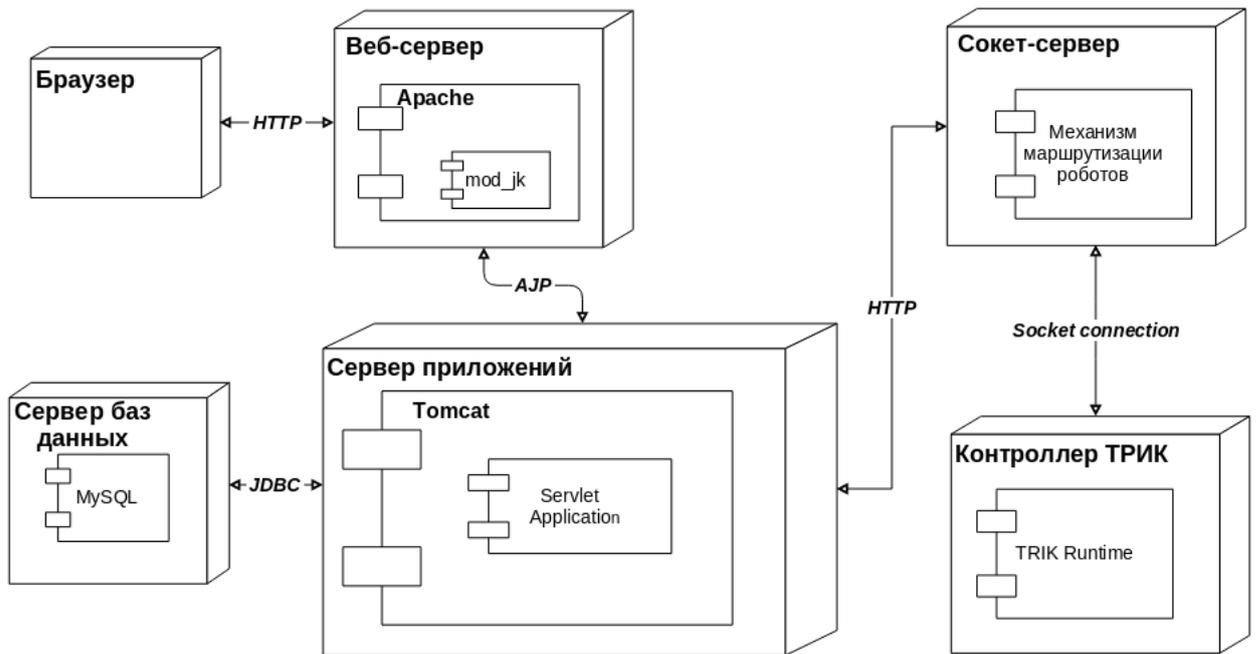


Рис. 2: Архитектура системы

- Контроллер ТРИК

На контроллере реализована функциональность, способная запускать присланные пользователем программы и обновлять конфигурацию, полученную от пользователя.

3.2. Компоненты

3.2.1. Веб-приложение

Используемые технологии

Для реализации веб-приложения было решено использовать следующие технологии.

Основным языком программирования был выбран Java⁴ [8], он удобен и надежен при программировании веб-приложений, для него есть множество фреймворков, которые позволяют быстро и качественно создавать хорошие веб-приложения.

⁴Язык Java, URL: <https://www.java.com> (дата обращения 17.05.2015)

В качестве веб-фреймворка используется Spring-MVC⁵ [5]. Этот фреймворк позволяет создавать приложения, используя шаблон проектирования Model-View-Controller [3]. Для построения механизма систем аутентификации, авторизации и для безопасности приложения используется библиотека Spring-Security⁶ [11].

Для сборки проекта используется Maven⁷ [2]. Этот фреймворк позволяет легко управлять зависимостями, декларативно описывать проект.

В качестве СУБД используется MySQL⁸ [4], а для решения задач объектно-реляционного отображения данных используется фреймворк Hibernate⁹ [7].

Одной из основных компонент системы управления и конфигурирования роботов является веб-приложение. В ходе данной работы разработано приложение, позволяющее с помощью пользовательского интерфейса задавать конфигурацию робота, отправлять на него программы. Внешний вид веб-приложения представлен на рис. 3.

