

Министерство науки и высшего образования Российской Федерации  
ФГАОУ ВО «Волгоградский государственный университет»  
Институт Математики и информационных технологий  
Кафедра компьютерных наук и экспериментальной математики

УТВЕРЖДАЮ:

Зав. кафедрой КНЭМ

Клячин В.А.

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ на ПРОИЗВОДСТВЕННУЮ  
ПРАКТИКУ (ПРЕДДИПЛОМНАЯ)

Студент	Курбанов Эльдар Ровшанович (ФИО)	МОСМ-201 (группа)
Руководитель практики от ВолГУ	Клячин В.А. (ФИО)	зав. кафедрой КНЭМ, профессор, д.ф.-м.н. (должность, ученое звание и степень)
Ответственный за организацию практики от кафедры	Клячин В.А. (ФИО)	зав. кафедрой КНЭМ, профессор, д.ф.-м.н. (должность, ученое звание и степень)
Место прохождения практики	Лаборатория «Математического и программного обеспечения ЭВМ» кафедры КНЭМ ИМИТ ФГАОУ ВолГУ (наименование учреждения, структурного подразделения)	
Сроки прохождения практики	с «29» апреля 2022 г.	по «16» мая 2022 г.

1. Содержание и задания практики:

№ п/п	Этапы практики	Содержание работы и задания этапов	Коли- чество ча- сов	Календар- ные сроки проведе- ния	Форма отчётно- сти
1	Подгото- витель- ный этап	Решение организационных вопросов: установочная конференция, знакомство с задачами и программой практики, требованиями к отчетной документации; знакомство с объектами и особенностями предстоящей деятельности; инструктаж по технике безопасности	10	29.04.2022	Собесе- дование
2	Ориенти- ровочный этап	Знакомство с базовой организацией практики, изучение и анализ/обзор нормативно-правовой документации; знакомство с методами работы; изучение/обзор литературы; знакомство с методами исследования.	18	30.04.2022- 04.05.2022	Собесе- дование. Пись- менный отчет (часть).

3	Основной этап	Подготовка научно-аналитического обзора по выбранной тематике на основе последних публикаций в российских и зарубежных журналах для анализа существующих решений по заданной предметной области. Построение функциональной модели или диаграммы классов для описания поставленной задачи (Описание структурных элементов исследования, их связи, возможные форматы представляемых в системе данных. Анализ особенностей решаемой задачи.). Представление методов оценки качества проектного решения (например, результатов тестирования).	60	05.05.2022-12.05.2022	Письменный отчёт (часть).
4	Заключительный этап	Подготовка отчета о прохождении практики. Выступление с докладом-презентацией.	20	13.05.2022-16.05.2022	Письменный отчет (оформление). Представление/защита результатов практики.

## 2. Планируемые результаты практики:

*студент должен знать:* основы в области математики, программирования и информационных технологий; методы построения научной работы, современные методы сбора и анализа полученного материала, способы аргументации; основы построения научных обзоров, публикаций, рефератов и библиографий по тематике проводимых исследований на русском и английском языках; особенности распоряжения правами на результаты интеллектуальной деятельности; формы и методы правовой охраны результатов интеллектуальной деятельности; современными технологиями проектирования и производства программного продукта; направления развития: компьютеров с традиционной (нетрадиционной) архитектурой; современных систем программных средств, операци-

онных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности; концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений; современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования; основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов.

*студент должен уметь:* находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий; решать научные задачи в связи с поставленной целью и в соответствии с выбранной методикой; решать задачи, связанные с использованием результатов интеллектуальной деятельности и средств индивидуализации для создания инновационной продукции и услуг, в том числе ориентированных на зарубежные рынки; использовать технологии проектирования при создании программных продуктов; программировать для компьютеров с различной современной архитектурой; программировать в рамках функционального, логического, объектно-ориентированного и визуального направлений; разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования; подготовить техническую документацию программных продуктов.

*студент должен владеть умениями:* научно-исследовательской деятельности в области программирования и информационных технологий; выступлений и научной аргументации и профессиональной деятельности; выполнять оценку преимуществ новой технологии по сравнению с аналогами; применения технологий проектирования при создании программных продуктов; выбора архитектуры и комплексирования современных компьютеров, систем, комплексов и сетей системного администрирования; разработки программ в рамках функционального, логического, объектно-ориентированного и визуального направлений; разработки и реализации алгоритмов их на базе языков и пакетов прикладных программ моделирования; подготовки технической документации.

