

# СОДЕРЖАНИЕ

Введение.....	6
1 Обзор литературы .....	7
1.1 Микроконтроллерные платформы .....	7
1.2 Радиомодули для беспроводной связи.....	8
1.3 Системы привода и драйверы моторов.....	8
1.4 Датчики .....	9
1.5 Индикация и пользовательский интерфейс.....	10
1.6 Датчики освещенности .....	10
2 Структурное проектирование .....	12
2.1 Общая структура системы.....	12
2.2 Структура пульта управления.....	12
2.3 Структура передвижной модели.....	13
2.4 Взаимодействие модулей .....	13
2.5 Преимущества выбранной структуры.....	14
2.6 Перспективы развития структуры .....	14
3 Функциональное проектирование .....	16
3.1 Обоснование выбора микроконтроллеров.....	16
3.2 Обоснование выбора датчика освещенности .....	16
3.3 Обоснование выбора ультразвукового датчика расстояния .....	16
3.4 Обоснование выбора модулей радиопередачи.....	17
3.5 Обоснование выбора драйвера моторов .....	18
3.6 Обоснование выбора мотор-редукторов.....	18
3.7 Система освещения .....	19
4 Разработка принципиальной электрической схемы устройства .....	20
4.1 Расчёт мощности элементов схемы.....	21
4.2 Расчёт нагрузки светодиодов .....	21
4.3 Микроконтроллеры .....	22
4.4 Датчик освещенности .....	23
4.5 Ультразвуковой датчик расстояния .....	23
4.6 Модуль радиопередачи .....	23
4.7 Драйвер моторов .....	24
4.8 Мотор-редукторы .....	24
4.9 Схема расширения ввода-вывода на регистре сдвига .....	24
5 Разработка программного обеспечения .....	25
5.1 Требования к разработке программного обеспечения .....	25
5.2 Блок-схема алгоритма.....	25
5.3 Исходный код программы для устройства управления .....	26
5.4 Исходный код программы для передвижного устройства .....	27
6 Разработка корпуса устройства .....	29
6.1 Концепция и материалы .....	29
6.2 Конструкция передвижного устройства .....	29
6.3 Конструкция пульта управления .....	30
7 Руководство пользователя .....	31

7.1 Назначение устройства .....	31
7.2 Комплектация .....	31
7.3 Подготовка к работе.....	31
7.4 Порядок работы .....	31
7.5 Меры предосторожности.....	32
7.6 Уход и техническое обслуживание .....	32
Заключение .....	33
Список использованных источников .....	34
Приложение А .....	35
Приложение Б .....	40
Приложение В.....	41
Приложение Г .....	42
Приложение Д.....	43
Приложение Е .....	44
Приложение Ж.....	45

## ВВЕДЕНИЕ

Современные робототехнические системы находят всё более широкое применение как в промышленности, так и в быту. Они используются для автоматизации рутинных процессов, проведения исследовательских работ, образовательных целей и развлечений. Одним из наиболее перспективных направлений в образовательной робототехнике является разработка небольших мобильных роботов, управляемых при помощи микроконтроллеров и радиомодулей. Такие проекты позволяют изучить основы электроники, программирования встроенных систем, принципы построения беспроводных сетей и работу исполнительных механизмов.

Образовательные проекты на базе платформ Arduino стали неотъемлемой частью инженерной подготовки, позволяя студентам освоить полный цикл создания устройств – от проектирования до программирования и тестирования.

Настоящий проект посвящён созданию прототипа мобильного робота с дистанционным управлением. Управление осуществляется с отдельного пульта при помощи радиоканала на базе модуля NRF24L01. Робот оснащён системой привода на основе колёсных электродвигателей, датчиками для ориентации в пространстве и модулем обратной связи. Конструкция предполагает возможность расширения функционала – добавление новых сенсоров или усовершенствование системы управления.

Основная цель проекта заключается в разработке структуры системы, обеспечивающей взаимодействие всех модулей: блока питания, микроконтроллера, исполнительных устройств, датчиков, радиомодуля и пользовательского интерфейса.

Разработка курсового проекта будет происходить поэтапно. В первую очередь необходимо подобрать элементы устройства, учитывая их надежность, стоимость, функциональность и размеры. Затем необходимо собрать устройство и разработать программное обеспечение для корректной обработки информации и поддержания связи между элементами схемы. В конце устройство подлежит тестированию, чтобы проверить правильность сборки и исключить сбои при эксплуатации.

