

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименования института полностью)

Кафедра «Промышленная электроника»  
(наименование)

11.03.04 Электроника и наноэлектроника  
(код и наименование направления подготовки, специальности)

Промышленная электроника  
(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Система поиска пути мобильного робота»

Студент	<u>С.Н. Стычев</u> (И.О. Фамилия)	_____	(личная подпись)
Руководитель	<u>к.т.н., Е.С. Глибин</u> (ученая степень, звание, И.О. Фамилия)	_____	
Консультант	<u>к.п.н., доцент, Кириллова А.В. Кириллова</u> (ученая степень, звание, И.О. Фамилия)	_____	

Тольятти 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

Кафедра Промышленная электроника  
(наименование)

**ЗАДАНИЕ**  
**на выполнение бакалаврской работы**

Студент Стычев Семен Николаевич, Элб-1601а

1. Тема Система поиска пути мобильного робота

2. Срок сдачи студентом законченной бакалаврской работы «14» июня 2020 г.

3. Исходные данные к бакалаврской работе \_\_\_\_\_

Обозреваемая и разрабатываемая система поиска пути мобильного робота должна строиться на известных промышленных решениях в сфере автономных мобильных роботов, а также отвечать всем стандартам безопасности и критериям современных роботизированных систем.

4. Содержание бакалаврской работы (перечень подлежащих разработке вопросов, разделов)

Аннотация

Введение

1 Анализ рынка и актуальность разработки

2 Выбор и обоснование решения для навигации робота

2.1 Выбор микроконтроллера

2.2 Выбор датчика расстояния

2.3 Выбор двигателей постоянного тока

2.4 Выбор драйвера управления двигателями постоянного тока

2.5 Выбор датчика акселерометра-гироскопа

3 Разработка структурной схемы

4 Разработка электрической схемы соединений

5 Разработка алгоритма движения робота

5.1 Описание работы с датчиком расстояния

5.2 Управление двигателями платформы

5.3 Описание работы с датчиком акселерометра-гироскопа

5.4 Определение перекрестков лабиринта

5.5 Способы задания маршрута

5.6 Алгоритм работы платформы

6 Результаты экспериментальных испытаний

7 Безопасность и экологичность проекта

8 Экономическая эффективность

Заключение

Список используемой литературы

5. Ориентировочный перечень графического и иллюстративного материала

1. Структурная схема устройства

2. Схема электрическая соединений

3. Вариант сооружения с поворотами и перекрестками

4. Блок-схема алгоритма программы

5. Таблица показаний датчиков и логика определения перекрестков и поворотов

6. Блок-схема алгоритма программы

6. Консультант по разделам А.В. Кириллова

7. Дата выдачи задания «28» февраля 2020 г.

Руководитель бакалаврской работы

\_\_\_\_\_  
(подпись)

Е.С. Глибин

(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

Кафедра Промышленная электроника  
(наименование)

**КАЛЕНДАРНЫЙ ПЛАН**  
**выполнения бакалаврской работы**

Студент С.Н. Стычев

по теме Система поиска пути мобильного робота

Наименование работ	Плановый срок выполнения	Фактический срок выполнения	Отметка о выполнении	Подпись руководителя / консультанта
Анализ актуальности проекта, написание введения и обзорной главы.	21.02.2020	20.02.2020		
Разработка структурной схемы разрабатываемого устройства.	13.03.2020	13.03.2020		
Разработка схемы электрической соединений.	24.04.2020	24.04.2020		
Составление алгоритма, написание программы и оформление остальных разделов работы.	15.05.2020	12.05.2020		
Оформление пояснительной записки и плакатов.	10.06.2020	10.06.2020		

Руководитель бакалаврской работы

\_\_\_\_\_

(подпись)

Е.С.Глибин

(И.О. Фамилия)

## Аннотация

Выпускная работа состоит из введения, восьми глав, заключения, 4 таблиц, 41 рисунка, списка литературы, включая зарубежные источники. Объем выпускной работы составил 63 страницы.

Объектом выпускной работы является разработка мобильной робототехнической платформы.

Характерной чертой данной разработки является выполнение задачи прохождения робототехнической платформой пути по заданному маршруту с использованием ультразвуковых датчиков расстояния.

Работа начинается с постановки задачи, целей и определения актуальности темы, а затем происходит переход к возможным решениям.

Целью работы является разработка колёсной роботизированной платформы и алгоритма поиска пути в сооружении с поворотами и перекрестками с учетом заданного маршрута.

Выпускная работа может быть разделена на следующие разделы: выбор и обоснование каждого отдельно взятого элемента, разработка структурной схемы и перечня оборудования, разработка схемы электрической соединений, экспериментальная часть, проверка безопасности и экологичности проекта, экономический расчет. Основная часть посвящена написанию программы для микроконтроллера и блок-схемы алгоритма с подробным описанием.

Сначала описываются существующие решения. Сегодня в мире активно ведутся поиски решений, позволяющих создавать мобильные робототехнические комплексы, имеющие возможность определять свое положение и находить путь в замкнутом пространстве. Некоторые компании уже имеют опыт применения робототехнических платформ с высокой степенью автономности в различных сферах. Отмечается, что поиск решения активнее всего ведется в области складской логистики.

Проблемы позиционирования и поиска пути подробно описаны в основной части работы. Платформа должна уметь считывать показания датчиков, делать выводы о своем текущем местоположении и двигаться согласно маршруту. Представлен вариант сооружения с поворотами и перекрестками, в котором ориентируется робот. Помимо этого, роботизированная платформа оснащена камерой и одноплатным компьютером Raspberry Pi 2 Model B, при помощи которых распознает дорожные знаки. Работа является комплексной, поэтому часть действий здесь не описана.

Экспериментальная часть работы показывает результаты тестирования алгоритма поиска пути. Полученные результаты соответствуют поставленной цели.

## ABSTRACT

The graduation work consists of an introduction, eight parts, a conclusion, four tables, forty one pictures, list of references including foreign sources.

The object of the senior thesis is the development of a mobile robotic platform.

The key issue of the development is the fulfillment of the task of a robot traversing a path along a given route using ultrasonic distance sensors.

We start with the statement of the problem, objectives and determining the relevance of the topic. The aim of the work is to develop a wheeled robotic platform and an algorithm for finding the path in a structure with turns and crossroads.

The graduation work may be divided into several parts which are: selection and explanation of each individual element, development of a structural diagram and a list of equipment, development of electrical connections, check of safety and environmental friendliness of the project, economic calculation. The general part of the project gives details about writing a program for the microcontroller and flowcharts with a detailed description.

We first discuss existing solutions. At the moment, the world is actively looking for solutions to create mobile robotic systems that can navigate in a confined space. The author points out that the developed solution can be widely used in the field of warehouse logistics.

The issues of positioning and finding path are highlighted in the project's general part. The platform should be able to read the readings of sensors and move according to the route. We also present a variant of the structure with turns and intersections, in which the robot is oriented. In addition, the robotic platform is equipped with a camera that recognizes traffic signs.

Finally, we report the results of experiments conducted to explore ways to path finding. This result is consistent with the aim.

## Содержание

Введение.....	3
1 Анализ рынка и актуальность разработки.....	7
2 Выбор и обоснование решения для навигации робота .....	10
2.1 Выбор микроконтроллера .....	13
2.2 Выбор датчика расстояния.....	18
2.3 Выбор двигателей постоянного тока .....	23
2.4 Выбор драйвера управления двигателями постоянного тока ....	24
2.5 Выбор датчика акселерометра-гироскопа.....	27
3 Разработка структурной схемы .....	28
4 Разработка электрической схемы соединений.....	31
5 Разработка алгоритма движения робота.....	34
5.1 Описание работы с датчиком расстояния .....	34
5.2 Управление двигателями платформы .....	36
5.3 Описание работы с датчиком акселерометра-гироскопа.....	39
5.4 Определение перекрестков лабиринта .....	44
5.5 Способы задания маршрута .....	47
5.6 Алгоритм работы платформы.....	49
6 Результаты экспериментальных испытаний .....	54
7 Безопасность и экологичность проекта .....	56
8 Экономическая эффективность .....	58
Заключение .....	59
Список используемой литературы .....	61



## Введение

Роботы – автоматические системы, программируемые с помощью компьютеров и предназначенные для воспроизводства двигательных и интеллектуальных функций человека. Они могут быть автономными, полуавтономными и управляться людьми с помощью компьютеров и других устройств. Отличаются от традиционных автоматов большей самостоятельностью и способностью адаптироваться к окружающей среде [1].

Робототехника - это пересечение науки, техники и технологий, которые производят роботов. По мере развития технологий растет и объем робототехники. В 2005 году 90% всех роботов помогало собирать автомобили на заводах. Эти роботы состоят в основном из механических рычагов, выполняющих сварку или завинчивание определенных частей автомобиля. Сегодня мы видим усовершенствованное и расширенное определение робототехники, которое включает в себя разработку, создание и использование ботов, которые исследуют самые суровые условия Земли, роботов, которые помогают правоохранительным органам, и даже роботов, которые помогают почти во всех аспектах здравоохранения. Зачастую роботов стали использовать для выполнения вредной и монотонной работы, чтобы упростить и обезопасить жизнь человека [2].

Как и все молодые отрасли науки, робототехника сталкивается со многими трудностями. Специалистам приходится разрабатывать каждую систему практически с нуля, имея минимум наработок. Это связано с тем, что пока нет каких-то общепринятых стандартов. С подобной ситуацией столкнулись разработчики электронно-вычислительных машин много лет назад.

Однако это не уменьшает количество разработчиков и суммы денег, вкладываемые в разработку робототехнических систем. Это по-прежнему одна из наиболее активно развивающихся отраслей. Поэтому можно сказать,

что в ближайшее время интерес человечества к этой сфере будет только расти, и, вместе с тем, роботы будут становиться более доступными для пользования в быту.

Индустрия робототехники еще относительно молода, но уже добилась удивительных успехов. От самых глубоких глубин наших океанов до высот космического пространства можно найти роботов, выполняющих задачи, о которых люди не могли и мечтать.

Робот обладает рядом постоянных характеристик:

Все роботы состоят из какой-то механической конструкции. Механика позволяет ему выполнять задачи в среде, для которой он предназначен.

Роботам нужны электронные компоненты, которые контролируют и приводят в действие оборудование. Аккумуляторы нужны роботам для работы, а датчики для реагирования на изменения в окружающей среде.

Помимо этого, роботы содержат хоть какой-то уровень компьютерного программирования. Без набора кода, указывающего, что делать, робот будет просто еще одним элементом простого механизма. Вставка программы в робота дает ему возможность знать, когда и как выполнять задачу.

Важнейшие классы роботов широкого назначения — манипуляционные и мобильные роботы.

Манипуляционный робот — автоматическая машина (стационарная или передвижная), состоящая из исполнительного устройства в виде манипулятора, имеющего несколько степеней подвижности, и устройства программного управления, которая служит для выполнения в производственном процессе двигательных и управляющих функций. Такие роботы производятся в напольном, подвесном и порталном исполнениях (рисунок 1). Получили наибольшее распространение в машиностроительных и приборостроительных отраслях.



Рисунок 1 – Применение манипуляционных роботов в автомобилестроении

Мобильный робот — автоматическая машина, в которой имеется движущееся шасси с автоматически управляемыми приводами. Такие роботы могут быть колёсными, шагающими и гусеничными (существуют также ползающие, плавающие и летающие робототехнические системы) [3].

Подобные роботы постепенно внедряются во многие сферы человеческой деятельности. Сейчас робот-пылесос (рисунок 2) продается практически в любом магазине электроники.



Рисунок 2 – Робот-пылесос Xiaomi Mi Robot

Такие роботы ориентируются в пространстве, определяя препятствия вокруг себя и выстраивая свой маршрут должным образом.

Подобные роботы, как правило, оснащены датчиками расстояния. Информация, полученная с датчиков, обрабатывается с помощью встроенного микроконтроллера и надвигающиеся части робота-пылесоса подаются управляющие сигналы.

Но если нужно не объезжать препятствия, а следовать вдоль них, поворачивая в нужном направлении, заранее заданном пользователем? Для успешной навигации в пространстве робот должен правильно интерпретировать сведения об окружающем мире, получаемые датчиками, отслеживать угол поворота, считать перекрестки и уметь правильно управлять моторами.

## **1 Анализ рынка и актуальность разработки**

Складская робототехника - это использование автоматизированных систем, роботов и специализированного программного обеспечения для транспортировки материалов, выполнения различных задач и оптимизации автоматизации складских процессов. Это направление является актуальным в современной мире постоянно растущих человеческих потребностей в товарах и услугах. В последние годы робототехника завоевала известность в цепочках поставок, распределительных центрах и кругах управления складами и продолжает играть важную роль в автоматизации склада.

Первые роботы, способные перевозить грузы, были использованы в производстве еще в прошлом веке. Джордж Девол подал первый патент на робототехнику в 1954 году, а его компания Unimation произвела первого промышленного робота в 1956 году [5].

Первые роботы были большими роботизированными манипуляторами, движения которых можно было запрограммировать. Изобретатели понимали, что людям не понравятся машины, занимающие рабочие места. Поэтому они сначала сосредоточились на переводе опасных или вредных рабочих мест на роботизированные технологии, таких как сварка и подъем тяжелой техники.

На сегодняшний день уже несколько крупных промышленных компаний имеют опыт разработки и использования автоматизированных складских роботов.

Китайская компания Syrius Robotics запустила робота (рисунок 1.1.), использующего сразу несколько разных датчиков, благодаря которым воспринимается окружающая среда. Это позволяет роботам запускать алгоритм объединения нескольких датчиков и воспринимать окружающую среду.



Рисунок 1.1 – роботизированная платформа компании Syrius Robotics

Основные характеристики:

- Активная бинокулярная 3D камера и камера с объективом «рыбий глаз» в голове AMR;
- Ультразвуковой датчик, однолинейный инфракрасный дальномер ToF и датчик LiDAR в корпусе AMR;
- Программное обеспечение для управления автопарком и облачные сервисы;
- Срок службы батареи: 10 часов; Время зарядки: 2 часа;
- Максимальная полезная нагрузка: 50 кг;
- Функции безопасности: кнопки аварийной остановки;
- Размер: 600ммx420ммx1200мм;
- Вес: 45 кг [6].

В 2012 году Amazon потратила 775 миллионов долларов на покупку молодой робототехнической компании под названием Kiva Systems, которая передала ей право собственности на новое поколение мобильных роботов, которые могли переносить полки с товарами от рабочего к рабочему, читая штрих-коды на местах для дальнейших указаний.

Но это также дало Amazon техническую базу, на которой она могла бы создавать новые версии складской робототехники на долгие годы, создавая основу для потенциального будущего, где единственные люди на предприятиях Amazon - это те, кто занят обслуживанием и ремонтом этих роботов (рисунок 1.2).



Рисунок 1.2 – Роботизированная платформа для перевозки грузов на складе Amazon

На сегодняшний день Amazon имеет более 200 000 мобильных роботов, работающих внутри своей складской сети, наряду с сотнями тысяч людей. Большое количество роботов помогло компании выполнить свои постоянно растущие обещания скорых поставок клиентам Amazon Prime [7].