Использование веб-приложения позволяет в удобной форме взаимодействовать с роботом, без необходимости к нему напрямую подключаться и править его конфигурацию вручную. А создание программ с помощью графического редактора позволяет разрабатывать программы без кодирования на текстовых языках, что позволяет делать это даже людям, не владеющим программированием. Онлайн среда графического программирования роботов разрабатывается в рамках бакалаврской работы студента 444 группы Захарова Владимира. Для перевода диаграмм в язык, понятный роботам, используется интерпретатор диаграмм, реализованный в рамках курсовой работы студентки 371 группы Лады Гагиной.

Веб-приложение обладает следующим функционалом:

- Регистрация и авторизация пользователей.

⁵Веб-фреймворк Spring-MVC, URL: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html> (дата обращения 17.05.2015)

⁶Веб-фреймворк Spring-Security, URL: <http://projects.spring.io/spring-security> (дата обращения 17.05.2015)

⁷Инструмент сборки Maven, URL: <https://maven.apache.org/> (дата обращения 17.05.2015)

⁸СУБД MySQL, URL: <https://www.mysql.com/> (дата обращения 17.05.2015)

⁹ORM-Фреймворк Hibernate, URL: <http://hibernate.org/> (дата обращения 17.05.2015)

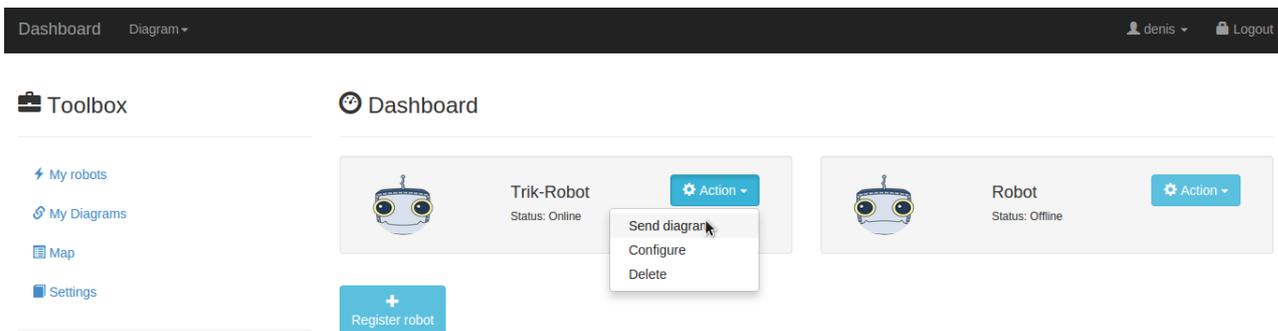


Рис. 3: Веб-приложение, разработанное в рамках данного проекта

- Регистрация и удаление роботов.
- Просмотр роботов, которые подключены к серверу маршрутизации.
- Интерпретация и отображение системной конфигурации робота, которая описывает все его периферийные устройства.
- Конфигурирование периферийных устройств роботов.
- Взаимодействие с сервером маршрутизации:
 - Получение информации о подключенных роботах пользователя системы.
 - Получение информации о системных конфигурациях подключенных роботов пользователя.
 - Отправка конфигурации на заданного робота.
 - Отправка скомпилированной диаграммы на заданного робота.

Создание роботов пользователем

Для работы с сервисом управления и конфигурирования роботов пользователь должен зарегистрироваться в системе. После регистрации пользователю становится доступна возможность регистрации своих роботов, а также создания программ для них с помощью графической среды. Данные о зарегистрированных пользователях и об их проектах хранятся в репозитории на основе системы управления базой данных MySQL. Информация о роботах пользователя также хранится в этой базе данных.

Архитектура приложения

Приложение разработано с использованием шаблона проектирования Model-View-Controller. Такая архитектура предоставляет чёткое разделение логики представления (интерфейса пользователя) и логики приложения. Это добавляет возможность менять компоненты с минимальными изменениями в других.

Конфигурирование робота

От робота на веб-приложение приходит его системная конфигурация, которая описывает, какие устройства у него есть, к каким портам их можно подключать и некоторые параметры запуска. К тому же от робота приходит текущая модель. Эта информация обрабатывается и отображается в удобном для пользователя виде. Пример отображения модели робота показана на рис. 4. Кроме того, пользователь имеет возможность конфигурировать конкретные параметры каждого устройства, подключенного к роботу. Пример конфигурации сервомотора указан на рис. 5.