Студент	_____	_____
	(подпись)	(расшифровка подписи)
Руководитель практики от ВолГУ	_____	_____
	(подпись)	Клячин В.А. (расшифровка подписи)
Ответственный за организацию практики от кафедры	_____	_____
	(подпись)	Клячин В.А. (расшифровка подписи)

Министерство науки и высшего образования Российской Федерации  
ФГАОУ ВО «Волгоградский государственный университет»  
Институт математики и информационных технологий  
Кафедра компьютерных наук и экспериментальной математики

УТВЕРЖДАЮ:

Зав. кафедрой *КНЭМ*

Клячин В.А.

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

ОТЧЕТ  
О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ  
(ПРЕДДИПЛОМНАЯ)

Студент	<u>Курбанов Эльдар Ровшанович</u> (ФИО)	<u>МОСМ-201</u> (группа)
Руководитель практики от ВолГУ	<u>Клячин В.А.</u> (ФИО)	<u>зав. кафедрой КНЭМ, профессор., д.ф.-м.н.</u> (должность, ученое звание и степень)
Ответственный за организацию практики от кафедры	<u>Клячин В.А.</u> (ФИО)	<u>зав. кафедрой КНЭМ, профессор., д.ф.-м.н.</u> (должность, ученое звание и степень)
Место прохождения практики	<u>Лаборатория «Математического и программного обеспечения ЭВМ» кафедры КНЭМ ИМИТ ФГАОУ ВолГУ</u> (наименование учреждения, структурного подразделения)	
Сроки прохождения практики	с «29» апреля 2022 г.	по «16» мая 2022 г.

## 1. Ход выполнения практики

№ п/п	Этап практики	Дата	Описание выполненной работы	Отметки руководителя о выполнении
1	Подготовительный этап	29.04.2022	Решение организационных вопросов: установочная конференция, знакомство с задачами и программой практики, требованиями к отчетной документации; знакомство с объектами и особенностями предстоящей деятельности; инструктаж по технике безопасности	
2	Ориентировочный этап	30.04.2022- 04.05.2022	Знакомство с базовой организацией практики, изучение и анализ/обзор нормативно-правовой документации; знакомство с методами работы; изучение/обзор литературы; знакомство с методами исследования.	
3	Основной этап	05.05.2022	Подготовка научно-аналитического обзора по выбранной тематике на основе последних публикаций в российских и зарубежных журналах для анализа существующих решений по заданной предметной области.	
		06.05.2022- 11.05.2022	Построение функциональной модели или диаграммы классов для описания поставленной задачи (Описание структурных элементов исследования, их связи, возможные форматы представляемых в системе данных. Анализ особенностей решаемой задачи.).	

		12.05.2022	Представление методов оценки качества проектного решения (например, результатов тестирования).	
4	Заключительный этап	13.05.2022-16.05.2022	Подготовка отчета о прохождении практики. Выступление с докладом-презентацией.	

Студент

\_\_\_\_\_  
(подпись)

Курбанов Э.Р.

\_\_\_\_\_  
(расшифровка подписи)

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

(по 5-балльной шкале)

(по 100-бальной шкале)

(подпись)

Клячин В.А.  
 (расшифровка подписи)

(подпись)

Клячин В.А.  
(расшифровка подписи)



## Содержание

<b>1. Описание предметной области</b>	<b>10</b>
1.1. Основы ROS . . . . .	10
1.2. Концепции ROS . . . . .	11
<b>2. Реализуемые узлы</b>	<b>12</b>
2.1. Узел камеры . . . . .	12
2.2. Узел LIDAR . . . . .	13
2.3. Узел движения . . . . .	14
2.3.1. Аппаратное управление двигателями . . . . .	15
2.3.2. Одометрия . . . . .	15
2.4. Узел SLAM . . . . .	16

---

<sup>1</sup>Приложения к отчету о прохождении практики: (приводятся материалы, указанные в индивидуальном плане на практику в графе «Форма отчетности», например, научно-исследовательская работа, презентации, конспект занятия и т.д.).

# Введение

Темой выпускной квалификационной работы является «Система управления и формирования поведенческой стратегии автономного мобильного робота на основе визуального анализа окружающего пространства».

Целью ВКР<sup>2</sup> является создание автономной системы управления мобильным роботом, а также системы самостоятельно принимающей решения о движении данного робота на основе визуальных данных, собираемых с лазерного сканера LiDAR, а также видеокамеры, установленной на шасси робота.

Задачей ВКР является построение роботизированной системы, и разработка программного обеспечения, выполняющую цель выпускной квалификационной работы.