Таким образом, разработкой автономных складских роботизированных платформ занимаются инженеры со всего мира. Данное направление развития необходимо для того, чтобы сократить количество людей, работающих на складах и других крупных производственных помещениях.

## **2 Выбор и обоснование решения для навигации робота**

Существует две тенденции развития: одна гласит, что проект любой автономной системы должен начинаться с точного определения ее задачи, а другая – что робот должен быть способен выполнять любую задачу в различных средах и ситуациях. Сейчас чаще всего робототехнические системы выполняют именно заранее определенные задачи. Это обусловлено тем, что себестоимость робота общего назначения высока, поскольку для этого требуются сложные механические, электронные и вычислительные системы.

Навигация - это наука управления мобильным роботом. Проблемы, связанные с навигацией, могут быть кратко определены в трех вопросах: «Где я нахожусь», «Куда я иду» и «Как туда добраться». Если разработать систему, которая будет состоять из датчиков, своевременно поставляющую информацию об окружающем мире роботу, то ответы на первые два вопроса будут получены. Что касается, третьего вопроса, здесь недостаточно просто получать данные, нужно хорошо поработать над определением изменения положения робототехнической системы в пространстве. То есть разработать алгоритм действий и реакцию на эти действия. Выбор навигационной системы зависит от среды и задач робота. Навигацию можно разделить на три системы: система на основе координат, система на основе поведения и гибридная система.

Координатная система основана на знании своего положения в глобальной системе координат окружающей среды. Такими методами являются картирование, навигация по сетке заполняемости и потенциальные поля.

Система, основанная на поведении, требует, чтобы робот распознавал особенности окружающей среды через свои датчики и использовал собранную информацию для поиска своих целей. Например, робот должен



уметь распознавать двери и коридоры и знать действия, которые приведут его в нужное место. В этом случае система координат является локальной.

Мобильные роботы могут перемещаться с использованием относительного или абсолютного измерения положения. Относительное позиционирование использует одометрию или инерциальную навигацию. Одометрия – простая и недорогая навигационная система, однако его недостатком является накапливание ошибок. Инерциальная навигация использует меры вращения и ускорения для получения информации о местоположении. Наиболее важной проблемой движения мобильного робота на основе неабсолютных датчиков является накопление ошибок, обнаруженных в относительных датчиках [8].

Система абсолютного позиционирования в отличие от относительной использует геометрические соотношения между положением робота и объектов окружающей среды. Эту систему можно поделить на несколько подтипов: определение положения с помощью активных маяков, ориентация по магнитным компасам и глобальные способы определения положения.

Магнитный компас – это датчик, который способен ориентироваться без создания или поиска каких-либо специальных объектов вокруг себя. Это возможно благодаря тому, что он использует естественное электромагнитное поле Земли. Это и является его преимуществом, ведь можно обойтись без изменений в окружающей среде. Однако недостатком является то, что металл и линии электропередач вносят помехи в показания датчика.

Активные навигационные системы маяков являются наиболее распространенными навигационными средствами на судах и самолетах, а также на коммерческих мобильных роботизированных системах. Активные маяки предоставляют точные сведения о местоположении в короткий срок. В результате этот подход обеспечивает высокую частоту дискретизации и высокую надежность. Но недостатком является высокая стоимость установки и обслуживания таких маяков.

Аналогичным образом, система ориентиров использует характеристики окружающей среды для оценки своего положения. Преимуществом такой системы является доступность ориентиров, ведь они могут быть уже созданы природой или человеком. Это может быть куст, холм или элемент домашнего интерьера, например, дверь. Помимо этого, можно использовать и такие ориентиры как дорожные знаки.

Наконец, метод сопоставления моделей использует особенности среды для построения карты или для распознавания среды в ранее известной карте. Основные проблемы связаны с поиском соответствия между локальной картой, обнаруженной с помощью датчиков робота, и известной глобальной картой.

Другой подход для навигации мобильного робота - это биомиметическая навигация. Многие считают, что классические навигационные методы, разработанные в последние десятилетия, не достигли того же успеха, что и навигационные механизмы муравьев или пчел. Это привело исследователей к изучению и внедрению навигационного поведения насекомых и животных.

Мобильным роботам нужна информация о мире, чтобы они могли относиться к окружающей среде, как животные. Для этой цели они используют различные датчики, которые преобразуют данные об окружающей среде в электрический сигнал. Эти сигналы представляют собой электрические данные, которые представляют состояние мира и должны интерпретироваться роботом для достижения его целей. Для этого используется широкий спектр датчиков.

По типу получаемой информации датчики можно разделить на две группы: проприоцептивные или экстероцептивные. Проприоцептивные датчики связаны с внутренними элементами робота. Они контролируют состояние его внутренних механизмов и устройств, в том числе положения двигателей. А экстероцептивные датчики собирают информацию из среды, в которой находится робот, и, как правило, связаны с навигацией и

применением робота. С точки зрения метода измерения датчики подразделяются на активные и пассивные датчики. Активные датчики подают определенный сигнал в окружающую среду и получают каким-то образом измененный сигнал. А пассивные датчики не посылают сигналы, так как они способны получать «сигналы», естественно излучаемые окружающим миром. Датчики также могут быть классифицированы по электрическому выходному сигналу: на цифровые и аналоговые датчики. В целом, данные, полученные с помощью датчиков, обычно неточны, что повышает сложность использования информации, предоставленной ими [9].

Выбор датчика определяется различными аспектами, основными из которых являются: цели робота, точность моделей робота и среды, неопределенность данных датчика, общая стоимость устройства, количество собранной информации, время, доступное для обработки данных, физические размеры датчиков.

Можно использовать систему позиционирования, при которой все датчики сосредоточатся на подвижной платформе, положение которой нас интересует. Для этого можно совместить ультразвуковые датчики расстояния, для оценки расстояния от платформы до стен, и датчик акселерометра-гироскопа для отслеживания угла поворота.

## **2.1 Выбор микроконтроллера**

На сегодняшний день на рынке представлено большое количество различных микроконтроллеров. Но многие из них имеют достаточно высокую стоимость, поэтому слишком дорогие варианты не рассматривались.

Платформа Netduino (рисунок 2.1.1) построена на микроконтроллерах класса ARM с запрограммированным программным кодом .NET Framework.

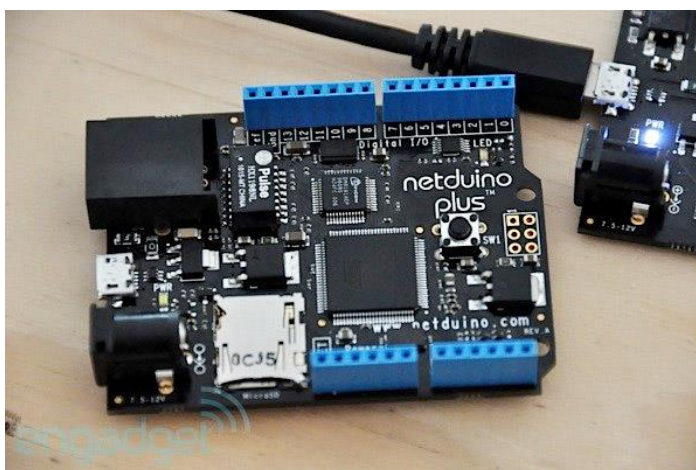


Рисунок 2.1.1 – микроконтроллер Netduino

Сегодня можно найти достаточное количество модификаций подобных плат, но существенным недостатком является то, что это не бюджетный вариант.

Микроконтроллер Ti MSP430 LaunchPad (рисунок 2.1.2) – это достойный внимания микроконтроллер, который стоит отнести к малобюджетной серии устройств. При этом заметьте, что это не китайская копия, а вполне себе самостоятельное решение.

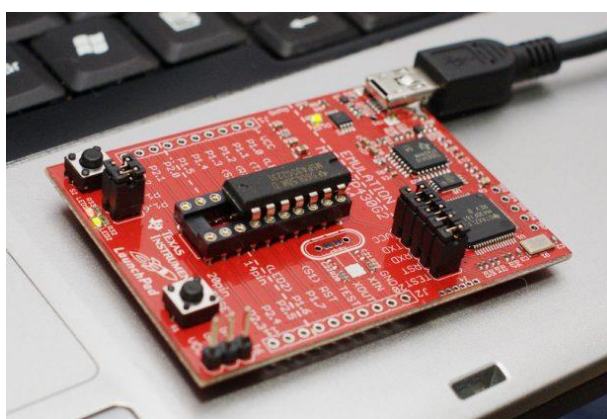


Рисунок 2.1.2 – микроконтроллер Ti MSP430 LaunchPad

Платформа LaunchPad имеет возможность использования различных плат расширений, предназначенных для увеличения функционала.

Еще одним хорошим вариантом является микроконтроллер Teensy(рисунок 2.1.3), что в переводе с английского означает «крошечный». Он имеет небольшие размеры и схож с Arduino. Teensy основаны на микропроцессоре серии Freescale ARM Cortex-M4.



Рисунок 2.1.3 – микроконтроллер Teensy

Teensy построен на базе Arduino IDE, что позволяет использовать большое количество библиотек, предназначенных для микроконтроллера Arduino. Доступны для установки add-on платы, если вы планируете их использовать.

Еще одним популярным микроконтроллером является Particle Photon (рисунок 2.1.4). Его особенность заключается в том, что он уже имеет встроенный модуль Wi-Fi, что позволяет работать с ним даже с помощью мобильных устройств на базе Android, iOS и Windows.

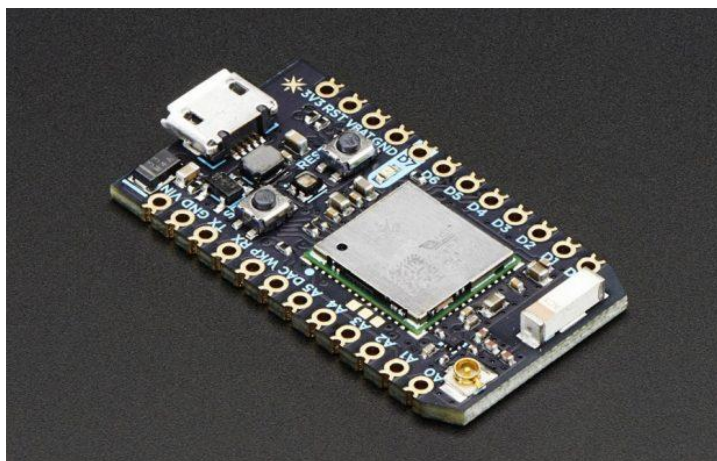


Рисунок 2.1.4 – микроконтроллер Particle Photon

Рассмотренные варианты отличаются высокой производительностью и большим количеством доступных плат расширений и устанавливаемых библиотек [10].

Одним из наиболее популярных микроконтроллеров среди новичков является Arduino (рисунок 2.1.5). Он имеет множество модификаций и плат расширений, обладает огромным сообществом, постоянно публикующим полезные библиотеки в свободный доступ.

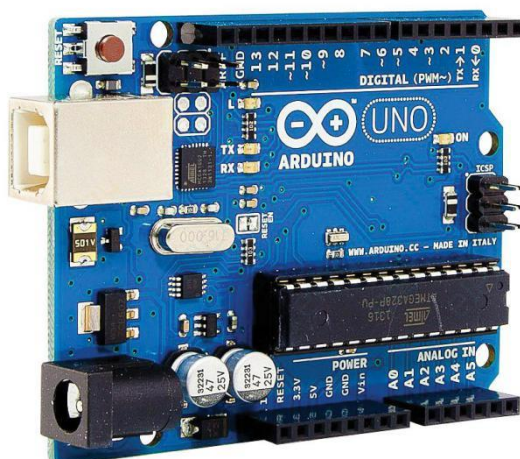


Рисунок 2.1.5 – микроконтроллер Arduino

Данный микроконтроллер программируется на языке C++. Платы Arduino оснащены наборами цифровых и аналоговых выводов. Платы

оснащены интерфейсами последовательной связи, включая универсальную последовательную шину (USB) на некоторых моделях, которые также используются для загрузки программ с персональных компьютеров [11].

Сложные вычисления, связанные с распознаванием дорожных знаков, выполняются на одноплатном компьютере Raspberry Pi 2 Model B. Поэтому было решено использовать именно микроконтроллер Arduino. Его производительности вполне достаточно для считывания показаний датчиков, а большой выбор плат расширений позволяет легко управлять моторами. А стоимость микроконтроллера Arduino ниже, чем стоимость практически любой подобной платы.

Линейка плат Arduino включает в себя множество модификаций. Выбор осуществлялся между моделями Arduino Nano, UNO R3 и Mega. Их стоимости практически одинаковы, поэтому можно не брать их во внимание. Основные характеристики представлены в таблице 2.1.1:

Таблица 2.1.1 – Характеристики Arduino Mega, UNO R3 и Nano.

Характеристики	Arduino Mega	Arduino UNO	Arduino Nano
Микроконтроллер	ATmega1280	ATmega328	ATmega328
Тактовая частота	16 МГц	16 МГц	16 МГц
Аналоговые контакты	8	8	8
Цифровые контакты	54	22	22
Диапазон аналоговых значений	От 0 до 5 В	От 0 до 5 В	От 0 до 5 В
Есть ли защита от коротких замыканий	Да	Да	Нет

Из таблицы 2.1.1 видно, что Arduino Mega отличается большим количеством контактов, однако 22 контактов для данного проекта вполне достаточно. Основное преимущество Arduino Nano в маленьких размерах. Но решающую роль в выборе модели сыграла защита от коротких замыканий. В

процессе разработки и тестирования робототехнической платформы, нужно быть уверенным в том, что если произойдет разрыв изоляции проводов, микроконтроллер самостоятельно отключится, а не выйдет из строя. В Arduino UNO R3 в отличие от Arduino Nano такая защита есть. Таким образом, для работы была выбрана плата Arduino UNO R3.

## 2.2 Выбор датчика расстояния

Для выполнения задач, связанных с определением роботизированной платформой поворотов и перекрестков, было решено использовать датчики расстояния или приближения. Было рассмотрено несколько вариантов, среди которых можно выделить три датчика.

Датчик препятствия YL-63 (рисунок 2.2.1) испускает инфракрасное излучение и принимает его. Используя переменный резистор можно изменить чувствительность и изменить расстояние срабатывания. Управлять им довольно просто – два выхода питания (плюс VCC и минус GND) и выходной сигнал OUT, который показывает появление объекта в зоне действия датчика.

