

КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-  
ПЕТЕРБУРГА

**САНКТ- ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ «КОЛЛЕДЖИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ»**

Специальность 09.02.01 «Компьютерные системы и комплексы»

**УТВЕРЖДАЮ**

**Зам. директора по УМР**

\_\_\_\_\_ **Н.А.Бондарчук**

«\_\_\_» \_\_\_\_\_ **2022 г.**

**ДИПЛОМНЫЙ ПРОЕКТ**

**Пояснительная записка**

Тема: Проектирование и макетирование робота-пылесоса

Группа 481

Рецензент \_\_\_\_\_

Нормоконтроль \_\_\_\_\_ **О.Н. Венедиктова**

Консультант по экономической части \_\_\_\_\_ **Т.В.Львова**

Руководитель дипломного проекта \_\_\_\_\_ **Д.В.Некрасов**

Выполнил \_\_\_\_\_ **Д.А.Тимченко**

Санкт-Петербург 2022

## Оглавление

ВВЕДЕНИЕ .....	2
1. ОБЩАЯ ЧАСТЬ .....	3
1.1 Назначение проекта.....	3
1.2 История создания робота пылесоса.....	3
1.3 Принцип работы робота-пылесоса .....	4
1.4 Сравнение моделей аналогов робота-пылесоса .....	9
1.4.1 Garlyn SR-600.....	9
1.4.2 Roborock S6/T6 .....	11
1.4.3 iCLEBO O5 WiFi.....	12
1.4.4 iRobotRoomba e5.....	13
2. ПРАКТИЧЕСКАЯ ЧАСТЬ .....	16
2.1 Алгоритм работы.....	16
2.2 Макетирование устройства .....	18
2.2.1 Arduino Mega.....	18
2.2.2 Инфракрасный датчик препятствий YL-63 .....	23
2.2.3 Ультразвуковой датчик HC-SR04.....	25
2.2.4 Коллекторный двигатель постоянного тока .....	27
2.2.5 Драйвер L298N .....	29
2.3 Создание и отладка программы для Arduino Mega .....	31
2.4 Разработка устройства в эмуляторе Tinkercad .....	36
3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ.....	42
3.1 Организационно-экономическое обоснование проекта .....	42
3.2 Расчет себестоимости .....	42
3.2.1 Расчет трудоемкости.....	42
3.2.2 Расчет затрат на материалы.....	44
3.2.3. Расчет затрат на электроэнергию амортизации оборудования .....	44
3.2.5. Расчет начислений на заработную плату.....	45
3.2.6 Общие расходы на создание проекта .....	46
3.3 Расчет экономического эффекта.....	46
Заключение.....	47
Список литературы .....	48
Приложение 1 .....	50

## **ВВЕДЕНИЕ**

В представленном проекте был спроектирован и разработан прототип робота-пылесоса и программа для его работы. Так же в проекте будет проведено сравнение прототипа с его аналогами на рынке и сравнение комплектующих. Работа будет осуществляться с помощью контроллера ArduinoUno, с написанной для неё управляющей программой в среде разработки ArduinoIDE.

При выполнении работы были использованы навыки и компетенции, полученные в рамках изучения дисциплин следующих программных модулей:

ПМ.02 Микропроцессорные системы:

- ПК 2.2 Производить тестирование и отладку микропроцессорных систем;
- ПК 2.1 Создавать программы на языке ассемблера для микропроцессорных систем.

ПМ.01 Проектирование цифровых устройств

- ПК 1.1 Выполнять требования технического задания на проективное цифровых устройств;
- ПК 1.3 Использовать средства и методы автоматизированного проектирования при разработке цифровых устройств;
- ПК 1.5 Выполнять требования нормативно-технической документации.

# 1. ОБЩАЯ ЧАСТЬ

## 1.1 Назначение проекта

Целью представленного проекта является разработка прототипа робота-пылесоса на основе анализа существующих аналогов.

В задачи проекта входит выявление типовых функциональных узлов и выбор элементной базы, а также разработка алгоритма управления устройством.

## 1.2 История создания робота пылесоса

Начало эры роботов пылесосов датируется 1996-м. Тогда BBC показали зрителям макет первого интеллектуального уборщика от всемирно известного бренда ELECTROLUX. Чуть позднее, в 2001-м свет увидела первая модель «умного» пылесоса ELECTROLUX Trilobite, а в свободную продажу устройство поступило в 2002-м. Данный уникальный прибор и стал отправной точкой в развитии этой разновидности быт техники.

В то же самое время (в 2002-м) инициативу подхватила фирма iRobot. Американцы выпустили свою первую модель роботов-пылесосов для сухой очистки помещений из линейки Roomba, однако повсеместно продавать его начали лишь в 2004 г. Основание компании Aerobot датировано 1990 годом, однако до выпуска роботов для бытового использования они преимущественно производили роботизированную технику для NASA и военных ведомств.

Тем временем, фирма-первооткрыватель Electrolux вывела на рынок модифицированную улучшенную версию своего первого робота-пылесоса - ElectroluxTrilobite 2.0. И в этом же 2004-м на рынок выходят новые участники:

- LG со своим ROBOKING;
- Samsung с аппаратом модели Samsung VC-RP30W;
- Siemens с новым роботом-пылесосом SensorCruiser;
- Karcher, представивший прибор под названием RoboCleaner RC 3000.

Переломный момент в развитии данной индустрии наступил в 2005-м. Тогда на рынок вышли именитые игроки, не сходящие с позиций лидеров и до сегодняшнего дня – iClebo (Корея) и NeatoRobotics (США).

Начиная с 2007-го, ниша стала пополняться роботами-пылесосами от различных китайских изготовителей. Наряду с этим свои разработки презентовали такие известные фирмы, как:

- INFINUVO;
- PANASONIC;
- DYSON;
- HITACHI.

Начиная с 2017-го под брендом Xiaomi вышло еще свыше десятка интеллектуальных уборщиков от бюджетных до премиальных, предназначенные для решения различных задач. Важно подчеркнуть, что сегодня востребованность роботов пылесосов Xiaomi (за исключением продуктов RoboRock) затмила даже компанию iRobot, удерживавшую лидирующие позиции и ни с кем их не делившую с 2010-го. И этому есть веские основания: приемлемая стоимость товаров, оптимальный набор функций, исключительно положительные отклики покупателей о технике.

В 2016-м покупателям стали доступны мойщики бренда Everybot. А компания Lenovo своего первого робота выпустила только в 2019. Это была модель под названием Lenovo X1. Тогда же на свет появился и первый в мире робот для мойки пола от компании Narwal модели T10.

До сего момента особых переворотов в мире бытовой робототехники отмечено не было. Неизменно ожидаем новые флагманские продукты iRobot, Xiaomi, Nobot и прочих не менее популярных фирм.

### **1.3 Принцип работы робота-пылесоса**

Одним из основных составляющих устройства считается навигационная система. Благодаря ей робот способен легко ориентироваться в помещении. От того, как работает данная система, зависит, будет ли пылесос застревать или

теряться в пространстве, а также зависит его непосредственная работа. Эта навигационная система может быть различной, в зависимости от компании-производителя и модели аппарата. Но все же существуют основные системы навигации, которые пользуются большим спросом. К ним относят:

- лазер;
- внутренние датчики (в том числе гироскоп);
- внешние датчики;
- камера.

Каждая система навигации обладает своими преимуществами и недостатками, но принцип работы у них практически одинаковый. Принципы работы систем навигации будут описаны дальше.

Самой продвинутой считается навигация с помощью лазера. По такому принципу работают известные модели XiaomiMiRoborockSweepOne, NeatoBotvacConnected, EcovacsDeebotOzmo 930 и многие другие. Принцип действия лазера заключается в следующем: пылесос оснащается специальными лазерным дальномером (лидар), с помощью которого определяется наличие определенных предметов в комнате и расстояние до них (рисунок 1). Эти данные откладываются в памяти аппарата и сохраняются в виде карты помещения. После составления в памяти карты, прибор приступает к уборке каждой комнаты, при этом передвигаться он будет по прямым размеченным линиям. Возвращение на базу происходит по такому же принципу.



Рисунок 1 –Демонстрация работы лазерного дальномера

Если система работает по датчикам, то принцип действия состоит в следующем: благодаря датчикам, которые находятся под корпусом и в бампере прибора, робот-пылесос обнаруживает любые изменения в помещении и самостоятельно принимает решение, как продолжать дальнейшую уборку (рисунок 2). Это так называемые маяки, которые координируют движение аппарата.

К примеру, устройство обнаружило на своем пути кресло, тогда оно замедлит свое движение и при касании к мебели начнет очистку вдоль нее. Также устройство способно фиксировать наиболее загрязненные участки на ковре или полу. Их робот-пылесос чистит более тщательно. Это все зависит от того, сколько датчиков есть у аппарата, их типа, а также от того как быстро обрабатывается процессором сигнал. Каждая компания-производитель закладывает на свое усмотрение движения и действия, которые будет осуществлять аппарат в той или иной ситуации.



Рисунок 2 – Демонстрация работа-пылесоса с датчиками

Отдельно следует рассказать о том, что такое гироскоп в роботе-пылесосе и как он помогает ориентироваться в пространстве.

Гироскопом называется специальный механизм, который реагирует на изменение углов ориентации объекта (в данном случае робота-пылесоса), относительно инерциальной системы отсчета. Т.е. простыми словами гироскоп измеряет ускорение и угловую скорость пылесоса, что дает сведения о координатах в пространстве. Еще проще — робот запоминает откуда приехал, куда он движется и куда нужно вернуться (в нашем случае на базу после окончания уборки).

Внешние датчики – это, по сути, та же самая навигация с датчиками, только с применением внешних приборов для деления комнаты на определенные зоны. Благодаря этому уборка может быть более качественной, а также быстрой. Такие датчики еще называют виртуальными стенами, так как с их помощью ограничивается передвижение пылесоса.

Внешний датчик, это небольшая коробочка из пластика, которая подает невидимый инфракрасный сигнал. Если применяется маяк, то посылается



дополнительно два ИК-луча, благодаря которым робот-пылесос определяет нахождение маяка. Работа основного маяка – это перемещение прибора в другую комнату. Принцип работы заключается в прекращении подачи ИК-сигнала. Принцип действия внешнего датчика проще, чем маяка. Если робот фиксирует внешний датчик, то он просто меняет свое направление.

