

ROBOT OPERATING SYSTEM

Alexei CAZIUC

Департамент компьютерных наук и системной инженерии, CRI-231M, факультет вычислительной техники, информатики и микроэлектроники, Технический университет Молдовы, Кишинев, Республика Молдова

Автор корреспонденции: Alexei CAZIUC, e-mail: caziuc.alexei@iis.utm.md

Îndrumătorul/coordonatorul științific **BRAGARENCO Andrei**,

Резюме. Robot Operating System (ROS) это платформа с множеством возможностей, связанных как программной так и физически. В ROS находятся компоненты для разработки 3D модель, движок для разработки физики модели, сервер-клиент для взаимосвязи программной и физически для взаимодействия и анализа данных удалённо способ реализации и взаимодействия неограничен. Способы добавит как III и другие дополнительный софт Работа с такими языками программирования как C++ и Python. Анализ 3D модель происходит от STL файлов и созданного пространства и передачи данных от датчиков есть возможность реализовать без подключения STL файлов. Данные получаемые операционной системой образуются в NODE, TOPIC, SERVIC,PARAMETERS и ACTION-это основная часть инструментов связи Подписчика и Редактора. В реализации проекта есть и механические части где можно написать собственную библиотеку и собственный драйвер для отдельных электронных компонентов. Условное выполнение пространства использует данные из датчиков и переводит в модель на карте представленного пространства и общий анализ окружения как и представление его на карте пространства для взаимодействия которого и был представлен ROS.

Ключевые слова: Robot Operating System, Драйвер, 3D модель, Сервер-Клиент(подписчик-редактор)

Введение

Robot Operating System (ROS) - это, по сути, промежуточное программное обеспечение, которое обеспечивает эффективную связь между аппаратным и программным обеспечением. Оно позволяет повторно использовать код для робототехники. На самом деле ROS не является операционной системой. ROS 1 работает на Linux (debian или ubuntu), а в ROS 2 добавлена поддержка различных других операционных систем (macOS, Windows, RTOS и т.д.). ROS состоит из 4 частей:

- Основные функции: управление процессами, межпроцессная связь, драйверы устройств (в основном, на уровне сборки)
- Инструменты: для моделирования, визуализации, графического интерфейса пользователя, регистрации данных
- Возможности: управление, восприятие, планирование, отображение и манипулирование
- Экосистема: относится к управлению пакетами, распространению программного обеспечения, документации, учебным пособиям и, в целом, к сообществу ROS

ROS - это распределенный P2P-фреймворк, в котором есть библиотеки для множества различных языков (например, Python, C++, Matlab, Java и т.д.). ROS имеет открытый исходный код и бесплатен для использования.

Для чего используется ROS?

- Используется многими университетами и компаниями в области робототехники для исследований и создания прототипов
- Программное обеспечение для промышленных роботов
- НАСА использует ROS в некоторых космических роботах (например, Astrobbee)
- BMW использует ROS для автоматизированного вождения

Узел ROS - это универсальная исполняемая программа, которая индивидуально компилируется, выполняется и управляется. Узлы организованы в пакеты. В ROS 1 главный узел ROS управляет связью между узлами. Он регистрирует каждый узел при запуске. Узлы организованы в виде древовидной структуры.

Примечание: Узлы не могут иметь одинаковые имена. Чтобы запустить несколько экземпляров одного и того же узла, добавьте префикс/суффикс или объявите его анонимным.

ROS использует следующие инструменты для облегчения взаимодействия между программами:

- Действия: Позволяет пользователям давать указания, не зависящие от времени, для мониторинга состояния сервера.
- Темы: поток сообщений, передаваемых между узлами. Узлы могут публиковать тему или подписываться на нее. Обычно от одного издателя приходится п подписчиков.
- Сообщение: Структура данных, определяющая тип темы. Определяется в msg-файлах
- Сервисы: создает синхронный канал между узлами, позволяющий пользователям запрашивать определенные действия у роботов и изменять настройки.

1. Узел

Таким образом, узел - это большая платформа, которая может публиковать различные темы, сервисы и действия. Это может быть узел камеры, который публикует изображения и может предоставлять другую информацию в качестве сервиса. Это может быть любой способ, которым, по вашему мнению, вы можете логически организовать свой проект [1].