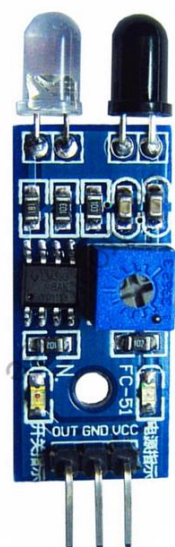


Рисунок 2.2.1 – Датчик препятствия YL-63



Датчик собран на чипе LM393 (компаратор) с относительно низким энергопотреблением. Напряжение питания составляет от 3,3 до 5 В. Ширина зоны чувствительности составляет около 35 градусов, а обнаружение препятствия происходит на расстоянии до 4 метров. Размеры датчика составляют 43 x 16 x 7 мм. Стоимость датчика начинается с 30 рублей. Датчик хорошо подходит для обнаружения препятствий, но в данной работе для плавного и ровного движения робототехнической платформы необходимо не только определять, есть ли вблизи стена, но также необходимо измерять расстояние до этой стены, чтобы регулировать скорость моторов. Поэтому датчик препятствий YL-63 в данном случае не подходит [12].

Для измерения расстояний можно использовать инфракрасный датчик расстояния GP2Y0A21YK0F фирмы Sharp. Датчики расстояния Sharp являются популярным выбором для многих проектов, которые требуют точных измерений расстояния. Этот ИК-датчик обеспечивает гораздо лучшую производительность, чем другие ИК-датчики. Рассматриваемый датчик состоит из инфракрасного светодиода с линзой и фотоэлемента. Светодиод излучает узкий пучок света, который отражается от объекта и направляется через другую линзу на позиционно-чувствительный фотоэлемент. От того, в какую точку упадет отраженный луч, зависит проводимость фоторезистора. Проводимость преобразуется в напряжение и, к примеру, оцифровывая его аналого-цифровым преобразователем микроконтроллера, можно вычислить расстояние. Интерфейс подключения прост: аналоговый выход нужно подключить к аналого-цифровому преобразователю для измерения расстояния. Диапазон обнаружения составляет приблизительно от 10 см до 80 см.



Рисунок 2.2.2 – Инфракрасный датчик расстояния

Среди недостатков такого датчика можно выделить высокую стоимость (около 300 рублей), а также влияние освещения и типа препятствия на результаты. Показания датчика могут отличаться днем и ночью, поэтому роботизированная платформа должна иметь собственный источник освещения либо находиться в помещении с постоянным освещением [13].

Лазерный дальномер VL53L0X – еще один датчик, способный измерить расстояние до объекта. Он отличается от инфракрасного датчика GP2Y0A21YK0F тем, что способен измерять расстояния более точно и менее зависим от типа поверхности объекта.

Датчик VL53L0X (рисунок 2.2.3) стал широко известным после того, как он был впервые использован в iPhone 7 для 3D визуализации. Он может обнаружить объект в пределах своего диапазона, даже если поверхность объекта обладает высокой отражающей способностью. Используемый здесь лазер представляет собой лазер на поверхности с вертикальной полостью 940 нм, который невидим и безопасен для глаз человека. Встроенные ИК-фильтры увеличивают дальность действия, повышают устойчивость к окружающей среде и обеспечивают хорошую защиту от оптических перекрестных помех на стекле.



Рисунок 2.2.3 – Лазерный дальномер VL53L0X

Этот модуль имеет 4 контакта VCC, GND, SCL и SDA. Плата имеет линейный регулятор 2,8, поэтому она может выдерживать напряжение в диапазоне от 2,6 до 5,5 В. Он использует протокол I2C для связи, поэтому контакты Serial Clock Line (SCL) и Serial Data Address (SDA) предназначены для связи с микроконтроллером. Среди преимуществ модуля можно отметить его маленькие размеры и высокую точность измерения [14].

Складские помещения зачастую имеют большую площадь, а, следовательно, не равномерное и возможно тусклое освещение. Поэтому в качестве навигационной системы можно использовать ультразвуковые датчики расстояния, показания которых не зависят от освещения. Одним из таких датчиков является ультразвуковой дальномер HC-SR04 (рисунок 2.2.4).



Рисунок 2.2.4 – Ультразвуковой дальномер HC-SR04

По сути ультразвуковой датчик расстояния HC-SR04 состоит из двух ультразвуковых преобразователей. Он действует как передатчик, который преобразует электрический сигнал в ультразвуковые звуковые импульсы 40 кГц. Приемник прослушивает передаваемые импульсы. Если он принимает их, он генерирует выходной импульс, ширина которого может использоваться для определения расстояния, пройденного импульсом.

Датчик небольшого размера, прост в использовании в любом проекте робототехники и обеспечивает превосходное бесконтактное обнаружение дальности от 2 до 400 см с точностью до 3 мм. Поскольку он работает от 5 вольт, его можно подключить непосредственно к Arduino или любым другим 5V логическим микроконтроллерам.

Сравнить характеристики и сделать окончательный выбор в пользу одного из датчиков помогла таблица 2.2.1, в которой представлены характеристики описанных выше датчиков [15].

Таблица 2.2.1 – Сравнительная таблица характеристик датчиков расстояния.

Характеристики	GP2Y0A21YK0F	VL53L0X	HC-SR04
Рабочее напряжение	4.5 – 5.5 В	2.6 – 5.5 В	5 В
Потребляемый ток	30 мА	10 мА	15 мА
Диапазон измерения расстояния	10 – 80 см	до 200 см	2 – 400 см
Зависимость от освещения и типа поверхности	Да	Да	Нет
Цена	300	300	50

Таким образом, среди рассмотренных датчиков расстояния выделяется ультразвуковой датчик расстояния HC-SR04. К недостаткам можно отнести большие габариты и меньшую точность. Однако он дешевле аналогов в 6 раз,

а на его показания не влияет освещение и тип отражаемой поверхности. Поэтому в данной работе будет использоваться именно этот датчик.

### 2.3 Выбор двигателей постоянного тока

Для того чтобы привести мобильного робота в движение необходимы двигатели постоянного тока, способные вращать колеса. В таблице 2.3.1 представлены основные характеристики нескольких моделей моторов (рисунок 2.3.1), доступных в любом специализированном магазине.



Рисунок 2.3.1 – TT motorOperating слева и Motor N20 справа.

Таблица 2.3.1 – Сравнительная таблица характеристик двигателей постоянного тока

Характеристики	ТТ MotorOperating	Motor N20
Рабочее напряжение	3 – 12 В	3 – 6 В
Ток нагрузки	70 мА	40 мА
Скорость без нагрузки	40 об/мин	55 об/мин
Защита от помех	Да	Нет
Цена	120	150

Из таблицы характеристик видим, что стоимость и скорость первого мотора чуть меньше второго. Но преимуществом первого мотора является встроенная защита от помех, что очень важно, если моторы крутятся с большой скоростью. Помимо этого, данные моторы обладают специальными отверстиями для удобства их крепления на платформе. По этим причинам окончательный выбор был сделан в пользу мотора модели TT MotorOperating [16].

## **2.4 Выбор драйвера управления двигателями постоянного тока**

Микроконтроллер Arduino при многочисленных достоинствах имеет и некоторые недостатки, например, ограничение по силе тока присоединенной к ней нагрузки. Для всей платы лимит – это 800 мА, а для каждого отдельно взятого контакта – 40 мА. Поэтому подключить напрямую к Arduino даже самый маленький двигатель постоянного тока не представляется возможным.

Чтобы подключить двигатель постоянного тока, можно воспользоваться несколькими способами:

Можно использовать реле. Для этого нужно подключить двигатель к реле, а подачей команд с микроконтроллера можно управлять включением и отключением мотора. Схема подключения представлена на рисунке 2.4.1 и отличается простотой подключения.

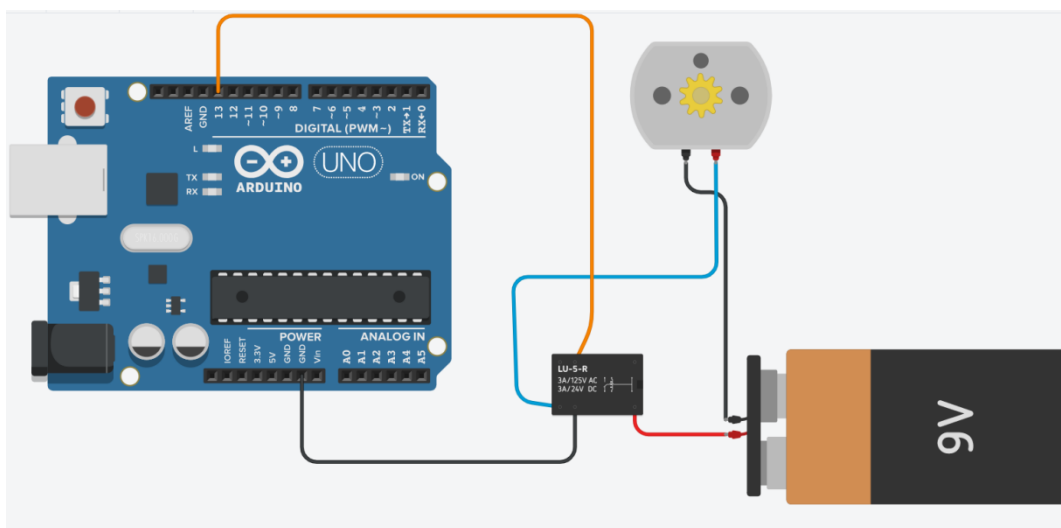


Рисунок 2.4.1 – Схема подключения двигателя постоянного тока с помощью реле

Недостатком такой схемы является то, что мы не сможем управлять скоростью и направлением вращения двигателей. Таким образом, робототехническая платформа не сможет двигаться назад и будет ехать с постоянной скоростью.

Поэтому было решено использовать специальную плату расширения для управления двигателями. Наиболее часто используемые драйверы управления двигателями относятся к сериям L293 и L298 (рисунок 2.4.2). Эти платы предназначены для одновременного управления двумя двигателями постоянного тока [17].

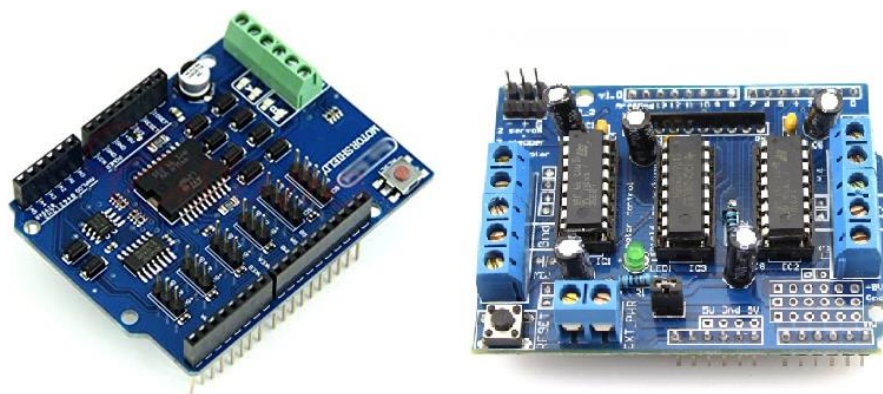


Рисунок 2.4.2 – Плата расширения для управления двигателями постоянного тока L298P слева и L293D справа

Платы очень схожи в подключении. Их конструкция такова, что контакты платы расширения соответствуют контактам микроконтроллера Arduino, поэтому чтобы их подключить, необходимо просто надеть плату на микроконтроллер сверху.

Основные характеристики указанных плат расширения представлены в таблице:

Таблица 2.4.1 – Сравнительная таблица характеристик плат расширения для управления двигателями

Характеристики	L298P	L293D
Максимально допустимое количество одновременно подключенных двигателей	2	4
Входное напряжение логики	5 В	5 В
Входное напряжение привода	4 – 24 В	4.5 – 36 В
Максимальное количество одновременно подключенных двигателей	2	4
Припаянные выводы готовы к использованию	Да	Нет
Цена	300	100

Так, плата L293D может управлять сразу 4-мя двигателями одновременно, когда драйвер L298P – только 2-мя. Цена такого драйвера в три раза меньше, чем у другой платы. Это можно объяснить тем, что L298P проще в использовании, так как все контакты уже припаяны и можно работать с платой сразу после подключения. А драйвер L293D имеет лишь специальные клеммы для подключения моторов и сервоприводов. А чтобы управлять обычными контактами микроконтроллера, необходимо допаять



нужные выводы. Однако не смотря на это неудобство, в остальном приобретение и использование данной платы показалось более целесообразным [18].

## 2.5 Выбор датчика акселерометра-гироскопа

Роботизированная платформа должна уметь поворачивать на нужный угол. Для того чтобы считывать текущий угол поворота необходим специальный датчик. Для этой роли подойдет датчик акселерометра-гироскопа. Имеется множество моделей данного типа датчиков. Они отличаются характеристиками, принципом работы, но самое главное – ценой. Целесообразно было выбрать наиболее дешевый датчик, потому что нам нужно лишь измерять угол в одной оси. С этой задачей отлично справится трех осевой акселерометр-гироскоп GY-521, представленный на рисунке 2.5.1.



Рисунок 2.5.1 – Датчик акселерометр-гироскоп GY-521

Подробное описание датчика и работа с ним описывается в разделе 5.3.

Таким образом, подбор элементов схемы происходил тщательным образом, были рассмотрены преимущества и недостатки каждого компонента. Обоснован выбор каждого элемента.

### 3 Разработка структурной схемы

Для выполнения задач, связанных с обработкой информацией и подачей сигналов, будет использован микроконтроллер Arduino. Микроконтроллер принимает сигналы с трех ультразвуковых датчиков, расположенных в передней части платформы: один слева, второй по центру и третий с правой стороны. Полученные данные обрабатываются микроконтроллером и на двигатели подаются определенные управляющие сигналы.

Но микроконтроллер Arduino ограничен по максимальному току, поэтому сверху к Arduino присоединяется плата расширения с драйвером L293D.

Во избежание помех, силовая и управляющая цепи разделены. Arduino питается напрямую от аккумулятора 5 вольт. А вторым источником питания являются аккумуляторы 3.7 вольт, представленные на рисунке 3.1. Они подключаются к плате контроля заряда аккумуляторов (рисунок 3.2), чтобы мобильный робот мог заряжаться даже во время работы.



Рисунок 3.1 – Литий-ионные аккумуляторы

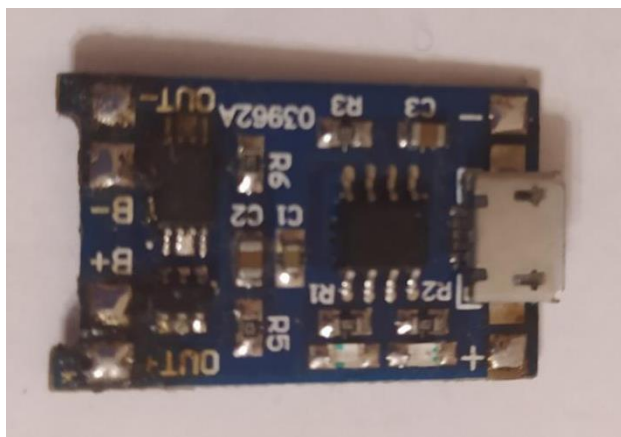


Рисунок 3.2 – Плата контроля заряда литий-ионных аккумуляторов

Однако напряжение 3.7 вольт недостаточно высокое для работы двигателей. Поэтому аккумуляторы подключаются к преобразователю напряжения, представленному на рисунке 3.3.