Один из вариантов навигации — магнитная лента для робота-пылесоса, которая создает так называемый «барьер», виртуальную стену, дальше которой аппарат проехать не сможет. На рисунке 3 наглядно демонстрируется принцип работы магнитной ленты:

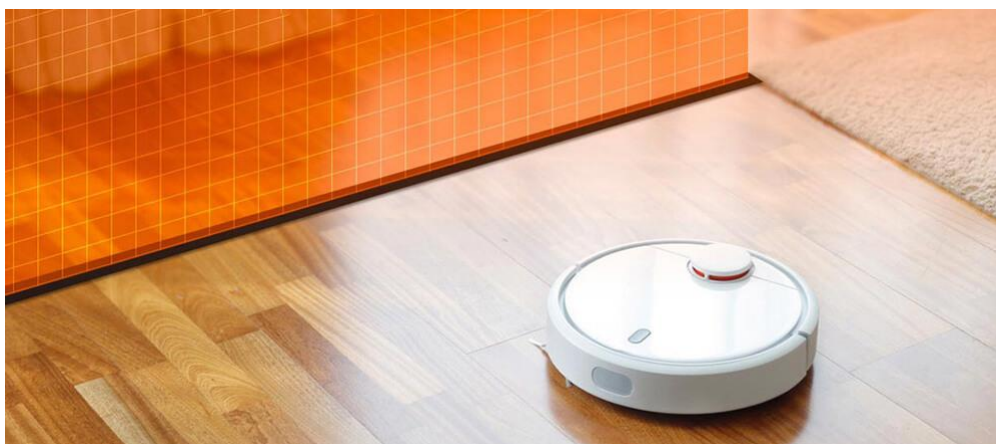


Рисунок 3 – Демонстрация работы магнитной ленты

Как работает навигация с помощью камеры? В большинстве случаев камера устанавливается сверху корпуса. Показания робот-пылесос снимает со стен и потолка. Такой принцип работы заложен в популярные роботы-пылесосы iCleboOmega, новый iClebo O5, iLife A8, флагман iRobotRoomba i7 и другие.

## 1.4 Сравнение моделей аналогов робота-пылесоса

Современный рынок насыщен роботами-пылесосами с различным функционалом и набором возможностей. Различаться они могут и скоростью работы, и мощностью, и степенью автономности, и конечно же ценой.

### 1.4.1 Garlyn SR-600

Garlyn SR-600 (рисунок 4) выполняет и сухую, и влажную уборку, при этом может похвастаться множеством дополнительных возможностей, включая функцию УФ-обеззараживания, особенно полезную в современных условиях.



Рисунок 4 – Пылесос Garlyn SR-600

Возможность непрерывной работы до 2,5 часов и пылесборник большого объема (0,6 л) позволяют использовать робот даже в больших по площади помещениях, а если этого времени не хватит, то после подзарядки робот сам вернется на точку, в которой уборка была прервана, чтобы продолжить ее с того же места. Его мощность подойдет не только для очистки гладких напольных покрытий, но и ковров со средним ворсом.

Для влажной уборки предусмотрен отдельный резервуар с системой автоподачи жидкости, он обеспечивает автономную работу прибора в данном режиме. Порадует также возможность настройки интенсивности подачи воды

на фибру. Пользователь может выбрать один из вариантов, наиболее подходящий под тот или иной тип покрытия.

За навигацию и автономность в построении маршрута отвечает система из 31 датчика и встроенной смарт-камеры с широким углом обзора 120 градусов. Карта помещения выстраивается максимально качественно, отображается в приложении и сохраняется в памяти. За счет использования оптического датчика робот считывает даже препятствия из стекла (в отличие от роботов с лазерными датчиками). Управление со смартфона максимально удобно реализовано, включает в себя помимо всего прочего возможность зонирования помещения для уборки.

Функция обеззараживания реализована с помощью УФ-лампы, которая в процессе уборки нейтрализует вредоносные частицы на поверхности пола. Такая функция будет полезна семьям с маленькими детьми и животными.

Среди дополнительных плюсов отметим компактность и проходимость, высота робота — менее 8 см., а значит он с легкостью может осуществлять очистку даже под мебелью на ножках. Максимальная комплектация (включает даже магнитную ленту) — еще одно преимущество в копилке этого робота-пылесоса.

#### Достоинства:

- сухая и влажная уборка;
- 31 датчик и смарт-камера для качественной навигации;
- построение карт и уборка по расписанию;
- непрерывная работа до 150 минут;
- функция УФ-обеззараживания;
- управление со смартфона;
- регулировка мощности работы.

#### Недостатки:

- маркая глянцевая поверхность корпуса;
- уровень шума выше среднего на максимальной мощности работы.

### 1.4.2 Roborock S6/T6

Робот-пылесос Roborock S6/T6 (рисунок 5) также порадует продвинутым функционалом, включающим как построение подробных карт помещения, так и удобное управление через приложение на смартфоне.



Рисунок 5 – Пылесос Roborock S6/T6

Заряда аккумулятора хватит на уборку в течение 150 минут, а скорость передвижения позволяет осуществлять за это время уборку больших по площади помещений. При этом стоит отметить небольшую вместимость пылесборника, что в целом все же может сказываться на производительности очистки.

Для измерения расстояния до имеющихся препятствий в конструкции в отличие от остальных участников здесь используется лазерный датчик. За счет него корпус робота имеет большие габариты в высоту, что стоит учитывать при планировке уборки труднодоступных мест.

Влажная уборка также включается в список функциональных возможностей робота-пылесоса. При этом важно указать, что салфетки для ее осуществления — одноразовые, а значит базового комплекта хватит на несколько циклов уборок, а затем необходимо будет докупать расходные материалы для дальнейшего использования функции влажной уборки пола.

Достоинства:

- сухая и влажная уборка;
- лазерный датчик для улучшенной навигации;
- построение карт с сохранением;
- управление со смартфона;
- непрерывная работа до 150 минут;
- уборка по расписанию.

Недостатки:

- небольшая вместимость пылесборника;
- салфетки для влажной уборки – одноразовые.

### 1.4.3 iCLEBO O5 WiFi

Робот-пылесос iCLEBO O5 WiFi (рисунок 6) — современный гаджет с оригинальным дизайном и широкими возможностями.

Для сухой уборки предусмотрено несколько режимов, отличающихся по мощности. С учетом высокой скорости передвижения робот может использоваться для уборки больших помещений, так как аккумулятор позволяет ему непрерывно работать до 120 минут. Управлять роботом можно с помощью смартфона, однако стоит отметить не самую удобную реализацию этой системы настройки робота.



Рисунок 6 – Пылесос iCLEBO O5 WiFi

Влажная уборка также присутствует в функционале, однако смачивать насадки придется вручную, сменного резервуара для влажной уборки не предусмотрено.

За навигацию отвечают датчики и камера, однако стоит отметить, что многие пользователи отмечают не самую лучшую отработку системы навигации и как следствие быструю порчу корпуса при столкновении с предметами мебели.

Достоинства:

- сухая и влажная уборка;
- оптические датчики для улучшенной навигации;
- построение карт с сохранением;
- управление со смартфона;
- непрерывная работа до 120 минут;
- высокая скорость движения.

Недостатки:

- влажная уборка без автоподачи воды;
- не самое комфортное управление со смартфона.

#### **1.4.4 iRobotRoomba e5**

Робот-пылесос iRobotRoomba e5 (рисунок 7) еще одна популярная модель, которая может похвастаться функцией управления со смартфона, уборки по расписанию и удобством пользования.



Рисунок 7 – Пылесос iRobot Roomba e5

В отличие от предыдущего участника эта модель предназначена только для сухой уборки, для чего оснащена пылесборником емкостью выше среднего (0,6 л).

Отличительная особенность — двойные щетки для собирания пыли выполненные без использования щетинок. Это в целом исключает риск наматывания, при этом слегка снижая производительность очистки.

В основном данный пылесос предназначен для твёрдых напольных покрытий.

Достоинства:

- программирование уборки;
- вместительный пылесборник 0,6 л;
- управление со смартфона;
- виртуальная стена в комплекте;
- непрерывная работа до 90 мин.

Недостатки:

- отсутствует функция влажной уборки;
- шумная работа.

**Таблица 1 – Сравнение характеристик представленных пылесосов**

Модель	Тип уборки	Режим уборки	Управление	Особенности	Навигация	Средняя цена
Garlyn SR-600	Сухая и влажная	Авто, По периметру, Точечная, Влажная, Нон-стоп, УФ-обеззараживание	Пульт ДУ, смартфон	Функция УФ-обеззараживания; Улучшенная картография; Влажная уборка с автоподачей воды; Регулировка мощности работы; Магнитная лента в комплекте	Камера, Датчики определения препятствий, перепадов высоты	39900
Roborock S6/T6	Сухая и влажная	Авто, По периметру, Точечная, Влажная	Пульт ДУ, смартфон	Качественная картография; Пониженный уровень шума; Салфетки для влажной уборки - одноразовые	Датчики определения препятствий, перепадов высоты, Лазерный датчик	49451
iCLEBO O5 WiFi	Сухая и влажная	Авто, По периметру, Точечная, Влажная	Пульт ДУ, смартфон	Влажная уборка без автоподачи воды; Высокая скорость передвижения	Камера, Датчики определения препятствий, перепадов высоты	44900
iRobot Roomba e5	Сухая	Авто, локальная	Смартфон	Двойная прорезиненная щетка для лучшего захвата мусора, возможность опустошения пылесборника при покупке специальной базы	Датчики определения препятствий, перепадов высоты	25490

В таблице 1 представлены основные характеристики на которые обращают внимание люди при покупке роботов-пылесосов. Сравнив эти характеристики и цены можно сделать вывод, что цена в основном зависит от типа уборки и от присутствующих режимов уборки, а так же от способа навигации.



## 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

### 2.1 Алгоритм работы

Работа робота-пылесоса проходит следующим образом. При подаче питания робот-пылесос начинает движение вперёд до тех пор, пока инфракрасный датчик не подаст сигнал об отсутствии пола под роботом или пока не сработают ультразвуковые датчики. Если сработал инфракрасный датчик, то робот-пылесос отъедет назад, а затем повернёт налево, после чего продолжит движение вперёд до следующего препятствия. Если же сработают ультразвуковые датчики, то робот повернёт в противоположную сторону сработанного датчика или отъедет и повернёт налево, если сработал ультразвуковой датчик спереди.

На рисунке 8 изображена структурная схема работы устройства.

