

Санкт-Петербургский государственный университет

Программная инженерия  
Кафедра системного программирования

Тюляндин Иван Владимирович

Поддержка компьютерного зрения для  
задач Олимпиады НТИ

Курсовая работа

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

Санкт-Петербург  
2018

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Постановка задачи</b>	<b>4</b>
<b>2. Обзор</b>	<b>5</b>
2.1. TRIK Runtime . . . . .	5
<b>3. Разработка графического кода и алгоритма анализа</b>	<b>7</b>
3.1. Графический код . . . . .	7
3.2. Алгоритм анализа маркера . . . . .	7
<b>4. Архитектура</b>	<b>10</b>
<b>5. Особенности реализации</b>	<b>11</b>
5.1. Имитация фотографирования . . . . .	11
5.2. Выбор реализации . . . . .	11
5.3. Медианный фильтр . . . . .	11
5.4. Ускорение производительности . . . . .	12
<b>Заключение</b>	<b>13</b>
<b>Список литературы</b>	<b>14</b>

# Введение

В рамках Национальной технологической инициативы проводится Всероссийская Инженерная Олимпиада для школьников. Эти соревнования проходят по различным научным профилям, в том числе есть профиль «Интеллектуальные робототехнические системы» (ИРС) [4]. Финал профиля ИРС проводится на базе конструктора ТРИК и одноименного контроллера [5], а предыдущие этапы помогают подготовиться к решению финальной задачи. Участникам необходимо оснастить мобильную наземную робототехническую платформу набором датчиков, настроить получение достоверной информации с них и выполнить основное задание.

В сезоне 2017-2018 организаторы хотели предложить участникам Олимпиады запрограммировать алгоритмы компьютерного зрения, чтобы стимулировать изучение соответствующих областей информатики в школах и кружках. Технически, захват изображения и обработка возможны на контроллере ТРИК, но конечному пользователю предлагалось выбирать из нескольких готовых алгоритмов, что не удовлетворяло требованиям методистов Олимпиады. На заключительном этапе команды, используя камеру и алгоритмы компьютерного зрения, должны будут реализовать считывание роботом графических кодов, определяющих маршрут перемещения.

Для организации Олимпиады нужно было реализовать возможность обработки непосредственно в пользовательском коде на работе изображения, получаемого с камеры, подключенной к контроллеру ТРИК, а также предложить формат маркера, который участники будут декодировать на соревнованиях. ТРИК — это не только контроллер, но и среда 2D моделирования TRIK Studio, которая работает на Windows, GNU/Linux и macOS. Для участников Олимпиады нужна возможность отладки программ в этой среде.

# 1. Постановка задачи

Учитывая требования организаторов Олимпиады были сформулированы следующие задачи:

1. Реализовать функциональность для получения фотографии из пользовательского кода, исполняемого на роботе.
2. Создать формат графического кода для соревнования.
3. Предложить алгоритм анализа изображения (как пример для методистов).
4. Разработать механизм имитации фотографирования для среды моделирования TRIK Studio.

## 2. Обзор

### 2.1. TRIK Runtime

Для контроллера TRIK на основе фреймворка Qt написана библиотека TRIK Runtime. На рисунке 1 представлена диаграмма компонент.

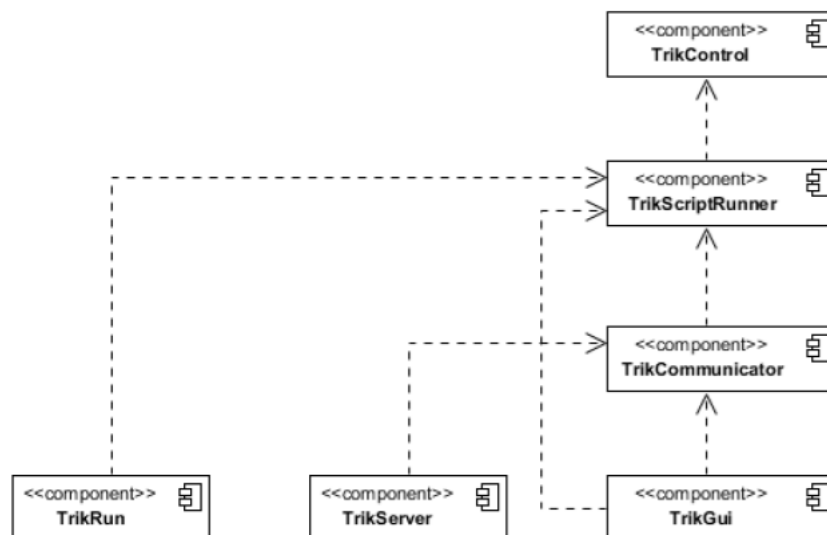


Рис. 1: Диаграмма компонент

На диаграмме прямоугольниками обозначены отдельные собираемые модули, стрелками — кто к кому компонуется. Модуль TrikControl предоставляет API для программирования робота. Модуль TrikScriptRunner — это интерпретатор, который выполняет программы, написанные для TrikControl на скриптовом языке программирования QtScript — это расширение стандарта ECMAScript и языка JavaScript специфичными для фреймворка Qt особенностями, например, поддержка сигналов и слотов. Рассмотрим подробнее архитектуру TrikControl (рис. 2).

