



Open Source Robotics Foundation

Обзор свободного фреймворка ROS

Касьяник Валерий

Брест, Беларусь

vvkasyanik@bstu.by

Брестский государственный технический
университет, СНИЛ «Робототехника»

Open source robotics

- The mission of OSRF is "to support the development, distribution, and adoption of open source software for use in robotics research, education, and product development."
- An **open-source robot** is a robot whose blueprints, schematics or source code are released under an open-source model.



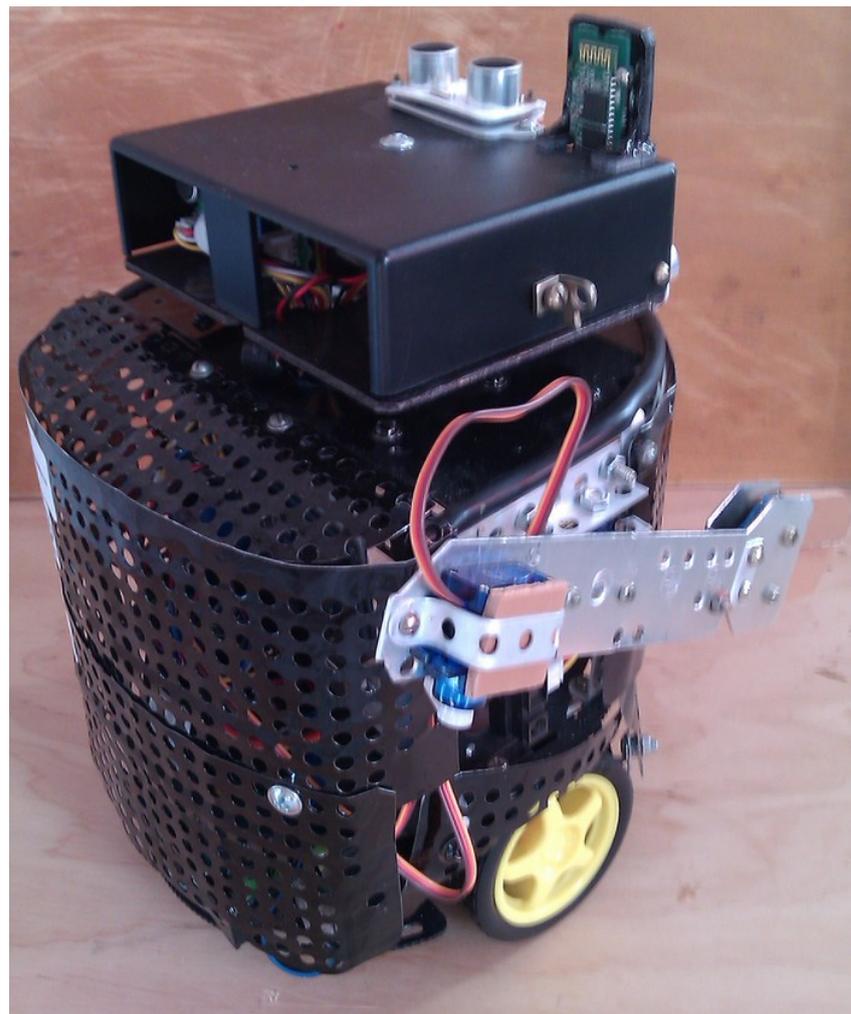
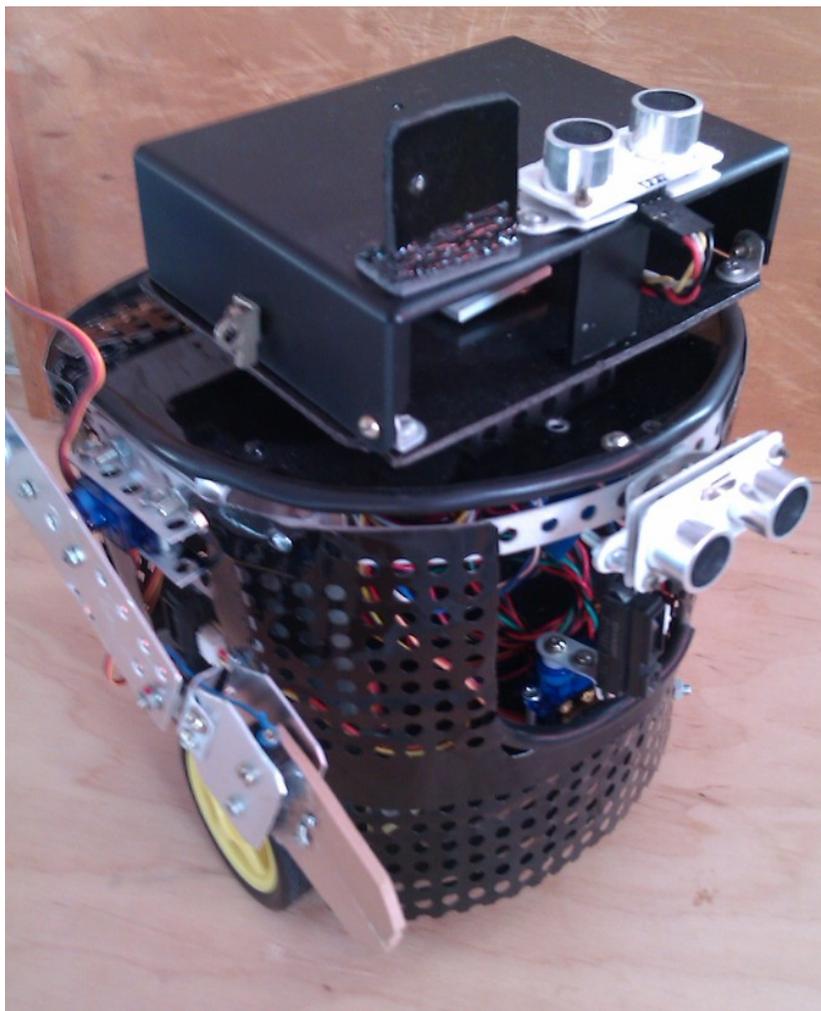
Open Source Robotics Foundation