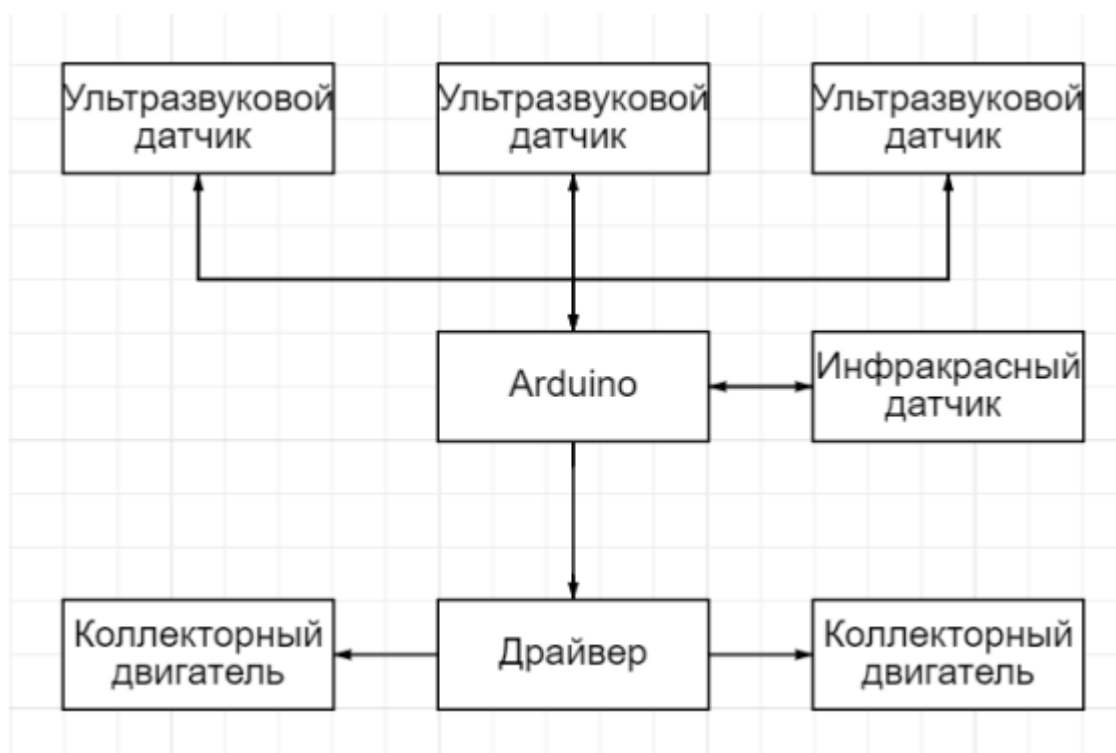


Рисунок 8 – Структурная схема принципа работы робота-пылесоса

Разработанная структурная схема рассматривает только управляющие сигналы и не рассматривает систему питания. Предполагается осуществлять питание от аккумуляторной батареи.

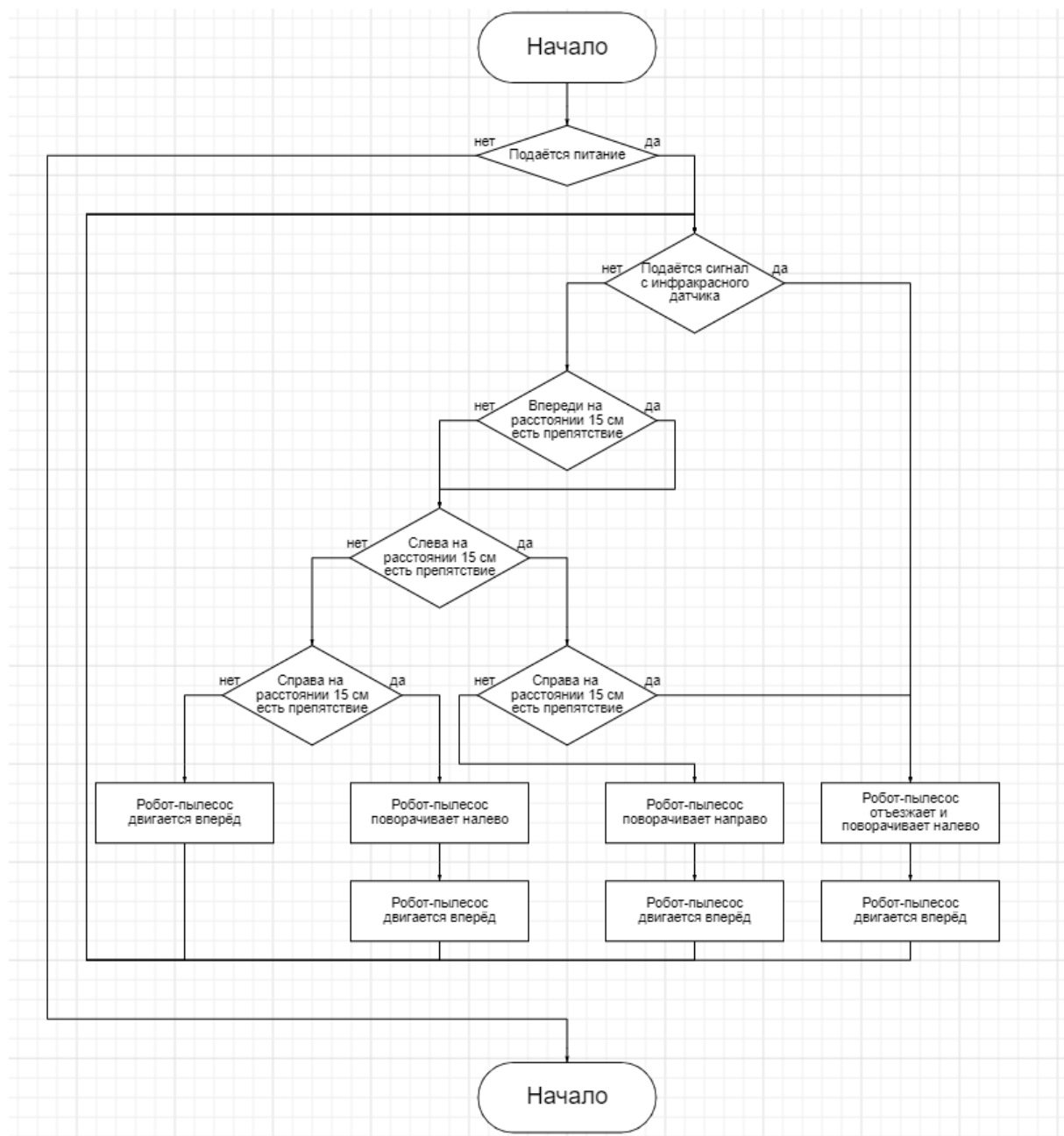


Рисунок 9 – Блок-схема алгоритма работы робота-пылесоса

Для удобства была разработана блок-схема алгоритма работы робота-пылесоса (рисунок 9). На её основе далее будет написана управляющая программа для микроконтроллера.

## 2.2 Макетирование устройства

При разработке электроники основной акцент делается на подбор комплектующих, ниже будут представлены компоненты, соответствующие разработанной структурной схеме.

### 2.2.1 Arduino Mega

Основой разрабатываемого прототипа робота-пылесоса является отладочная плата Arduino Mega (рисунок 10).



Рисунок 10 – Отладочная плата Arduino Mega 2560.

Arduino Mega построена на микроконтроллере ATmega2560 (техническое описание). Плата имеет 54 цифровых входа/выходов (14 из которых могут использоваться как выходы ШИМ), 16 аналоговых входов, 4 последовательных порта UART, кварцевый генератор 16 МГц, USB коннектор, разъем питания, разъем ICSP и кнопка перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареей. Arduino Mega 2560 совместима со всеми платами расширения, разработанными для платформ Uno или Duemilanove.

В таблице 2 показаны технические параметры отладочной платы Arduino 2560.

**Таблица 2 – Технические параметры отладочной платы Arduino Mega 2560**

Микроконтроллер	ATmega2560
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	54 (14 из которых могут работать также как выходы ШИМ)
Аналоговые входы	16
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	256 КВ (из которых 8 КВ используются для загрузчика)
ОЗУ	8 КВ
Энергонезависимая память	4 КВ
Тактовая частота	16 МГц

Для подключения устройств к Arduino, необходимо знать назначение и расположение выводов, для правильного подключения (Рисунок 12).

## ARDUINO MEGA 2560 PINOUT

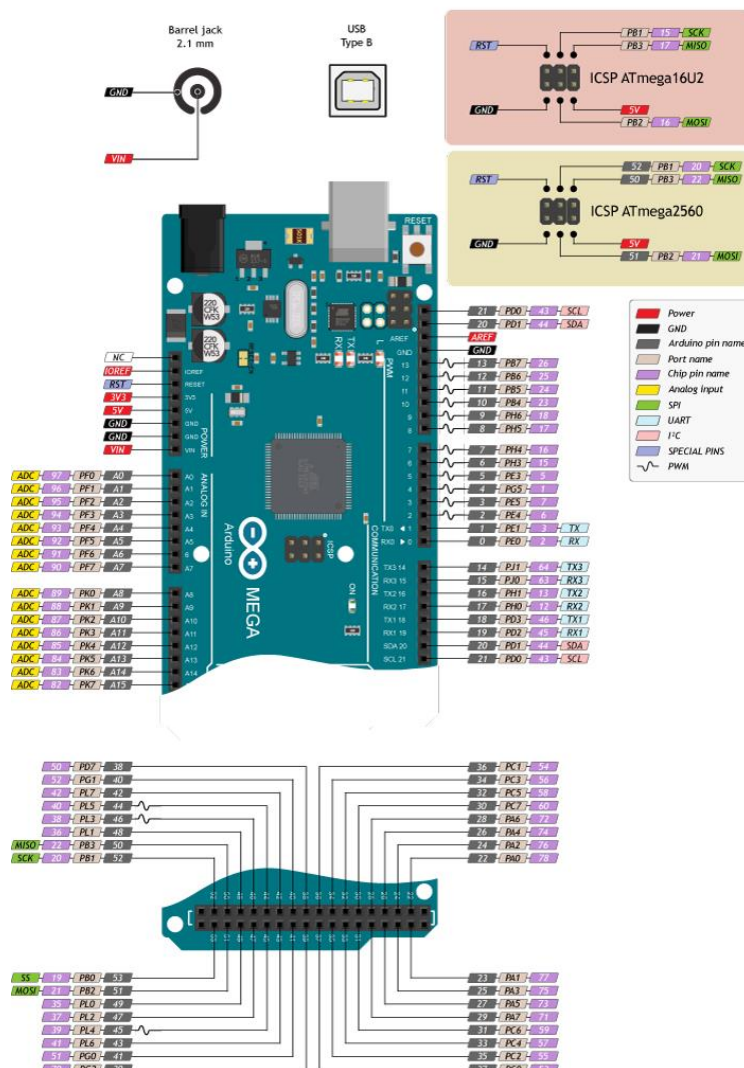


Рисунок 12 – Расположение выводов Arduino Mega 2560

Arduino Mega может получать питание как через подключение по USB, так и от внешнего источника питания. Источник питания выбирается автоматически.

Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с положительным полюсом на центральном контакте. Провода от батареи подключаются к выводам Gnd и Vin разъема питания (POWER).

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом

платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Плата Mega2560, в отличие от предыдущих версий плат, не использует FTDI USB микроконтроллер. Для обмена данными по USB используется микроконтроллер Atmega8U2, запрограммированный как конвертер USB-to-serial.

Выводы питания:

- VIN. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод;
- 5V. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В;
- 3V3. Напряжение на выводе 3.3 В генерируемое микросхемой FTDI на платформе. Максимальное потребление тока 50 мА;
- GND. Выводы заземления.

Микроконтроллер ATmega2560 имеет: 256 кБ флеш-памяти для хранения кода программы (4 кБ используется для хранения загрузчика), 8 кБ ОЗУ и 4 Кб EEPROM (которая читается и записывается с помощью библиотеки EEPROM).