# 1 ОБЗОР ЛИТЕРАТУРЫ

## 1.1 Микроконтроллерные платформы

Существует огромное разнообразие плат с разными микроконтроллерами. Все они отличаются размерами, параметрами, предустановленными интерфейсами и выполняемыми задачами.

Наибольшую популярность в образовательных и любительских проектах получили микроконтроллеры семейства Arduino. Их преимущества: открытая архитектура, наличие большого количества библиотек и примеров, а также низкая стоимость.

Для исполнительного устройства в данном проекте используется Arduino Uno, обладающий достаточной производительностью и количеством выводов для подключения моторов, датчиков и радиомодуля. В пульте управления лучше применять Arduino Nano – компактная версия, удобная для интеграции в портативные устройства, но в данном проекте будет использована Arduino Uno. Микроконтроллеры:

1 Arduino Uno. 16 МГц, 32 КБ Flash, 2 КБ RAM. Преимущества: богатая периферия, простота программирования.

2 Arduino Nano. Аналогичные характеристики в компактном корпусе.

3 STM32. Более производительные аналоги, требующие глубоких знаний.

4 ESP32. Интеграция Wi-Fi и Bluetooth, но избыточность для данной задачи.

Таблица 1.1 – Сравнение микроконтроллеров

Параметр	Arduino Uno	STM32F103	ESP32
Тактовая частота	16 МГц	72 МГц	240 МГц
Память Flash	32 КБ	64 КБ	4 МБ
Сложность разработки	Низкая	Средняя	Средняя

Для получения более подробной информации о рассмотренных микроконтроллерах использовались источники [1, 2, 3].

В ходе практической реализации проекта выявилось ключевое ограничение платформы Arduino Uno — количество доступных цифровых и аналоговых выводов. Для подключения всего запланированного набора периферии (драйвер двигателей L298N, радиомодуль NRF24L01, датчик расстояния HC-SR04, датчик освещенности, светодиодная индикация) ресурсов стандартного контроллера оказалось недостаточно. Это исключило возможность использования еще более компактной Arduino Nano,

обладающей аналогичным количеством пинов. Для решения проблемы была применена схема расширения ввода-вывода с использованием регистра сдвига 74HC595. Данный подход позволил управлять несколькими выходными устройствами (например, светодиодами индикации) через всего 3 линии контроллера (данные, тактовая частота, защелка), эффективно увеличив количество доступных цифровых выходов.

## 1.2 Радиомодули для беспроводной связи

Для дистанционного управления чаще всего применяются следующие решения:

1 Инфракрасные модули (ограниченная дальность и работа только в прямой видимости).

2 Bluetooth-модули (HC-05, HC-06) – просты в использовании, но имеют ограниченную дальность (до 10–15 м).

3 Wi-Fi модули (ESP8266, ESP32) – обеспечивают большую скорость передачи данных, но требуют сетевой инфраструктуры и сложнее в настройке.

4 Модули NRF24L01 – обеспечивают работу на частоте 2,4 ГГц, поддерживают разные скорости передачи данных (250 Кбит/с, 1 Мбит/с, 2 Мбит/с), обладают низким энергопотреблением и хорошей дальностью связи.

5 В проекте выбран NRF24L01, так как он оптимален по сочетанию дальности, простоты и стоимости. Для робота рекомендуется версия NRF24L01 + PA + LNA с внешней антенной, что увеличивает радиус действия и устойчивость соединения.

Важным аспектом выбора NRF24L01 стала его способность работать в помехозащищённом окружении благодаря использованию частотного скачка (FHSS). Для обеспечения надёжного канала управления на платформе применён модуль с внешней активной антенной и усилителями (PA + LNA), что компенсирует возможные потери сигнала и увеличивает эффективную дальность до 70-100 метров на открытом пространстве. На стороне пульта достаточно модуля без усилителя, что оптимизирует энергопотребление.