Рисунок 3.3 – Преобразователь напряжения XL6009

Для лучшего понимания связи и взаимодействия отдельных элементов была составлена структурная схема мобильного робота (рисунок 3.4).

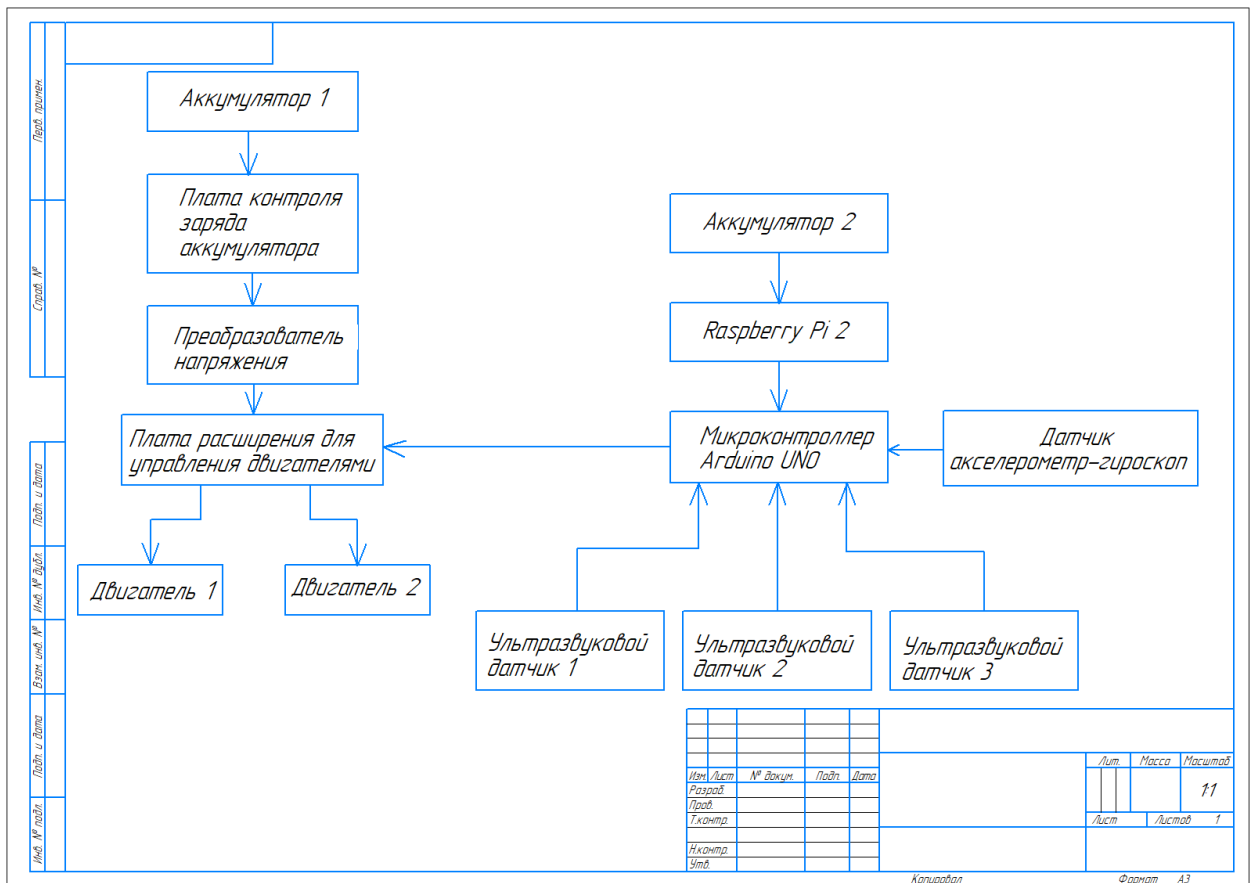


Рисунок 3.4 – Структурная схема мобильной робототехнической платформы

Таким образом, в данном разделе была разработана структурная схема мобильной робототехнической платформы и обоснован выбор отдельного питания для силовой и логической цепи.

#### **4 Разработка электрической схемы соединений**

Схема электрическая соединений включает в себя следующие элементы:

1. Микроконтроллер Arduino UNO R3;
2. Драйвер управления двигателями L293D;
3. Плата контроля заряда литий-ионных аккумуляторов TP4056;
4. Аккумуляторы 3.7 В;
5. Аккумулятор 5 В;
6. Регулируемый DC-DC преобразователь напряжения XL6009;
7. Двигатели постоянного тока;
8. Модуль акселерометра-гироскопа GY-521;
9. Ультразвуковые датчики HC-SR04.
10. Raspberry Pi 2 Model B

На рисунке 4.1 представлена схема электрическая соединений, спроектированная в системе КОМПАС-3D V18.

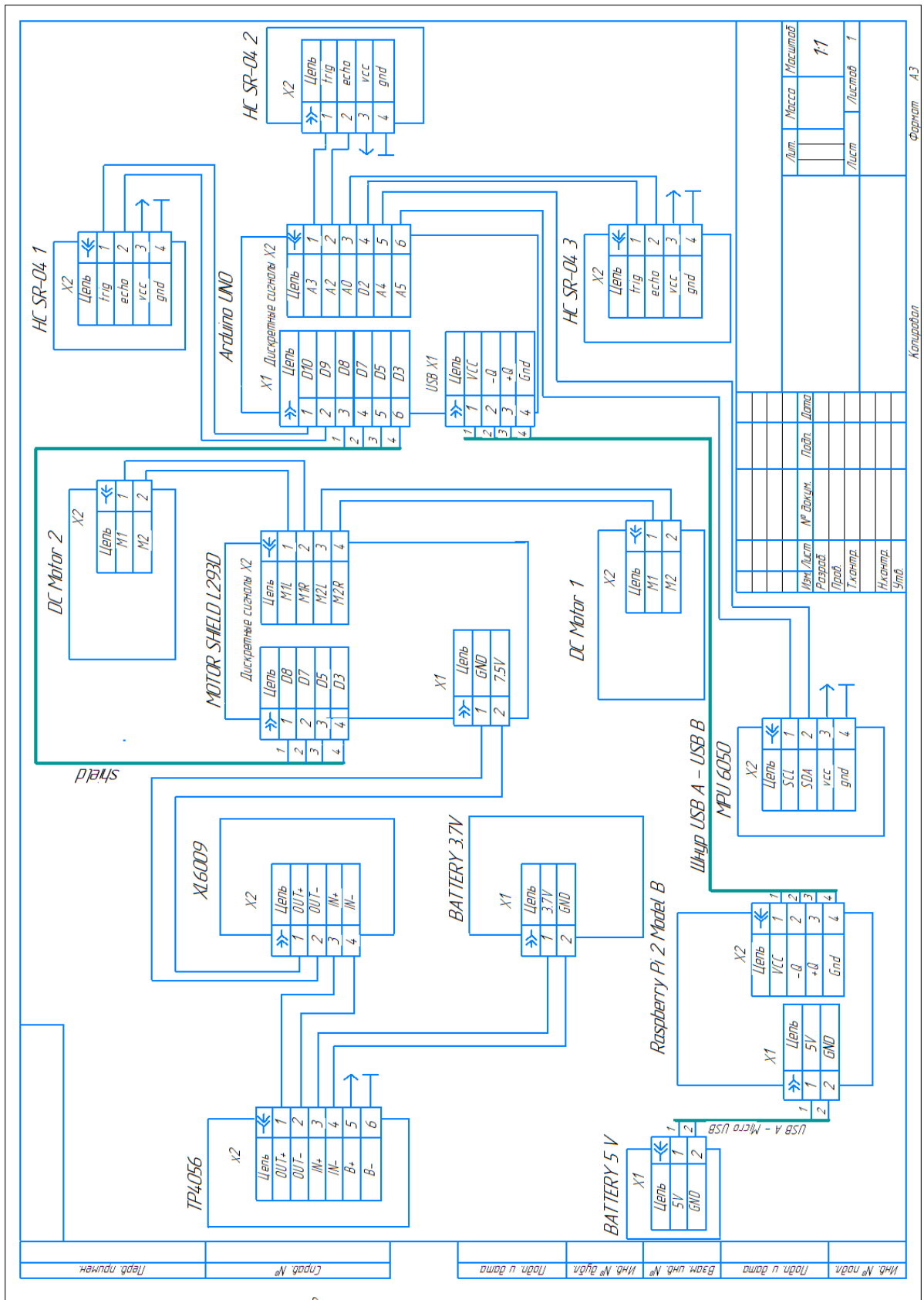


Рисунок 4.1. – Схема электрическая соединений

Соединение аккумулятор 5 В – Raspberry Pi 2 осуществляется при помощи шнура USB-A – micro-USB, а микроконтроллер подключается к одноплатному компьютеру при помощи шнура с контактами вилка USB A – вилка USB B. Это соединение необходимо для того, чтобы питать микроконтроллер Arduino UNO R3, а также для налаживания последовательной связи между одноплатным компьютером и микроконтроллером. Так как работа является комплексной, то часть компонентов, присоединяемых к одноплатному компьютеру Raspberry Pi 2 здесь не описывается.

Остальные соединения выполняются при помощи проводов.

В данном разделе был представлен перечень всех используемых электронных компонентов проекта и разработана схема электрическая соединений. Было обосновано подключение микроконтроллера к одноплатному компьютеру.

## 5 Разработка алгоритма движения робота

### 5.1 Описание работы с датчиком расстояния

В качестве измерителя расстояний до объектов в данной работе используются три ультразвуковых датчиков расстояния HC-SR04. Дальномер излучает ультразвук с частотой 40 000 Гц. Звук движется по воздуху до тех пор, пока не встретит на своем пути какой-либо объект. После столкновения он отражается и возвращается обратно к модулю (рисунок 5.1.1).

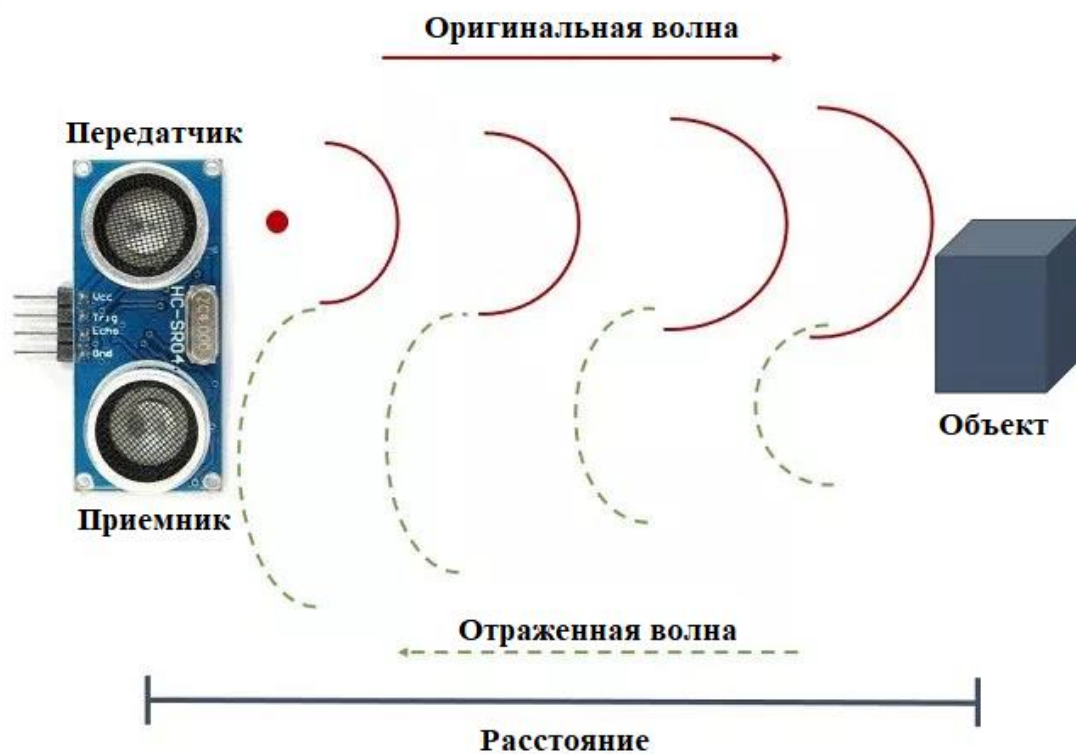


Рисунок 5.1.1 – Наглядное изображение передачи и приема ультразвуковой волны датчиком

Время между передачей и приемом сигнала позволяет рассчитать расстояние до объекта. Это возможно, потому что скорость звука в воздухе заранее известна.



Ультразвуковой датчик HC-SR04 имеет 4 контакта: GND, VCC, Trig и Echo. Контакты GND и VCC датчика должны быть подключены к GND и контакту 5 В на плате Arduino соответственно, а контакты trig и echo к любым доступным выводам микроконтроллера. Схема подключения представлена на рисунке 5.1.2.

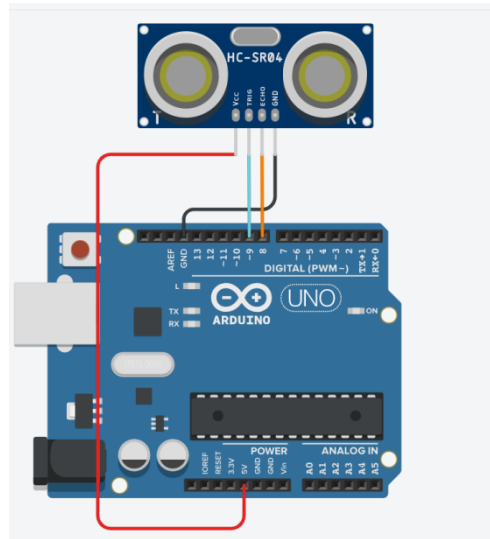


Рисунок 5.1.2 – Схема подключения ультразвукового датчика к микроконтроллеру

В начале программы необходимо установить триггерный вывод в качестве выхода, а вывод эхо-сигнала в качестве входа:

```
pinMode(trigPin, OUTPUT);
```

```
pinMode(echoPin, INPUT);
```

Затем мы подаем напряжение на контакт trig, тем самым отправляя импульс в течение 10 микросекунд. Но перед этим подаем короткий НИЗКИЙ импульс, чтобы убрать возможные помехи:

```
digitalWrite(trigPin, LOW);
```

```
delayMicroseconds(5);
```

```
digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);
```

Затем с помощью специальной функции мы получаем время в микросекундах с момента отправки сигнала до приема его эхо-сигнала:

`duration = pulseIn(echoPin, HIGH);`

Наконец, вам просто нужно преобразовать продолжительность в расстояние. Мы можем рассчитать расстояние по следующей формуле (1):

$$S=V*T/2, \tag{1}$$

Где S – расстояние между датчиком и препятствием;

V – скорость движения звуковой волны;

T – время, за которое звук дошел до препятствия и вернулся обратно на модуль датчика.