Каждый из 54 цифровых выводов Mega, используя функции pinMode(), digitalWrite(), и digitalRead(), может настраиваться как вход или выход. Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать до 40 мА.

Некоторые выводы имеют особые функции:

- последовательная шина: 0 (RX) и 1 (TX); Последовательная шина 1: 19 (RX) и 18 (TX); Последовательная шина 2: 17 (RX) и 16 (TX); Последовательная шина 3: 15 (RX) и 14 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Выводы 0 и 1 подключены к соответствующим выводам микросхемы последовательной шины ATmega8U2;
- внешнее прерывание: 2 (прерывание 0), 3 (прерывание 1), 18 (прерывание 5), 19 (прерывание 4), 20 (прерывание 3), и 21 (прерывание 2). Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции `attachInterrupt()`;
- PWM: 2 до 13 и 44-46. Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции `analogWrite()`;
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Посредством данных выводов осуществляется связь SPI, например, используя библиотеку SPI. Также выводы SPI могут быть выведены на блоке ICSP, который совместим с платформами Uno, Duemilanove и Diecimila;
- LED: 13. Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит;
- I2C: 20 (SDA) и 21 (SCL). Посредством выводов осуществляется связь I2C (TWI). Для создания используется библиотека Wire (информация на сайте Wiring). Расположение выводов на платформе Mega не соответствует расположению Duemilanove или Diecimila.

На платформе Mega2560 имеется 16 аналоговых входов, каждый разрешением 10 бит (т.е. может принимать 1024 различных значения).

Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее имеется возможность изменить верхний предел посредством вывода AREF и функции `analogReference()`.

Дополнительная пара выводов платформы:

- AREF. Опорное напряжение для аналоговых входов. Используется с функцией `analogReference()`;
- Reset. Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на плате расширения, закрывающей доступ к кнопке на самой плате Arduino.

### **2.2.2 Инфракрасный датчик препятствий YL-63**

Цифровой инфракрасный датчик обхода препятствий YL-63 (или FC-51) (рисунок 13) применяется тогда, когда нужно определить наличие объекта, а точное расстояние до объекта знать необязательно. Датчик состоит из инфракрасного излучателя, и фотоприемника. ИК источник излучает инфракрасные волны, которые отражаются от препятствия и фиксируются фотоприемником. Датчик обнаруживает препятствия в диапазоне расстояний от нуля до установленной предельной границы. Он построен на основе компаратора LM393, который выдает напряжение на выход по принципу: обнаружено препятствие – логический уровень HIGH, не обнаружено – логический уровень LOW, данное состояние показывает и находящийся на датчике красный светодиод. Пороговое значение зависит от настройки датчика и регулируется с помощью установленного на модуле потенциометра. Для индикации питания на датчике установлен зеленый светодиод. Датчик применяется в робототехнике для обнаружения препятствий при движении колесных или гусеничных роботов.



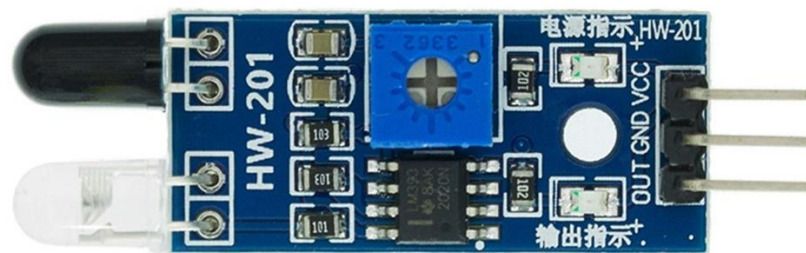


Рисунок 13 - Инфракрасный датчик препятствий YL-63

Технические характеристики датчика препятствия YL-63:

- модель: YL-63 (или FC-51);
- напряжение питания: 3.3 – 5 В;
- тип датчика: диффузионный;
- компаратор: LM393;
- расстояние обнаружения препятствий: 2 – 30 см;
- эффективный угол обнаружения препятствий: 35°;
- потенциометр для изменения чувствительности;
- светодиод индикации питания;
- светодиод индикации срабатывания;
- размеры: 43 x 16 x 7 мм.

Модуль имеет 3 вывода:

- VCC — питание 3-5 В;
- GND — земля;
- OUT — цифровой выход.

Подключение датчика к отладочной плате, на примере Arduino Uno показано на рисунке 14.

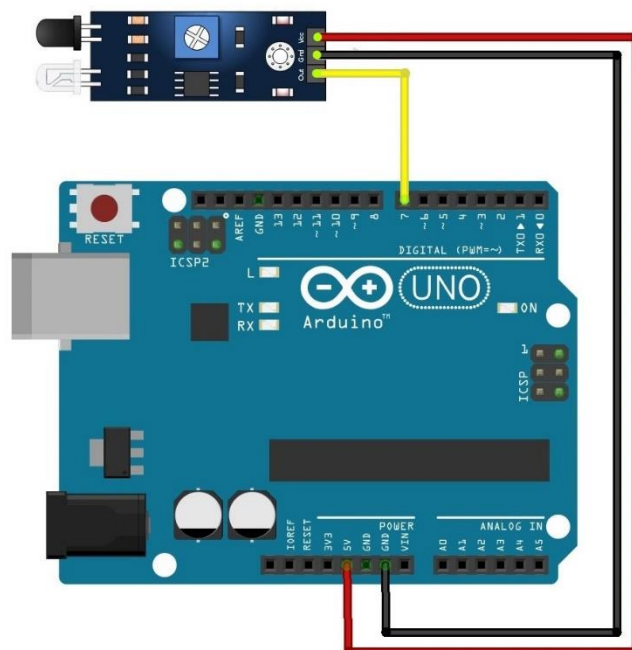


Рисунок 14 – Подключение ИК датчика к Arduino Uno

### 2.2.3 Ультразвуковой датчик HC-SR04

Датчик расстояния Arduino (рисунок 15) является прибором бесконтактного типа, и обеспечивает высокоточное измерение и стабильность. Диапазон дальности его измерения составляет от 2 до 400 см. На его работу не оказывает существенного воздействия электромагнитные излучения и солнечная энергия. В комплект модуля с HC SR04 arduino также входят ресивер и трансмиттер.



Рисунок 15 – Ультразвуковой датчик HC-SR04

Ультразвуковой дальномер HC SR04 имеет такие технические параметры:

- питающее напряжение 5В;
- рабочий параметр силы тока – 15 мА;
- сила тока в пассивном состоянии  $< 2$  мА;
- обзорный угол – 15°;
- сенсорное разрешение – 0,3 см;
- измерительный угол – 30°;
- ширина импульса –  $10^{-6}$  с.

Для получения данных, необходимо выполнить такую последовательность действий:

- подать на выход Trig импульс длительностью 10 микросек;
- в ультразвуковом дальномере hc sr04 подключенном к arduino произойдет преобразование сигнала в 8 импульсов с частотой 40 кГц, которые через излучатель будут посланы вперед;
- когда импульсы дойдут до препятствия, они отразятся от него и будут приняты приемником R, что обеспечит наличие входного сигнала на выходе Echo;
- на стороне контроллера полученный сигнал при помощи формул следует перевести в расстояние.

При делении ширины импульса на 58.2, получим данные в сантиметрах, при делении на 148 – в дюймах.

Выполнить подключение ультразвукового датчика расстояния к плате Arduino достаточно просто. Схема подключения показана на рисунке 16.

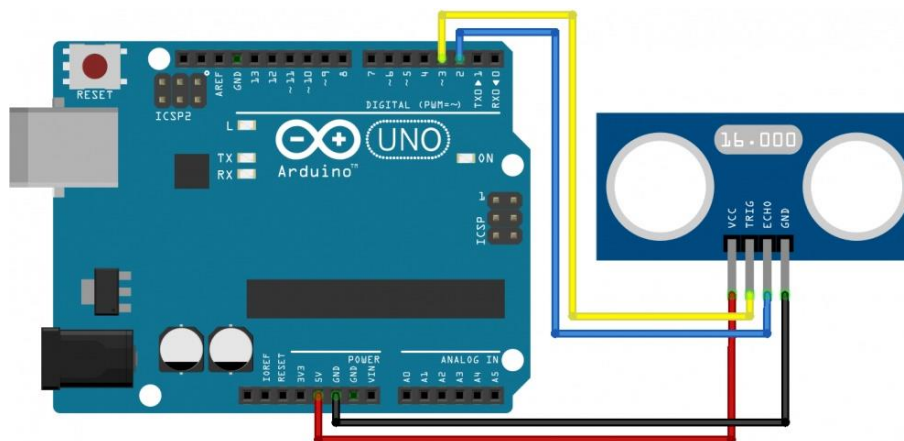


Рисунок 16 – Схема подключения датчика к Arduino

#### 2.2.4 Коллекторный двигатель постоянного тока

Коллекторный электродвигатель (рисунок 17) — электрическая машина, в которой датчиком положения ротора и переключателем тока в обмотках является одно и то же устройство — щёточно-коллекторный узел.

Щёточно-коллекторный узел — узел электрической машины, обеспечивающий электрическое соединение цепи ротора с цепями, расположенными в неподвижной части машины. Состоит из коллектора (набора контактов, расположенных на роторе) и щёток (скользящих контактов, расположенных вне ротора и прижатых к коллектору).



Рисунок 17 – Простой коллекторный двигатель

В коллекторном электродвигателе щёточно-коллекторный узел одновременно выполняет две функции:

- является датчиком углового положения ротора (датчик угла) со скользящими контактами;
- является переключателем направления тока со скользящими контактами в обмотках ротора в зависимости от углового положения ротора.

Принцип работы заключается в том, что при подаче питающего напряжения ток проходит сквозь обмотку возбуждения. У смежных полюсов возникает противоположная полярность, из-за чего образуется своеобразный магнит с постоянным магнитным полем. На якорь двигателя через коллектор подаётся постоянный ток, на который со стороны магнитного поля статора действует электромагнитная индукция. В результате создаётся вращающий момент, поворачивающий ротор на 90 электрических градусов. После этого щёточно-коллекторный узел коммутирует обмотки ротора, и вращение продолжается.

Подключение к Arduino Mega производится через специальный драйвер, а схема подключения показана на рисунке 18.