Отправка программ на робота

При работе с приложением пользователь может создавать программы для робота, используя графическую среду программирования. Со-

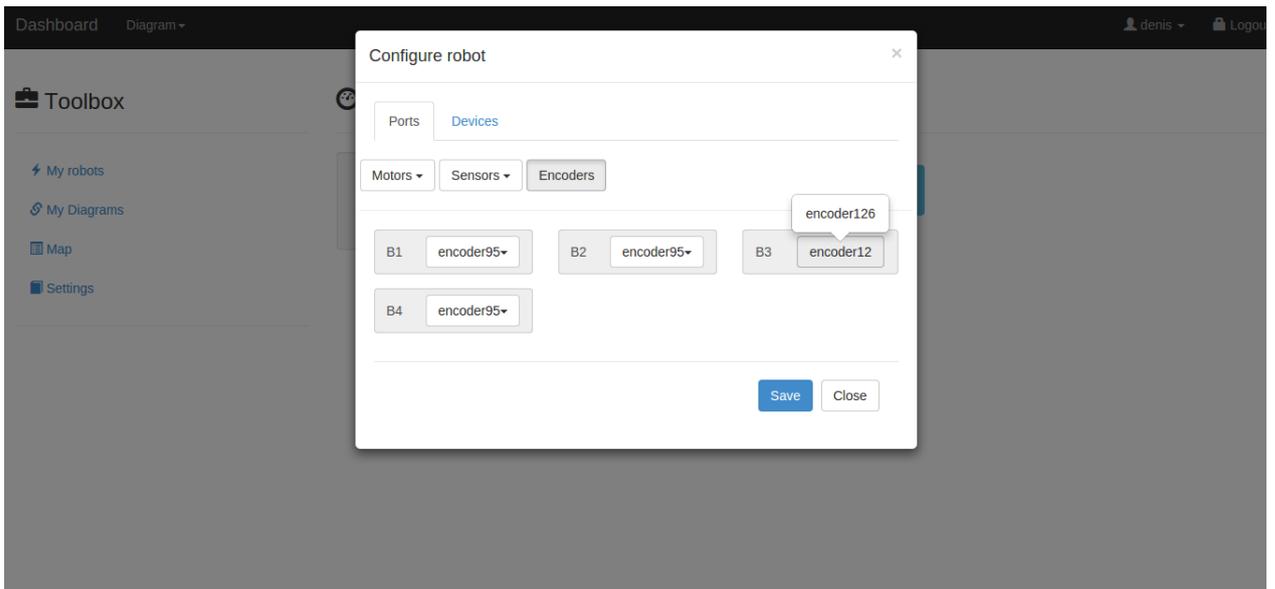


Рис. 4: Конфигурирование портов на работе

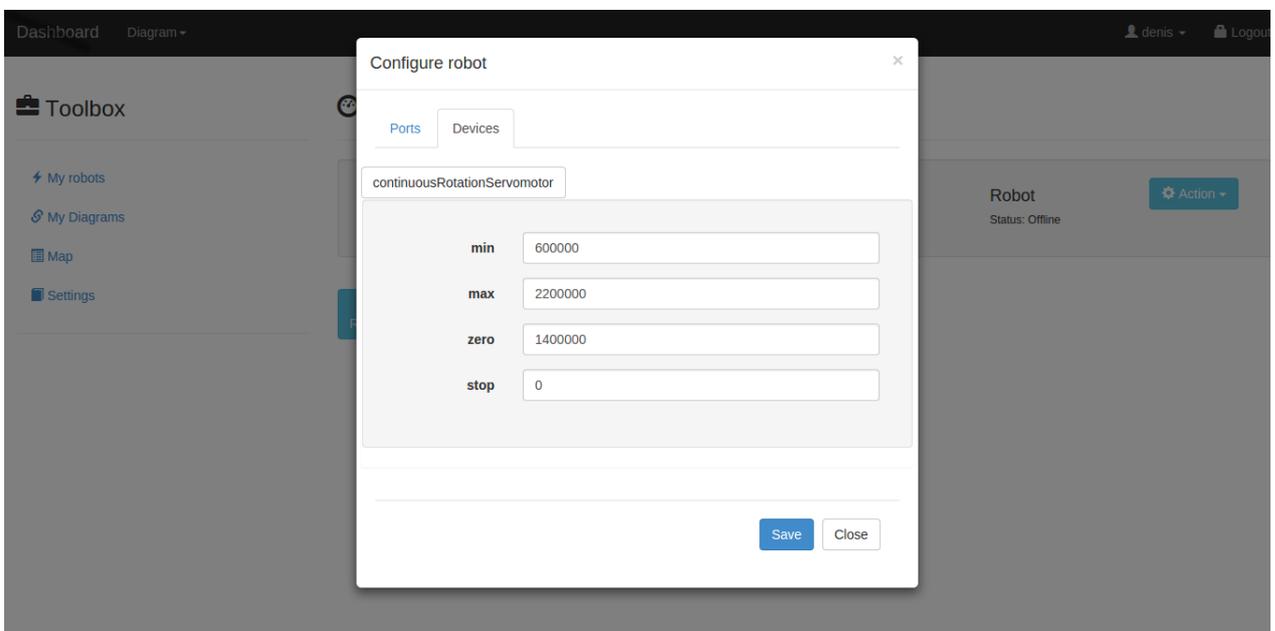


Рис. 5: Конфигурирование сервомотора на работе

храненные в систему диаграммы будут доступны для отправки на робота. На сервере предусмотрена возможность отправки диаграмм, скомпилированных в язык, для которого есть среда исполнения на контроллере. При получении роботом скомпилированной диаграммы, она автоматически запускается на нем.

Взаимодействие с сервером маршрутизации

Для взаимодействия с сервером маршрутизации используется специально разработанный протокол, использующий данные в формате JSON, все это работает поверх REST-сервиса [9].

Протокол общения между компонентами системы

Протокол представляет из себя данные в формате JSON. В этих данных указывается, кто является отправителем — робот или веб-приложение, тип сообщения и необходимая служебная информация.

Пример запроса онлайн роботов пользователя:

```
"from": "WebApp",
"type": "getOnlineRobots",
"userRobots": [ { "ssid": "ssid1" },
                 { "ssid": "ssid2" }
               ]
```

3.2.2. Сервер маршрутизации роботов

Сервер маршрутизации предназначен для обработки подключенных роботов, а также для взаимодействия с веб-сервером. Он хранит информацию о текущих подключениях роботов, определяет, какому роботу нужно отправить сообщение, полученное от веб-приложения. Наличие этого сервера позволяет разделить логику взаимодействия с роботом и логику веб-приложения. Кроме того, он снижает нагрузку с веб-приложения, позволяя взаимодействовать в формате "запрос-ответ", а не держать открытыми подключения через сокет-соединение.