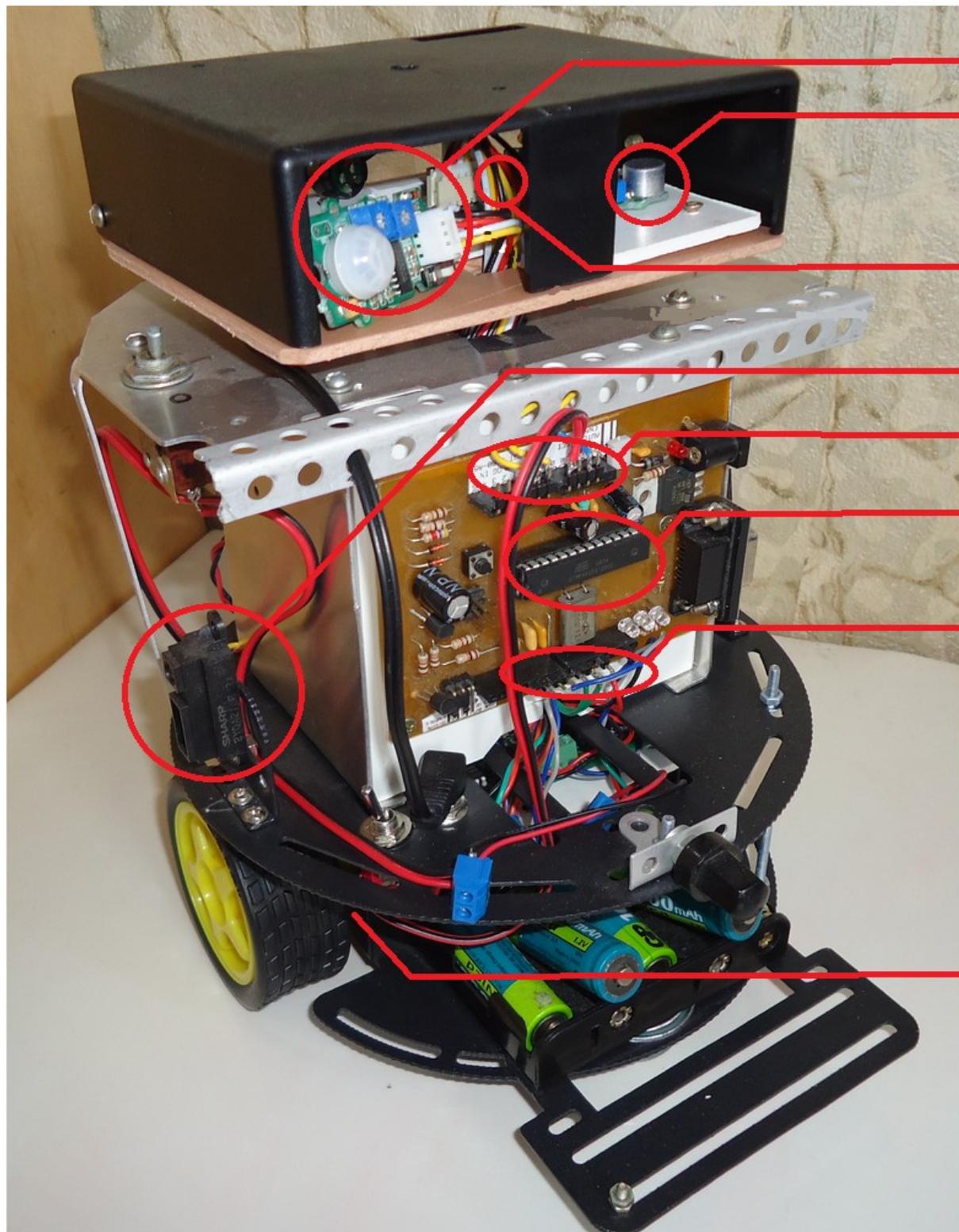
Open source robotics in Belarus

- Roborace
- Visegrad Robotics Workshop
- Прикладные роботы



MARVIN





PIR-motion sensor

sound sensor

gyro-sensor

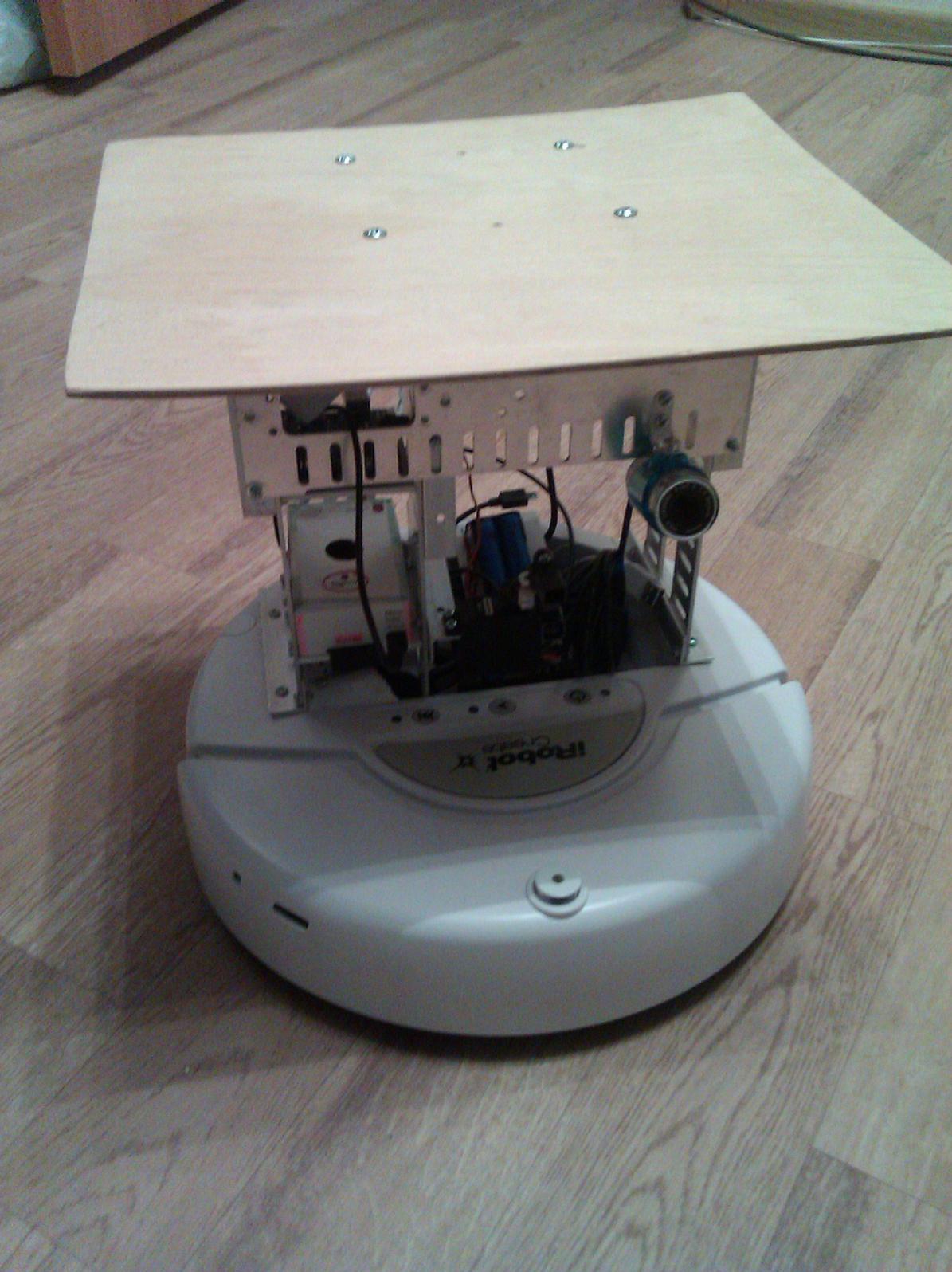
infrared range finder

analog input

Arduino Atmega 168P

digital I/O for
peripherals devices

2 wheels with
encoders

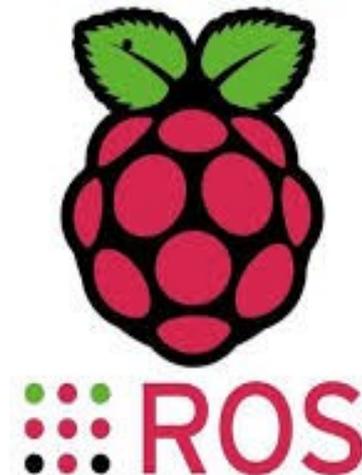


Аппаратная структура робота



ROS и Raspberry Pi

- ROS (Ubuntu-based ARMv7)
- Onboard Computer — Raspberry Pi (ARM v6)
- Решение: ROSbian



Почему появилась ROS?

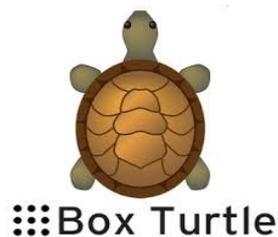
- Писать ПО для роботов сложно, потому что
 - Есть много разных роботов и оборудования для них
 - Нужно большое количество стороннего кода для различных устройств робота
 - Нужны знания в множестве областей для реализации функциональности робота

Что такое ROS?