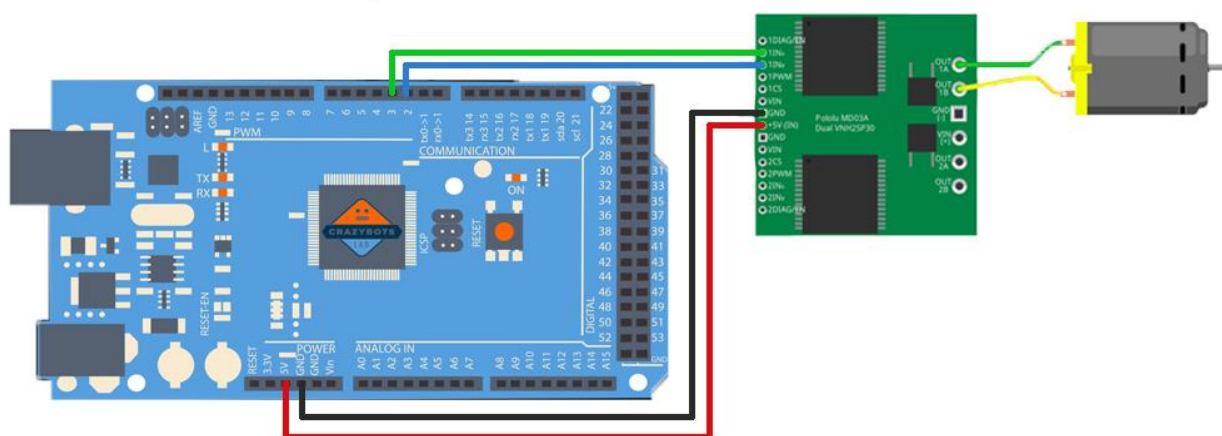


Рисунок 18 – Схема подключения коллекторного двигателя через драйвер

### 2.2.5 Драйвер L298N

Напрямую подключить электродвигатель к выводам Arduino нельзя, есть риск сжечь вывод, к которому подключён двигатель. Для безопасного подключения электродвигателей разных типов к Arduino необходим драйвер двигателей. Драйверы бывают разные, для их работы часто используются микросхемы типа HG788, L9110S, L293D, L298N и другие. Драйверы двигателей имеют выводы подачи питания, выводы для подключения электродвигателей, а также управляющие выводы.

Драйвер L298N используется радиолюбителями для многофункционального управления двигателями постоянного тока (рисунок 19). Схема модуля, состоящая из двух Н-мостов, позволяет подключать к нему один биполярный шаговый двигатель или одновременно два щёточных двигателя постоянного тока. При этом есть возможность изменять скорость и направление вращения моторов. Управление осуществляется путём подачи соответствующих сигналов на командные входы, выполненные в виде штыревых контактов. Технические характеристики драйвера показаны в таблице 3.

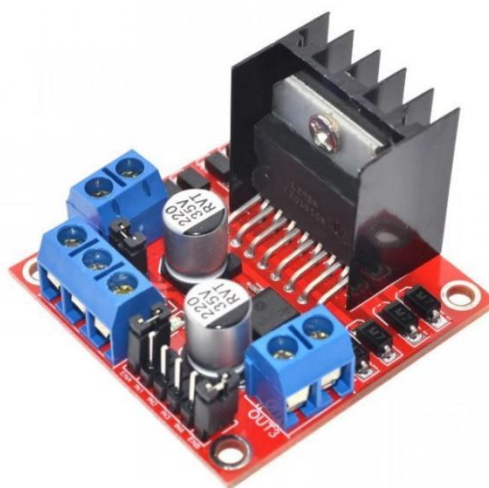


Рисунок 19 – Драйвер моторов L298N

**Таблица 3 – Технические параметры драйвера L298N**

Напряжение питания логики модуля	5 В
Потребляемый ток встроенной логики	36 мА
Напряжение питания драйвера	5 В – 35 В
Рабочий ток драйвера	2 А (пиковый ток 3 А)
Габариты	43.5 мм х 43.2мм х 29.4мм

Работать с модулем L298N довольно просто и комфортно. Все сводится к простому переключению логических уровней на выводах Arduino плюс к генерации ШИМ, если требуется управлять скоростью вращения. Так как драйвер способен управлять двумя типами моторов (щёточным или шаговым), то и работа с ним строится по-разному. Схема подключения драйвера и моторов показана на рисунке 20.

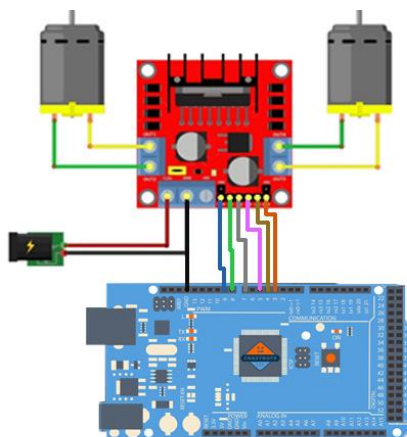


Рисунок 20 – Схема подключения драйвера с коллекторными двигателями к Arduino

## 2.3 Создание и отладка программы для Arduino Mega

Для создания программы для Arduino используется среда разработки Arduino IDE (см. рисунок 21).

Arduino IDE — интегрированная среда разработки для Windows, MacOS и Linux, разработанная на Си и C ++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей.

Arduino IDE предоставляет библиотеку программного обеспечения из проекта Wiring, которая предоставляет множество общих процедур ввода и вывода. Для написанного пользователем кода требуются только две базовые функции для запуска эскиза и основного цикла программы, которые скомпилированы и связаны с заглушкой программы `main ()` в исполняемую циклическую программу с цепочкой инструментов GNU, также включённой в дистрибутив IDE. Использует программу `avrdude` для преобразования исполняемого кода в текстовый файл в шестнадцатеричной кодировке, который загружается в плату Arduino программой-загрузчиком во встроенном программном обеспечении платы.

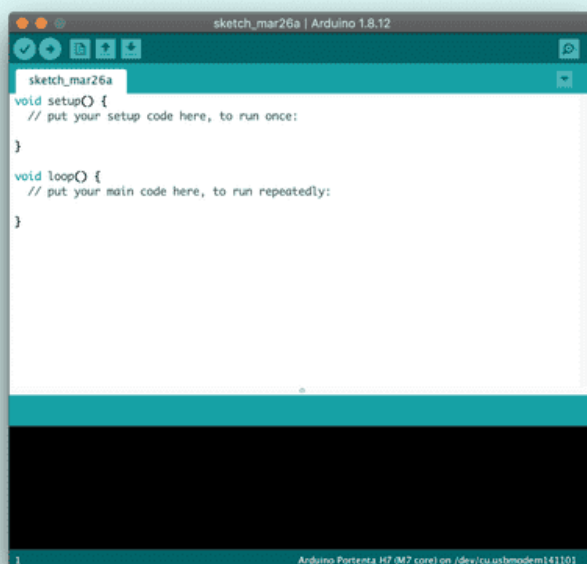


Рисунок 21 – Интерфейс Arduino IDE



В данной программе была написана управляющая программа для работы прототипа робота-пылесоса.

В коде программы не используются никакие внешние библиотеки, поскольку взаимодействие с датчиком HC-SR04 осуществляется достаточно просто.

Первым делом в программе объявляются переменные для взаимодействия контактами Echo и Trigger ультразвуковых датчиков (рисунок 22). Первый датчик стоит слева робота, второй – спереди, а третий – справа робота.

```
const int trigPin1 = 3;
const int echoPin1 = 5;
const int trigPin2 = 6;
const int echoPin2 = 9;
const int trigPin3 = 10;
const int echoPin3 = 11;
int irpin =2;
```

Рисунок 22 – Программа объявления переменных для контактов Echo и Trigger

Также мы необходимо объявить переменные для хранения значений расстояния от датчиков до препятствий (типа int), переменные для хранения длительностей распространения ультразвуковой волны (типа long) и переменную для хранения состояния движения (рисунок 23).

```
long duration1;
long duration2;
long duration3;
int distanceleft;
int distancefront;
int distanceright;
int a=0;
```

Рисунок 23 – Программа объявления переменных датчика препятствий

Далее необходимо задать режимы работы для всех используемых контактов, на ввод или вывод данных, с помощью функции pinMode(). Для считывания информации от инфракрасного датчика используется контакт irpin. И контакты, с которых осуществляется управление драйвером двигателей, конфигурируются на вывод данных OUTPUT (рисунок 24).

```

void setup() {
    pinMode(trigPin1, OUTPUT);
    pinMode(trigPin2, OUTPUT);
    pinMode(trigPin3, OUTPUT);
    pinMode(echoPin1, INPUT);
    pinMode(echoPin2, INPUT);
    pinMode(echoPin3, INPUT);
    pinMode(irpin, INPUT);
    pinMode(4, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(12, OUTPUT);
}

```

Рисунок 24 – Программа объявления режимов работы для контактов

В основной функции loop программы будет определяться расстояние до препятствия с помощью ультразвуковых датчиков (рисунок 25). Фрагменты кода для всех трех датчиков будут аналогичны. Для определения расстояния до препятствия сначала подаётся на контакт trigger датчика уровень LOW на 2 мкс, чтобы очистить его. Далее для излучения ультразвуковой волны передатчиком датчика подаётся на его контакт trigger уровень HIGH на 10 мкс. Далее время, в течение которого распространялась ультразвуковая волна до препятствия и обратно, определяется с помощью функции pulseIn() и сохраняется в переменной duration. У функции pulseIn() два параметра – контакт, на котором производится измерение (echo) и уровень (HIGH или LOW). Если установлен параметр HIGH, то функция ждет пока на контакте не появится состояние HIGH, после появления этого состояния она начинает счет и заканчивает она счет когда на контакте снова будет состояние LOW. Функция возвращает длительность импульса в микросекундах. Для расчета расстояния необходимо умножить измеренную длительность на 0.034 (эта константа получена исходя из скорости распространения звука в воздухе – 340 м/с) и поделить её на 2 (потому что измерена длительность распространения волны до препятствия и обратно). Определенное значение расстояние до препятствия сохраняется в соответствующей переменной.

```

void loop() {
    digitalWrite(trigPin1, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin1, LOW);
    duration1 = pulseIn(echoPin1, HIGH);
    distance1 = duration1 * 0.034 / 2;
}

```

Рисунок 25 – Программа определения расстояния

После определения расстояния до препятствия для всех ультразвуковых датчиков можно начать управление двигателями с помощью операторов if (рисунок 26). Расстояние, когда препятствие будет считаться близко расположенным к роботу, поставлено 15 см. К примеру, если левый датчик показываем расстояние меньше или равное 15 см, а расстояния с двух остальных датчиков большие, то даётся команда на поворот робота вправо. Также непрерывно проверяется состояние инфракрасного датчика. Если робот находится на полу, то состояние выходного контакта датчика будет равным LOW, в противном случае оно будет равно HIGH. Это состояние сохраняется в переменной s – ее значение также используется для управления движением робота.

```

if(s==HIGH)
{
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(1000);
    a=1;
}

```

Рисунок 26 – Программа движения робота вперед или назад

Когда робот обнаружит отсутствие пола, он будет двигаться назад, потом снова будет двигаться вперед и, таким образом, заикнется. Для преодоления этой проблемы используется переменную a, в которую записывается значение 1, которое будет свидетельствовать о том, что впереди робота нет пола, то есть он стоит на краю лестницы. В дальнейшем будет

использоваться значение этой переменной для формирования условий движения робота.

Таким образом, при обнаружении отсутствия пола робот не будет двигаться вперед – вместо этого он будет двигаться влево, этим и решится проблема его заклинивания.