## 1. Описание предметной области

Программное обеспечение мобильного автономного робота будет работать на базе **Robot Operating System (ROS)** - гибкого фреймворка<sup>3</sup>, предоставляющего различные инструменты и библиотеки для написания роботизированного программного обеспечения.



Рис. 1. Логотип Robot Operating System

ROS предоставляет несколько мощных функций, помогающих разработчикам в таких задачах, как передача сообщений, распределение вычислений, повторное использование кода и реализация современных алгоритмов для роботизированных приложений[2].

### 1.1. Основы ROS

В общем случае, ROS представляет собой инструмент, позволяющий связывать несколько независимых программных модулей при помощи сервисов и узлов, которые могут передавать друг другу сообщения в различном формате.

Большими преимуществами использования данного фреймворка является возможность передачи сообщений по локальной сети<sup>4</sup> и обширная библиотека уже реализованного ПО, которое можно без относительно больших затрат по времени интегрировать в

---

<sup>2</sup>Выпускная квалификационная работа

<sup>3</sup>Программная среда для выполнения чего-либо, своеобразный каркас, используемый для того, чтобы существенно облегчить процесс объединения определённых компонентов программного обеспечения в зависимости от потребностей[1].

<sup>4</sup>LAN

свой проект. На момент написания данного отчёта по практике глобальный репозиторий ROS Index, находящийся по адресу <https://index.ros.org/stats/> насчитывает 2455 подключённых к нему сторонних репозиторийев и 6927 пакетов. Диаграмму распределения пакетов по разным версиям ROS можно увидеть на Рисунке 2 .

FiXme  
Note:  
цитат  
<https://>

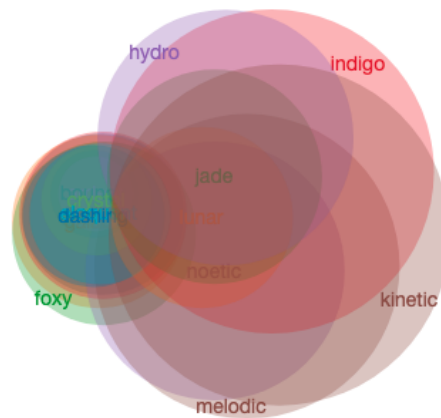


Рис. 2. Диаграмма, показывающая статистику распределения пакетов для различных версий ROS из глобального репозитория ROS Index

## 1.2. Концепции ROS

Ниже приведён список концепций рассматриваемого фреймворка для понимания описываемой в данном отчёте:

- **Узел** - это процесс, выполняющий вычисления. Каждый узел написан с использованием клиентских библиотек ROS. Используя методы связи, узлы могут общаться друг с другом и обмениваться данными. Для этого создаются узлы-подписчики, и узлы-публикаторы.
- **Мастер** - обеспечивает регистрацию и работоспособность запущенных узлов.
- **Сообщение** - простая структура данных, содержащая типизированное поле, которое может содержать целый набор данных, отправляемых на другой узел. Помимо стандартных типов сообщений<sup>5</sup> возможна отправка заранее обозначенных собственных типов сообщений.
- **Топик** - именованная шина данных, используемая узлами для отправки сообщений. Публикующий и подписанный узел не знают о существовании друга друга. Благодаря тому что каждая тема имеет уникальное имя, любой узел может получить доступ к данному топiku и отправляет через него данные, при условии соблюдения заранее оговорённых передаваемых типов, данным топиком
- **Сервисы** - реализация удалённого вызова процедур<sup>6</sup> в ROS. В некоторых случаях модель связи публикации и подписки может не подходить. В этих случаях и применяют взаимодействия в виде сервисов (схема запрос/ответ), при котором один

<sup>5</sup>Такие как целые, с плавающей точкой, логические, строковые...

<sup>6</sup>RPC

узел может запросить выполнение процедуры для другого узла, ожидая какого-то обязательного ответа<sup>7</sup>[2].

Схема взаимодействия ROS изображена на Рисунке 3 .