- ROS — это свободная мета-операционная система для Вашего робота (C) www.ros.org
 - Пакеты различного софта и утилит для его сборки
 - Архитектура для распределенного межпроцессного и межмашинного взаимодействия
 - Средства визуализации и анализа данных
 - Независимость от языка разработки

Существующие дистрибутивы

- Box Turtle
- C Box Turtle
- Daimondback
- Fuerte
- Groovy Galapagos
- Hydro Medusa



Чем не ROS не является

- Операционной системой в традиционном понимании
- Языком программирования
- Интегрированной средой разработки
- Сложной архитектурой исполнения кода в режиме реального времени
- Только библиотекой кода

Главные достоинства ROS

- Распределенность
- Повторное использование кода
- Независимость от языка разработки
- Простое тестирование
 - На реальном роботе
 - В симуляторе
- Масштабирование

ROS предоставляет разработчику сервисы

- Уровень аппаратной абстракции
- Низкоуровневое управление устройствами
- Наиболее часто используемая функциональность робота
- Передача сообщений
- Управление пакетами ПО

Инструментальная мультиязычность

- Нативная поддержка популярных языков:
 - C++, Python, Octave, LISP, Matlab
- Стандартизованные спецификации на уровне обмена сообщениями
- Собственный язык описания сообщений

Утилиты ROS

- Микроядерная архитектура
- Большое число утилит
 - Сборка и запуск компонент ROS
 - Навигация по исходным кодам
 - Управление конфигурацией системы
 - Визуализация топологии соединений узлов
 - Измерение скоростей обмена сообщениями
 - Графическое отображение данных сообщений
 - Автогенерация документации

УТИЛИТЫ

- Визуализация (rviz)
- Запуск вычислительного графа (roslaunch)
- Диагностика (roswhf :))
- Логирование данных (rosbag)
- Отрисовка данных (rplot)
- Визуализация графа вычислений (rgraph)
- Симуляция роботов и окружения (gazebo/stage)
- И многие другие

Функциональность робота

- Мобильность
- Восприятие
- Манипуляторы
- Планирование пути
- Построение карты
- Алгоритмы
- Телеуправление