## 1.3 Системы привода и драйверы моторов

Передвежные устройства чаще всего используют колёсное шасси с электродвигателями постоянного тока (DC motors). Для управления такими двигателями применяются драйверы на основе Н-моста. Наиболее распространённые:

1 L298N. Недорогой и простой модуль, но с низким КПД и нагревом при больших токах.

2 TB6612. Более современный и эффективный драйвер.

3 VNH2SP30. Рассчитан на большие токи, подходит для мощных двигателей.

В проекте применяются два драйвера L298N, каждый из которых управляет парой двигателей. Один драйвер L298N содержит два независимых

Н-моста, что позволяет управлять парой моторов. При подключении четырёх моторов к одному драйверу пришлось бы объединять их попарно на каждом канале, что привело бы к значительным ограничениям:

1 Потеря независимого управления левой и правой сторонами. Это сделало бы невозможным выполнение базового манёвра – поворота на месте за счёт вращения колёс левого и правого борта в разные стороны.

2 Снижение отказоустойчивости. Выход из строя одного канала драйвера парализовал бы сразу два мотора.

Реализация полноценного дифференциального (танкового) привода:

- драйвер №1 управляет двумя моторами левого борта;
- драйвер №2 управляет двумя моторами правого борта.

Такая схема позволяет независимо задавать скорость и направление вращения для левой и правой сторон платформы. Это является основой для всех типов манёвров:

1 Движение вперёд/назад. Оба драйвера получают идентичные команды.

2 Поворот на месте (разворот). Драйверы получают команды на вращение в противоположных направлениях.

3 Плавный поворот в движении. Одному драйверу задаётся большая скорость, чем другому.

4 Повышенная мощность и надёжность. Нагрузка (ток) распределяется между двумя микросхемами, что снижает тепловыделение на каждом драйвере и позволяет им работать в более щадящем режиме. Выход из строя одного драйвера выведет из строя только один борт, оставив платформе возможность совершить аварийный разворот или остановку, управляя оставшимися исправными моторами.

## 1.4 Датчики

Бортовые датчики являются «органами чувств» передвижной платформы, позволяя ей взаимодействовать с окружающей средой. Для навигации и безопасности в подобных проектах используются:

1 Ультразвуковые дальномеры (например, HC-SR04). Принцип работы основан на измерении времени между отправкой ультразвукового импульса и приёмом его эха. Отличаются простотой интерфейса (Trigger-Echo), достаточной для проектов точность (до 3 мм), измерением расстояния до объектов любой фактуры и цвета. Из минусов обладают узким углом конуса измерения ( $\sim 15^\circ$ ), возможными ошибками при измерении до мягких или мелких объектов, зависимостью скорости звука от температуры воздуха. В проекте HC-SR04 используется как основной датчик безопасности для предотвращения столкновений.

2 Инфракрасные датчики (ИК-датчики). Разделяются на датчики линии/расстояния (например, TCRT5000). Сочетают в себе ИК-светодиод и фотодиод. Измеряют интенсивность отражённого сигнала. Используются для следования по контрастной линии (чёрная линия на белом фоне или наоборот) или как простой датчик приближения.

3 ИК-приёмники (например, VS1838). Принимают кодированные сигналы от стандартных пультов ДУ, что может быть использовано для создания альтернативного канала управления.

4 Инкрементальные энкодеры. Устанавливаются на вал мотора или колеса и генерируют импульсы при вращении. Позволяют с высокой точностью измерять пройденное расстояние, скорость вращения и направление движения, что является основой для одометрии и точного позиционирования. В данной версии проекта не используются, но зарезервированы как ключевое направление для модернизации.

5 Инерциальные измерительные модули (IMU), такие как MPU-6050 (гироскоп + акселерометр). Позволяют определять ориентацию платформы в пространстве, угловую скорость и линейные ускорения. Критически важны для реализации алгоритмов стабилизации, балансировки или сложной навигации с учётом поворотов.

В текущем проекте реализован базовый сенсорный набор: УЗ-дальномер для безопасности и ИК-датчики линии (опционально) для реализации классического алгоритма движения по трассе. Такая комбинация покрывает основные задачи обнаружения препятствий и следования по маршруту.

## **1.5 Индикация и пользовательский интерфейс**

Эффективное взаимодействие между пользователем и устройством обеспечивается интерфейсом на стороне пульта и системой обратной связи на стороне платформы.

На стороне пульта управления:

1 Джойстик аналоговый (2 оси с кнопкой). Является оптимальным органом управления для телекомандных платформ. Преимущества перед кнопками – плавное и интуитивное управление скоростью и направлением.