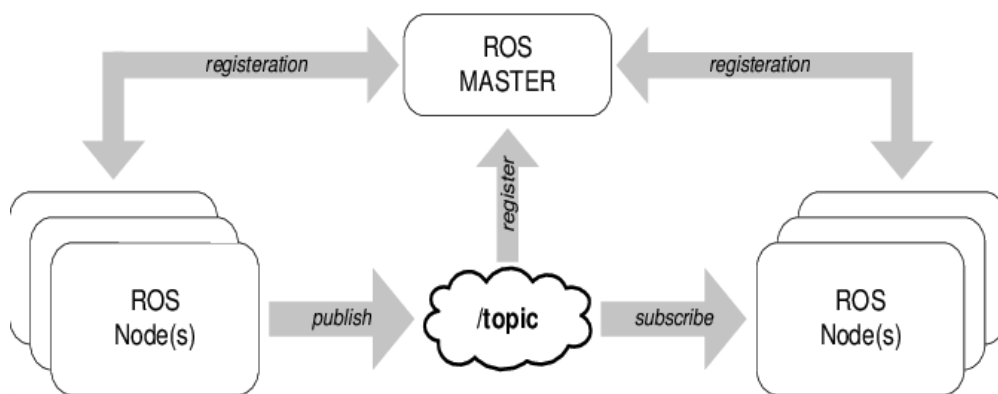


Рис. 3. Схема взаимодействия ROS

## 2. Реализуемые узлы

В терминах ROS для решения задачи, поставленной в ВКР необходимо реализовать ряд узлов, которые будут взаимодействовать друг с другом.

### 2.1. Узел камеры

Для поиска объектов, встречающихся на пути следования робота, необходимо озаботиться реализацией одним из "органов чувств" робота - зрением.

Если на изображении будет обнаружен целевой объект, то роботу необходимо будет его локализовать, подъехать и обозначить его местоположения на карте местности, которая будет строиться исходя из показаний другого сенсора, установленного на роботе (подробнее в разделе 2.2. Визуализация такого обнаружения объектов на изображении показано на Рисунке 4.

Компьютер, которым оснащён робот предоставляет высокоскоростной интерфейс CSI, позволяющий читать изображение высокого разрешения с миниатюрной видеокамеры. В качестве видеокамеры выбрана Sony IMX219 с разрешением 3280 x 2464 пикселей и углом обзора 70 градусов, изображённая на Рисунке 5 .

Задачи данного узла:

- Чтение «сырого» изображения с видеокамеры с шины CSI;

<sup>7</sup>В случае использования схемы с подписчиками и публикаторами доставка сообщений и ответ не гарантируются

FiXme  
Note:  
цитата  
Corral  
Tercero  
Gonzal  
Macías  
Miquel  
(2018).  
Trainin  
drone  
using F  
and  
OpenA  
Gym.

FiXme  
Note:  
цитата  
FiXme  
Note:  
цитата

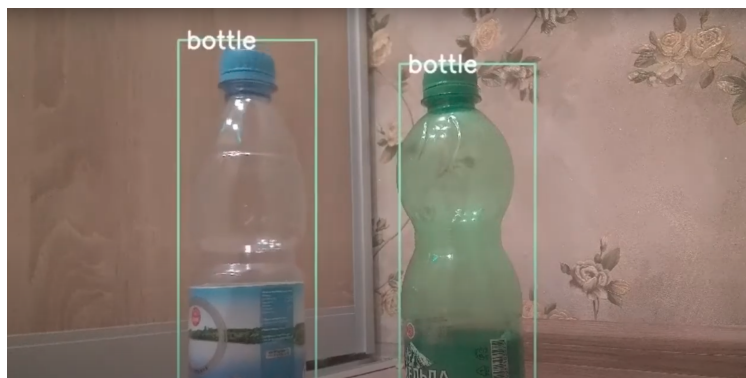


Рис. 4. Пример распознавания двух целевых объектов роботом



Рис. 5. CSI камера для Nvidia Xavier NX

- Преобразование считанных данных в формат «`sensors_msgs/Image`», приведённой в Листинге 1 ;
- Публикация в соответствующий ROS топик `"/image"`.

FiXme  
Note:  
цитата  
<http://>

Листинг 1. Формат сообщения `sensors_msgs/Image`

```
std_msgs/Header header
uint32 height
uint32 width
string encoding
uint8 is_bigendian
uint32 step
uint8 [] data
```

Результат работы данного узла можно увидеть в программе RVIZ на Рисунке .

FiXme  
Note:  
рисуно  
тут

## 2.2. Узел LIDAR

Данный узел будет являться вторым органом зрения робота, обеспечивая его информацией об окружающем пространстве на расстоянии от 10 см до 10 метров. Обеспечи-

вать такое зрение будет лазерный сканер YDLIDAR X4, изображённый на рисунке 6[3]. Он подключается по серийному порту и способен считывать облако точек с частотой до 12 Гц[3].



Рис. 6. YDLIDAR X4

В задачи данного узла входит:

- Чтение «сырых» данных об интенсивности отражения лазера от объекта;
- Перевод этих данных в формат «sensor\_msgs/LaserScan», приведённой в Листинге 2 ;
- Публикация в ROS топик "/scan".