В вышеприведенной формуле происходит деление на 2, так как ультразвук проходит путь до препятствия и затем такой же путь от препятствия до датчика [20].

## 5.2 Управление двигателями платформы

Плата расширения L293D – это четырехканальный драйвер, который позволяет работать с большими токами и напряжениями. С помощью микросхемы L293D можно управлять двигателями постоянного тока и использовать источники питания до 36 вольт, микросхема L293D также обеспечивает максимальный ток до 600 мА на канал. Микросхема L293D (рисунок 5.2.1) представляет собой разновидность H-мостов.



Рисунок 5.2.1 – Микросхема L293D

Н-мост - это электронная схема, которая позволяет приложить напряжение к нагрузке в противоположном направлении (рисунок 5.2.2). Эти схемы часто используются в робототехнике, чтобы вращать двигатели постоянного тока вперед или назад.

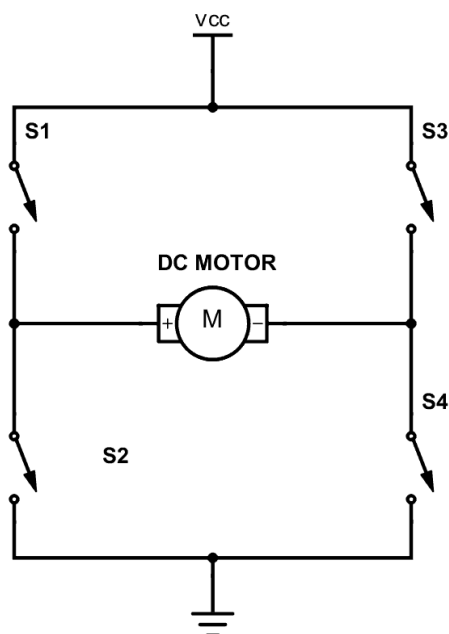


Рисунок 5.2.2 – Схема Н моста

Если замкнуть оба ключа S1 и S2, произойдет короткое замыкание. То же самое касается и ключей S3 и S4.

Если же замкнуть выключатели S1 и S4, то ток начнет протекать через двигатель слева направо. Если замкнуть S3 и S2, ток потечет справа налево. В этих двух условиях направление вращения моторов будет разным.

Именно этот принцип и используется в микросхемах, предназначенных для управления двигателями.

Для удобства подключения сразу две микросхемы L293D расположены на специальной плате расширения для микроконтроллера Arduino. Поэтому нет необходимости что-то припаивать, достаточно просто подключить двигатели в любые доступные клеммы для двигателей (рисунок 5.2.3).

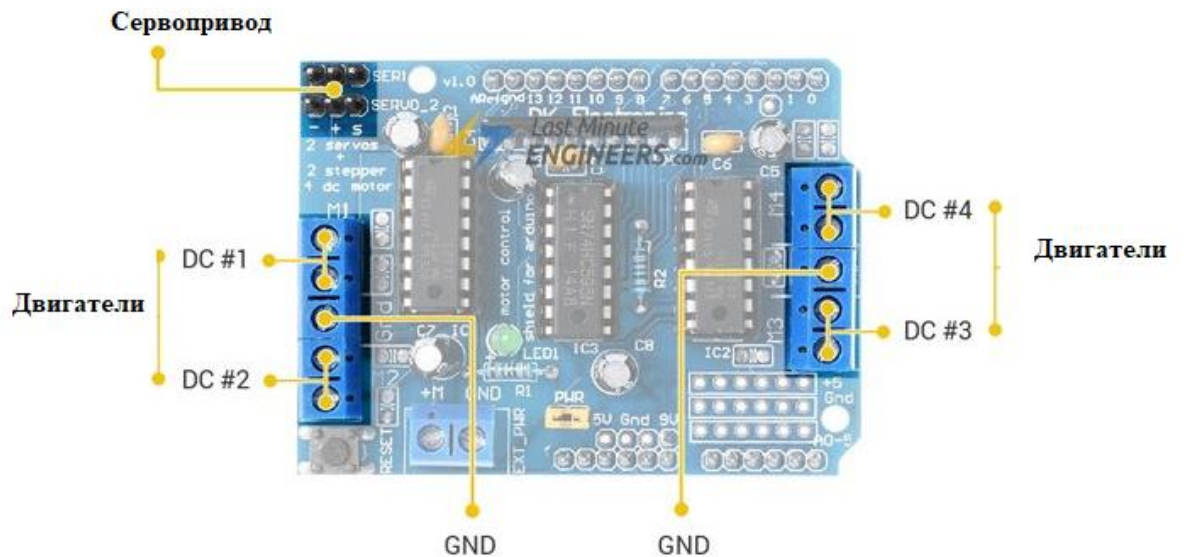


Рисунок 5.2.3 – Описание доступных клемм платы L293D

Выходные каналы обеих микросхем L293D выведены на край экрана с помощью двух 5-контактных винтовых клемм, а именно. DC #1, DC #2, DC #3 и DC #4. К этим клеммам можно подключить четыре двигателя постоянного тока напряжением от 4,5 до 25 В. Клемма GND нужна для того чтобы подключать однополярный шаговый двигатель.

Для управления платой нам необходимо установить и подключить библиотеку AFMotor.h, чтобы мы могли использовать простые команды для управления двигателями постоянного тока:

```
#include <AFMotor.h>
```

Затем нужно указать в программе, что мы подключили моторы к клеммам 2 и 3 и дать им названия, например, motorR и motorL:

```
AF_DCMotor motorR(3);
```

```
AF_DCMotor motorL(2);
```

Далее, чтобы управлять моторами достаточно знать лишь две простые функции:

```
motorL.run(FORWARD);
```

```
motorR.run(FORWARD);
```

```
motorL.setSpeed(255);
```

```
motorR.setSpeed(255);
```

Данный фрагмент кода предназначен для вращения двух двигателей постоянного тока вперед на максимальной скорости. Контакты с широтно-импульсной модуляцией (ШИМ) могут принимать значения от 0 до 255, где 0 – это 0 Вольт, а 255 – 5 Вольт.

Если необходимо вращать мотор в другую сторону, нужно передать в функцию другой аргумент:

```
motorR.run(BACKWARD);
```

Для остановки моторов нужно передать в функцию `setSpeed` значение 0 или в функцию `run` передать аргумент `RELEASE`.

### 5.3 Описание работы с датчиком акселерометра-гироскопа

Для расчета угла поворота платформы используется датчик MPU-6050, расположенный на модуле GY-521. Датчик имеет следующие характеристики:

- Диапазон акселерометра:  $\pm 2, \pm 4, \pm 8, \pm 16$  g, где  $g=9.8\frac{M}{c^2}$ ;
- Диапазоны гироскопа:  $\pm 250, 500, 1000, 2000$  градусы / с;
- Диапазон напряжения: 3,3 В - 5 В (модуль включает в себя регулятор напряжения).

MPU-6050 обменивается данными с Arduino по протоколу I2C. MPU-6050 подключен к Arduino, как показано на рисунке 5.3.1. Затем заземление Arduino подключается к заземлению MPU-6050.

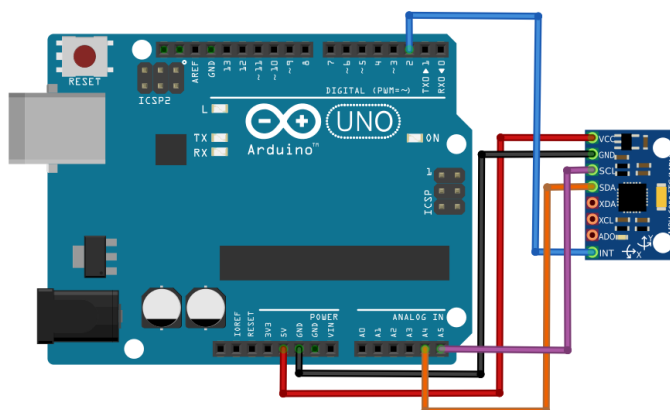


Рисунок 5.3.1 – Подключение датчика MPU-6050 к микроконтроллеру

Цифровой контакт 2 Arduino (контакт прерывания 0) подключается к контакту, помеченному как INT на MPU-6050. Это нужно для некоторых программ, где используются прерывания.

Также необходимо настроить протокол I2C. Для этого нужно присоединить вывод с маркировкой SDA на MPU-6050 к аналоговому выводу Arduino 4 (SDA), а вывод с маркировкой SCL на MPU-6050 к аналоговому выводу 5 Arduino (SCL).

Перед тем как начать работу с датчиком, необходимо установить библиотеки, упрощающие получение данных с модуля: I2cdev и MPU6050, которые нужны для быстрой настройки датчика и для удобства приема и передачи данных.

Перед тем как начать получать показания с модуля, нужно провести коррекцию датчика. Для начала подключаем описанные выше библиотеки:

```
#include "I2Cdev.h"
#include "MPU6050.h"
```

Объявляем объект:

```
MPU6050 accgyro;
```

Затем запускаем последовательный порт, благодаря которому будет происходить вывод данных на экран, интерфейс I2C и сам датчик:

```
void setup() {
  Serial.begin(9600);
```

```
Wire.begin();  
accgyro.initialize();  
}
```

Затем нужно объявить переменные, в которых будут храниться все получаемые значения:

```
int16_t ax, ay, az, gx, gy, gz;
```

В основной части программы, которая является бесконечным циклом, каждую секунду получаем новые данные с датчика и выводим их на экран:

```
void loop() {  
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);  
  Serial.print("a/g: \t");  
  Serial.print(ax); Serial.print("\t");  
  Serial.print(ay); Serial.print("\t");  
  Serial.print(az); Serial.print("\t");  
  Serial.print(gx); Serial.print("\t");  
  Serial.print(gy); Serial.print("\t");  
  Serial.println(gz);  
  delay(1000);  
}
```

На рисунке 5.3.2 приведены данные, получаемые с датчика MPU-6050.

a/g:	798	200	16748	138	110	131
a/g:	842	220	16860	158	96	115
a/g:	818	222	16844	165	100	108
a/g:	810	300	16764	174	97	107
a/g:	834	312	16776	198	129	92
a/g:	858	374	16782	187	148	98
a/g:	808	336	16756	180	156	91
a/g:	826	394	16732	169	176	99
a/g:	816	370	16780	157	169	110
a/g:	762	314	16790	131	175	115
a/g:	764	330	16764	121	162	128
a/g:	772	284	16708	113	141	136
a/g:	742	254	16798	122	138	122
a/g:	736	192	16782	121	137	114
a/g:	792	250	16756	124	127	122
a/g:	814	264	16800	146	111	107

Рисунок 5.3.2 – Данные, полученные с датчика MPU-6050

На рисунке 5.3.2 первые три столбца это ускорение по осям X, Y и Z соответственно, а вторые три столбца – это угловые скорости вокруг осей X, Y и Z.

Однако, посмотрев на рисунок 5.3.2, можно заметить, что гироскоп находится в движении. На самом деле он не двигается, а смещение показаний обусловлено погрешностью самого датчика, ведь он преобразует механическое воздействие в электрический сигнал. Величина смещения у всех датчиков индивидуально и поэтому каждый датчик настраивается отдельно. Чтобы скомпенсировать эти смещения, можно воспользоваться специальными регистрами. В них нужно записать величину смещения оси, чтобы затем датчик сам отнимал заданную величину и записывал в свои регистры данных уже скорректированные значения. Для этого используются функции `setAccelOffset` и `setGyroOffset`.

Изменим нашу программу, добавив функции `setAccelOffset()` и `setGyroOffset()` для каждой из осей акселерометра и гироскопа:

```
accgyro.setXAccelOffset(0);
accgyro.setYAccelOffset(0);
```



```
accgyro.setZAccelOffset(0);  
accgyro.setXGyroOffset(0);  
accgyro.setYGyroOffset(0);  
accgyro.setZGyroOffset(0);
```

В эти функции мы будем передавать целые числа, чтобы скорректировать показания датчика. Чтобы выбрать правильные смещения, нужно проделать несколько шагов:

- Сначала смотрим полученные данные с акселерометра и гироскопа при нулевых значениях коррекции;
- Максимально приближаем значения угловых скоростей вокруг осей X, Y и Z были к нулю методом подбора коэффициентов;
- То же самое проделываем для акселерометра. По двум осям его значения также должны быть близки к нулю, а по третьей – должны быть близко к величине 1g.

В результате настройки получили следующие коэффициенты:

```
accgyro.setXAccelOffset(39); // Коррекция смещения акселерометра по  
оси X
```

```
accgyro.setYAccelOffset(44); // Коррекция смещения акселерометра по  
оси Y
```

```
accgyro.setZAccelOffset(18); // Коррекция смещения акселерометра по  
оси Z
```

```
accgyro.setXGyroOffset(-1714); // Коррекция смещения гироскопа по  
оси X
```

```
accgyro.setYGyroOffset(-71); // Коррекция смещения гироскопа по оси  
Y
```

```
accgyro.setZGyroOffset(456); // Коррекция смещения гироскопа по  
оси Z
```

На рисунке 5.3.3 изображены данные, присылаемые с датчика, после коррекции, описанной сверху.

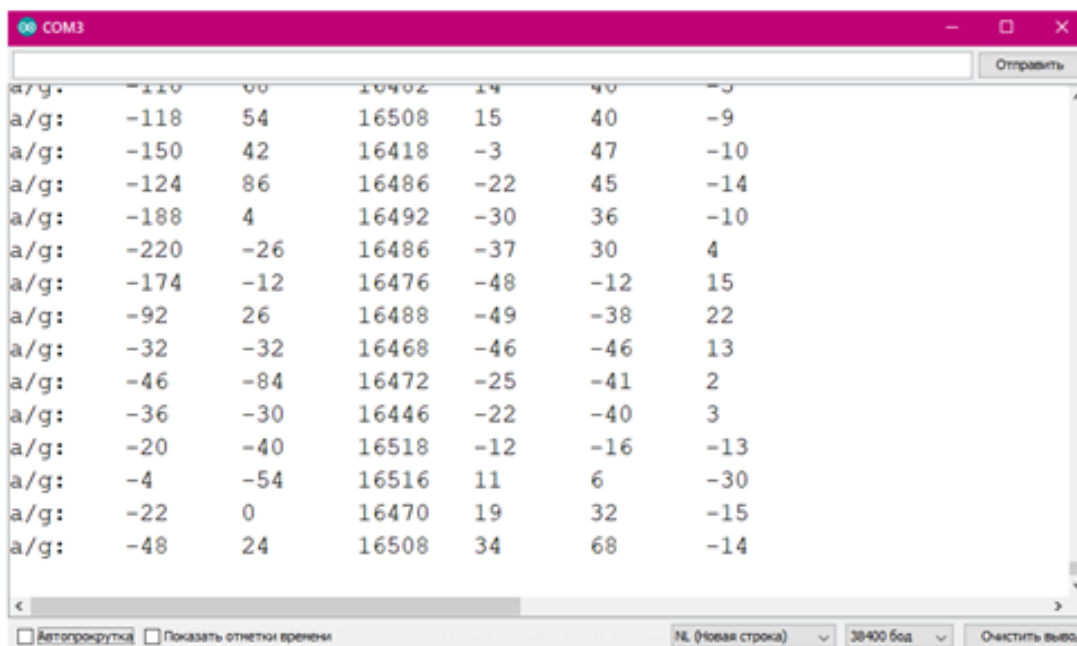


Рисунок 5.3.3 – Данные, полученные с датчика MPU-6050 после коррекции

Если сравнить значения на рисунке 5.3.3 со значения на рисунке 5.3.2 можно увидеть, что показания имеют меньший разброс и больше приблизились к идеальным значениям.