2 Светодиодная индикация. При смене положения джойстика светодиод начнет загораться.

На стороне передвижной платформы есть светодиодная индикация. Решает две основные задачи:

- функциональное освещение;
- сигнальная индикация.

Отказ от сложных дисплеев (LCD, OLED) на данном этапе является осознанным решением, направленным на снижение сложности и энергопотребления системы. Вся необходимая информация может быть закодирована в состояниях светодиодов. Возможно добавление в будущей модернизации.

## **1.6 Датчики освещенности**

Наиболее распространёнными моделями датчиков освещенности являются фоторезистивный датчик освещенности на чипе LM393, датчик освещенности CJMCU-ТЕМТ6000 и модуль освещения GY-302 на чипе

ВН1750FVI. В таблице 1.2 приведены их сравнительные характеристики.

Таблица 1.2 – Сравнение датчиков освещенности

Параметры сравнения	Фоторезистивный датчик освещенности	Датчик освещенности СJMCU-ТЕМТ6000	Модуль освещения GY-302
Принцип действия	Изменение сопротивления фоторезистора	Фотодиод в фото – гальваническом режиме	Спец. ИС-датчик
Тип выходного сигнала	Аналоговый (A0) и Цифровой (D0)	Аналоговый (A0)	Цифровой (I2C)
Точность и линейность	Низкая, нелинейная характеристика	Средняя, хорошая линейность	Высокая, калибранный цифровой выход в люксах
Скорость отклика	Низкая (десятки миллисекунд)	Высокая	Высокая
Энергопотребление	Среднее (~15 мА)	Низкое (~0.2 мА)	Очень низкое (~120 мкА)
Зависимость от спектра света	Высокая (близка к глазу)	Высокая (близка к глазу)	Скорректирована под чувствительность глаза
Ключевое преимущество	Два типа выхода, простота	Быстрый отклик, линейность	Высокая точность, цифровой интерфейс
Ключевой недостаток	Низкая точность, инерционность	Требует доп. компонентов	Высокая стоимость

Для получения точной информации о данных датчиках использовалась техническая спецификация [4, 5] и источник [6].

Для поставленной задачи – автоматического включения фар при снижении уровня окружающей освещенности – был выбран фоторезистивный датчик на компараторе LM393. Данный выбор является оптимальным компромиссом и обусловлен следующими факторами:

1 Адекватность точности. Функция носит пороговый, а не измерительный характер («темно/светло»). Высокая точность измерения в люксах, которую предоставляет ВН1750, является избыточной и не оправдывает увеличения стоимости.

2 Гибкость и надежность интерфейса. Наличие как аналогового, так и цифрового выходов предоставляет уникальную гибкость.

## **2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

### **2.1 Общая структура системы**

Общая структура спроектированной системы построена по классической схеме «управляющее устройство (пульт) – исполнительное устройство (передвижная платформа)» с двунаправленным беспроводным каналом связи. Данная архитектура обеспечивает чёткое разделение функций, модульность и лёгкость масштабирования.

Основные компоненты системы:

1 Пульт дистанционного управления. Его ключевая функция – преобразование физических воздействий пользователя (перемещение джойстика, нажатие кнопок) в цифровые управляющие команды и их гарантированная передача на исполнительное устройство по радиоканалу.

2 Передвижная платформа. Является мобильным исполнительным механизмом. Она отвечает за приём и декодирование команд с пульта, обработка данных с бортовых датчиков (освещённость, расстояние до препятствий), формирование управляющих воздействий на двигатели на основе комбинации внешних команд и данных телеметрии (приоритет безопасности), управление системой световой индикации для обратной связи и автономных функций.

Ключевые принципы, заложенные в структуру:

1 Модульность. Каждый функциональный блок (питание, управление, связь, привод, сенсорика) может быть модернизирован или заменён независимо.

2 Централизованное управление. На каждом устройстве микроконтроллер Arduino Uno выступает в роли центрального процессора, координирующего работу всех подключённых модулей по чёткому алгоритму.

3 Приоритет безопасности. В алгоритме работы платформы заложен приоритет сигналов от датчиков безопасности (например, команда «вперёд» будет проигнорирована при обнаружении препятствия ультразвуковым датчиком).