Листинг 2. Формат сообщения sensor\_msgs/LaserScan

```
std_msgs/Header header
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32 [] ranges
float32 [] intensities
```

Результатом работы узла является облако точек, которое показывает препятствия вокруг робота[4]. Пример можно увидеть из программы RVIZ на Рисунке .

## 2.3. Узел движения

Для того чтобы приводить шасси робота, изображённое на Рисунке в движение необходимо контролировать два установленных на него электродвигателя. С этой задачей необходимо правильно работать как с аппаратной, так и с программной стороны. Для этого необходимо:

FiXme  
Note:  
цитата  
http://

FiXme  
Note:  
Рисун  
тут

- Одновременно управлять силой тока двух электродвигателей;
- Определять расстояние и направление, пройденное каждой гусеницей;
- Без задержек взаимодействовать с остальными участниками сети ROS.

### 2.3.1. Аппаратное управление двигателями

Для того чтобы обеспечить неукоснительное команд от управляющего узла ROS в виде показанного на Листинге формата сообщения «geometry\_msgs/Twist» и соблюдать заданную скорость и направление движения необходимо уметь регулировать уровень электрического входного напряжения на каждом двигателе по отдельности при помощи ШИМ-модуляции, схематично изображённой на Рисунке . При этом устройство, выполняющее данную задачу должно быть интерфейсно совместимо с управляющим компьютером.

В качестве такого устройства было выбран двухканальный драйвер двигателя Grove Motor Driver, изображённый на Рисунке . В его сердце лежит управляющий чип L298 и он может управлять двигателями с уровнем напряжения от 2.5В до 13.5В. Интерфейс управления - I2C, совместимый с управляющим компьютером робота.

### 2.3.2. Одометрия

Далее определение расстояния и направления, пройденного каждой гусеницей будет называться одометрией. Для решения данной задачи необходимо получать информацию о фактических оборотах электродвигателей робота, так как по каким-либо физическим причинам она может отличаться от ожидаемой.

Для подсчёта фактических оборотов будут использоваться датчики Холла, встроенные в электродвигатель робота, изображённый на Рисунке .

Датчики Холла выдают электрический импульс (в нашем случае 3.3В) при прохождении колесом двигателя определённой точки. Осциллограмму датчика Холла можно увидеть на Рисунке .

С программной стороны в реальном времени мы должны подсчитывать данные импульсы и на основе очередности сигналов с двух датчиков Холла, установленных на оба электродвигателя делать вывод о направлении движения, а также публиковать сообщение одометрии в стандартном формате сообщений ROS «nav\_msgs/Odometry», описанном в Листинге . Заниматься этой работой будет микрокомпьютер Teensy 4.0, изображённый на Рисунке .

FiXme  
Note:  
листинг  
прикре-  
пить  
FiXme  
Note:  
цитату  
прикре-  
пить  
FiXme  
Note:  
сноска  
здесь  
FiXme  
Note:  
цитата  
здесь  
FiXme  
Note:  
рисунк  
тут  
FiXme  
Note:  
рисунк  
тут  
FiXme  
Note:  
цитата  
https://  
I2C-Mo  
Driver-  
with-  
L298.h  
FiXme  
Note:  
рисунк  
тут  
FiXme  
Note:  
цитата  
книги  
https://  
FiXme  
Note:  
рисунк  
FiXme  
Note:  
листинг  
цитата

## 2.4. Узел SLAM

Программным сердцем данной работы является узел SLAM<sup>8</sup>, реализуемый открытой программной реализацией Google Cartographer. SLAM (одновременная локализация и картографирование) — это метод, используемый для автономных транспортных средств, который позволяет одновременно создавать карту и локализовать себя на этой карте. Алгоритмы SLAM позволяют картографировать неизвестные среды и обычно это используется для выполнения таких задач, как планирование пути и обход препятствий.

Для определения препятствий будет использоваться облако точек, приходящее от LiDAR. На основе получаемого облака и текущего местоположения робота, которое высчитывается исходя из данных одометрии, мы будем строить карту местности, изображённую на Рисунке .

Эта карта позволит роботу самостоятельно прокладывать маршрут до следующей планируемой точки. Выбор планируемой точки осуществляется следующим образом:

- 1) Если в кадре есть целевой объект и ранее он не был обозначен на карте, мы стремимся к нему;
- 2) Если мы находимся достаточно близко к целевому объекту, помечаем его на карте и выбираем следующую точку;
- 3) Во всех остальных случаях мы выбираем следующую случайную точку близкой к неисследованной области с возможным беспрепятственным проездом туда.