Используемые технологии

В качестве основного языка для создания сервера маршрутизации был выбран Groovy¹⁰[6]. Поскольку Groovy является компилируемым в байт-код, он полностью совместим со всеми Java-библиотеками. Этот язык удобен для сериализации и десериализации типизированных данных, таких как XML или JSON.

Для сборки и запуска сервера используется система сборки Maven.

Архитектура сервера

Сервер предназначен для управления взаимодействием пользователя с роботами. При получении сообщения SocketServer проверяет его на корректность заданному протоколу. Если сообщение удовлетворяет заданным условиям, сервер запускает отдельный процесс RobotConnectionProcessor или WebAppConnectionProcessor с соответствующим обработчиком для каждого типа соединения — работа или веб-приложения.

При обработке соединения от робота в общий пул подключенных роботов (RobotConnectionManager) добавляется информация о текущем подключении. После этого соединение переходит в режим ожидания сообщений от веб-приложения.

При получении сообщения от веб-приложения, оно сразу отправляется на робота, если он находится в сети, при этом соединение продолжает ожидать новые сообщения. Если же соединения с роботом нет, данное сообщение добавляется в очередь сообщений для данного робота. При подключении робота, ему будут доставлены все сообщения из очереди.

Поскольку в сокет-соединении нет возможности узнать состояние подключения робота, ему отправляется с определенной периодичностью заранее заготовленное сообщение. Такой метод проверки соединения называется Heartbeat-mechanism¹¹.

Диаграмма классов описанных компонент представлена на рис. 6.

