

Санкт-Петербургский Государственный Университет
Кафедра системного программирования

Сергеев Евгений Дмитриевич

Реализация сетевой инфраструктуры стенда
«Танчики Роботов»

Курсовая работа

Научный руководитель:
к.т.н., доцент каф. СП, Литвинов Ю.В.

Санкт-Петербург

2017

Оглавление

Введение	5
1. Постановка задачи	7
2. Обзор	8
2.1. ТРИК	8
2.2. ROS	10
2.3. SSL-Vision	11
3. Реализация	12
3.1. Реализация сборки пользовательских узлов ROS средствами SDK для контроллера TRIK	12
3.2. Написание плагина к SSL-Vision	14
3.3. Интеграция SSL-Vision с ROS	15
3.4. Документация и примеры узлов ROS	16
4. Полученные результаты	17
5. Список литературы	18

Введение

Робототехника является активно развивающейся областью науки и с каждым годом находит все более широкое практическое применение в различных областях, таких как медицина, военное дело, различные области промышленности, быт.

К робототехнике проявляют большой интерес как опытные инженеры и программисты, так и ученики школ. Для учащихся это возможность приобщиться к перспективной области, а также наглядно обучиться программированию и алгоритмизации. Важной задачей является обучение и подготовка квалифицированных специалистов, что осуществляется, в том числе, внедрением робототехники в образовательный процесс в школах [1, 2].

Для того, чтобы сделать образовательный процесс интереснее, в сфере робототехники организуются различные соревнования. Одной из наиболее распространенных дисциплин является робофутбол¹. Однако многие дисциплины, включая робофутбол, требуют значительной подготовки и плохо пригодны для обучения школьников. Поэтому было решено создать новую соревновательную дисциплину “Танчики роботов”, где две команды роботов, находясь на поле с препятствиями, пытались бы условно “уничтожить” команду противников, поразив датчик вражеского робота лазерным лучом. Чтобы сделать возможным участие детей школьного возраста в таких соревнованиях, потребовалось разработать программную платформу, которая бы выполняла сложные задачи низкоуровневого управления аппаратной частью робота, компьютерного

¹ RoboCup Small Size League, URL: <http://www.robocup.org/leagues/7> (дата обращения: 13.05.2017)

зрения и сетевой коммуникации команды и предоставляла интерфейс для реализации высокоуровневых алгоритмов.

В качестве основы для реализации был взят существующий робототехнический стенд для проведения соревнований по робофутболу и кибернетический конструктор ТРИК².

²Блог ТРИК, URL: <http://blog.trikset.com> (дата обращения: 13.05.2017)

1. Постановка задачи

В рамках работы были поставлены следующие задачи:

- выбрать платформу связующего программного обеспечения и интегрировать ее с робототехническим стендом;
- реализовать слой сетевого взаимодействия роботов внутри команды и роботов со стендом;
- модифицировать использующуюся на стенде программную платформу для использования в новой дисциплине.

2. Обзор

2.1. ТРИК

ТРИК — это кибернетический конструктор, который позволяет собирать множество различных робототехнических моделей из составных частей. На плате контроллера ТРИК находится процессор OMAP-L138³, который объединяет процессор общего назначения ARM и цифровой сигнальный процессор (DSP). Контроллер ТРИК работает под управлением операционной системы на основе ядра Linux, которая собирается с использованием фреймворка OpenEmbedded⁴.

Для сборки прошивки контроллера используется инструмент BitBake⁵. BitBake — это система сборки, используемая для компиляции встраиваемой ОС Linux. Она считывает рецепты (наборы инструкций) и выполняет их — скачивает необходимые пакеты, компилирует их и создает результирующие загрузочные образы

Управление контроллером и предоставление доступа к датчикам робота осуществляется при помощи набора библиотек trikRuntime⁶.

