

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование
информационных систем

Гуданова Варвара Сергеевна

Реализация возможности взаимодействия с
TRIK роботом с помощью сред разработки
Visual Studio Code и PyCharm

Курсовая работа

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

Санкт-Петербург

2019

Содержание

1. Обзор	5
1.1. Существующие решения	5
1.2. Вывод	5
2. Описание решения	6
2.1. PyCharm	6
2.2. Visual Studio Code	6
2.2.1. Выбор технологий	6
2.2.2. Выбор способа взаимодействия	6
2.2.3. Архитектура	7
3. Тестирование	9
4. Заключение	10
Список используемой литературы	11

Введение

Грамотное знакомство детей школьного возраста с программированием — довольно серьезная задача. Важно предоставить возможность легкого вхождения в данную область и сохранить интерес обучающегося.

Благодаря работе студентов и преподавателей СПбГУ был создан контроллер TRIK и кибернетический конструктор, которые позволяют создавать несложные робототехнические модели. Кроме того, была создана удобная среда разработки TRIK Studio, которая упрощает задачу создания программ для TRIK роботов. Данная среда позиционирует себя как среду для обучения основам программирования и кибернетики. Создание программ происходит с помощью диаграмм, что делает данный процесс интуитивно понятным даже для детей младшего возраста. Однако необходимо предоставить возможность обучающимся развиваться дальше, когда они «вырастают» из TRIK Studio. То есть предоставить возможность обучаться программированию на сложном широко используемом языке в известных средах разработки.

Так как для обучения часто используются такие языки как Python и JavaScript, и данные языки поддерживаются TRIK-контроллером, то было решено в качестве сред разработки выбрать PyCharm и Visual Studio Code. Данные среды в настоящее время являются наиболее широко используемыми для разработки программ на этих языках [1] [2].

В связи с этим была сформулирована цель данного проекта — разработка расширений для Visual Studio Code и PyCharm, предоставляющих возможность взаимодействовать с TRIK роботом непосредственно из этих сред разработки.

Для достижения данных целей были поставлены следующие личные задачи:

- 1) В рамках работы с PyCharm разработать визуальную часть плагина — расширение пользовательского интерфейса данной среды.
- 2) В рамках работы с Visual Studio Code разработать полноценное расши-

рение, предоставляющее возможность

- а) запускать открытую в редакторе программу на работе;
- б) запускать программу на работе по имени;
- в) останавливать выполнение программы;
- г) сохранять открытый в редакторе файл на работе;
- д) проверять состояние подключения.

1. Обзор

1.1. Существующие решения

В настоящий момент подобных решений именно для TRIK роботов не существует. Но реальным конкурентом данного проекта является непосредственно сама TRIK Studio. Однако в рамках поставленной цели данная среда разработки имеет ряд недостатков.

- 1) TRIK Studio не является широко используемой средой разработки, а важной частью работы является желание познакомить обучающихся с популярными продуктами в данной области.
- 2) Так как TRIK Studio предлагает разрабатывать программы с помощью языка диаграмм, то это делает ее неудобной для разработки с использованием сложных текстовых языков.

Помимо TRIK Studio в качестве аналога данного проекта можно рассмотреть похожий плагин для Visual Studio Code, но для LEGO MINDSTORMS EV3. Данное расширение не подходит для взаимодействия с TRIK роботом. Кроме того, для начала работы потребуется установка и настройка, которые достаточно сложны и потребуют некоторого времени [3].

1.2. Вывод

В данный момент не существует решений, которые отвечали бы поставленным требованиям. Достаточно близкие решения имеют значительные недостатки и либо не приспособлены для работы с TRIK роботом, либо не решают важных проблем, которые стали основой данной работы.

2. Описание решения

2.1. PyCharm

В рамках работы с PyCharm было необходимо разработать визуальную часть плагина. Для ее создания использовалась графическая библиотека Swing. При построении пользовательского интерфейса было решено использовать паттерн Model-View-Presenter, который позволяет отделить друг от друга логику, данные и отображение [4].