Класс Brick представляет контроллер и дает доступ ко всей периферии робота. На данной диаграмме показаны следующие периферийные устройства: сервомотор, моторный двигатель и сенсор. Для расширения списка устройств, с которыми контроллер может взаимодействовать, нужно сделать класс, который будет отвечать за работу определенного периферийного модуля. Инициализация периферии происходит на основе данных из конфигурационного файла, полученных с помощью

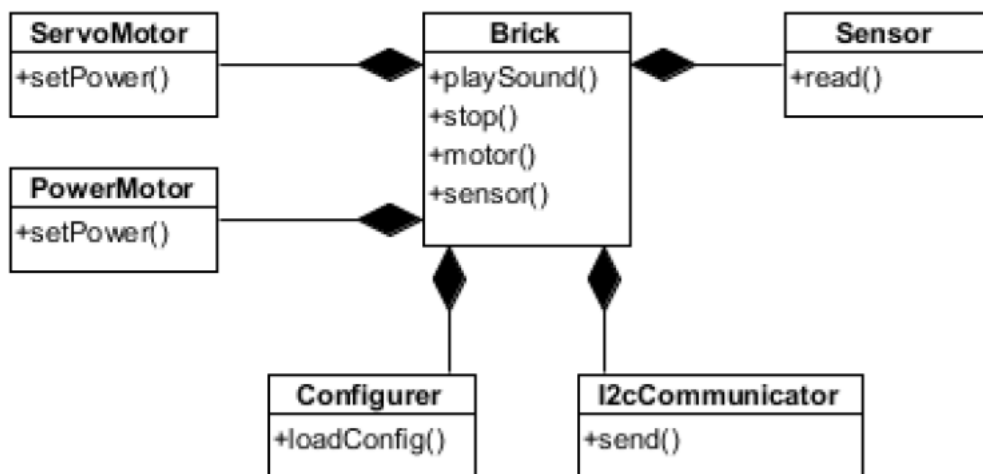


Рис. 2: Архитектура модуля TrikControl

класса Configurer. Подключаемые модули описываются в конфигурационном файле с помощью логического порта — это строка, идентифицирующая периферийное устройство в прикладной программе. Любое устройство, которое можно подключить к контроллеру ТРИК, имеет свой формат логического порта.

## 3. Разработка графического кода и алгоритма анализа

### 3.1. Графический код

В качестве примеров графических кодов были рассмотрены маркеры для дополненной реальности ARTag [1] (рис. 3) и ArUco [3] — это двумерные бинарные штрихкоды с монохромной границей. У ARTag 6х6 кодирующих битов, у ArUco нет ограничения на размер маркера.

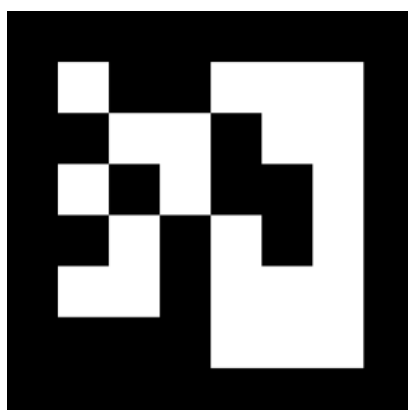


Рис. 3: Маркер ARTag

На основе маркеров ARTag и ArUco был предложен следующий формат маркера: размер 5х5, снаружи граница из чёрных клеток для лучшего распознавания. Такой формат вариативен. Можно использовать все оставшиеся 9 битов для кодирования или оставить заранее определенные биты для ориентации или выделить бит для проверки четности. При необходимости можно увеличить размер маркера для возможности закодировать больше информации. Пример маркера в предложенном формате на рисунке 4.

### 3.2. Алгоритм анализа маркера

После фотографирования пользователю выдается массив пикселей — каждый пиксель кодируется 3 байтами в формате RGB. Размер картинки 320х240. Пример фотографии с ТРИКа на рисунке 5.