На рисунках 27 и 28 показана программа поворота робота на право, таким же способом осуществляются повороты в другую сторону.

```
if ((a==0)&&(s==LOW)&&(distanceleft <= 150 && distancefront > 150 && distanceright <= 150) |  
{  
    digitalWrite(4, HIGH);  
    digitalWrite(7, LOW);  
    digitalWrite(8, HIGH);  
    digitalWrite(12, LOW);  
}  
  
| (a==0)&&(s==LOW)&&(distanceleft > 150 && distancefront > 150 && distanceright > 150))
```

Рисунок 27/28 – Программа поворота робота на право

Полный код программы представлен в приложении 1.

## 2.4 Разработка устройства в эмуляторе Tinkercad

TinkerCAD —это online-сервис и среда моделирования для работы с 3D объектами и электронными схемами, принадлежащий в настоящее время компании Autodesk. Интерфейс программы можно посмотреть на рисунке 29.

Отличительными особенностями являются:

- онлайн-платформа;
- открытость;
- бесплатный доступ;
- богатые функциональные возможности редактора.

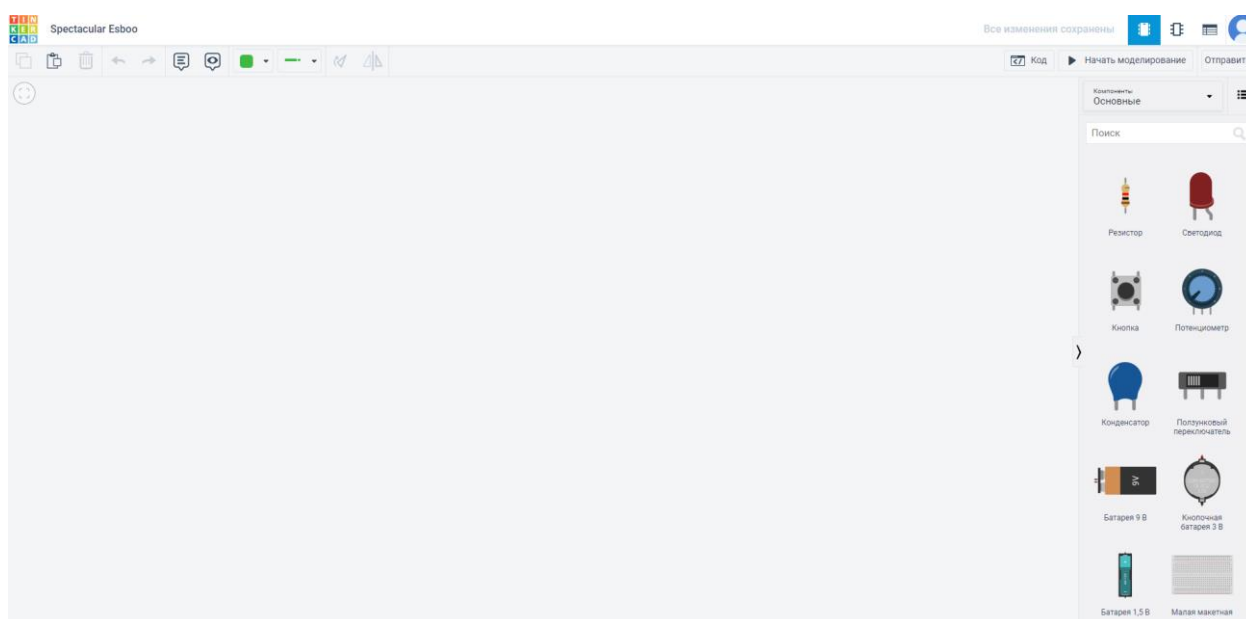


Рисунок 29 – Интерфейс Tinkercad

Для создания проекта в Tinkercad есть все необходимые элементы, кроме драйвера двигателей, но вместо него и обычных коллекторных двигателей будут использоваться моторы-редукторы.

После подключения всех элементов и загрузки кода управляющей программы, можно начать тестирование (рисунок 30).

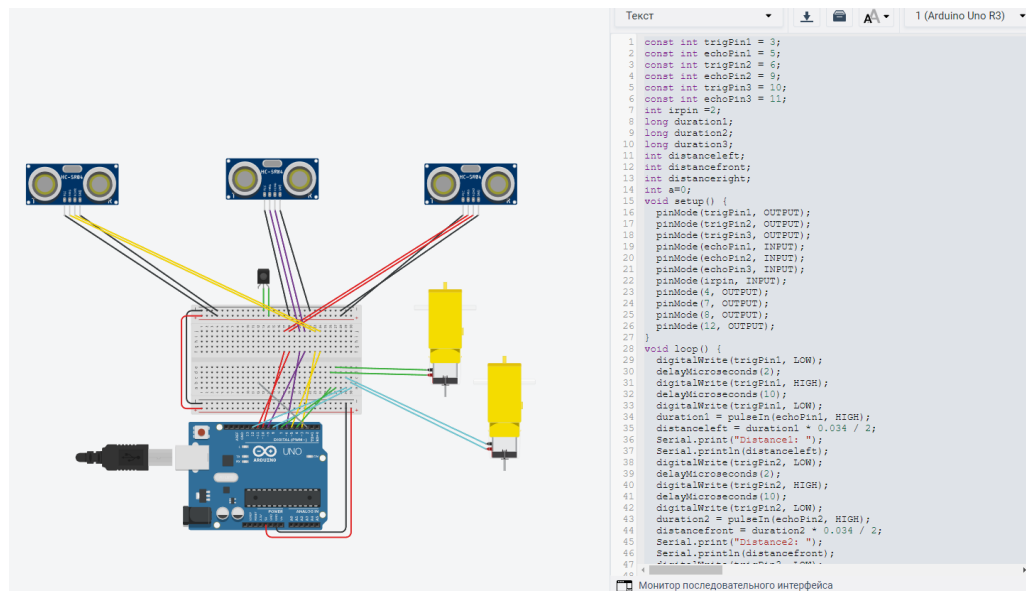


Рисунок 30 – Собранная электронная схема проекта

После начала моделирования на моторах будет отображаться их скорость вращения, а ультразвуковым датчикам можно будет настроить расстояние до предполагаемого объекта (рисунок 31).

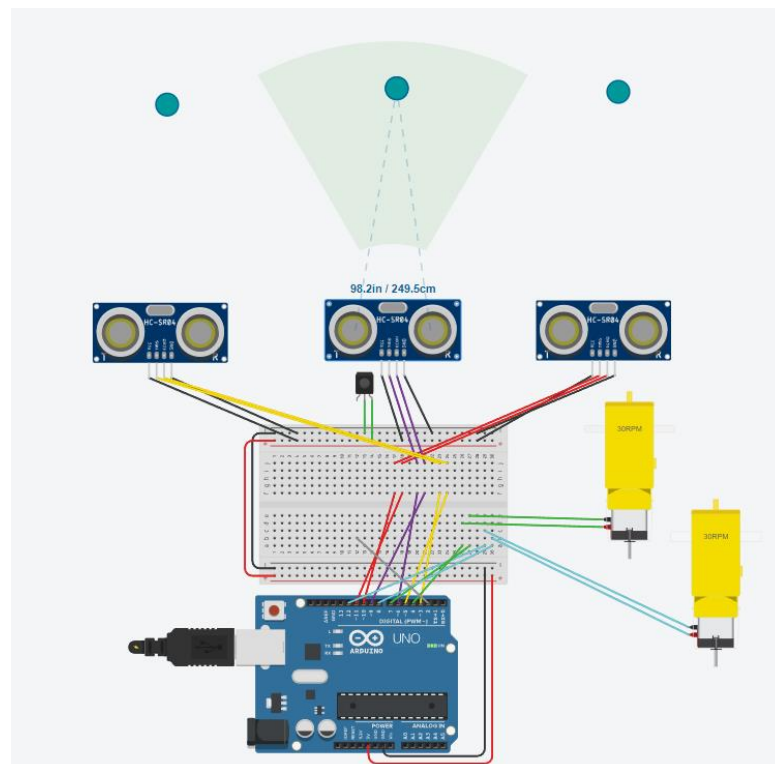


Рисунок 31 – Электронная схема в режиме моделирования

Для удобства тестов, в коде были изменены значения обнаружения препятствия с 15 см до 150 см. Так же изначально к Arduino не подключен инфракрасный датчик, что будет означать что сигнал с него не подаётся (рисунок 32).

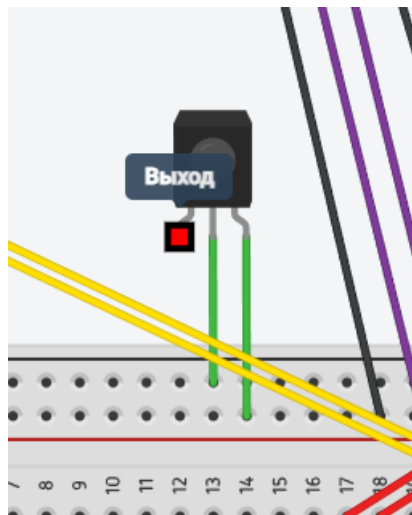


Рисунок 32 – Не подключённый инфракрасный датчик

Для начала проверим работу моторов при обнаружении препятствия спереди от робота-пылесоса (рисунок 33).

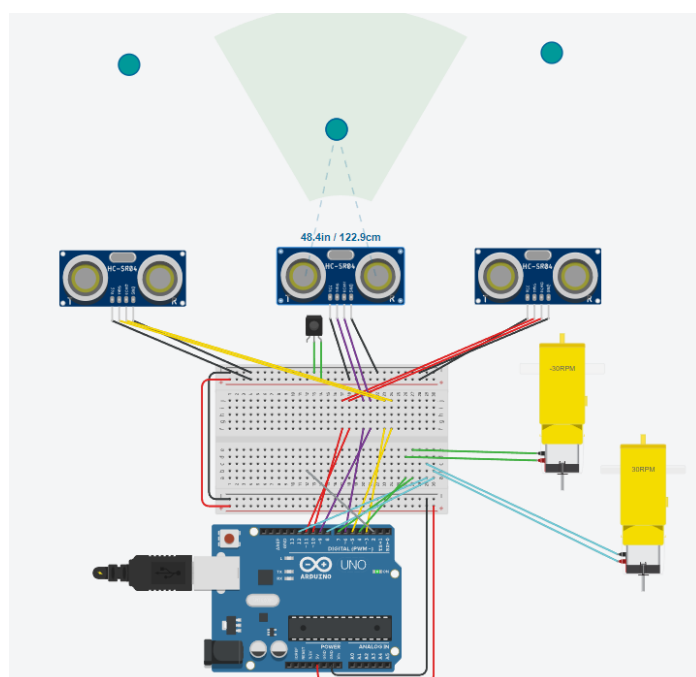


Рисунок 33 – Работа моторов при обнаружении препятствия спереди

На рисунке 34 показана работа моторов при обнаружении препятствий слева и спереди от робота-пылесоса.