### **2.2 Структура пульта управления**

Пульт управления спроектирован как компактное, эргономичное и автономное устройство, основной задачей которого является надежное преобразование намерений оператора в цифровые команды и их передача на подвижную платформу. Его структура оптимизирована для решения этой задачи с акцентом на надежность радиоканала и удобство использования.

Ядро системы и интерфейс оператора:

1 Микроконтроллер Arduino Uno. Выполняет функции центрального процессора. Обрабатывает аналоговые сигналы с джойстика, опрашивает

состояние кнопок, формирует структурированные управляющие пакеты данных и управляет периферией.

2 Аналоговый джойстик (2 оси). Основной орган управления. Потенциометры джойстика выдают напряжение, пропорциональное углу отклонения рукоятки. Программный алгоритм преобразует эти значения в команды направления (вперёд/назад, влево/вправо).

3 Модуль связи и обеспечение надежности – радиомодуль NRF24L01 + + PA + LNA.

4 Система индикации и питания.

5 Источник питания.

## 2.3 Структура передвижной платформы

Передвижная платформа включает в себя:

- 1 Arduino Uno – основной контроллер.
- 2 NRF24L01 +PA+LNA – модуль радиосвязи.
- 3 Драйверы моторов L298N – обеспечивают управление четырьмя колесными двигателями.
- 4 DC моторы с редуктором – исполнительные устройства для движения.
- 5 Датчики линии – используются для навигации.
- 6 Светодиодная индикация – отображение состояния питания и связи.
- 7 Источник питания (аккумулятор 7,4 В 2×18650) – питание машины.

## 2.4 Взаимодействие модулей

Взаимодействие модулей строится по иерархическому принципу с центральным управляющим алгоритмом на каждом микроконтроллере. На пульте:

1 Опрос ввода. Циклическое чтение аналоговых значений с джойстика и цифровых состояний кнопок.

2 Обработка. Нормирование значений, формирование компактной командной структуры (байт или два байта).

3 Передача. Отправка пакета данных через радиомодуль в эфир с минимальной необходимой периодичностью (20-50 мс).

На передвижной платформе:

1 Приём и дешифрация. Получение пакета по радиоканалу, проверка его целостности, извлечение команд.

2 Обработка датчиков. Параллельный опрос УЗ-дальномера и датчика освещённости. Данные с датчиков имеют приоритет над командами с пульта (например, команда «вперёд» будет проигнорирована при обнаружении препятствия).

3 Формирование выходных сигналов. На основе актуальной команды и состояния датчиков микроконтроллер генерирует управляющие последовательности для драйвера моторов.

4 Пульт передаёт управляющие команды по радиоканалу.

5 Исполнительное устройство принимает команды, обрабатывает их и изменяет направление движения.

6 Датчики передают информацию в микроконтроллер, который может использовать её для коррекции движения.

7 Светодиоды обеспечивают визуальную обратную связь пользователю.

## 2.5 Преимущества выбранной структуры

Выбранная модульная и централизованная структура системы предоставляет ряд значимых преимуществ как на этапе разработки и сборки, так и в процессе эксплуатации и возможной модернизации:

1 Высокая ремонтопригодность и простота отладки. В случае выхода из строя любого компонента его можно заменить, не затрагивая остальную часть системы. Это также упрощает пошаговую отладку: сначала проверяется работа микроконтроллера и индикации, затем добавляется управление моторами, после чего подключаются датчики и радиосвязь.

2 Масштабируемость и гибкость. Архитектура допускает легкое расширение функционала, добавление новых датчиков. Можно использовать оставшиеся свободные аналоговые входы или увеличить количество цифровых.

3 Переход на более мощные двигатели или драйверы потребует замены только соответствующего силового модуля без переработки всей схемы.

## 2.6. Перспективы развития структуры

Предложенная структура системы представляет собой прочный фундамент, который допускает целенаправленное развитие в нескольких перспективных направлениях, трансформируя учебный прототип в более сложное и функциональное устройство:

1 Система технического зрения. Замена или дополнение ультразвукового датчика компактной камерой (например, ESP32-CAM или подключенной по UART камерой на чипе OV7670) открывает путь для реализации алгоритмов распознавания образов, следования за объектом или движения по сложной траектории.