Таким образом, мы разобрались в том, как настраивать базовые параметры датчика. Теперь с его помощью можно получать информацию о текущих углах поворота платформы [19].

#### 5.4 Определение перекрестков лабиринта

В пункте 5.1 мы разобрались, как работать с ультразвуковым дальномером. Теперь рассмотрим, каким образом роботизированная платформа определяет, в каком месте она находится.

Датчики на платформе располагаются таким образом, чтобы можно было одновременно измерять расстояния слева, справа и спереди платформы. Показания считываются каждые 100 мс. Эта задержка нужна, чтобы исключить некорректные показания датчиков и дерганья тележки.

Вариант замкнутого пространства с множеством поворотов и перекрестков представлен на рисунке 5.4.1.

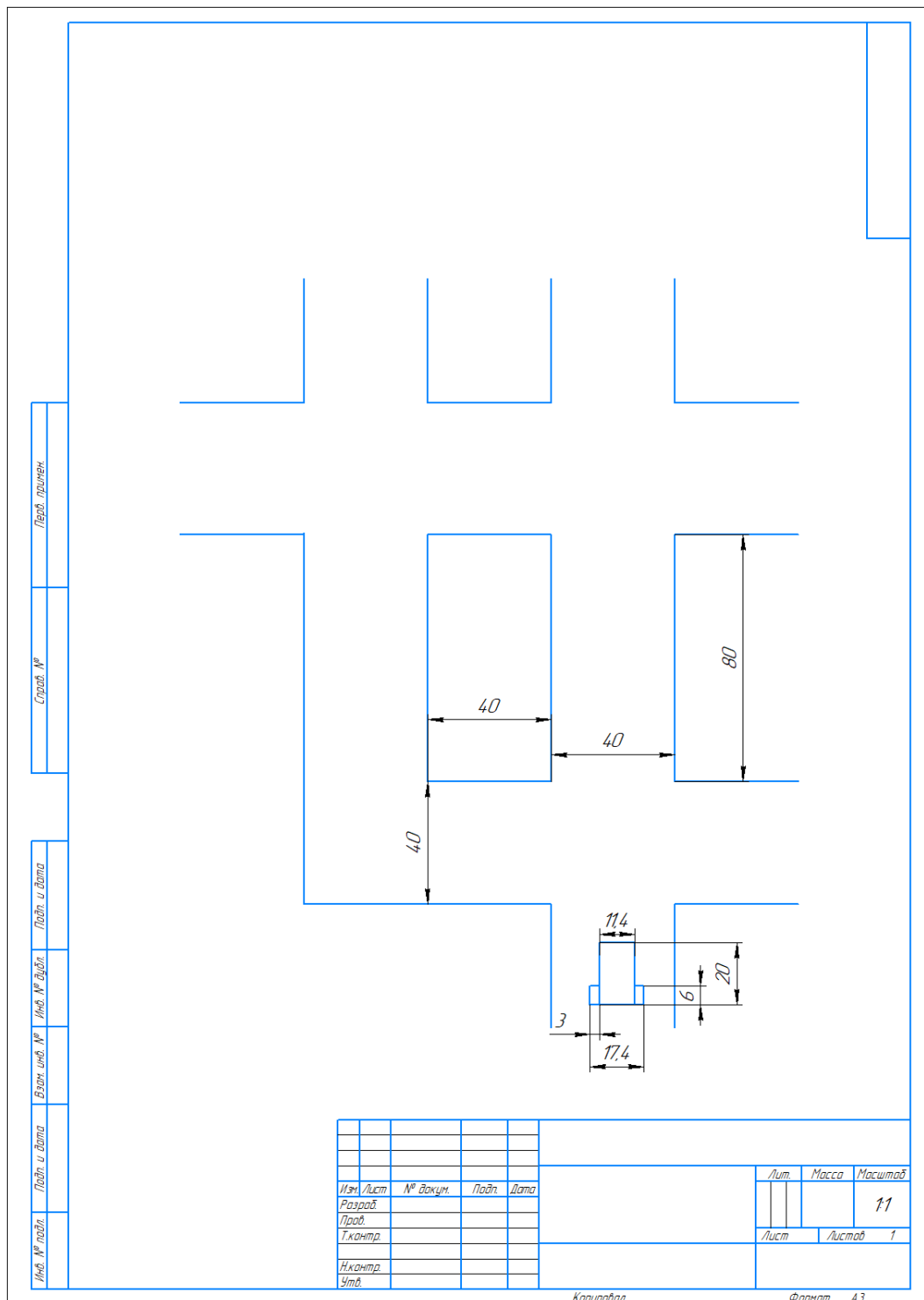


Рисунок 5.4.1 – Вариант замкнутого пространства с множеством поворотов и перекрестков

Из рисунка можно видеть, что платформа обладает шириной 17,4 см, а расстояние между стенами – 40 см. Следовательно, если тележка в результате программной или аппаратной ошибки сместится максимально близко к стене, расстояние до стены будет приблизительно равняться 24 см. Исходя из этого величину максимального расстояния, на котором может находиться стена, приняли за 25 сантиметров. На рисунке 5.4.2 представлены все возможные варианты показаний датчиков расстояния.

<i>Показания левого датчика, см</i>	<i>Показания центрального датчика, см</i>	<i>Показания правого датчика, см</i>	<i>Место</i>	<i>Выбор пути</i>
$<25$	$\geq 25$	$<25$	<i>прямой коридор</i>	<i>ехать прямо</i>
$\geq 25$	$<25$	$<25$	<i>поворот налево</i>	<i>поворачивать налево</i>
$<25$	$<25$	$\geq 25$	<i>поворот направо</i>	<i>поворачивать направо</i>
$\geq 25$	$\geq 25$	$<25$	<i>T-образный перекресток лево-прямо</i>	<i>выбрать путь согласно маршруту</i>
$<25$	$\geq 25$	$\geq 25$	<i>T-образный перекресток право-прямо</i>	<i>выбрать путь согласно маршруту</i>
$\geq 25$	$<25$	$\geq 25$	<i>T-образный перекресток лево-право</i>	<i>выбрать путь согласно маршруту</i>
$\geq 25$	$\geq 25$	$\geq 25$	<i>четырёхсторонний перекресток</i>	<i>выбрать путь согласно маршруту</i>
$<25$	$<25$	$<25$	<i>Тупик</i>	<i>Развернуться</i>

Рисунок 5.4.2 – Таблица показаний датчиков и логика определения перекрестков и поворотов

Если хотя бы два из трех датчиков показали, что стены около них нет, значит, робот доехал до перекрестка. Если только один датчик показывает отсутствие стены, значит платформа движется в направлении расположения этого датчика.

## 5.5 Способы задания маршрута

В начале программы задается маршрут, по которому должна проехать платформа. Он задается массивом, состоящих из английских слов `left`, `right`, `forward`, которые в переводе на русский означают: налево, направо и вперед. Роботизированная платформа может проехать по абсолютно любому пути, который будет задан пользователем. Нужно лишь посмотреть на лабиринт и вписать необходимые повороты на перекрестках в массив:

```
long povorot[] = {'forw', 'left', 'left', 'left', 'right'};  
int per=0;
```

В данном случае маршрут будет состоять из пяти перекрестков. На первом – робот должен ехать прямо, на следующих трех повернуть налево, а на последнем свернуть направо. Счетчик `per` хранит количество пройденных перекрестков.

Помимо этого, маршрут может заполняться не только пользователем вручную, но и при помощи дорожных знаков (рисунок 5.5.1). Как распознаются знаки, здесь не описывается, так как это другая часть комплексной работы.



Рисунок 5.5.1 – Распознавание дорожного знака компьютером  
Raspberry Pi

После того как знак был распознан, с одноплатного компьютера Raspberry Pi 2 Model B по интерфейсу последовательной связи приходят данные. Алгоритм приема и обработки данных показан на рисунке 5.5.2.

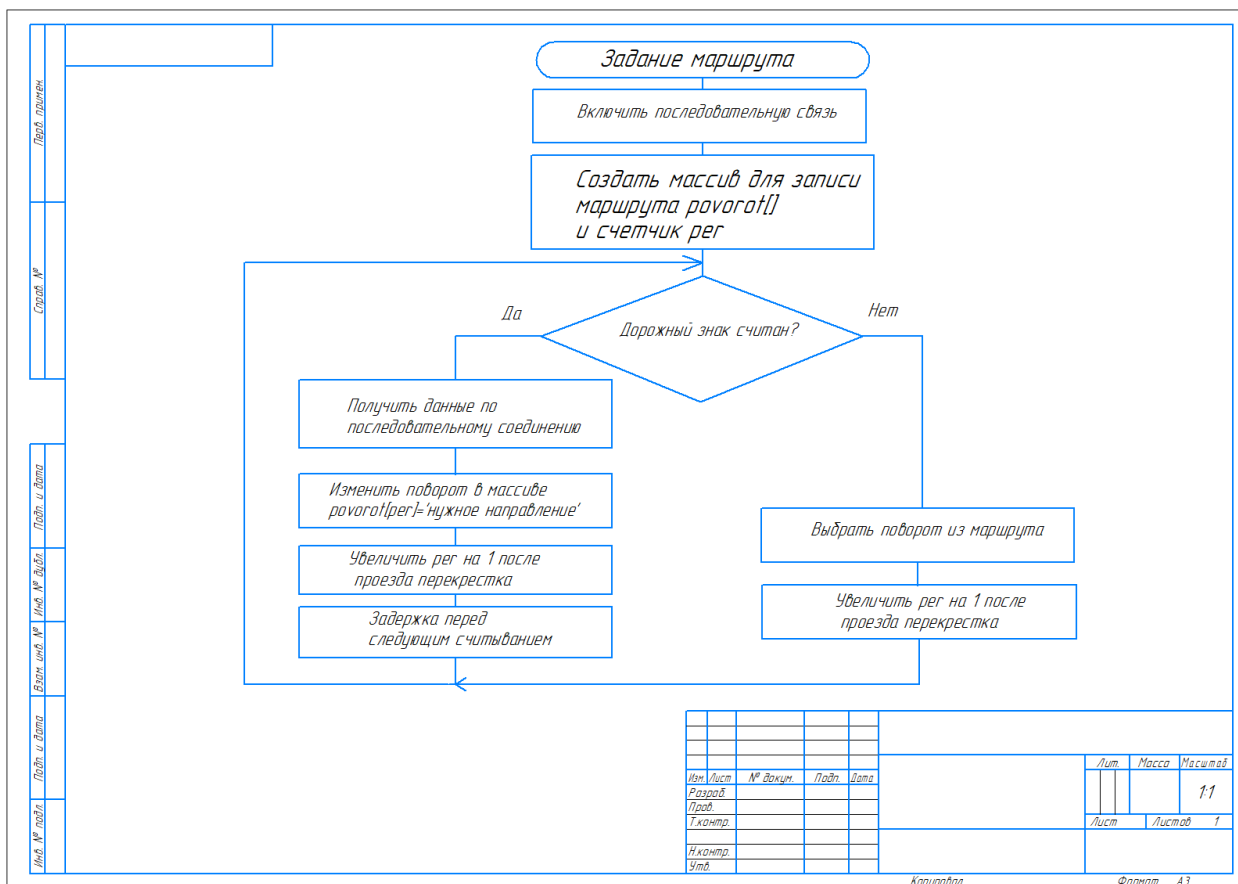


Рисунок 5.5.2 – Алгоритм приема и обработки данных маршрута

Для начала включаем последовательную связь с помощью функции:

`Serial.begin(9600);`

Принимать сообщения от компьютера будем командой:

`Serial.read();`

Полученное сообщение, например, left, означает, что текущий элемент массива перекрестков нужно изменить на требуемое направление:

`povorot[per]= 'left';`

Как только мы получили данные от знака и изменили направление движения для следующего перекрестка, перестаем обрабатывать данные, получаемые от камеры. Как только перекресток пройден, увеличиваем счетчик перекрестков на единицу и начинаем алгоритм заново. Если данные

с компьютера не приходят, значит, знаков нет, следовательно, ехать нужно согласно заранее заданному маршруту.

## 5.6 Алгоритм работы платформы

Основная блок-схема алгоритма работы мобильной робототехнической платформы представлена на рисунке 5.6.1.

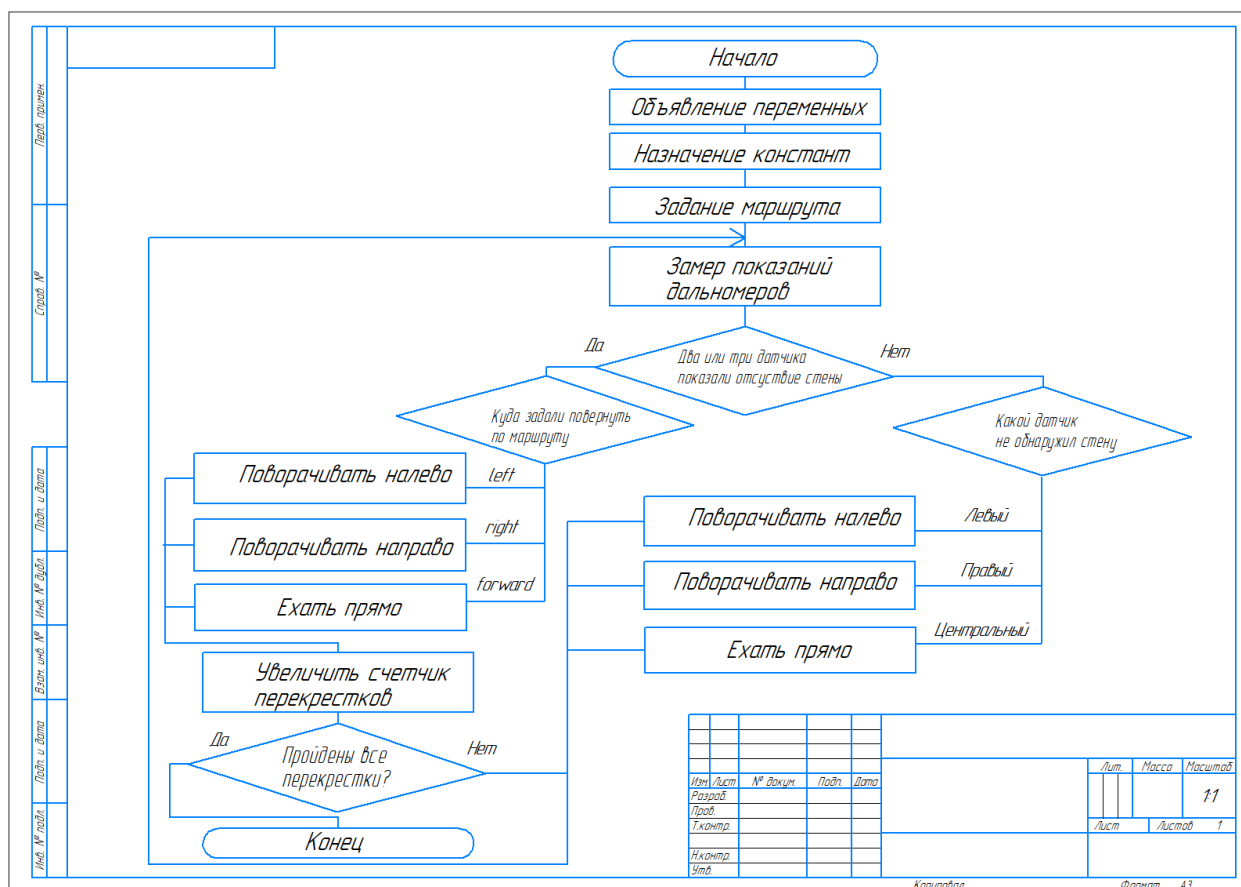


Рисунок 5.6.1 – Блок-схема алгоритма программы

В начале программы объявляются все необходимые переменные, назначаются константы скоростей и расстояний, а также задается маршрут.