2.2. Visual Studio Code

2.2.1. Выбор технологий

Важной частью процесса создания проекта является выбор технологий. Расширения для Visual Studio Code разрабатываются с помощью языков TypeScript и JavaScript. Однако в официальной документации по созданию расширений авторы сообщают, что по их мнению лучшим вариантом является использование TypeScript [5]. Именно поэтому он был выбран в качестве языка для данного проекта.

2.2.2. Выбор способа взаимодействия

Для взаимодействия с роботом пользователю предоставляется возможность набирать нужные команды в Command Palette Visual Studio Code. В зависимости от выбранной команды выполняется определенное действие, соответствующее названию команды и, соответственно, ожиданиям пользователя.

Возможные команды:

- Send file to TRIK – отправка файла на робота.
- Run file on TRIK – запуск открытой в редакторе программы на работе.
- Check if robot is active – проверка активности робота.
- Stop execution on TRIK – остановка выполнения программы на работе.

Такой выбор способа взаимодействия был следствием нескольких причин.

В разработке расширений для Visual Studio Code есть некоторые ограничения. Разработчики не могут создавать расширения с добавлением HTML-элементов к пользовательскому интерфейсу VS Code [6].

Кроме того, одной из задач является возможность познакомить пользователя со средой разработки. А создание новых интерфейсных элементов отвлекает пользователя от среды, создает новое рабочее пространство. Выбранный же метод способствует более глубокому изучению возможностей редактора Visual Studio Code.

2.2.3. Архитектура

На основе принятых решений была создана следующая архитектура проекта (рис. 1).

Таким образом можно выделить несколько основных модулей проекта:

- Extension – самый обширный модуль, отвечает за работу самого расширения, регистрирует все необходимые команды.
- Connection – отвечает за подключение к роботу.
- Command – отвечает за корректное общение с роботом в соответствии с протоколом.