2 Картирование и навигация. Комбинация данных с энкодеров на моторах, IMU и дальномера (лидара) позволит строить простые карты помещений и реализовывать алгоритмы автономного перемещения между заданными точками ( $A^*$  pathfinding).

3 Двухсторонняя телеметрия в реальном времени. Модернизация программного протокола для регулярной передачи с платформы на пульт данных о состоянии (напряжение батареи, ток мотора, показания всех датчиков) с отображением на TFT/OLED-дисплее пульта.

4 Резервирование каналов. Установка второго радиомодуля (например, работающего на другой частоте) для повышения отказоустойчивости критически важных систем управления.

5 Модернизация исполнительной части. Замена корпуса машинки и пульта на более прочный материал.

6 Голосовое управление. Интеграция простого модуля распознавания голосовых команд (на базе готовых решений типа LD3320) для альтернативного способа управления.

7 Система самодиагностики. Реализация встроенного теста ключевых узлов при старте, непрерывный мониторинг ключевых параметров (температура драйверов, уровень сигнала) и ведение журнала ошибок.

Первые две задачи для более мощных процессоров и сложного программного стека, где Arduino Uno может выступать лишь как подчиненное устройство. Придется использовать более мощную плату.

## **3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

### **3.1 Обоснование выбора микроконтроллеров**

В качестве управляющего ядра для обеих частей системы – пульта дистанционного управления и передвижной платформы – выбрана плата Arduino Uno. Несмотря на то, что для пульта управления логичным выбором казалась более компактная Arduino Nano, окончательное решение в пользу Uno. В ходе детального проектирования схемы мобильного робота выяснилось, что даже количество выводов Arduino Uno является недостаточным для прямого подключения всего комплекса исполнительных устройств и датчиков. Использование Nano, имеющей меньшее количество физических выводов, было бы изначально невозможно. Таким образом, выбор Uno стал компромиссом между достаточной (хотя и требующей расширения) пиновой базой, доступностью, наличием опыта разработки и необходимостью обеспечить полную функциональность устройства.

### **3.2 Обоснование выбора датчика освещенности**

Для реализации функции автоматического включения фар выбран фоторезистивный датчик освещенности на основе компаратора LM393. Основными критериями выбора стали:

1 Двойной интерфейс выхода. Наличие как аналогового, так и цифрового выходов предоставляет максимальную гибкость при программировании. Аналоговый выход (A0) позволяет измерять градации освещенности для реализации сложной логики (например, плавного изменения яркости фар). Цифровой выход (D0) с регулируемым порогом срабатывания (через потенциометр на модуле) может быть использован как простой и надежный триггер состояния «светло/темно» без нагрузки на АЦП микроконтроллера.

2 Низкое энергопотребление. Ток потребления модуля в активном режиме не превышает 15 мА, что является одним из минимальных показателей в своем классе. Это важно для автономной работы платформы от аккумулятора.

3 Простота интеграции и обработки сигнала. Модуль не требует сложных обвесов или дополнительной калибровки.

4 Стоимость и доступность. Модуль является одним из наиболее доступных решений на рынке для образовательных и любительских проектов.

### **3.3 Обоснование выбора ультразвукового датчика расстояния**

Для обеспечения функции обнаружения препятствий и предотвращения столкновений выбран ультразвуковой дальномер HC-SR04. Его выбор обоснован следующими преимуществами:

1 Оптимальный диапазон измерений. Датчик эффективно работает на расстояниях от 2 до 400 см, что полностью покрывает необходимую дистанцию для безопасного маневрирования платформы в помещении. Точность измерения ( $\pm 3$  мм) более чем достаточна для данной задачи.

2 Удобный цифровой интерфейс. Работа по принципу «запрос-ответ» (trigger-echo) упрощает взаимодействие с микроконтроллером. Для получения данных достаточно подать короткий импульс на вывод Trig и измерить длительность ответного импульса на выводе Echo. Это не требует использования специализированных протоколов связи (I2C, UART) и экономит соответствующие линии контроллера.

3 Широкий угол обзора. Угол конуса измерения составляет примерно 15 градусов, что обеспечивает надежное обнаружение препятствий по курсу движения без излишней чувствительности к объектам сбоку.

4 Стабильность работы и защита от помех: Датчик имеет неплохую устойчивость к акустическим помехам.