Далее все действия выполняются циклически. Каждые 100 миллисекунд датчики обновляют информацию об окружении вокруг себя, что было описано в разделе 5.4. Затем платформа выбирает поворот согласно

заданному маршруту и отмечает, что этот перекресток пройден, увеличивая специальную переменную-счетчик на единицу. И так далее, пока не будут пройдены все заданные перекрестки.

На рисунке 5.6.2 представлена блок-схема алгоритма движения мобильного робота вперед.

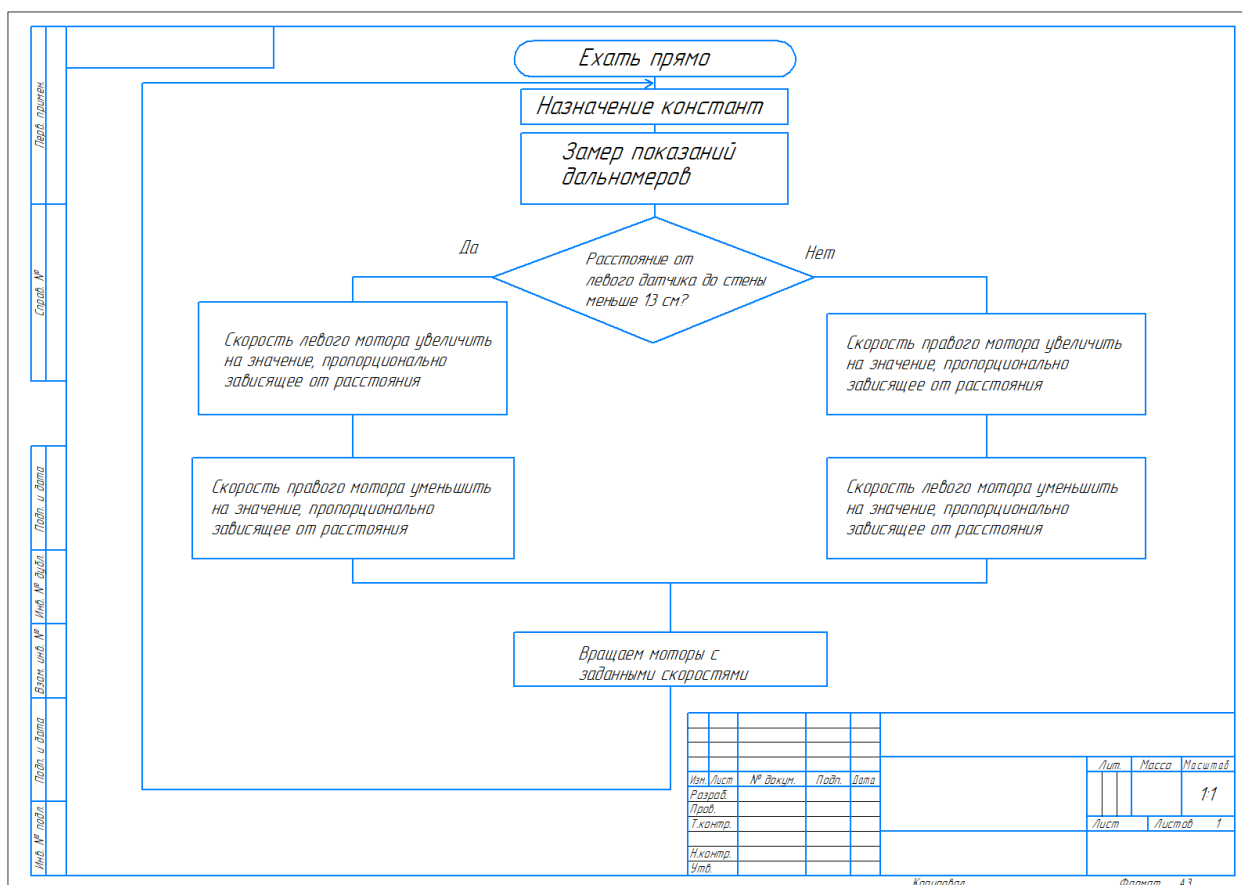


Рисунок 5.6.2 – Блок-схема алгоритма движения платформы вперед

Чтобы робот ехал строго по центру, необходима тонкая подстройка скоростей. Дальномеры постоянно присылают новые значения. Если расстояние от левого датчика до стены меньше 13 см, значит, робот начал съезжать влево, следовательно, скорость левого мотора нужно увеличить, а скорость правого уменьшить. Выбор коэффициентов изменения скоростей происходил с учетом веса платформы, мощностей двигателей и величины напряжения, подаваемой на плату управления мотора.



Если же расстояние от левого датчика до стены не меньше 13 см, следовательно, робот либо едет по центру, либо начал сдвигаться вправо. В этот раз нужно действовать следующим образом: увеличивать скорость правого мотора, а уменьшать скорость левого. Таким образом, платформа всегда стремится к центру коридора.

Поворот – это довольно сложный процесс. Платформа должна повернуть строго на 90 градусов, чтобы не уехать с трассы. Именно для этого мы и используем датчик акселерометра-гироскопа MPU-6050. На рисунке 5.6.3 представлена блок-схема алгоритма поворота робота налево.

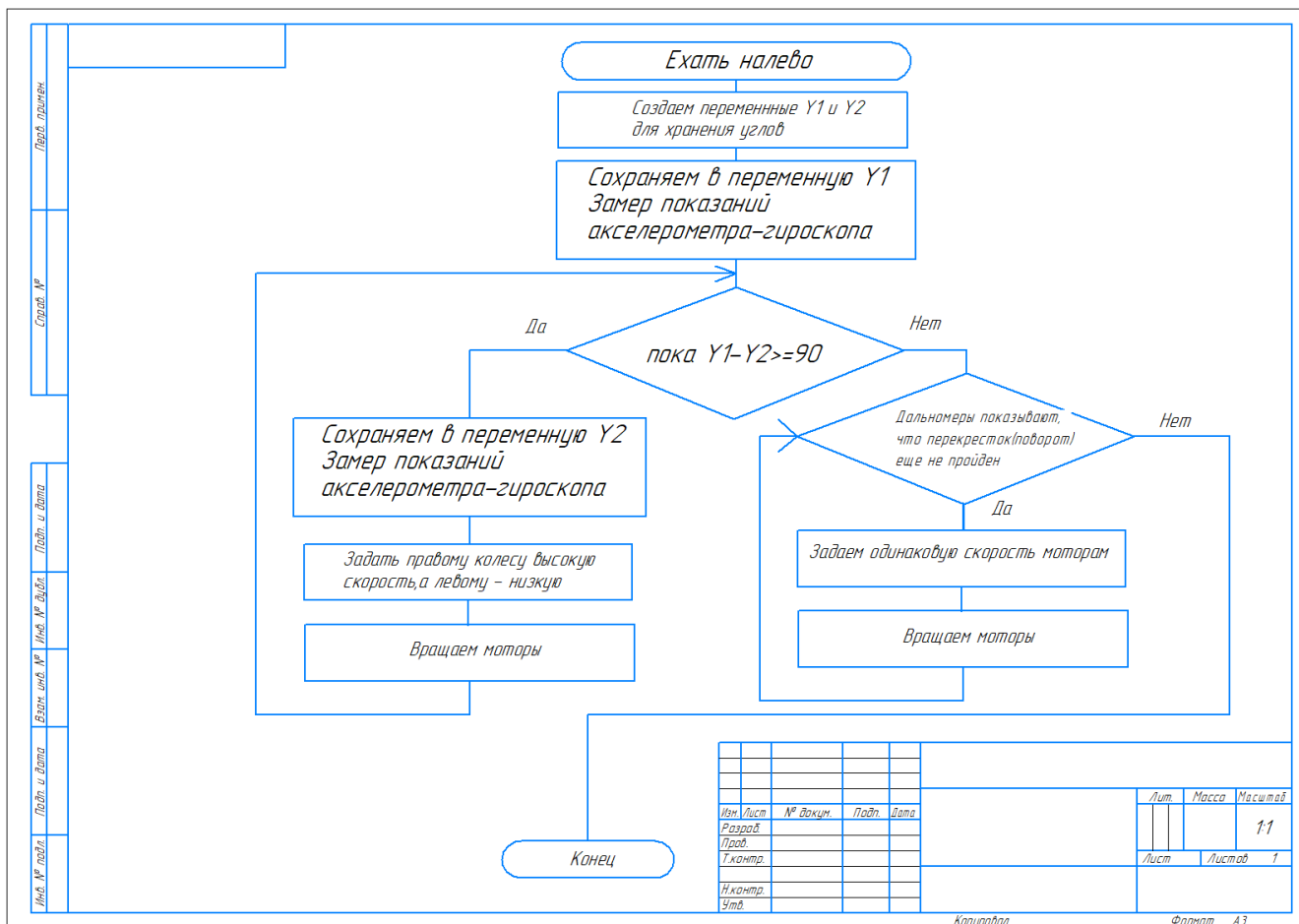


Рисунок 5.6.3 – Блок-схема алгоритма поворота платформы налево

Алгоритм поворота налево заключается в том, что прямо перед поворотом текущий угол мобильного робота сохраняется в переменную. Затем при повороте налево из этого угла вычитается новый угол платформы. Если полученная разница меньше 90 градусов, то правое колесо вращается с высокой скоростью, а левое с низкой или совсем не вращается. Этот цикл происходит до тех пор, пока разница между нынешним углом и углом, который был до поворота, больше или равен 90 градусам. После этого моторы крутятся с одинаковой скоростью до тех пор, пока датчики слева и справа не покажут, что с обеих сторон находятся стены, значит, поворот пройден успешно и можно вернуться к основному алгоритму.

На рисунке 5.6.4 представлена блок-схема алгоритма поворота робота направо.

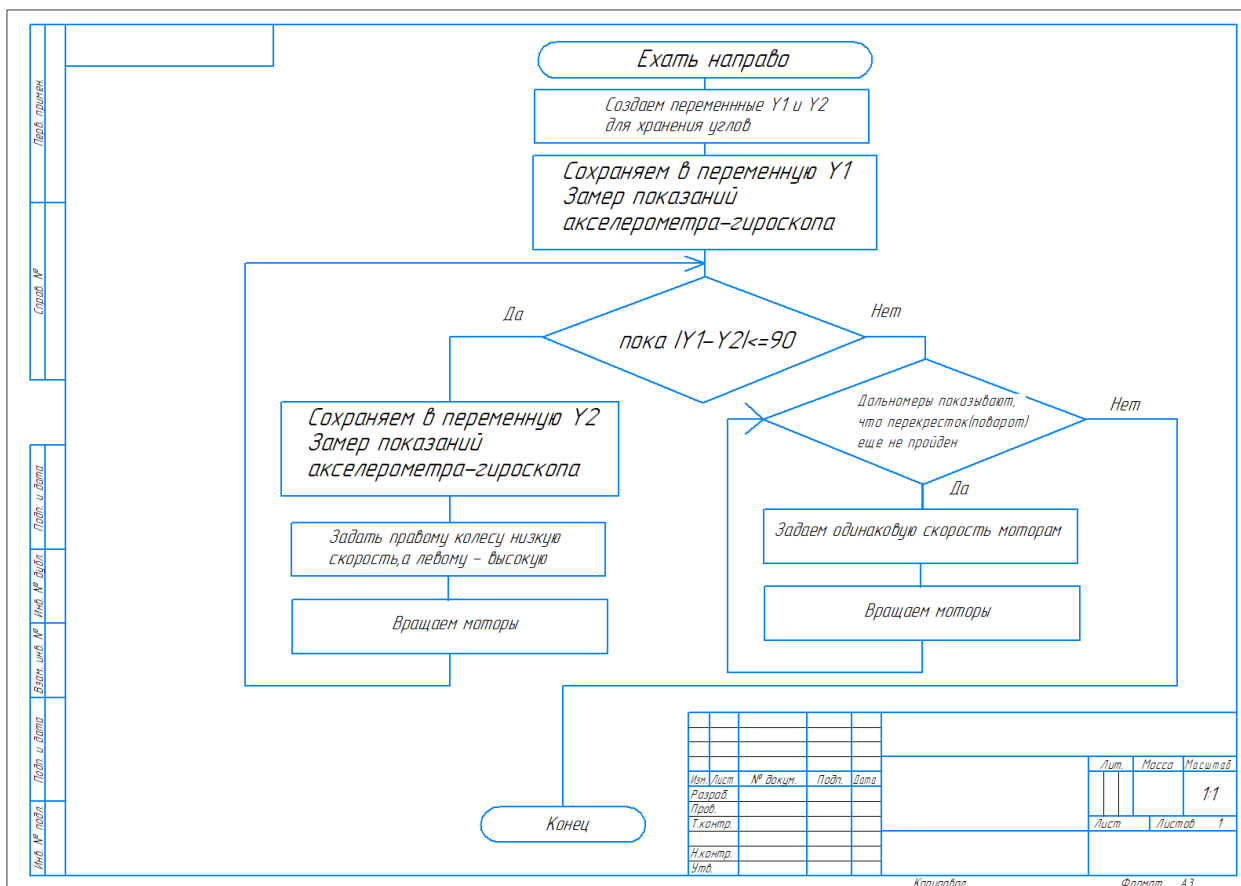


Рисунок 5.6.4 – Блок-схема алгоритма поворота платформы направо

Поворот направо происходит по такому же алгоритму. Мобильный робот выезжает на перекресток и начинает поворачивать направо. Отличие в том, что вычитание углов происходит по модулю, а итоговая величина необходимая для завершения поворота должна быть меньше или равна 90 градусам.