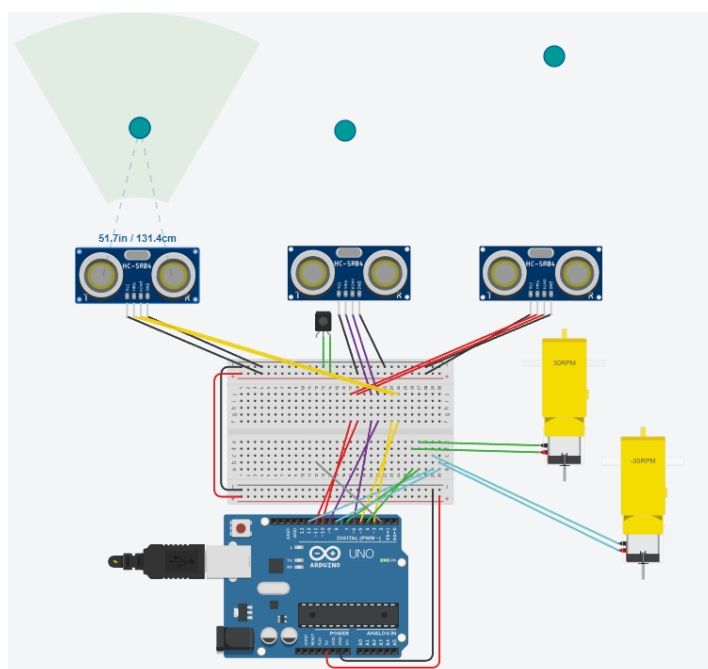


Рисунок 34 – Работа моторов при обнаружении препятствий слева и спереди

На рисунке 35 показана работа моторов при обнаружении препятствия только слева от робота-пылесоса.

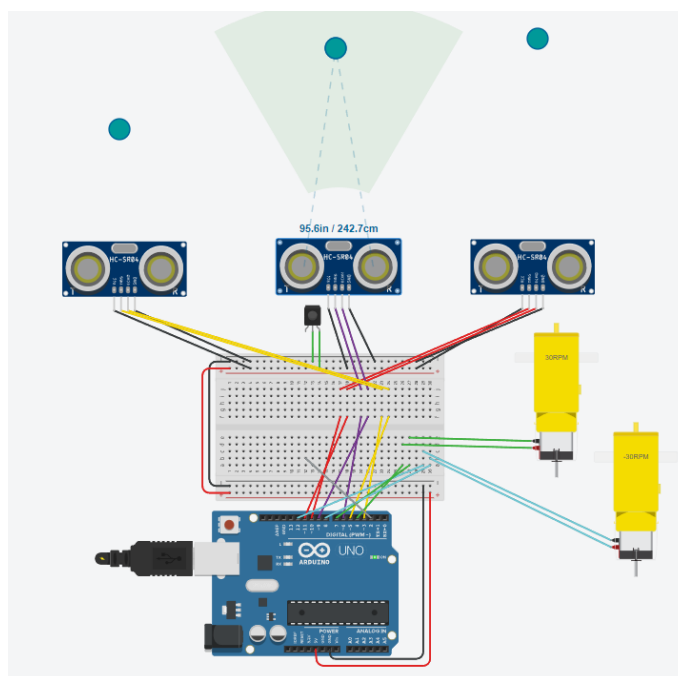


Рисунок 35 – Работа моторов при обнаружении препятствия слева



На рисунке 36 показана работа моторов при обнаружении препятствия только справа от робота-пылесоса.

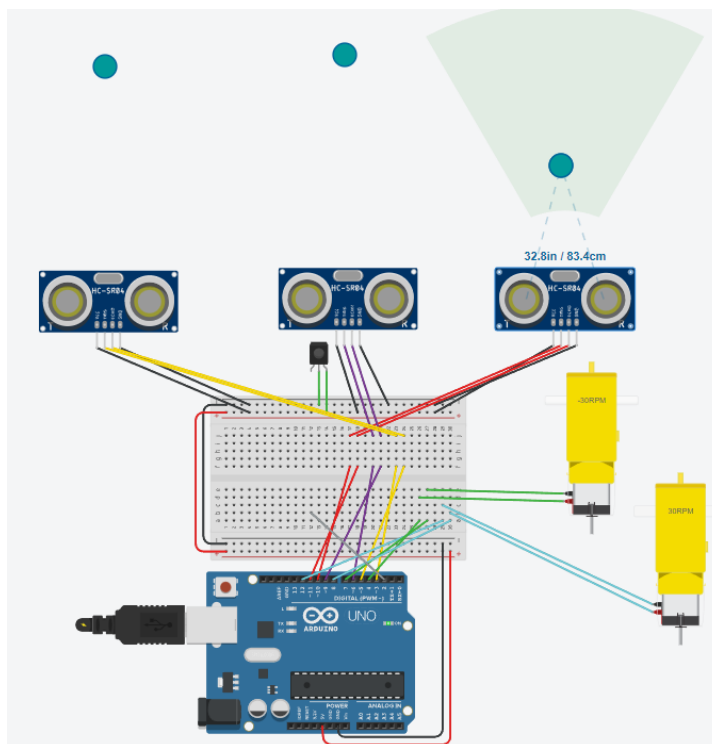


Рисунок 36 – Работа моторов при обнаружении препятствия справа

На рисунке 37 показана работа моторов если инфракрасный датчик будет подключён к Arduino, что означает что он подаёт высокий сигнал.

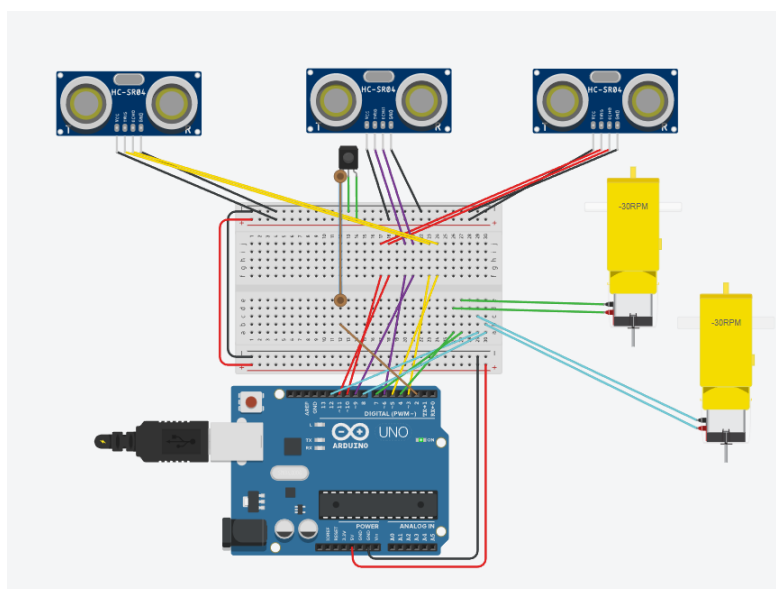


Рисунок 37 – Работа моторов при подаче сигнала с инфракрасного датчика

На рисунке 38 показана работа моторов если инфракрасный датчик будет подключён к Arduino, при этом ультразвуковые датчики тоже будут обнаруживать препятствия.

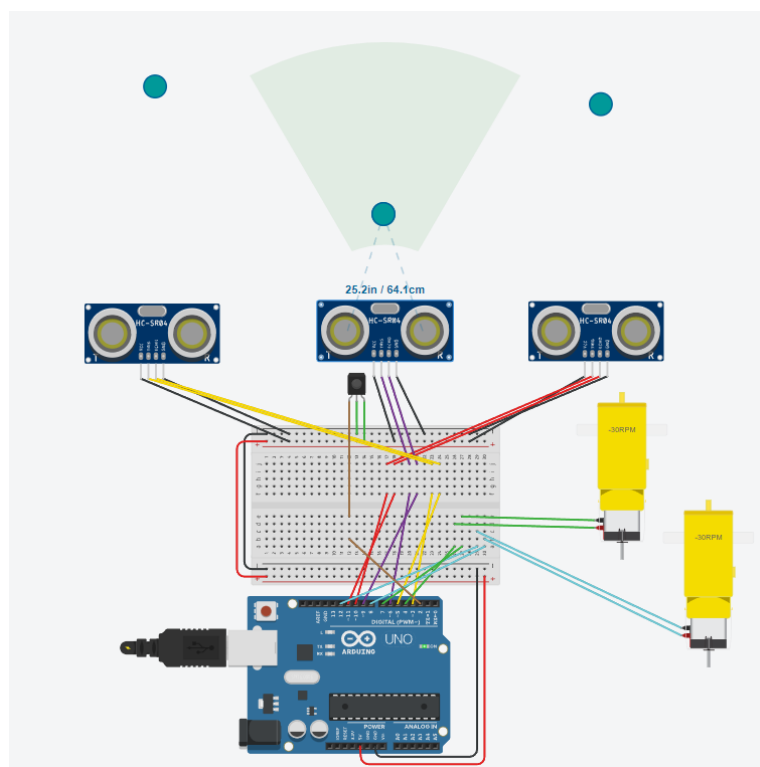


Рисунок 38 – Работа моторов при обнаружении препятствия и с подключённым инфракрасным датчиком

Таким образом в практической части была рассмотрена структурная схема, в соответствии выбрана элементная база, разработана блок-схема алгоритма, по которой написана управляющая программа для микроконтроллера. Отладка управляющей программы осуществлена в эмуляторе Tinkercad.

### **3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ**

#### **3.1 Организационно-экономическое обоснование проекта**

Для обоснования экономической целесообразности разработки и внедрения в практическую деятельность предложенного устройства ранее был проведён сравнительный анализ имеющихся аналогов.

Учитывая дороговизну аналогов, представленных на рынке, разработанное устройство будет способно занять достойную позицию для целевого потребителя. Плюсом данной системы является модульность элементной базы, что позволит пользователю усовершенствовать систему под свои конкретные задачи.

#### **3.2 Расчет себестоимости**

При расчете себестоимости стоит учитывать все затраты на создание проекта.

##### **3.2.1 Расчет трудоемкости**

Форма расположения работ по этапам:

1. Разработка технического задания (ТЗ):
  - получение ТЗ.
2. Подготовительный этап:
  - сбор информации;
  - обзор готовых решений;
  - обзор компонентов и их преимуществ.
3. Основной этап:
  - написание и отладка текста программы;
  - сборка модели устройства.
4. Завершающий этап:
  - тестирование и моделирование устройства;
  - подготовка технической документации;
  - сдача работы.

Расчет трудоемкости представлен в таблице 4

**Таблица 4 – Расчёт трудоёмкости**

Виды работ	Трудоемкость час.
Получение ТЗ	1
Сбор информации и ознакомление с предметной областью	15
Обзор готовых решений	5
Обзор компонентов и их преимуществ	15
Составление электронной схемы	5
Разработка и написание программы	20
Сборка модели устройства	4
Тестирование и моделирование устройства	10
Подготовка технической документации	30
Сдача продукта	2
Итого	107

Суммарная трудоемкость составляет 107 часов

### 3.2.2 Расчет затрат на материалы

Расчет затрат представлен в таблице 5.