## Заключение

Таким образом был разработан и описан мобильный автономный робот, изображённый на Рисунке с функцией поиска целевых объектов на карте, который может быть применён в задачах по исследованию опасных объектов, разведке и сборе данных. Безусловно, данная модель для применения в реальных задачах требует значительных доработок в плане точности движения и локализации, и не является идеальной.

## Вступление

В ходе практики я работал над корректным подсчётом пройденного роботом расстоянием. Это позволит роботу самому оценивать его текущее местоположение на карте, выстраиваемой при помощи лазерного сканера LiDAR. Сам робот представляет собой платформу на двух гусеницах и оснащён двумя электродвигателями, драйвером, лазерным сканером и компьютером, управляющий процессом движения. Он изображён на Рисунке 7.

---

<sup>8</sup>Simultaneous Localisation and Mapping

FiXme  
Note:  
рисуно  
тут

FiXme  
Note:  
рисуно  
тут





Рис. 7. Робот

Моей задачей стало исправление некорректного подсчёта числа оборотов колеса производимых на ведущей части гусеницы робота. Для успешного построения карты местности (пример изображён на Рисунке 8) при помощи лазерного сканера, изображённого на Рисунке 9 роботу необходимо решать задачу локализации в пространстве. Погрешностей в определении местоположения должно быть как можно меньше, они напрямую будут влиять на выстраиваемую карту местности. Будут возникать смещения или ещё хуже - артефакты<sup>9</sup>.

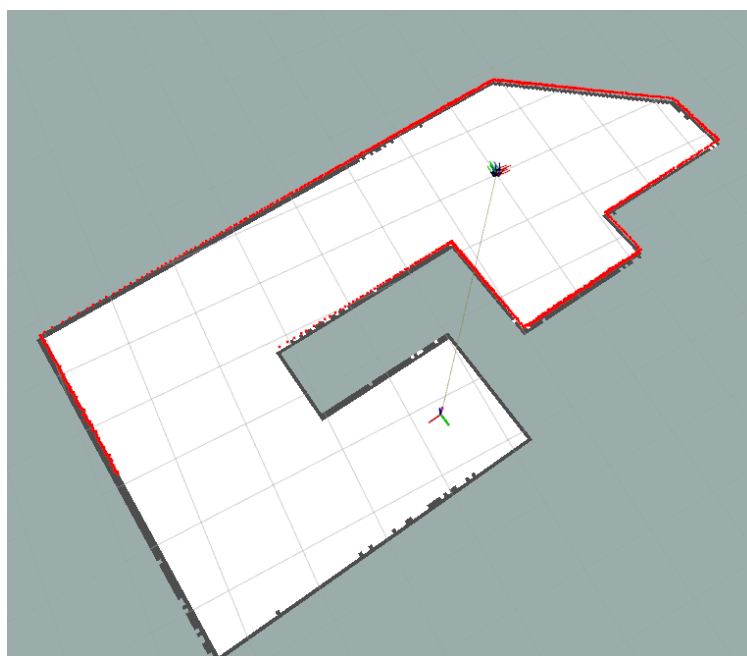


Рис. 8. Карта местности, построенная при помощи лазерного сканирования

На данном роботе возможно реализовать три способа локализации в пространстве:

- 1) Анализ смещения облака точек;

---

<sup>9</sup>объекты на карте, которых в реальности не существует



Рис. 9. Лазерный сканер YDLIDAR X4

- 2) Подсчёт одометрии;
- 3) Первые два способа вместе, корректирующие показатели друг друга.

Такие способы локализации, как триангуляция на основе заранее установленных радиомаяков и спутниковая связь Глонасс не рассматриваются ввиду требования полной автономности робота.

## Практика

Моей задачей является - "исправление расчета оборотов ведущего колеса гусеничного шасси робота". На основе этих оборотов считается фактически пройденное роботом расстояние после применения команды движения в определённую сторону. Обнаружилось, что получаемые значения оборотов отличались от ожидаемых при высокой загрузке управляющего компьютера. Изначально исправлению подлежала только программная часть робота, однако в ходе работы выяснилось, что природа ошибки кроется в операционной системе робота.

Принцип получения показателей пройденного роботом расстояния следующий:

- Робот включается и инициализирует среду ROS<sup>10</sup>;
- Включается система навигации робота, которая требует лазерный сканер и текущее расстояние, пройденное гусеницами;

---

<sup>10</sup>Robot Operating System

- Запускается лазерный сканер и происходит инициализация аппаратного интерфейса GPIO с цифровыми электрическими входами;
- Навигационная система по шине I<sup>2</sup>C даёт команду драйверам двигателя на движение;
- Датчики Холла, установленные на двигателях робота подают электрический сигнал 3.3 вольт в момент прохождения колесом одного оборота.
- Аппаратный интерфейс GPIO считывает данный сигнал и суммирует все такие обороты;
- На основе новых пройденных роботов подсчитывается пройденное роботом расстояние.

