

Санкт-Петербургский государственный электротехнический университет им.  
В.И. Ульянова (Ленина)

# Разработка дифференциального рулевого управления на базе ROS 2

Выполнил: Никитин Владислав Андреевич, гр. 5304

Руководитель: к.т.н., доцент Кринкин Кирилл Владимирович

Консультант: старший преподаватель Чернокульский Владимир Викторович

# Актуальность

ROS занимает лидирующие позиции в стандартах робототехники. За последнее десятилетие ROS вырос в несколько раз. ROS2 является новым витком в развитии роботизированных операционных систем.

## Проблема

На данный момент в ROS отсутствуют некоторые из наиболее важных требования, такие как режим реального времени, безопасность, сертификация, сохранность.

# Цель и задачи

**Цель:** реализовать дифференциальное рулевое управление робота с использованием библиотек ROS 2.

## **Задачи:**

- Исследовать возможности ROS2 и сравнить их с ROS
- Исследовать возможности Micro-ROS
- Описать схему и модель движения робота
- Реализовать программный код под микроконтроллер

# Задача 1. Возможности ROS2 и различия с ROS

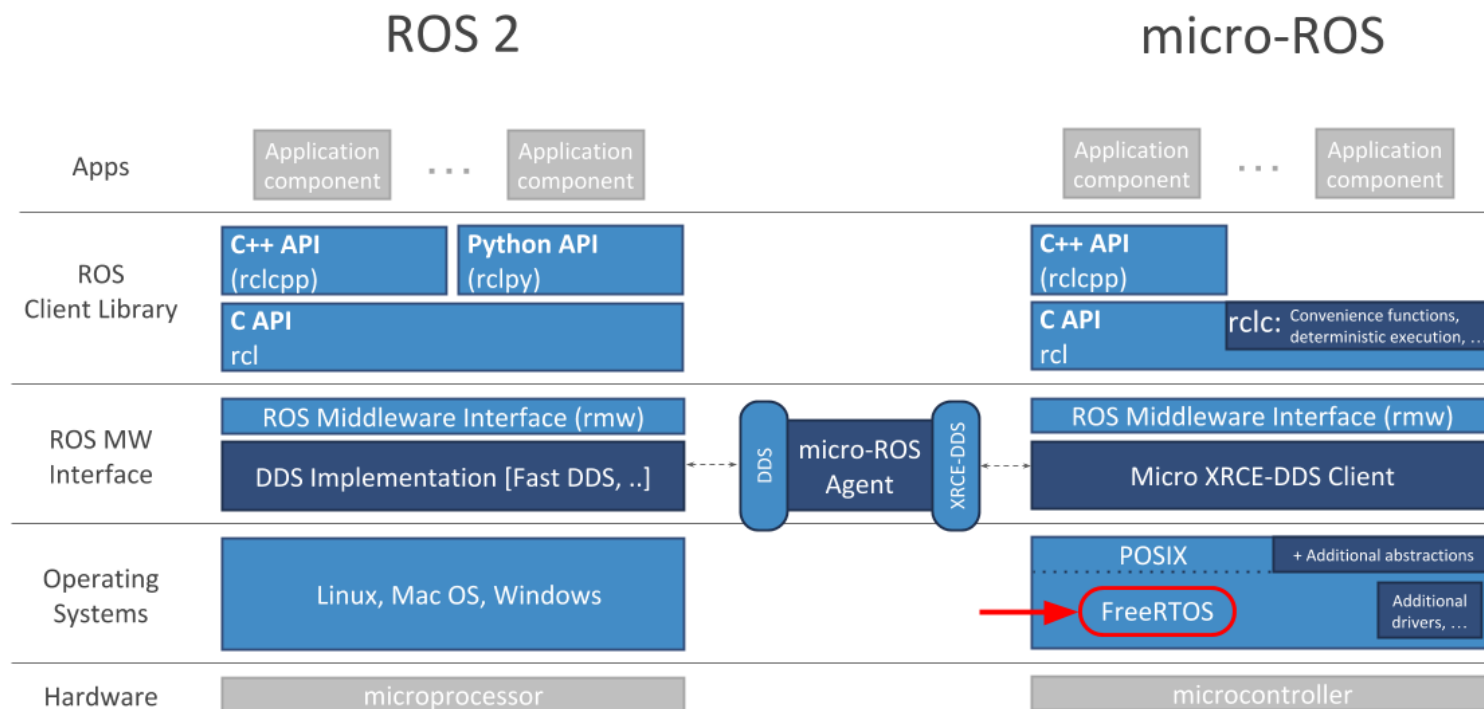
- Использование «компонент» для записи нод, в один контейнер.
- Использование службы DDS (Data Distribution Service) для публикации и подписки.
- Многослойность кода ROS 2.
- Возможность миграции существующего кода с ROS на ROS2 (ros1\_bridge)