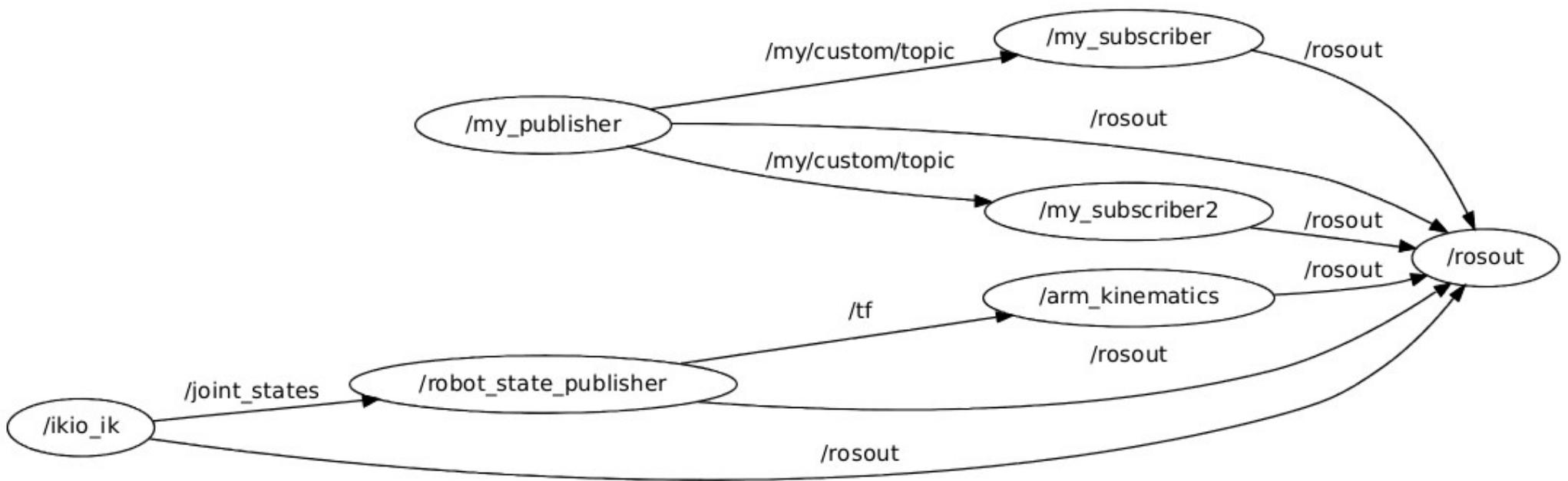
Свобода и открытость

- Весь исходный код публично доступен без ограничений
- BSD лицензия
- Модульность позволяет избежать проблем смешения лицензий

Уровни ROS

- **Вычислительный граф** или как ROS запускает код
- **Файловая система** или как ROS организует хранение кода и его сборку
- **Репозитории кода** или как ROS управляет обменом кода

Вычислительный граф



Узлы

- Исполняемые программы предоставляющие атомарную функциональность
 - Управление моторами
 - Обработка изображений камеры
 - и т.д.
- Каждый узел может быть скомпилирован, запущен и остановлен индивидуально

Сообщения

- Используются для связи узлов
- Строгая типизация структур данных
- Поддержка стандартных типов данных
- Поддержка массивов
- Поддержка вложений

Язык определения интерфейсов взаимодействия

- Собственный язык определения интерфейсов взаимодействия
- Сериализация и десериализация
- Стандартные типы сообщений (float, bool, ...)
- Пример сообщения для управления роботом:
 - Integer LeftVelocity
 - Integer RightVelocity

Модель публикуй/подписывайся

- Связь между узлами использует модель публикую / подписываюсь
- Узлы могут публиковать данные по темам
- Узлы могут подписываться на получение данных по заданной теме
- Узел master управляет информационными потоками

Темы

- Широковещательная архитектура
- Описывается именем и типом данных
- Узел публикует информацию всегда в определенную тему
- Узел получает информацию, подписавшись на соответствующие темы
- Допустима конкуренция множества подписчиков и публикаторов для одной темы

Сервисы

- Синхронные транзакции
- Описывается именем и двумя типами данных:
 - тип запроса
 - тип ответа
- Аналог веб-сервисов
- В отличие от тем, один сервис предоставляет только один узел

ROS master

- Координирующий узел системы, который сообщает другим узлам о доступных сервисах и темах
- Связь на основе XML/RPC
- Предоставление основных параметров ROS по url: ROS_MASTER_URI: http://host:port

Сервер параметров

- Запускается как служба master-узла
- Одни узлы могут хранить свои ключевые параметры в одном месте, чтобы другие имели к ним доступ
- Каждый параметр имеет свое уникальное имя и тип данных
- Параметры могут устанавливаться и изменяться во время выполнения

Файловая система

- Пакеты
 - Манифест пакета
- Стеки
 - Манифест стека
- Кластеры
 - Манифест кластера

Пакеты

- Содержат библиотеки, код узлов, описание сообщений, утилиты
- Выделяются исходя из функциональности, которая должна решать достаточно узкую задачу.
- Сообщество ROS уже имеет более 1000 пакетов

Стеки и кластеры

- Стеки содержат пакеты, необходимые для решения более сложной задачи.
 - Пример: навигация, управление манипулятором
- Стеки используются для версионирования кода
- Кластеры нужны для управления несколькими стеками.
 - Пример: Управление несколькими манипуляторами

Репозитории

- Содержат стеки для решения самых разных задач
- Как правило поддерживаются лабораториями робототехники
- Фермы портирования пакетов для различных аппаратных архитектур

Вопросы?