Логичным выводом из данного раздела может послужить разработанный программный алгоритм системы поиска пути мобильного робота. Подробно описана работа со всеми используемыми в проекте датчиками. Разработаны алгоритмы определения препятствий, поиска пути робота и поворота на заданный угол. Представлены и описаны подробные блок-схемы алгоритмов. Разработанная структура отвечает всем необходимым начальным условиям, которые предъявлялись к данной системе для обеспечения низкой стоимости и высокой надежности.

## 6 Результаты экспериментальных испытаний

Тестирование мобильного робота проводилось в несколько этапов:

- Проверка прямолинейного движения – предназначена для анализа выбранных коэффициентов изменения скорости;
- Проверка детектирования стен с трех сторон платформы препятствий – для определения работоспособности алгоритма обнаружения объектов;
- Проверка приема данных, полученных от одноплатного компьютера, и преобразования их в маршрут.
- Проверка поворота платформы на заданный угол – для определения точности показаний акселерометра-гироскопа;
- Проверка следования платформы по заданному маршруту;

Испытание на прямолинейном участке производилось в длинном коридоре, по бокам ограниченном стенами лабиринта. Производилась проверка изменения скоростей моторов в зависимости от расстояния до стен слева и справа. Платформа преодолела маршрут длиной в 10 м. В результате испытаний, платформа в среднем отклонялась от центра не более чем на 0.5 см.

Обнаружение объектов с помощью ультразвуковых датчиков тестировалось разными поверхностями с разной высотой и расстоянием до них. Некоторые объекты не определялись на высоком расстоянии, в том числе и стены лабиринта. Стены высотой 8 см на расстоянии более 22 см не всегда успешно считывались. В результате высота стен была увеличена на 2 см, что исправило эту ошибку. Датчики верно считывали расстояния на заявленных в документации диапазонах от 2 до 400 см.

Для проверки приема данных с одноплатного компьютера Raspberry Pi 2, в лабиринт были добавлены знаки поворотов, разрешающих проезд только в заданных направлениях. Дорожные знаки распознавались и данные о них передавались на микроконтроллер примерно за 50 см до знака. Знаки были

размещены за 20 см до перекрестков для того, чтобы мобильный робот успел изменить свой маршрут с учетом считанного знака.

Проверка точности поворота производилась на четырехстороннем перекрестке (рисунок 6.1).



Рисунок 6.1 – Мобильный робот находится на четырехстороннем перекрестке

Проверка следованию маршруту проводилась в лабиринте, представленном на рисунке 5.4.1. Маршрут был построен таким образом, чтобы платформа полностью проехала через лабиринт два раза, а на одном и том же перекрестке проехала в двух разных направлениях: направо и вперед.

Таким образом, в результате экспериментов было принято решение внести несколько поправок в программы. Чтобы скомпенсировать инерцию и скольжение колес по напольному покрытию, было принято задавать угол поворота в программе равным не 90 градусам, а 88, чтобы платформа начала торможение чуть раньше. В итоге, платформа поворачивалась под нужным углом. Для того, чтобы камера передавала более плавную картинку и знак считывался быстрее, было добавлено снижение скорости робота при возникновении знака на пути.

## **7 Безопасность и экологичность проекта**

При изготовлении платформы использовались такие материалы как полипропилен, алюминий, цинк. Перед тем как начать работу с такими материалами, нужно убедиться в знании техники безопасности при обработке материалов. Глаза человека наименее защищены и, следовательно, наиболее подвержены опасности, которая может внезапно возникнуть при резке или обработке металлических и полипропиленовых материалов. Поэтому следует надевать защитные очки перед началом работы. Затем нужно подготовить пространство для работы и проверить исправность необходимых для работы инструментов.

Во время выполнения любых работ нужно быть максимально сосредоточенным, следить за тем, чтобы рядом с инструментами не находились люди. Одежда должна быть удобной, чтобы не отвлекала от работы. Также нужно проверить исправность инструмента и только тогда приступать к резке или обработке материала. При поломке инструмент нужно сразу отложить в сторону и при необходимости обесточить.

При сборке электрических соединений нужно обеспечить изоляцию всех проводников с током. Для изоляции можно использовать специальные материалы, например, термоусадочную трубку или электроизоляционную ленту. При подключении электронных компонентов, необходимо соблюдать полярность элементов. Несоблюдение полярности электронных компонентов и разрыв изоляции проводника может привести к короткому замыканию, возгоранию и задымлению комнаты.

Пайка элементов схемы должна осуществляться с осторожностью и соблюдением всех вышеперечисленных мер безопасности. Припой олова или свинца при плавлении может выделять вредные для здоровья пары, поэтому приступать к пайке можно только убедившись, что вы находитесь в хорошо проветриваемом и освещенном помещении.

При работе с мобильным роботом нужно соблюдать следующее:

- В данной работе используются литий-ионные аккумуляторы, поэтому следует знать и соблюдать все меры безопасности. Заряд аккумуляторов необходимо осуществлять только специальными средствами, например, с помощью микросхемы контроля заряда литий-ионных аккумуляторов TP-4056. Следить за нагревом аккумуляторов, температура выше 75 градусов Цельсия является критической, следует немедленно прекратить работу и обесточить цепь. Несоблюдение названных требований способно привести к самовоспламенению аккумуляторов.

- Перед эксплуатацией нужно проверить, надежно ли подключены и зафиксированы все соединения. При обнаружении неподключенных проводников, необходимо подключить провода согласно документации и проверить надежность фиксации, а при обнаружении поврежденных или оголенных проводников необходимо произвести замену или их изоляцию.

- Для обеспечения полной безопасности подавать питание на силовую часть спустя некоторое время после подачи питания на микроконтроллер.

- Во время движения платформы исключить попадание на пути маршрута людей и препятствий, рассеивающих или поглощающих ультразвуковые волны.

- Для немедленного отключения платформы, установлена кнопка включения и выключения питания.

- Не допускать попадания жидкостей на элементы мобильного робота.

- Замена элементов платформы происходит только при полностью отключенном питании.

В работе используются двигатели постоянного тока, которые не наносят окружающей среде никакого вреда. Также высокий уровень экологичности обеспечивается тем, что, по истечении срока годности аккумуляторов, они сдаются в специальные пункты утилизации.

Таким образом, при соблюдении немногочисленных правил эксплуатации платформы, обеспечивается безопасность и экологичность системы.

## 8 Экономическая эффективность

Стоимость каждого отдельно взятого элемента и общая стоимость полученной роботизированной платформы приведены на рисунке 8.1.

<i>№</i>	<i>Наименование</i>	<i>Кол</i>	<i>Цена, р</i>
1	<i>Arduino UNO</i>	1	250
2	<i>Кабель USB A – USB-B</i>	1	100
3	<i>HC-SR04</i>	3	150
4	<i>Ultrasonic Sensor Mounting Bracket</i>	3	70
5	<i>Lii-34A 18650 3400mah</i>	4	700
6	<i>Держатель для аккумуляторов</i>	1	80
7	<i>XL-6009</i>	1	111
8	<i>TP-4056</i>	1	15
9	<i>Arduino shield L293D</i>	1	109
10	<i>TT motor</i>	2	150
11	<i>Внешний аккумулятор 5 В 2.5А</i>	1	1000
12	<i>Датчик акселерометра-гироскопа GY-521</i>	1	150
13	<i>Ведомое колесо</i>	1	50
14	<i>Ведущее колесо 1B48-14 16LSD6</i>	2	200
15	<i>Разделочная доска из полипропилена 29x20</i>	2	80
16	<i>Крепеж бытовой оцинкованный</i>	4	85
17	<i>Кнопка</i>	1	25
18	<i>Гайка (din 934) оцинкованная М3</i>	8	15
19	<i>Винт оцинкованный М3</i>	8	10
20	<i>Держатель для двигателей</i>	2	30
<i>Итого</i>			<i>3380</i>

Рисунок 8.1 – Экономический расчет

Таким образом, стоимость каждого отдельно взятого компонента невысока, что объясняется тщательным анализом рынка и сравнением всех доступных аналогов элементов. Общая сумма за полностью рабочего мобильного робота является приемлемой.



## Заключение

В результате выполнения бакалаврской работы разработан мобильный робот, способный перемещаться по заданному маршруту и менять его в соответствии с принимаемыми командами.

В первом разделе был проведен анализ существующих робототехнических систем, применяемых в различных областях. Доказана актуальность разработанного мобильного робота.

Во втором разделе были освещены наиболее известные методы позиционирования, применяемые в мире, показаны их преимущества и недостатки, предложено альтернативное решение, наиболее подходящее для робототехнических систем складской логистики. Обоснован выбор каждого элемента системы. Представлены аргументы в пользу выбора микроконтроллера Arduino. Отмечено преимущество ультразвуковой системы определения препятствия перед инфракрасной.

В третьем разделе была разработана структурная схема платформы и перечень всех используемых элементов.

В четвертом разделе была составлена и описана схема электрическая соединений, а также представлен перечень всех используемых в проекте электронных компонентов.

В пятом разделе реализована и описана связь микроконтроллера Arduino UNO с одноплатным компьютером Raspberry Pi по последовательному соединению через кабель USB. Подробно описана работа с каждым элементом проекта. Программа реализована для микроконтроллера Arduino. Составлены блок-схемы алгоритма основных функций программы.

В шестом разделе представлены результаты экспериментальных испытаний, выявлены и исправлены недочеты робототехнической платформы.

В седьмом разделе представлены указания по технике безопасности при работе с мобильным роботом.

Стоимость каждого элемента платформы, а также общая сумма представлены в восьмом разделе. Отмечается экономическая эффективность разработки мобильной платформы.

Таким образом, готовый комплексный проект представляет собой мобильного робота, который движется в соответствии с поставленным маршрутом и учитывает расположенные на некоторых перекрёстках дорожные знаки.

## Список используемой литературы

1. Состав и характеристики мобильных роботов: учеб, пособие по курсу «Управление роботами и робототехническими комплексами». / К. Ю. Машков, В. И. Рубцов, И. В. Рубцов. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2014. — 75, [1] с.: ил.
2. SIX COMPANIES SHAPING THE FUTURE OF AUTOMOTIVE ROBOTICS [Электронный ресурс].URL:<https://builtin.com/robotics/automotive-cars-manufacturing-assembly>
3. Основы робототехники: Введение в специальность. / Попов Е. П., Письменный Г. В. — М.: Высшая школа, 1990. — 224 с.
4. Создание роботов в домашних условиях / Брага Ньютон ; пер. с англ. Е. А. Добролежина. - М. : НТ Пресс, 2007. - 368 с. : ил. - (Робот - своими руками)
5. Warehouse Robotics: Everything You Need to Know in 2019 [Электронный ресурс].URL: <https://www.logiwa.com/blog/warehouse-robotics>
6. Chinese startup Syrius Robotics builds warehouse robots with JD Logistics [Электронный ресурс].URL: <https://www.therobotreport.com/chinese-startup-syrius-robotics-builds-warehouse-robots-with-jd-logistics/>
7. How robots are transforming Amazon warehouse jobs — for better and worse [Электронный ресурс].URL: <https://www.vox.com/recode/2019/12/11/20982652/robots-amazon-warehouse-jobs-automation>
8. Одометрия [Электронный ресурс].URL: <http://robocraft.ru/blog/technology/736.html>
9. Conceptual Bases of Robot Navigation Modeling, Control and Applications [Электронный ресурс].URL: <https://www.intechopen.com/books/advances-in-robot-navigation/conceptual-bases-of-robot-navigation-modeling-control-and-applications>

10. Достойные аналоги Ардуино: Teensy, Netduino и другие [Электронный ресурс]. URL: <https://arduinoplus.ru/vse-analogi-arduino/>
11. Глибин Е.С. Программирование электронных устройств: электронное учебное пособие / Е.С. Глибин, А.В. Прядилов, - Тольятти: Изд-во ТГУ, 2014.
12. Инфракрасный датчик препятствий YL-63 [Электронный ресурс]. URL: <https://3d-diy.ru/wiki/arduino-datchiki/infrakrasnyj-datchik-prepyatstvuj-yl-63/>
13. Чувства роботов: ИК-датчик расстояния Sharp GP2Y0A21YK [Электронный ресурс]. URL: <https://mysku.ru/blog/aliexpress/29357.html>
14. VL53L0X [Электронный ресурс]. URL: <https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html>
15. Документация на ультразвуковой датчик расстояния HC-SR04 [Электронный ресурс]: документация. – режим доступа: <http://www.micropik.com/PDF/HCSR04.pdf>, свободный.
16. DC GEARBOX MOTOR - "TT MOTOR" - 200RPM - 3 TO 6VDC [Электронный ресурс]. URL: <https://www.verical.com/datasheet/adafruit-brushless-dc-motors-3777-5912007.pdf>
17. Документация на драйвер L298n [Электронный ресурс]: документация. – режим доступа: [https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf), свободный.
18. Control DC, Stepper & Servo with L293D Motor Driver Shield & Arduino [Электронный ресурс]. URL: <https://lastminuteengineers.com/l293d-motor-driver-shield-arduino-tutorial/>
19. Invensense. [Электронный ресурс]: документация на микроэлектромеханический датчик акселерометра-гироскопа. – Электрон.текстовые дан. – режим доступа к документу: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

20. Стычев С.Н., Краснопевцева Н.А., Мальцев С.А. В сборнике: Интеграционные процессы в современной науке: Применение робототехнических платформ в области складской логистики. – Материалы Международной научно-практической конференции (г.-к. Анапа, 26 мая 2020 г.). [Электронный ресурс]. – Анапа: Изд-во «НИЦ ЭСП» в ЮФО, 2020. - 94 с. ISBN 978-5-95283-322-7

21. Усатенко С.Т., Каченюк Т.К., Терехова М.В. Выполнение электрических схем по ЕСКД: Справочник. – М.: Издательство стандартов, 1989. – 325 с.

22. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. – СПб.: БВХ-Петербург, 2015. – 336 с.: ил. ISBN 978-5-9775-3585-4

23. Программирование на языке С для AVR и PIC микроконтроллеров./ Сост. Ю.А. Шпак – К.: «МК-Пресс», 2006. – 400 с., ил. ISBN 966-8806-16-6

24. Программирование на языке С для AVR и PIC микроконтроллеров./ Сост. Ю.А. Шпак – К.: «МК-Пресс», 2006. – 400 с., ил. ISBN 966-8806-16-6

25. Шилдт, Герберт. Полный справочник по С++, 4-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 800 с. : ил. – Парал. тит. Англ. ISBN 5-8459-0489-7 (рус.)

26. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. – СПб.: БВХ-Петербург, 2015. – 336 с.: ил. ISBN 978-5-9775-3585-4

27. Усатенко С.Т., Каченюк Т.К., Терехова М.В. Выполнение электрических схем по ЕСКД: Справочник. – М.: Издательство стандартов, 1989. – 325 с.