¹⁰Язык Groovy, URL: <http://www.groovy-lang.org/> (дата обращения 17.05.2015)

¹¹Heartbeat-mechanism, URL: <http://docs.oracle.com/cd/E19206-01/816-4178/6madjde6e/index.html> (дата обращения 17.05.2015)

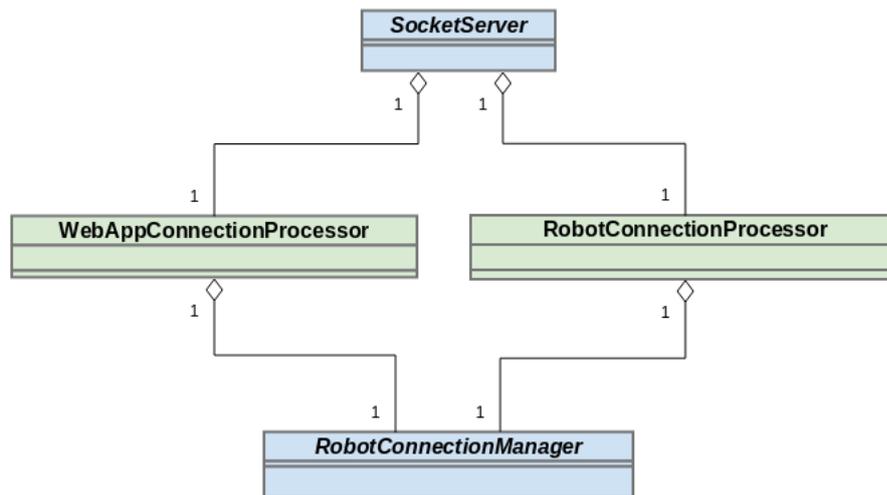


Рис. 6: Архитектура сервера маршрутизации роботов

3.2.3. Контроллер ТРИК

В рамках проекта ТРИК для контроллера был сделан графический пользовательский интерфейс для управления роботом, кроме того, имеется функциональность для запуска на контроллера программ. В связи с этим было решено расширять эту функциональность добавлением своих модулей. Был расширен модуль trikGUI, отвечающий за пользовательский интерфейс и переиспользован модуль, позволяющий запускать программы. К существующему рабочему окружению был добавлен модуль, который отвечает за подключение к серверу маршрутизации, также добавлен модуль, который позволяет обновлять конфигурацию робота. На рис.7 представлена архитектура добавленного в trikRuntime решения.

Используемые технологии

Большая часть рабочего окружения контроллера написана на C++ с активным использованием фреймворка Qt¹². Есть интерфейсы для запуска программ и конфигурирования робота. Поэтому логичным шагом было использование C++/Qt в качестве основного языка программирования для рабочего окружения контроллера ТРИК.

¹²Фреймворк Qt, URL: <http://www.qt.io/> (дата обращения 17.05.2015)

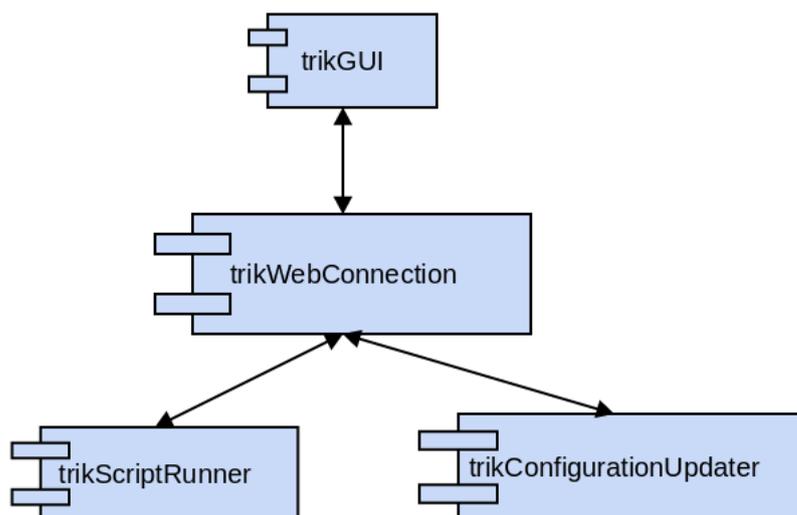


Рис. 7: Компоненты trikRuntime

Для работы с сериализацией и десериализацией данных в формат Json была использована библиотека с открытым исходным кодом qt-json¹³.

