

# **Интеграция поддержки Python в существующее C++ приложение**

Белков Р., СПбГУ, [roman.belkov@gmail.com](mailto:roman.belkov@gmail.com)

Костюков Ю., СПбГУ, [kostyukov.yurii@gmail.com](mailto:kostyukov.yurii@gmail.com)

## **Аннотация**

В докладе описывается опыт интеграции текстового программирования на языке Python в ПО контроллера ТРИК. Рассмотрены возможности, предоставляемые фреймворками для переиспользования C++ кода из языка Python. Данный опыт может быть использован при создании обёрток на языке Python над существующим C++ кодом.

## **Введение**

Значительное место в начале процесса обучения основам программирования занимает визуальное программирование, так как оно является наиболее понятным и доступным способом познакомиться с абстракциями выполнения программ. Визуальные программы, созданные учащимися, могут также помогать им на последующих этапах обучения. Например, при помощи генераторов визуальных диаграмм в код на текстовом языке возможно совершить плавный переход от визуального программирования к текстовому программированию, давая ученикам понять, как конструкции из диаграммы представляются на текстовом языке программирования.

## **Постановка задачи**

Для набирающего популярность кибернетического контроллера ТРИК [1] и одноимённого конструктора потребовалось создать возможность использования языка Python для программирования роботов. Поскольку Python является одним из наиболее популярных языков для обучения основам программирования [2] [3], то разработчикам контроллера ТРИК с открытым исходным кодом постоянно поступали просьбы и вопросы о поддержке языка Python. Тогда школьники, имеющие навыки работы с Python на персональном компьютере, смогут без проблем попробовать свои силы в программировании роботов. Поскольку для контроллера ТРИК уже существовал основной фреймворк trikRuntime,

написанный с использованием C++ и Qt, то требовалось переиспользовать существующий фреймворк для снижения затрат на поддержку и дальнейшую разработку.

## Python

В начале работы надо было определиться, какую из многочисленных реализаций языка Python стоит использовать. Рассматривались самые популярные и поддерживаемые реализации: PyQt, PySide, PythonQt, Boost.Python, SWIG. Основными критериями оценки фреймворков стали следующие:

- Возможность поддержать исполнение Python из C++ приложения.
- Поддержка интеграции с Qt.
- Активная поддержка фреймворка компанией или сообществом.
- Доступная и полная документация.
- Лицензия со свободными условиями.

Дистрибутив на основе GNU/Linux для контроллера ТРИК собирается при помощи Yocto/OpenEmbedded [4], как и для многих других контроллеров. К сожалению, на момент исследования отсутствовали готовые рецепты для сборки любого из рассматриваемых фреймворков, поэтому этот критерий не рассматривался.

	PyQt	PySide	PythonQt	SWIG	Boost.Python
встраивание Python в C++	-	-	+	-	+
интеграция с Qt	+	+	+	-	-
поддерживается	+	-	-	+	+
документация	±	±	±	±	±
обёртка C++ для Python	+	+	-	+	+
лицензия	GPL	LGPL	LGPL	GPL	Boost.License

Таблица 1: Сравнение фреймворков

По результатам обзора, приведённом на таблице 1, PythonQt или Boost.Python являлись наиболее подходящими фреймворками для решения данной задачи, но оба фреймворка обладали недостатками, для устранения

которых потребовалось бы значительное количество времени: Boost.Python не хватало поддержки Qt и её пришлось реализовать в рамках данной работы, а PythonQt были необходимы патчи к генератору. Однако, реализация поддержки Qt для Boost.Python по нашим оценкам требовала больше времени, чем реализация патчей к генератору PythonQt.

Поэтому для первого этапа был использован PyQt, который позволил в крайне сжатые сроки создать первую версию программного обеспечения для робота, хотя PyQt и не являлся оптимальным фреймворком для решения поставленной задачи.

Переход на PythonQt позволил добиться более естественной интеграции Python наподобие той, что предлагает JavaScript в составе trikRuntime. Так пользователям TRIK, привыкшим работать с JavaScript, не придётся привыкать к новым решениям в контексте использования АПИ периферии робота.

## **Заключение**

В результате работы была реализована и интегрирована в ПО контроллера TRIK поддержка языка Python для создания и последующего исполнения программ на кибернетическом контроллере TRIK. Опыт, описанный в работе, может быть полезен при необходимости переиспользовать уже существующий C++ код из Python-кода.

## **Литература**

1. Главная страница проекта TRIK [http://www.trikset.com/index\\_en.html](http://www.trikset.com/index_en.html)
2. Guo Philip. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. — 2014. — URL: <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-nowthe-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>
3. Limited QS Quacquarelli Symonds. QS Top Universities in Engineering and Technology 2017. — URL: <https://www.topuniversities.com/university-rankings/university-subject-rankings/2017/engineering-technology>.
4. Главная страница проекта OpenEmbedded [http://www.openembedded.org/wiki/Main\\_Page](http://www.openembedded.org/wiki/Main_Page)