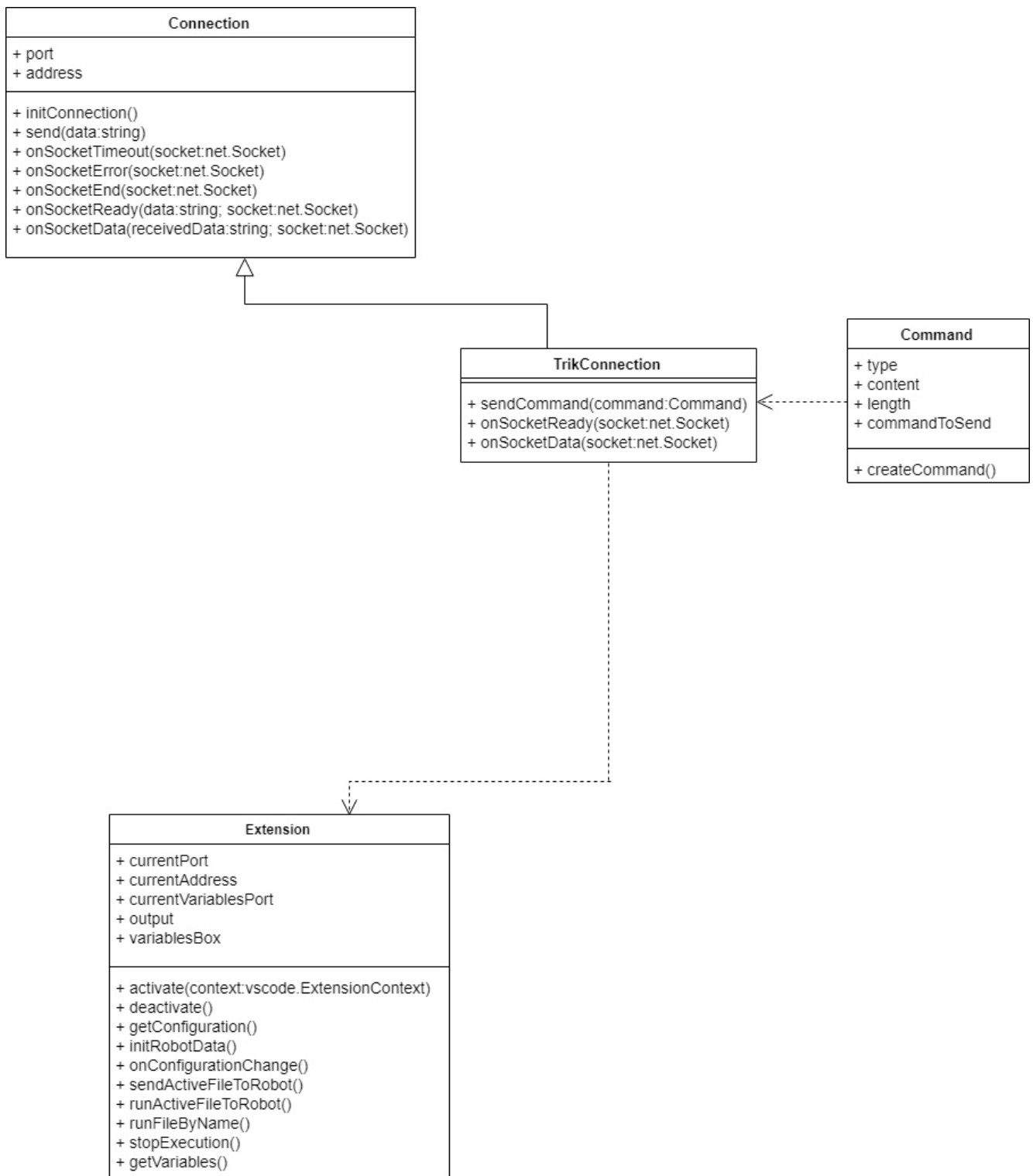


Рис. 1. Архитектура расширения для Visual Studio Code.

3. Тестирование

Тестирование функциональных возможностей проводилось вручную. Весь процесс можно разделить на два этапа:

- 1) Тестирование возможностей с помощью TRIK Runtime — прошивки робота — на той же машине, что и тестируемый проект.
- 2) Тестирование возможностей с использованием реального робота. Подключение производилось по сети Wi-Fi.

В обоих случаях результаты тестирования совпали с ожидаемыми и показали работоспособность созданного решения.

4. Заключение

В результате проделанной работы были получены следующие личные результаты:

- 1) В рамках работы с PyCharm была разработана визуальная часть плагина.
- 2) В рамках работы с Visual Studio Code были реализованы все основные заявленные возможности:
 - а) запуск открытой в редакторе программы на работе;
 - б) запуск программы на работе по имени;
 - в) остановка выполнения программы;
 - г) сохранение открытого в редакторе файла на работе;
 - д) проверка состояния подключения.

Список используемой литературы

[1] Best Python IDE or editors.

<https://www.slant.co/topics/366/~best-python-ides-or-editors>

(дата обращения: 01.06.2019)

[2] Best JavaScript IDE or editors.

<https://www.slant.co/topics/1686/~best-javascript-ides-or-editors>

(дата обращения: 01.06.2019)

[3] EV3 extension for Microsoft Visual Studio Code.

<https://sites.google.com/site/ev3python/setting-up-vs-code>

(дата обращения: 01.06.2019)

[4] Model View Presenter In 5 Minutes.

[https://www.dreamincode.net/forums/topic/](https://www.dreamincode.net/forums/topic/353210-swing-passive-model-view-presenter-in-5-minutes/)

[353210-swing-passive-model-view-presenter-in-5-minutes/](https://www.dreamincode.net/forums/topic/353210-swing-passive-model-view-presenter-in-5-minutes/)

(дата обращения: 01.06.2019)

[5] Visual Studio Code. TypeScript.

[https://code.visualstudio.com/api/get-started/](https://code.visualstudio.com/api/get-started/your-first-extension)

[your-first-extension](https://code.visualstudio.com/api/get-started/your-first-extension)

(дата обращения: 01.06.2019)

[6] Visual Studio Code. Extension Capabilities.

<https://code.visualstudio.com/api/extension-capabilities/overview>

(дата обращения: 01.06.2019)

[7] TypeScript in 5 minutes.

[https://www.typescriptlang.org/docs/handbook/](https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html)

[typescript-in-5-minutes.html](https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html)

(дата обращения: 01.06.2019)

[8] Visual Studio Code Extensions.

<https://code.visualstudio.com/api>

(дата обращения: 01.06.2019)