### 3.4 Обоснование выбора модулей радиопередачи

Для организации канала дистанционного управления выбраны модули NRF24L01 в двух модификациях: базовая версия на машинке и версия NRF24L01 + PA + LNA на пульте. Данная асимметричная конфигурация обеспечивает оптимальное соотношение дальности, помехоустойчивости и стоимости.

Обоснование выбора технологии NRF24L01:

1 Оптимальный баланс характеристик. По сравнению с альтернативами (Bluetooth, WiFi), модули NRF24L01 обеспечивают большую дальность связи (до 100 м в открытом пространстве с усилителем), меньшее энергопотребление и не требуют сложных процедур сопряжения или подключения к сети. Протокол с частотным скачком (FHSS) повышает устойчивость к помехам в переполненном диапазоне 2.4 ГГц.

2 Надёжность и предсказуемость связи. Архитектура с подтверждением доставки пакетов (ACK) и автоматической повторной отправкой гарантирует, что управляющие команды не будут потеряны. Возможность программной настройки скорости передачи данных позволяет выбрать оптимальный режим между скоростью (2 Мбит/с) и максимальной дальностью (250 Кбит/с).

На стороне пульта управления (NRF24L01 + PA + LNA):

1 Критическая важность доставки команды. Основная задача – гарантированно «доставить» управляющий пакет до платформы. Усилитель мощности (PA) увеличивает энергию излучаемого сигнала, что является самым эффективным способом повышения надёжности канала «вниз» (от пульта к платформе).

2 Компенсация помех. Усилитель низкого шума (LNA) на приёмной стороне пульта позволяет ему уверенно «слышать» даже слабый сигнал подтверждения (ACK) от базового модуля платформы, завершая цикл обмена.

3 Внешняя антенна обеспечивает лучшую эффективность излучения.

4 Приемлемое энергопотребление. Пульт питается от более ёмкого источника (power bank или 2x18650), что позволяет ему нести нагрузку от модуля с усилителем.

На стороне передвижной платформы (базовый NRF24L01):

1 Энергосбережение. Базовая версия модуля потребляет значительно меньше тока, особенно в режиме ожидания. Это критически важно для автономной платформы, чей аккумулятор также питает энергоёмкие двигатели.

2 Упрощение схемы. Отсутствие необходимости в усилителе снижает требования к стабильности и мощности источника 3.3В на плате платформы.

3 Достаточная чувствительность. Для приёма мощного сигнала от усиленного пульта чувствительности базового приёмника более чем достаточно.

### 3.5 Обоснование выбора драйвера моторов

Для управления четырьмя двигателями постоянного тока платформы выбран драйвер L298N. Несмотря на наличие более современных аналогов, его выбор был сделан по ряду практических и педагогических причин:

1 Высокая нагрузочная способность и надежность. Каждый канал драйвера L298N способен выдерживать постоянный ток до 2 А (пиковый до 3 А) и напряжение до 46 В, что с большим запасом перекрывает требования мотор-редукторов (0.6-1 А, 7.4 В).

2 Полная управляемость и простота интерфейса. Драйвер обеспечивает независимое управление направлением вращения двух моторов с помощью 4-х цифровых сигналов (IN1, IN2, IN3, IN4) и регулировку скорости с помощью двух ШИМ-сигналов (ENA, ENB). Такой прямой и понятный интерфейс идеально подходит для обучения основам управления двигателями.

3 Функция стабилизатора напряжения. Модуль содержит встроенный стабилизатор 5V/0.5A, который может быть использован для питания логической части схемы (Arduino, датчики), что упрощает построение системы электропитания и позволяет запитать контроллер от той же батареи, что и двигатели.

4 Конструкция модуля позволяет легко модернизировать систему. Например, заменив один L298N на два более эффективных драйвера TB6612FNG, без изменения общей архитектуры управления.

### 3.6 Обоснование выбора мотор-редукторов

В качестве исполнительных приводов для передвижной платформы выбраны прямые двухосевые мотор-редукторы с передаточным числом 1:48. Данный выбор сделан на основе комплексного анализа требований проекта, обеспечивая оптимальный баланс между ключевыми характеристиками:

1 Соответствие источнику питания. Мотор-редукторы имеют рабочий диапазон напряжения 3–8 В, что идеально согласуется с выходным

напряжением выбранной аккумуляторной батареи (номинальное напряжение 7.4 В). Это позволяет использовать энергию источника питания с максимальной эффективностью без необходимости применения дополнительных понижающих или повышающих преобразователей, которые усложнили бы схему и увеличили бы потери.