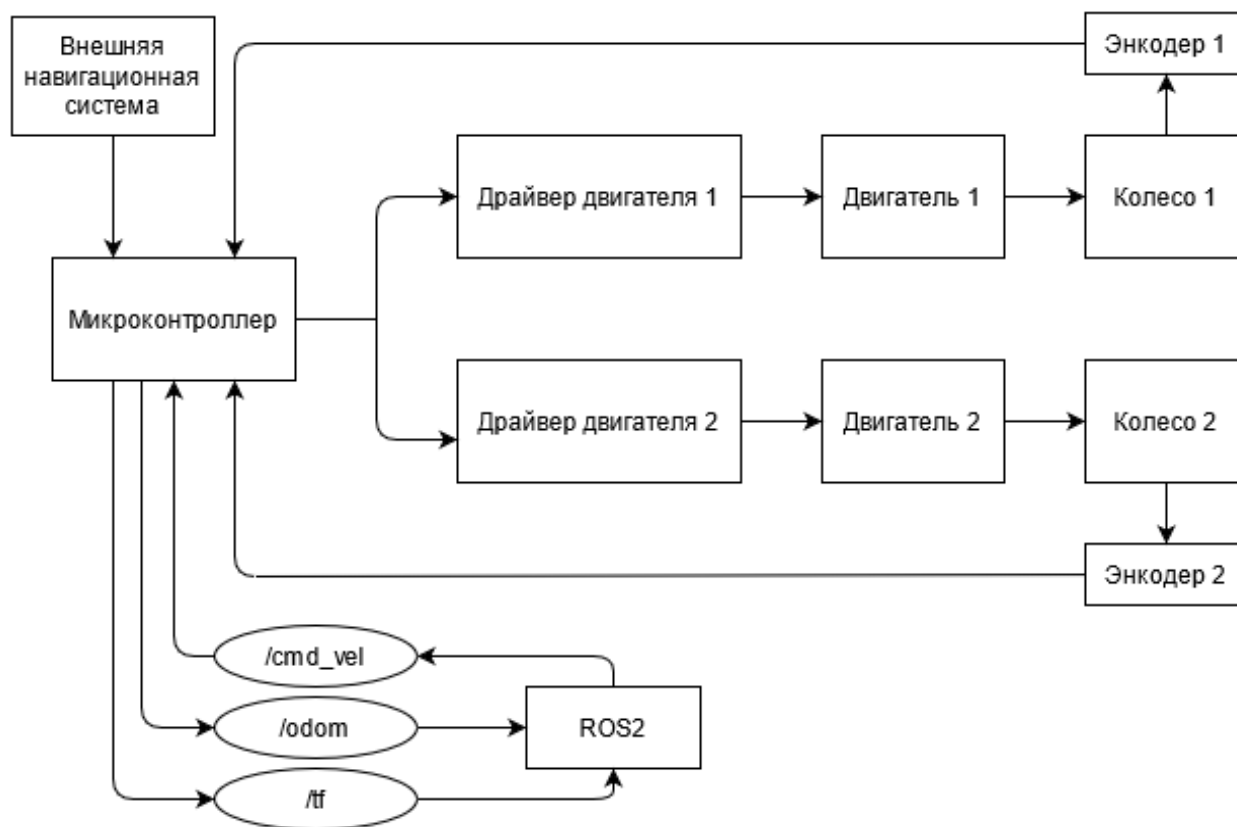
# Сравнение ROS и ROS2

	ROS	ROS 2
Поддерживаемые платформы	Ubuntu OS X	Ubuntu Xenial OS X El Capitan Windows 10 Micro-ROS: RTOS
Поддерживаемые языки программирования	C++ 03 Частично C++ 11 Python 2	C++ 11 Python 3 В планах: C++ 14, C++ 17
Промежуточное программное обеспечение	Собственный формат сериализации, транспортного протокола и центрального механизма обнаружения	Реализации интерфейса основаны на стандарте DDS
Модель потоковой передачи	Можно выбирать только однопоточное или многопоточное выполнение	Более детализированные модели выполнения, и возможность легкой реализации пользовательского исполнителя (executor)
Возможность создания нескольких нод в одном процессе	Нет	Да
Roslaunch	XML файл с ограниченными возможностями	Написан на Python с доступом к более сложной логике

# Задача 2. Micro-ROS



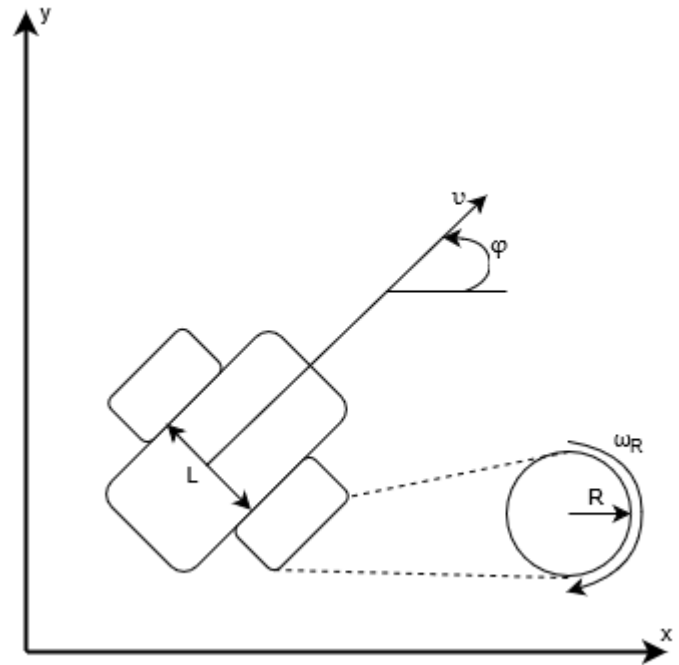
# Задача 3. Схема дифференциального рулевого управления



# Кинематическая модель движения

$$\begin{cases} \dot{x} = \frac{R}{2}(\omega_1 + \omega_2) \cos \varphi \\ \dot{y} = \frac{R}{2}(\omega_1 + \omega_2) \sin \varphi \\ \dot{\varphi} = \frac{R}{L}(\omega_1 - \omega_2) \end{cases} \quad \begin{cases} \omega_1 = \frac{1}{R}(v + \frac{\omega \cdot L}{2}) \\ \omega_2 = \frac{1}{R}(v - \frac{\omega \cdot L}{2}) \end{cases}$$

- где:  $\dot{x}, \dot{y}$  - координаты смещения базовой точки робота (середина оси вращения колес);
- $\varphi$  - угол между положительным направлением оси  $O_x$  и вектором скорости робота;
- $\omega_1, \omega_2$  - угловые скорости вращения колес;
- $R$  – радиус колес;  $L$  – расстояние между колесами;





## Задача 4. Результаты разработки

- Скомпилирована библиотека Micro-ROS для разработки в среде STM32CubeIDE.
- Собран проект под микроконтроллер Nucleo STM32F746ZG, на основе FreeRTOS.
- Связь между Micro-ROS и ROS2 осуществляется по Serial порту с использованием Micro XRCE-DDS.
- Максимальная стабильная скорость передачи данных равна 150 КБит/с, а средняя частота сообщений равна 2000 Hz.
- <https://github.com/Aludey/ROS2-SDSSMR>

# Заключение

- Рассмотрена интеграция библиотек с открытым исходным кодом ROS2 в программу, реализующую дифференциальное рулевое управление роботом на микроконтроллере.
- Составлены схема дифференциального рулевого управления, математическая модель движения и реализована модель управления с использованием ROS2.
- Оценена передача данных через serial порт платы с использованием Micro XRCE-DDS