2.2. ROS

В качестве связующего программного обеспечения (middleware) была выбрана система ROS [3], так как она поддерживается роботами ТРИК

³ OMAP-L138 C6000 DSP+ARM Processor, URL: <http://www.ti.com/product/omap-l138> (дата обращения: 13.05.2017)

⁴ Система сборки для встраиваемой ОС Linux, URL: http://www.openembedded.org/wiki/Main_Page (дата обращения: 13.05.2017)

⁵ Система сборки для компиляции встраиваемой ОС Linux, URL: <https://www.yoctoproject.org/docs/1.6/bitbake-user-manual/bitbake-user-manual.html> (дата обращения: 13.05.2017)

⁶ Домашняя страница TRIK Runtime, URL: <https://github.com/trikset/trikRuntime> (дата обращения: 13.05.2017)

(интеграция робототехнической ОС (ROS) с кибернетическим контроллером ТРИК была выполнена в работе Евгения Новожилова [4], интеграция робототехнических библиотек ROS с контроллером ТРИК была выполнена в работе Евгения Аудучинок [5]).

ROS (Robot Operating System) — это фреймворк для программирования роботов, предоставляющий функциональность для распределённой работы, имеющий множество готовых компонент.

С точки зрения архитектуры ROS представляет собой распределённую вычислительную систему, состоящую из множества независимых узлов, которые могут быть запущены на различных физических устройствах под управлением разных операционных систем, обменивающихся сообщениями с использованием TCP или UDP, используя модель издатель-подписчик, путем публикации сообщений и подписки на них (топики), либо путем удаленных вызовов (сервисы).

С точки зрения функциональности фреймворк ROS состоит из двух частей:

- 1) пакеты, которые предоставляют возможности для взаимодействия с другими узлами;
- 2) пакеты, реализующие различные алгоритмы и использующие данные, производимые другими пакетами.

Связь между узлами сети осуществляется с помощью мастер-узла, который регистрирует запущенные узлы и предоставляет информацию о топиках и сервисах. Перед началом публикации сообщений в канал каждый узел сообщает об этом мастер-узлу, для получения сообщений узлы подписываются на каналы. Узлы используют мастер-узел только для

того, чтобы узнать адреса друг друга, обмениваются сообщениями они друг с другом напрямую.

2.3. SSL-Vision

SSL-Vision[6] (The Shared Vision System for the RoboCup Small Size League) является системой компьютерного зрения, разделяемой несколькими командами роботов. Она состоит из одной или нескольких камер, расположенных над игровым полем, которые соединены с сервером. SSL-Vision обрабатывает в режиме реального времени изображения, снимаемые камерами: она определяет местоположения роботов, а затем передает каждому роботу координаты других членов его команды. Определение команды робота происходит на основе его "шапки" — разноцветного изображения, расположенного сверху на роботе, уникального для команды в рамках данной сессии.

SSL-Vision имеет модульную структуру, которая позволяет добавлять новые плагины и гибко конфигурировать систему под нужды конкретного робототехнического стенда.

3. Реализация

3.1. Реализация сборки пользовательских узлов ROS средствами SDK для контроллера TRIK

В рамках работы была модифицирована система сборки пользовательских ROS узлов. Ранее пользовательские узлы собирались как часть прошивки с помощью фреймворка BitBake. Для того, чтобы собрать узел, было необходимо скачать репозиторий с рецептами сборки⁷, добавить новый рецепт для сборки своего узла и произвести полную сборку прошивки, которая длилась несколько часов, в зависимости от вычислительной мощности компьютера, на котором она осуществлялась. Репозиторий с собранной прошивкой занимает более 30 Гб места на жестком диске. Был написан файл инструментария для кросс-компиляции для системы сборки cmake⁸, в котором указаны пути до необходимых разделяемых библиотек и пакетов в SDK для контроллера ТРИК, с помощью которого возможно собирать узлы при помощи SDK посредством утилиты catkin_make⁹ без полной сборки прошивки.