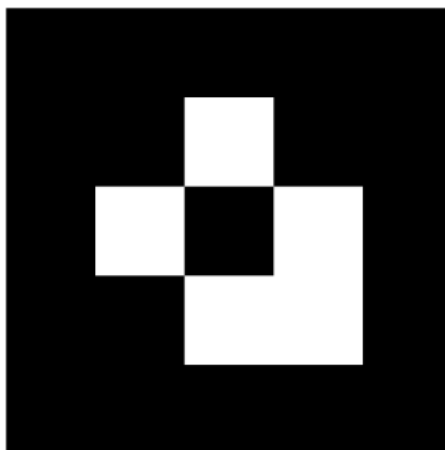


Рис. 4: Пример маркера

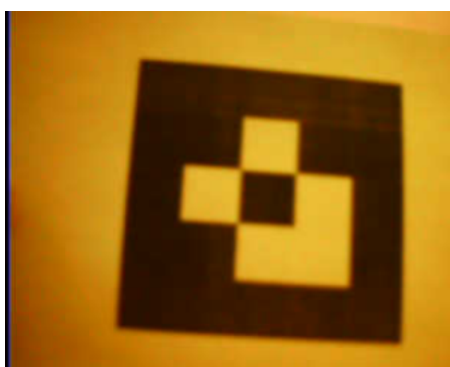


Рис. 5: Пример фотографии с ТРИКа

Первый шаг предложенного алгоритма — бинаризация изображения. Для этого пользователь должен сначала перевести изображение в серый цвет, а затем по интенсивности получить черно-белую картинку. Порог бинаризации можно получить с помощью алгоритма Отсу. Далее нужно найти углы маркера обходом по диагоналям от каждого угла картинке, пока не найдется черный цвет, как показано на рисунке 6. Для того, чтобы избежать ложного распознавания из-за шумов (например, слева на картинке есть черная полоска, которую алгоритм может ошибочно определить как угол), вычисляется плотность пикселей, находящихся рядом с этой точкой. Например, в случае с верхним левым углом верно следующее предположение: “если взять правый нижний квадрат от предполагаемой точки, то он должен быть почти черным”.

Аналогично производятся вычисления и для других углов (разница лишь в том, по каким диагоналям обходить и для какого квадрата вы-



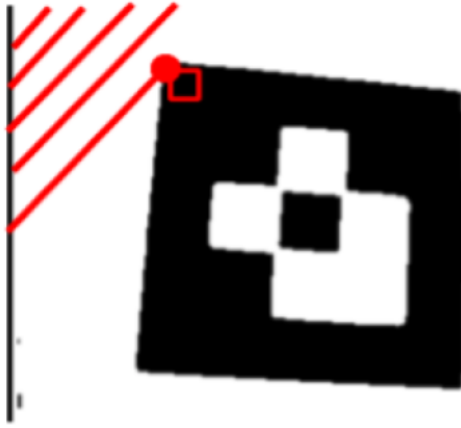


Рис. 6: Поиск углов маркера

числять плотность). Вычислив координаты углов и зная размер маркера, можно построить предполагаемую решетку маркера. Для этого стороны маркера пропорционально делятся в соответствии с его размером, после чего находятся все узлы. Зная решетку для маркера, можно определить сам маркер. Для этого для каждой клетки берем плотность ее центральных пикселей. Если плотность ближе к 0 — белый цвет, ближе к 1 — черный.

## 4. Архитектура

На диаграмме (рис. 7) показаны внесенные изменения класса Brick для реализации фотографирования.

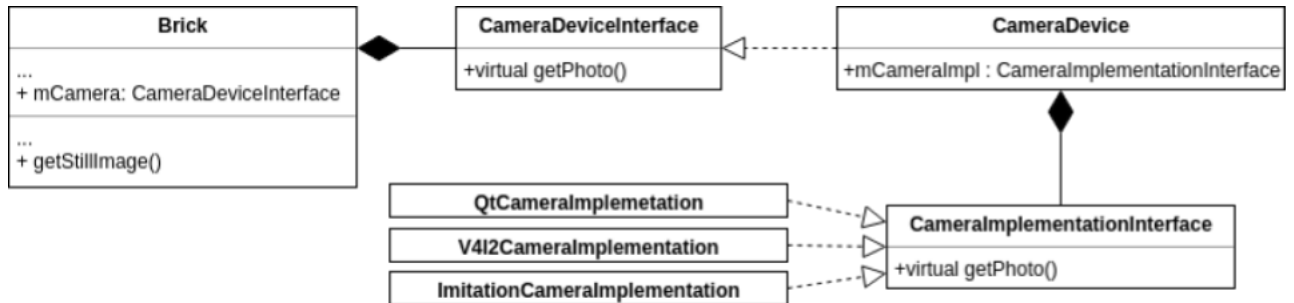


Рис. 7: Архитектура

Сделаны различные реализации работы камеры:

1. QtCameraImplementation — реализация средствами Qt.
2. V4l2CameraImplementation — реализация через API библиотеки видео-захвата v4l2 в ОС GNU/Linux [2].
3. ImitationCameraImplementation — механизм имитации фотографирования (необходим для отладки в 2D среде моделирования).