**Таблица 5 – Материальные затраты**

Наименование	Количество, шт.	Цена ед., руб.	Сумма, руб.
Плата Arduino Mega	1	2 640	2 640
Инфракрасный датчик YL-63	1	590	590
Ультразвуковой датчик HC-SR04	3	145	435
Коллекторный двигатель	2	260	520
Драйвер L298N	1	250	250
Переключатель	1	200	200
Литий-ионная батарея	1	1 200	1 200
Портативный вакуумный пылесос	1	1 000	1 000
Опорные ролики	3	350	1 050
Изготовление корпуса	1	3 000	3 000
Итого			10 885

### 3.2.3. Расчет затрат на электроэнергию амортизации оборудования

Стоимость компьютера 25 тыс. руб. Используем его в течении 3 года (36 мес.). Годовая амортизация составит в месяц 694 руб. в час 4,1 руб. Умножив на трудоёмкость, определим амортизационные отчисления:

$$АО = 4,1 \times 107 = 438,7 \text{ руб.}$$

Компьютер в среднем потребляет 0,24 кВт/час. Умножим это значение на общие часы трудоёмкости:

$$0,24 \times 107 = 25,68 \text{ кВт}$$

Примерная стоимость 1 кВт/ч – 4,98 рубля. Итого затраты на электроэнергию будут:

$$25,68 \times 4,98 = 127.87 \text{ руб.}$$

### **3.2.5. Расчет начислений на заработную плату**

На основе данных о трудоемкости и средней заработной плате по отрасли, рассчитываем основную заработную плату. Предположим, что заработная плата составляет 40 000 руб. в месяц. (22 рабочих дня, 8 часовой рабочий день) или 227,27 руб./час. Исходя из полученных данных, расходы на заработную плату составляют:

$$\text{ЗП} = 227,27 \times 107 = 24\,317,89 \text{ руб.}$$

Отчисления в пенсионный фонд (ПФ):

$$\text{Зпф} = 24\,317,89 \times 0,22 = 5\,349,94 \text{ руб.}$$

Отчисления в фонд социального страхования (ФСС):

$$\text{Зфсс} = 24\,317,89 \times 0,029 = 705,21 \text{ руб.}$$

Федеральный фонд обязательного медицинского страхования (ФФОМС):

$$\text{Зффомс} = 24\,317,89 \times 0,051 = 1\,240,21 \text{ руб.}$$

Отчисления на страхования от несчастных случаев на производстве:

$$\text{Зснсп} = 24\,317,89 \times 0,002 = 48.64 \text{ руб.}$$

Таким образом, страховые взносы составляют:

$$\text{Зстрах} = 5\,349,94 + 705,21 + 1\,240,21 + 48,64 = 7\,344 \text{ руб.}$$

### 3.2.6 Общие расходы на создание проекта

Смета всех затрат представлена в таблице 6.

**Таблица 6 – Смета всех затрат**

Наименование статей затрат	Сумма, руб.
Основная заработная плата	24 317,89
Затраты на материалы	10 885
Расходы на электроэнергию	127,87
Амортизационные отчисления	438,7
Начисления на ЗП	7 344
Итого	43 113,46

### 3.3 Расчет экономического эффекта

Экономическим эффектом (выгодой) является предполагаемая прибыль от реализации созданной разработки:

Предположим, что на данное устройство можно установить розничную цену в 55 000 рублей, при этом можно получить ощутимую прибыль с реализации даже одного изделия. Прибыль от реализации данного изделия составит:  $55\,000 - 43\,113,46 = 11\,886,54$ . На рынке безусловно существует огромное количество роботов-пылесосов и по меньшей цене, но в основном цена робота пылесоса варьируется в районе 50 000 – 60 000 рублей. В

Так же разработанное устройство может выделяться модульностью конструкции, так как пользователь имеет непосредственный доступ ко всем элементам.

## **Заключение**

В представленном проекте был спроектирован и макетирован прототип робота-пылесоса.

Было разработано устройство, которое отвечает поставленным задачам. В процессе разработки были применены теоретические и практические навыки проектирования цифровых устройств на базе микроконтроллеров, а также отладки микропроцессорных систем и написано программное обеспечение.

При запуске в массовое производство, устройство будет обладать относительно невысокой стоимостью, что делает его актуальным для среднестатистического потребителя.

Так же, учитывая наличие на отладочной плате порта USB, имеется возможность подключения её к ПК, что делает возможным ввод данных непосредственно с компьютера.



## Список литературы

1. Книга Веретехина, С. В. Модели, методы, алгоритмы и программные решения вычислительных машин, комплексов и систем / С. В. Веретехина, В. Л. Симонов, О. Л. Мнацаканян. – Москва ; Берлин : Директ-Медиа, 2021. – 307 с.
2. Книга Береснев, А. Л. Разработка и макетирование микропроцессорных систем / А. Л. Береснев, М. А. Береснев ; Южный федеральный университет, Инженерно-технологическая академия. – Таганрог : Южный федеральный университет, 2016. – 108 с.
3. Книга Бейктал, Д. Конструируем роботов от А до Я. Полное руководство для начинающих / Бейктал Д. - Москва :Лаборатория знаний, 2018. - 397 с.
4. Книга Ревич «Практическое программирование Микроконтроллеров AVR на языке ассемблера»
5. Книга Джереми Блум – Изучаем Arduino "Изучаем Arduino: инструменты и методы технического волшебства"
6. Статья на Википедии о роботах-пылесосах <https://ru.wikipedia.org/wiki/Робот-пылесос#История>
7. Статья «История создания роботов-пылесосов» <https://robot-store.com.ua/istoriya-sozdaniya-robotov-pylesosov/>
8. Статья «Лучшие роботы-пылесосы — ТОП-6» [https://obzor-tovarov.pro/luchshye-roboty-pylesosy-top-6/?utm\\_campaign=search\\_rob\\_pyl\\_%28reyting-reyting%29\\_rate&utm\\_medium=cpc&utm\\_source=yandex&utm\\_term=как%20автоматизированный%20пылесос&utm\\_content=k50id%7C0100000031033431263\\_%7Ccid%7C58671954%7Cgid%7C4531415765%7Caid%7C10586015705%7Cadp%7Cno%7Cpos%7Cpremium1%7Csrc%7Csearch\\_one%7Cdvc%7Cdesktop%7Cmain&etext=2202.JEu4fV97bM-](https://obzor-tovarov.pro/luchshye-roboty-pylesosy-top-6/?utm_campaign=search_rob_pyl_%28reyting-reyting%29_rate&utm_medium=cpc&utm_source=yandex&utm_term=как%20автоматизированный%20пылесос&utm_content=k50id%7C0100000031033431263_%7Ccid%7C58671954%7Cgid%7C4531415765%7Caid%7C10586015705%7Cadp%7Cno%7Cpos%7Cpremium1%7Csrc%7Csearch_one%7Cdvc%7Cdesktop%7Cmain&etext=2202.JEu4fV97bM-)

[pCerFtzMF9KZQ04SB8PGzdOVJbVacnbYF5awlq6zoY7Mx89-2jZdPeN70oyC665DEMEKlooO\\_rnBzb2V1Y2F5cXh0dmRkYXg.78266056a2a1f399ee33dfa911be0684fb68f01b&yclid=2686957596490912896](https://pCerFtzMF9KZQ04SB8PGzdOVJbVacnbYF5awlq6zoY7Mx89-2jZdPeN70oyC665DEMEKlooO_rnBzb2V1Y2F5cXh0dmRkYXg.78266056a2a1f399ee33dfa911be0684fb68f01b&yclid=2686957596490912896)

9. Статья «Как работает робот-пылесос для сухой и влажной уборки»

<https://tehnikoved.ru/poryadok-i-chistota/pylesos/rabota-robotapylesosa.html>

- 10.Статья «Что такое Arduino?»

[HTTPS://CLUB.DNS-SHOP.RU/BLOG/T-335-NABORYI-I-KONSTRUKTORYI/30666-CHTO-TAKOE-ARDUINO/?UTM\\_REFERRER=HTTPS%3A%2F%2FYANDEX.RU%2F](HTTPS://CLUB.DNS-SHOP.RU/BLOG/T-335-NABORYI-I-KONSTRUKTORYI/30666-CHTO-TAKOE-ARDUINO/?UTM_REFERRER=HTTPS%3A%2F%2FYANDEX.RU%2F)

- 11.Справочник функций и операторов для программирования Arduino

<HTTPS://ALL-ARDUINO.RU/PROGRAMMIROVANIE-ARDUINO/>

## Приложение 1

Полный код программы:

```
const int trigPin1 = 3;
const int echoPin1 = 5;
const int trigPin2 = 6;
const int echoPin2 = 9;
const int trigPin3 = 10;
const int echoPin3 = 11;
int irpin =2;
long duration1;
long duration2;
long duration3;
int distanceleft;
int distancefront;
int distanceright;
int a=0;
void setup() {
  pinMode(trigPin1, OUTPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(echoPin2, INPUT);
  pinMode(echoPin3, INPUT);
  pinMode(irpin, INPUT);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
}
void loop() {
```

```

digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distanceleft = duration1 * 0.034 / 2;
Serial.print("Distance1: ");
Serial.println(distanceleft);
digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distancefront = duration2 * 0.034 / 2;
Serial.print("Distance2: ");
Serial.println(distancefront);
digitalWrite(trigPin3, LOW);
delayMicroseconds(2);
digitalWrite(trigPin3, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin3, LOW);
duration3 = pulseIn(echoPin3, HIGH);
distanceright = duration3 * 0.034 / 2;
Serial.print("Distance3: ");
Serial.println(distanceright);
int s = digitalRead(irpin);
if(s==HIGH)

```

```

{
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(1000);
    a=1;
}

if ((a==0)&&(s==LOW)&&(distanceleft <= 15 && distancefront > 15 &&
distancefront > 15 && distanceright > 15) || (a==0)&&(s==LOW)&&(distanceleft > 15 && distancefront
> 15 && distanceright > 15))
{
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(12, LOW);
}

if ((a==1)&&(s==LOW)||s==LOW)&&(distanceleft <= 15 && distancefront <=
15 && distanceright > 15)||s==LOW)&&(distanceleft <= 15 && distancefront <=
15 && distanceright > 15)||s==LOW)&& (distanceleft <= 15 && distancefront >
15 && distanceright > 15)||distanceleft <= 15 && distancefront > 15 &&
distanceright > 15))
{
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(100);
    a=0;
}

```

```

    }
    if ((s==LOW)&&(distanceleft > 15 && distancefront <= 15 && distanceright <=
15) ||(s==LOW)&& (distanceleft > 15 && distancefront > 15 && distanceright <=
15) ||(s==LOW)&& (distanceleft > 15 && distancefront <= 15 && distanceright >
15) )
    {
        digitalWrite(4, LOW);
        digitalWrite(7, HIGH);
        digitalWrite(8, HIGH);
        digitalWrite(12, LOW);
    }
}

```