⁷ Helper scripts for setting development environment, URL:
<https://github.com/trikset/setup-scripts> (дата обращения: 13.05.2017)

⁸ CMake скрипт, который устанавливает переменные, необходимые для кросс-компиляции, URL:
https://cmake.org/Wiki/CMake_Cross_Compiling#The_toolchain_file (дата обращения: 19.05.2017)

⁹ Инструмент сборки пакетов ROS, URL:
http://wiki.ros.org/catkin/commands/catkin_make (дата обращения: 19.05.2017)

3.2. Написание плагина к SSL-Vision

В рамках работы был написан плагин для SSL-Vision, который позволяет судьям отмечать на статическом изображении неподвижные объекты (стены, препятствия, лабиринт и т.д.) в качестве списка пар точек, а затем посредством TCP отправлять местоположения этих объектов всем роботам в системе ROS. Также возможно отмечать точки на карте, что может быть полезно для отработки командных алгоритмов.

Классы статических объектов, такие как стена (Wall) и точка (Point), наследуются от одного абстрактного класса `StaticObjectInterface`, и все остальные классы взаимодействуют со статическими объектами через интерфейс этого абстрактного класса. Поэтому при добавлении возможности отмечать объекты новых типов не придется изменять основную логику плагина.

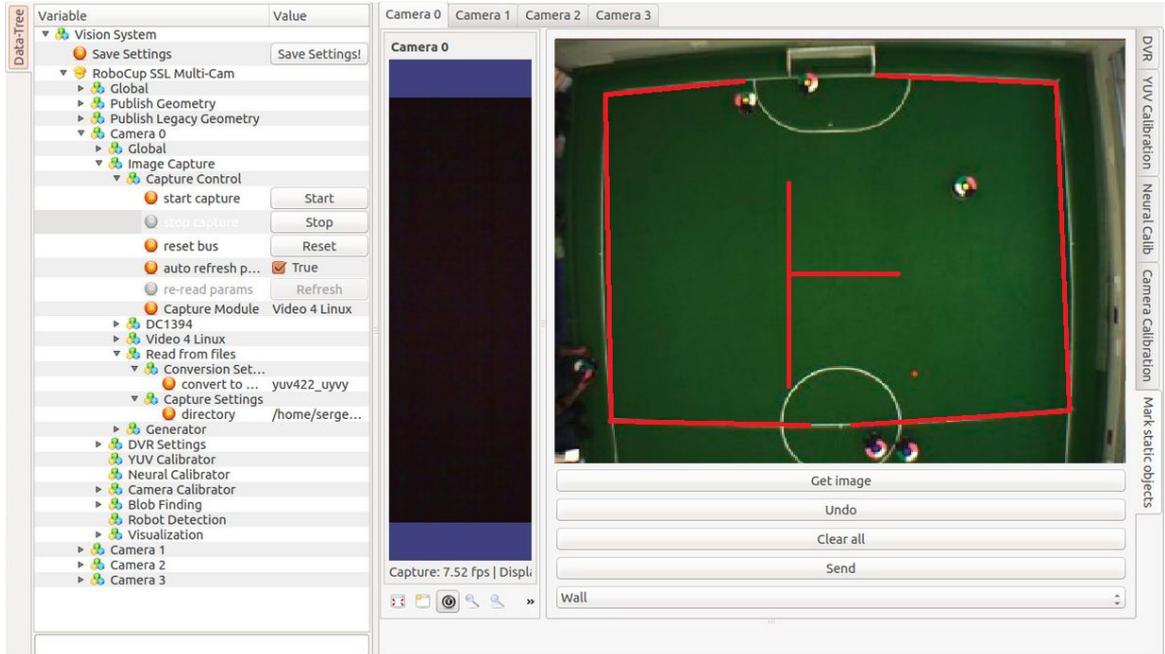


Рис. 1: Графический интерфейс плагина к SSL-Vision

3.3. Интеграция SSL-Vision с ROS

В системе ROS на стенде находятся команды роботов, сервер с SSL-Vision и, при необходимости, компьютеры судей. Роботы обмениваются друг с другом сообщениями внутри своих команд и получают сообщения от SSL-Vision с координатами других роботов своей команды.