## 1 этап

Первоначально я посчитал, что причиной расхождения показателей является подвешивание программы на каком-либо из циклов в программном коде и при высокой загрузке мы просто не успеваем исполнить код, отвечающий за чтение цифрового сигнала на интерфейсе GPIO. В таком случае вполне возможно мы могли недосчитать каких-то оборотов колеса и избавление от таких циклов станет решением проблемы.

Т.к. речь идёт о программном коде робота и мы имеем дело с Robot Operating System, оперирующей с входными данными, как с входящими в неё топиками, которые публикуют другие узлы, я нашёл какой узел отвечает за публикацию и суммирование текущих оборотов двигателя. Искать долго не пришлось, но никаких бесконечных циклов в коде узла и библиотеки Jetson.GPIO, которую он использует найдено не было. Каких-либо мест в коде, где исполнение узла могло бы застревать найдено не было.

Мною была выдвинута идея о том, что такие просчёты со стороны узла напрямую связаны с природой операционной системы Ubuntu, используемой на роботе. Данная ОС не является системой, нацеленной на исполнение команд в режиме реального времени, а это значит, что в момент прохождения ведущим колесом робота датчика Холла мы не можем гарантировать квант времени от операционной системы на исполнение программы нашего узла, а значит не можем и гарантировать подсчёт всех оборотов колеса. Примерная схема такого просчёта представлена на Рисунке 10

Заручившись поддержкой тематических интернет-форумов и своего научного руководителя, я приступил ко второму этапу...



Рис. 10. Пример импульса, который не будет подсчитан программой

## 2 этап

Выходом из данной ситуации стало бы использование операционной системы реального времени, такой как QNX<sup>11</sup>, но это стало не позволительной роскошью для данного робота в следствии отсутствия какой-либо рабочей реализации используемого фреймворка ROS для данной ОС, а также высокая стоимость лицензии.

По названным выше причинам было решено некоторый микроконтроллер, который удовлетворял следующим требованиям:

- 1) Принимает электрические сигналы в реальном времени без просчётов
- 2) Способна коммуницировать с Robot Operating System
- 3) Является компактным и энергоэффективным решением

Под эти требования отлично подошёл микроконтроллер Teensy 4.0 на базе 32 битного ARM процессора NXP MIMXRT1062DVL6A. Схематичное описание и внешний вид микрокомпьютера представлены на Рисунке

К нему были подсоединены датчики холла и внешние электропитание 5 вольт. В следствии планируется делегировать на данный микроконтроллер нагрузку, связанную с управлением драйвером электродвигателей робота.

После проверки цепей питания и удостоверившись в корректном прохождении сигналов к микроконтроллеру, я начал реализовывать программную часть.

---

<sup>11</sup>QNX (произносится «кьюникс», «кью-эн-экс») — POSIX-совместимая операционная система реального времени, предназначенная преимущественно для встраиваемых систем. Считается одной из лучших реализаций концепции микроядерных операционных систем.

