

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий

Отчет о прохождении производственной практики

Выполнил студент группы 22207:

Яскеляйнен Семён Дмитриевич

Направление подготовки:

Программная инженерия

Место прохождения практики:

Кафедра ИМО, ЦИИ

Сроки прохождения практики:

30.05.22 - 09.06.22

Руководитель практики:

к.т.н., доцент

Богоявленская Ольга Юрьевна

Оценка _____

Дата _____

Петрозаводск 2022 г.

Содержание

Введение	3
1 Теоретическая часть	4
1.1 Что такое ROS.	4
1.2 Механизм publisher/subscriber	4
2 Практическая часть	5
2.1 Создание собственных publisher и subscriber	5
2.2 Запуск своих узлов	5
2.3 Просмотр графа взаимодействий	7
Заключение	8
Список литературы	9

Введение

Робототехника — прикладная наука, занимающаяся разработкой автоматизированных технических систем и являющаяся важнейшей технической основой развития производства[1]. Робототехника опирается на такие дисциплины, как электроника, механика, кибернетика, телемеханика, мехатроника, информатика, а также радиотехника и электротехника. Выделяют строительную, промышленную, бытовую, медицинскую, авиационную и экстремальную (военную, космическую, подводную) робототехнику.

В своей работе я буду рассматривать такую технологию в робототехнике, как ROS(Robot Operating System) - операционная система для роботов, которая является одним из способов взаимодействия с роботом.

Главной целью моей работы является изучение механизма publisher/subscriber в ROS.

Основные задачи:

1. Изучение теоретических материалов.
2. Знакомство с механизмом publisher/subscriber.
3. Создание 2 publisher и 2 subscriber, которые могут "общаться" через 3 топика.
4. Запуск системы.
5. Проверка графа взаимодействий.

1 Теоретическая часть

1.1 Что такое ROS.

Операционная система для роботов (ROS) — это набор программных библиотек и инструментов, помогающих создавать приложения для роботов. От драйверов до самых современных алгоритмов, а также с мощными инструментами разработчика — в ROS есть все, что вам нужно для вашего следующего проекта по робототехнике. И все это с открытым исходным кодом.[\[2\]](#)

ROS обеспечивает стандартные службы операционной системы, такие как: аппаратную абстракцию, низкоуровневый контроль устройств, реализацию часто используемых функций, передачу сообщений между процессами, и управление пакетами. ROS основан на архитектуре графов, где обработка данных происходит в узлах, которые могут получать и передавать сообщения между собой. Библиотека ориентирована на Unix-подобные системы.[\[3\]](#)

1.2 Механизм publisher/subscriber

«Узел» — это термин ROS для исполняемого файла, подключенного к сети ROS. Для организации работы нам необходимо писать узлы, которые что-то сообщают в различных темах (такие узлы называются publisher), либо нам надо писать узлы, которые "слушают" происходящее топики (такие узлы называются subscriber). Узлы могут обозначать различные части нашего робота (датчики, сенсоры, светодиоды и т.д.)

Например, мы можем написать publisher, который будет посылать в некоторый топик значения, а некоторый subscriber будет читать получаемые значения в данном топике, обрабатывать их и подсвечивать соответствующий светодиод на плате.