SSL-Vision, а также вышеупомянутый плагин, работают как три различных узла в системе ROS. Первый узел, `ssl_server`, запускается на компьютере, на котором работает SSL-Vision. Он считывает данные, отсылаемые SSL-Vision на настроенный multicast адрес на указанный порт, и публикует их в топик с названием `ssl_data`. Второй узел, `staticObjectsServer`, запускается на том же компьютере, что и первый узел. Второй узел, по нажатию пользователем кнопки «Send» из графического интерфейса SSL-Vision, публикует в топик `staticObjects` координаты всех отмеченных объектов как структуру, включающую в себя список точек и список отрезков. Третий узел запускается на всех участвующих в соревновании роботах и компьютерах судей. Этот узел подписывается на топики `ssl_data` и `staticObjects` и получает все публикуемые системой SSL-Vision данные. Все данные отправляются сериализованными в формате `protobuf`¹⁰.

3.4. Документация и примеры узлов ROS

Так как соревновательная дисциплина ориентирована в том числе и на учащиеся школ средних и старших классов, важным моментом при

¹⁰Протокол сериализации и передачи структурированных данных, URL: <https://developers.google.com/protocol-buffers/> (дата обращения: 13.05.2017)

создании программной платформы было написание документации и создание наглядных примеров. Так, была написана подробная инструкция по сборке пользовательских узлов ROS¹¹ и выложен пример узла телеметрии¹², который, используя библиотеку `trikRuntime`, считывает данные с датчиков робота и публикует их в топик `telemetry`.

¹¹ Документация по сборке пользовательского узла ROS, URL: <https://github.com/yurii-litvinov/TrikABCL/wiki/How-to-create-a-custom-ROS-node-and-run-it-on-TRIK> (дата обращения: 13.05.2017)

¹² Пример написания пользовательского узла ROS, URL: <https://github.com/yurii-litvinov/TrikABCL/tree/telemetry> (дата обращения: 13.05.2017)

4. Полученные результаты

В рамках данной работы были получены следующие результаты.

1. Поддержана возможность сборки пользовательских узлов ROS средствами SDK для контроллера TRIK.
2. Произведена интеграция SSL-Vision с ROS.
3. Модифицирован SSL-Vision, добавлена возможность отмечать статические объекты и отправлять информацию о них роботам.
4. Результаты работы были представлены на конференции СПИСОК-2017.
5. Исходные коды проекта и документация выложены в свободный доступ^{13,14}.

¹³ Advanced Behaviors and Communication Library for TRIK robots, URL: <https://github.com/yurii-litvinov/TrikABCL> (дата обращения 13.05.2017)

¹⁴ Shared Vision System For The RoboCup Small Size League, URL: <https://github.com/SergeevED/ssl-vision> (дата обращения 13.05.2017)

5. Список литературы

- [1] Филиппов С.А. Уроки робототехники. Конструкция. Движение. Управление. — Лаборатория знаний, 2017. — 176 С.
- [2] М.М. Киселев, М.М, Киселев. Робототехника в примерах и задачах. Курс программирования механизмов и роботов. — Солон-Пресс, 2017. — 136 С.
- [3] M. Quigley et al. ROS: An open-source robot operating system. // in Proc. ICRA Open-Source Softw. Workshop. — 2009. — pp. 1-6.
- [4] Новожилов Евгений. Интеграция робототехнической ОС (ROS) с кибернетическим контроллером ТРИК — 2015. — URL: <http://se.math.spbu.ru/SE/diploma/2015/s/544-Novozhilov-report.pdf> (дата обращения: 13.05.2017).
- [5] Аудучинок Евгений. Интеграция робототехнических библиотек ROS с контроллером ТРИК — 2016. — URL: http://se.math.spbu.ru/SE/YearlyProjects/spring-2016/371/Auduchinok_report.pdf (дата обращения: 13.05.2017).
- [6] S. Zickler et al. SSL-vision: The shared vision system for the RoboCup Small Size League. // Robo Cup 2009: Robot Soccer World Cup XIII. — 2010. — pp. 425-436.