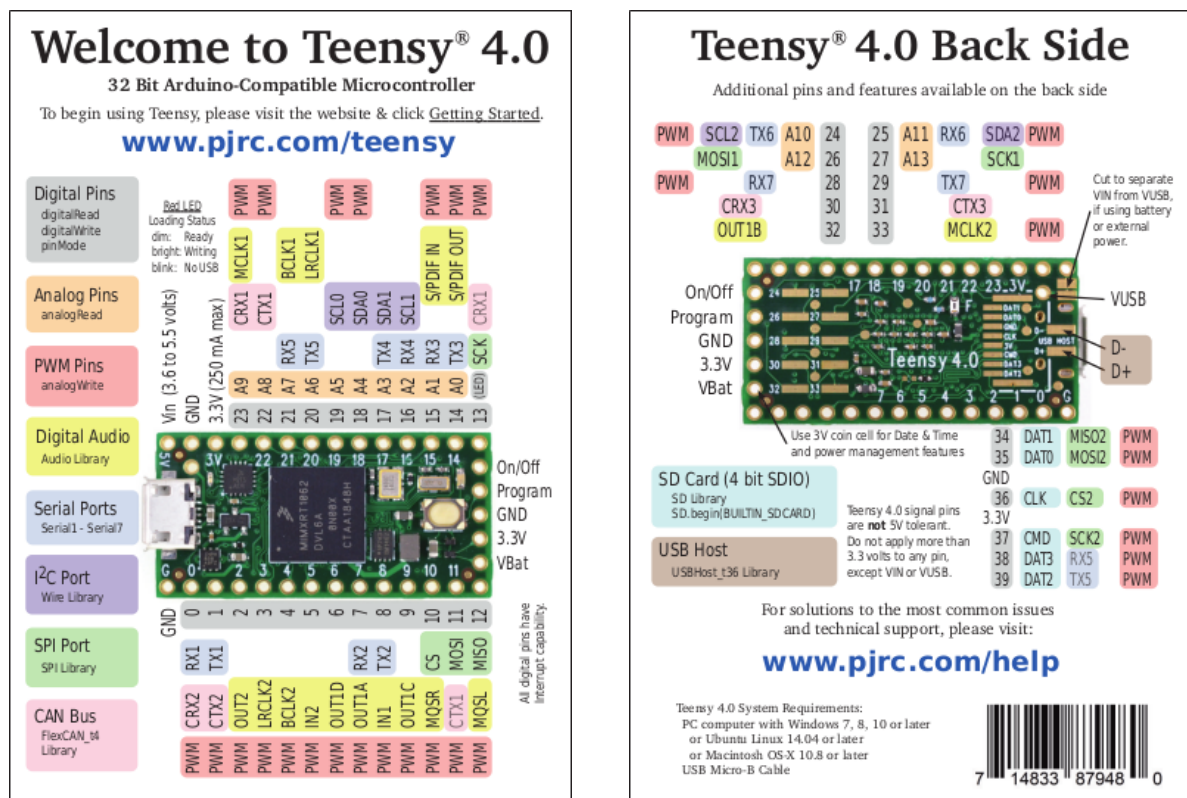


Рис. 11. Описание микрокомпьютера Teensy 4.0

### 3 этап

Для реализации программной части необходимо использовать систему разработки Arduino IDE с установленным дополнением TeensyDuino. Это позволяет использовать все библиотеки, доступные для Arduino доступными и для микроконтроллера Teensy 4.0.

Для коммуникации между основным компьютером NVIDIA Jetson Xavier NX и Teensy 4.0 было решено использовать предоставляемый фреймворком ROS инструмент roserial. Данный инструмент позволяет при помощи Arduino-совместимой библиотеки и подключения по серийному порту наладить полноценную в рамках ROS коммуникацию в режиме реального времени без необходимости вручную описывать взаимодействие между двумя компьютерами.

Идея взаимодействия будет следующая:

- 1) На основном компьютере запускается ROS, который при помощи roserial устанавливает соединение с Teensy
- 2) Микроконтроллер считает количество пришедших электрических сигналов
- 3) Каждый ROS цикл публикуется количество подсчитанных сигналов
- 4) Узел на стороне главного компьютера принимает и обрабатывает данные числа для подсчёта местоположения робота

Реализация скетча представлена в Листинге ...

### Листинг 3. Формат сообщения nav\_msgs/Odometry

```
std_msgs/Header header
string child_frame_id
geometry_msgs/PoseWithCovariance pose
geometry_msgs/TwistWithCovariance twist
```

После завершения работы мои проверки не показали расхождений в значении подсчитанных оборотов ведущих колёс робота и я посчитал данную задачу завершённой.

## Список литературы

- [1] FructCode. Фреймворк - важный инструмент программиста. Обзор HTML/CSS, PHP и Python-фреймворков. - URL: <https://fructcode.com/ru/blog/features-of-popular-frameworks-html-css-php-and-python-frameworks/> (дата обращения: 16.03.2020). - Текст: электронный.
- [2] Lentin Joseph. Mastering ROS for Robotics Programming: Second Edition / Lentin Joseph, Jonathan Cacace. - Birmingham - Mumbai: Packt Publishing Ltd, 2018. - 552 с. - Текст: непосредственный. - с. 7, 20.
- [3] Hefei Fidurobot Store. Лазерный Радар-сканер EAI YDLIDAR X4 LIDAR модуль датчика диапазона 10 метров 5 кГц Диапазон частоты EAI YDLIDAR-X4 для ROS. / AliExpress - URL: <https://aliexpress.ru/item/32908156152.html> (дата обращения: 16.03.2020). - Текст: электронный.
- [4] Lentin Joseph. Mastering ROS for Robotics Programming / Birmingham - Mumbai: Packt Publishing Ltd, 2015. - 451 с. - Текст: непосредственный. - с. 270