Подключение к серверу маршрутизации роботов

Перед подключением к серверу маршрутизации роботов, пользователю нужно на работе в специальном конфигурационном файле указать адрес сервера маршрутизации. Далее пользователю необходимо зарегистрировать робота в системе. Сделать это можно с помощью веб-приложения, указав там уникальный идентификатор робота, который можно посмотреть на нем в соответствующем пункте меню. После регистрации на работе выбрать пункт "Подключиться к серверу". Автоматически сгенерированная информация отправится на сервер, и робот будет ожидать сообщений от веб-сервера.

3.3. Проверка работоспособности системы

Для проверки была собрана робототехническая модель на базе контроллера ТРИК с установленным рабочим окружением, позволяющим подключаться к серверу маршрутизации роботов.

¹³Json библиотека qt-json, URL: <https://github.com/gaudecker/qt-json> (дата обращения 17.05.2015)

Мы зарегистрировались на созданном сайте, также добавили робота. Далее подключились роботом к онлайн-системе. С помощью графической среды программирования создали программу, которую сохранили в системе. После этого программу отправили на робота, который, обработав пришедшее сообщение, запустил полученную программу. Кроме того, была изменена роботехническая модель робота и заново сконфигурирована его модель с помощью данной инфраструктуры.

Данная система была предложена в пользования двум людям, которые при использовании системы высказали несколько замечаний по улучшению пользовательскому интерфейса. Например, группирование портов по типу подключаемых устройств: моторы, сенсоры или видео порты. Данное замечание было учтено и поправлено в следующей итерации разработки веб-приложения.

4. Результаты

В ходе работы было разработано архитектурное решение для системы управления и конфигурирования роботов. Было разработано веб-приложение, которое позволяет в удобной форме удаленно конфигурировать периферийные устройства робота, а также отправлять на него скомпилированные диаграммы, созданные в онлайн среде графического программирования роботов.

Ниже перечислены результаты данной квалификационной работы.

- Разработана общая архитектура системы для управления и конфигурирования роботов.
- Разработана система для управления и конфигурирования роботов.
- Проведена интеграция с онлайн средой графического программирования роботов.
- Проверена работоспособность разработанного решения на робототехнической модели, собранной на базе контроллера ТРИК.

5. Дальнейшее развитие

Созданная в ходе данной работы инфраструктура для конфигурирования и управления роботом может использоваться для дальнейших исследований в рамках проекта ТРИК.

- Внедрение системы в образовательный процесс, это существенно упростит взаимодействие школьников с роботами.
- Дальнейшее развитие инфраструктуры, добавление новой функциональности:
 - Возможность онлайн обновления программного обеспечения робота.
 - Возможность управления роботом сразу несколькими пользователями.

Список литературы

- [1] A.N. Terekhov, R.M. Luchin, S.A. Filippov. Educational Cybernetical Construction Set for Schools and Universities. — 2012. — Vol. 9. — P. 430–435.
- [2] Company Sonatype. Maven: The Definitive Guide. — O'Reilly Media, 2008. — P. 470.
- [3] Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. — Addison-Wesley Professional, 1994. — P. 395.
- [4] DuBois Paul. MySQL. — Addison-Wesley Professional, 2008. — P. 1176.
- [5] Expert Spring MVC and Web Flow (Expert's Voice in Java) / Colin Yates, Seth Ladd, Steven Devijver, Darren Davison. — Apress, 2007.
- [6] Groovy in Action / Dierk Koenig, Andrew Glover, Paul King et al. — Manning Publications, 2007. — P. 700.
- [7] Konda Madhusudhan. Just Hibernate. — O'Reilly Media, 2014. — P. 140.
- [8] Oracle. Java SE documentation. — 2015. — URL: <https://docs.oracle.com/javase/8/> (дата обращения: 20.05.2015).
- [9] Richardson Leonard, Amundsen Mike, Ruby Sam. RESTful Web APIs. — 2013. — P. 406.
- [10] TRIK. Проект TRIK // Всё о ТРИК. — 2015. — URL: <http://trikset.com/> (дата обращения: 20.05.2015).
- [11] Winch Robert, Mularien Peter. Spring Security 3.1. — 2012. — P. 456.

- [12] А.Н. Терехов, Ю.В. Литвинов, Т.А. Брыксин. Среда для обучения информатике и робототехнике QReal:Robots. Девятая независимая научно-практическая конференция «Разработка ПО 2013» (СЕЕ SEC(R)-2013). — Москва, 2013.