При проектировании были использованы паттерны «фабрика», «прокси» и «стратегия». Такой подход позволяет пользователю выбрать реализацию, которая ему необходима, через конфигурационный файл. Первоначально предполагалось, что будут только две реализации — с помощью Qt и имитация камеры, но из-за особенностей прошивки контроллера ТРИК реализация через Qt не работала на работе, поэтому была добавлена реализация средствами v4l2. Применение паттернов упростило задачу добавления новой реализации в архитектуру.

## 5. Особенности реализации

### 5.1. Имитация фотографирования

Для возможности отладки на Олимпиаде в 2D среде моделирования была реализована имитация фотографирования: пользователь делает несколько фотографий реальных маркеров и затем использует их в имитационной модели. При вызове функции фотографирования возвращается одно из этих изображений. Изображения выдаются циклически. В конфигурационном файле пользователь указывает путь к папке и регулярное выражение, по которому отбираются файлы, которые будут использоваться в качестве изображения с камеры.

### 5.2. Выбор реализации

Среда 2D моделирования работает на разных операционных системах: Windows, GNU/Linux и macOS. На данный момент среди указанных реализаций на всех операционных системах и на роботе может выполняться только имитация фотографирования. Реализация через Qt не работает на роботе. Реализация с использованием v4l2 осуществима только на ОС GNU/Linux. Пользователь по ошибке может выбрать неподходящую реализацию для конкретной платформы, поэтому при конструировании класс Brick проверяет соответствие платформы и запрашиваемой реализации. Если указанная реализация не подходит, то об этом сообщается пользователю и создается механизм имитации фотографирования.

### 5.3. Медианный фильтр

Так как компьютерное зрение только появляется в задачах Олимпиады, то необходимо обеспечить качественное изображение с низким уровнем шума. После ряда экспериментов было решено использовать медианный фильтр, а пользователь получает уже очищенное от шумов изображение.

## 5.4. Ускорение производительности

При проверке работоспособности фотографирования на контроллере было замечено, что получение изображения занимало до 20 секунд, что очень долго для соревнований, по истечении такого времени пользователь может подумать, что его программа зациклилась. При запуске аналогичного кода вне среды исполнения, фотографирование занимало около 1 секунды. После анализа кода было замечено, что дольше всего работает конвертация изображения из среды исполнения TrikControl в тип, доступный в пользовательском коде. Так как это функция, реализованная в фреймворке Qt, то нужно посмотреть в исходный код и понять причину долгой работы. Выяснилось, что с типом QList (аналог `std::list` языка C++ в фреймворке Qt) конвертация работает неэффективно из-за множественного выделения, перемещения и освобождения памяти. Был найден тип QVariantList, для которого такой проблемы не было. После замены типа выполнение фотографирования на работе занимало 1-1,5 секунды.

# Заключение

Итоги работы:

1. Сделана возможность получения изображения из пользовательского кода.
2. Создан формат графического кода.
3. Разработан алгоритм анализа предложенного графического кода.
4. Реализован механизм имитации фотографирования для работы в 2D среде моделирования.

Основные тезисы этой работы были опубликованы в сборнике конференции «Современные технологии в теории и практике программирования», а за доклад получено III место.

С 23 по 28 февраля 2018 года в г. Сочи успешно прошёл финал Всероссийской Инженерной Олимпиады Национальной технологической инициативы. Используя камеру и алгоритмы компьютерного зрения, команды реализовывали считывание роботом графических кодов, определяющих маршрут перемещения. Реализованная функциональность теперь доступна всем пользователям ПО ТРИК.

## Список литературы

- [1] Fiala Mark. ARTag, a fiducial marker system using digital techniques.— URL: <https://inside.mines.edu/~whoff/courses/EENG512/lectures/other/ARTag.pdf>.
- [2] LinuxTV.org, What is V4L or DVB.— URL: [https://linuxtv.org/wiki/index.php/What\\_is\\_V4L\\_or\\_DVB%3F](https://linuxtv.org/wiki/index.php/What_is_V4L_or_DVB%3F) (дата обращения: 24.03.2018).
- [3] S. Garrido-Jurado R. Muñoz-Salinas F.J. Madrid-Cuevas M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion.— URL: <https://pdfs.semanticscholar.org/2d90/537b709ae92e07856b93aab70614ac5af561.pdf>.
- [4] Профиль «Интеллектуальные робототехнические системы» Олимпиады Национальной технологической инициативы.— URL: <http://nti-contest.ru/profiles/irs/> (дата обращения: 24.03.2018).
- [5] Сайт производителя ТРИК.— URL: [http://www.trikset.com/index\\_en.html](http://www.trikset.com/index_en.html) (дата обращения: 24.03.2018).