2. Темы

Темы - это то, что вы о них думаете. Это названия, под которыми передаются сообщения. Сообщения могут быть строкой, например, "Привет, революция роботов!", значением Int, например, 42, или пользовательским, которое вы придумали в виде изображения. Насколько нам известно, вы можете отправлять данные в любых форматах, таких как массивы с изображениями и строками, облако точек и т.д.

Эти пользовательские сообщения создаются с использованием msg-файлов. В нем вы можете указать структуру вашего сообщения, и после компиляции и небольшого количества кодирования их можно будет использовать [1].

3. Услуги

Сервис - это, в общем, услуга. Один узел может запросить у другого узла что-то сделать на основе данных. Я могу присвоить узлу номера 2 и 3, а узел вернет мне 5. Магия, я знаю! Они работают так же, как и сообщения, по способу их создания. Для их создания можно использовать файл .srv [1].

4. Действия

Вы приказываете роботу двигаться вперед. Но потом забываете! Вы можете запросить у узла действий возврат статуса. Подождите, робот вот-вот упадет со скалы! Вы также можете отменить действие. Это более сложное действие, но это основы. В любом

случае, главный сервер обрабатывает все сообщения, службы и действия. Узел - это сообщество кода, содержащее разделы, службы и действия. Есть 2 вещи, которые могут выполнять узлы [3].

5.ROSCLI

Это также сложная часть ROS. Существует несколько команд, но вот как они работают.

- `roscore` - для запуска главного сервера.
- `ros node` - для управления
- узлами ROS, `rostopic` - для управления темами,
- `rosservice` - для управления сервисами,
- `ros action` - для управления действиями,
- `roslaunch` - это команда, которую мы используем чаще всего. Она предназначена для запуска вашего кода после его компиляции.

У вас есть команда для каждого из основных разделов, о которых я говорил выше.

Вы видите, как каждая из команд разбита на более мелкие разделы, основанные на различных аспектах ROS. Обратите внимание, что команд больше, чем показано здесь. У каждой из этих команд есть вложенные команды. Позже я расскажу подробнее о том, что все они делают, а пока знайте, что вы можете читать (воспроизводить) сообщения, запускать службы и действия, измерять частоту и размер тем и многое другое [1].

6.RVIZ

RVIZ - это палочка-выручалочка для нас, простых смертных. Только боги ROS могут обойтись без него. Это инструмент визуализации, с помощью которого можно увидеть любую тему, сообщение или информацию, поступающую или исходящую от пользователя. Облака точек, одометрия, изображения, видео, цифры и многое другое. Его можно запустить, просто набрав `rviz` или `roslaunch rviz rviz` [2].

Вывод

В ходе выполнения работы был разработан и реализован комплекс программного обеспечения на основе Robot Operating System (ROS) для управления роботом. Этот комплекс включает в себя модули для навигации, восприятия окружающей среды, планирования движения и взаимодействия с пользователем.

Основные компоненты системы включают в себя:

Навигация: Реализованы алгоритмы SLAM (Simultaneous Localization and Mapping) для построения карты окружающей среды и определения местоположения робота в реальном времени.

Восприятие: Интегрированы сенсоры, такие как камеры, лидары и датчики расстояния, для сбора данных о окружающей среде.

Планирование движения: Разработаны алгоритмы для планирования оптимального пути и управления движением робота по заданной траектории с учетом препятствий.

Взаимодействие с пользователем: Создан интерфейс для управления роботом с помощью команд пользователя, а также для предоставления обратной связи о состоянии системы.

В ходе тестирования системы было подтверждено ее эффективное функционирование в различных условиях и сценариях использования. Робот успешно справляется с навигацией в неизвестной среде, обнаружением препятствий и выполнением заданных команд.

В целом, разработанный комплекс программного обеспечения на основе ROS представляет собой надежное и гибкое решение для управления роботом, обеспечивая высокий уровень автономности и возможность дальнейшего расширения функциональности.

Библиография:

- [1] „ROS-tutorial”, [Online]. Available: <https://docs.ros.org/en/iron/Tutorials.html#>
- [2] Course "Introduction to Mobile Robotics and Robot Operating System (ROS)", [Online]. Available:
<https://www.youtube.com/watch?v=ve3cwG8UHjQ&list=PL2PmRem6srUn6jc7Q6ahjL8x2qJg150mZ&index=2>
- [3] „Проект ROBOT Anno (ROS)”, [Online]. Available: <https://robotanno.com/en/ros-system-controller/>