2 Практическая часть

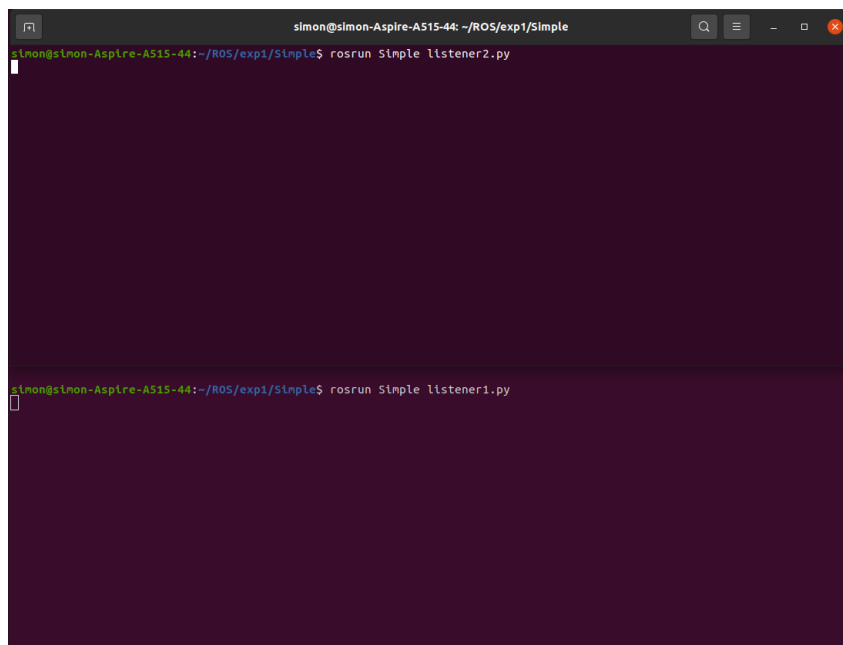
2.1 Создание собственных publisher и subscriber

В рамках моей практической части мной было решено реализовать 2 подписчика и 2 издателя, которые будут обмениваться информацией в 3 топиках. Первый издатель был сделан таким образом, что каждую третью секунду он отправлял данные во второй топик, а в остальное время его сообщения отправляются в topic1. Второй издатель всё время отправляет все свои сообщения в topic3.

Подписчики были организованы следующим образом: первый подписчик получает информацию из topic2 и topic3, а второй подписчик получает сообщения только из topic1. Тогда получается, что второй subscriber будет получать сообщения только от первого publisher, а первый subscriber будет получать сообщения от обоих издателей. Проверим сообщения, запустив наши узлы и проанализировав сообщения.

2.2 Запуск своих узлов

Пробуем сперва просто запустить наших подписчиков и убеждаемся, что ничего не происходит, ведь никто не отправляет данные в топики.



```
simon@simon-Aspire-A515-44: ~/ROS/exp1/Simple
simon@simon-Aspire-A515-44:~/ROS/exp1/Simple$ rosrn Simple listener2.py

simon@simon-Aspire-A515-44:~/ROS/exp1/Simple$ rosrn Simple listener1.py
```

Рис. 1: Запуск одних лишь subscriber

Затем запускаем первый publisher и видим, что сообщения пошли как в subscriber1, так и в subscriber2, что мы и предсказывали.

[illegible]

Рис. 2: Запуск одних лишь subscriber

И теперь нам осталось только запустить `publisher2`. После его запуска в работе `subscriber2` ничего не изменилось, а вот у `subscriber1` стали чаще появляться сообщения и в моей работе они специально отличаются визуально от сообщений, которые отправляет `publisher1`.

[illegible]

Рис. 3: Изменения в работе subscriber1 при запуске publisher2

2.3 Просмотр графа взаимодействий

Теперь пропишем команду `gqt_graph` и посмотрим граф, который отображает взаимодействие наших узлов через топики.

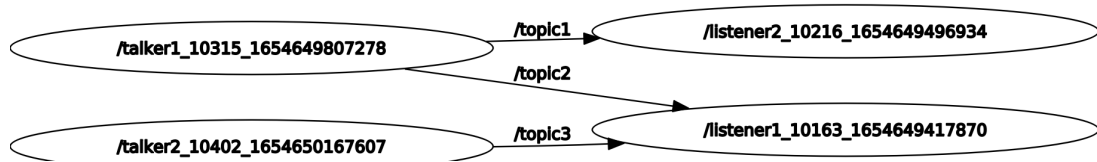


Рис. 4: Граф взаимодействий наших узлов

На графе всё отображается так же, как мы и предполагали сделать, задача выполнена.

Заключение

В ходе работы были изучены теоретические материалы по ROS. Подробнее изучен механизм publisher/subscriber, который представляет собой обмен сообщениями между узлами, которые могут являться различными частями нашего робота. Были написаны свои узлы, которые взаимодействовали через 3 топика и был получен граф взаимодействия узлов.