2 Баланс скорости и тягового усилия. Передаточное число 1:48 является оптимальным компромиссом для мобильных платформ данного класса. Оно обеспечивает достаточный крутящий момент на валу колеса для уверенного старта, преодоления мелких неровностей и движения по поверхностям с умеренным сопротивлением (например, ковровое покрытие).

3 Совместимость с системой управления. Выходные параметры выбранных моторов (рабочий ток в нагрузке до 0.6–1.0 А на каждый) полностью укладываются в возможности драйвера L298N. Это гарантирует стабильную работу силового тракта без перегрузок и перегрева.

### **3.7 Система освещения**

Внедрение системы освещения в конструкцию передвижной платформы обусловлено имитацией реальных транспортных средств и визуальной индикации состояния и режимов работы. Работа системы построена на трёх ключевых режимах:

1 Режим инициализации при старте. Происходит последовательность мигания светодиодами. После завершения этой последовательности (длительностью 1-2 секунды) фары переходят в режим, определяемый логикой основного алгоритма.

2 Автоматический режим (основной). После инициализации система переходит в автоматический режим работы, при котором состояние передних фар (зеленые светодиоды) напрямую зависит от уровня окружающей освещённости, измеряемого датчиком LM393. Если значение с аналогового выхода датчика опускается ниже программно заданного порога, система интерпретирует это как «темноту» и включает фары. При увеличении уровня освещённости выше порога фары автоматически выключаются.

3 Режим при обнаружении препятствия перед устройством. При срабатывании ультразвукового дальномера HC-SR04 на критически близком расстоянии светодиоды переключаются в режим попеременного мигания, формируя хорошо заметный предупредительный сигнал.

## **4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА**

Разработка принципиальной электрической схемы является одним из ключевых этапов проектирования любого электронного устройства. Эта схема служит основным руководством для монтажа, прототипирования и последующего анализа работы системы. Она должна однозначно отображать все электрические соединения между компонентами, обеспечивая корректное функционирование каждого модуля и устройства в целом. В данном разделе проводится не только описание подключений, но и важные расчеты, подтверждающие правильность выбора элементов и источников питания. Таким образом, тщательная проработка принципиальной схемы напрямую влияет на надежность, производительность и стоимость конечного продукта.

### **4.1 Расчёт мощности элементов схемы**

Расчёт мощности является критически важным этапом для корректного выбора источника питания, оценки тепловыделения и обеспечения стабильности работы системы. Мощность рассчитывается как произведение напряжения питания на потребляемый ток для каждого активного элемента. В таблицах 4.1 и 4.2 приведён расчёт для пульта управления и передвижной платформы соответственно. При расчётах учитывался номинальный (средний) ток потребления компонентов. Пиковые токи (например, в момент старта моторов или передачи радиомодуля) могут превышать указанные значения в 1.5-2 раза. В связи с этим введён поправочный коэффициент запаса 20%, который учитывает разброс параметров элементов и неидеальность источников питания.

Таблица 4.1 – Расчет мощности элементов схемы устройства управления

Блок	U, В	I, мА	Кол-во	P, мВт
Микроконтроллер Arduino UNO R3	5	22	1	110
Модуль радиопередачи NRF24LO1 + PA + LNA	5	120	1	600
Джойстик	5	10	1	50
Светодиод	5	20	1	100
Суммарная мощность, мВт				810

В реализованной схеме используются микроконтроллер Arduino UNO R3, модуль радиопередачи NRF24LO1 + PA + LNA, джойстик и светодиод.

Таким образом потребляемая мощность будет равна:

$$P = 5 \cdot 22 + 5 \cdot 120 + 5 \cdot 10 + 5 \cdot 20 = 810 \text{ мВт.}$$

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 972 мВт.

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{0.972}{5} = 0.1944 \approx 0.19 \text{ А.}$$

Таблица 4.2 – Расчет мощности элементов схемы устройства

Блок	U, В	I, мА	Кол-во	P, мВт
Микроконтроллер Arduino UNO R3	5	22	1	110
Модуль радиоприема NRF24LO1	5	12.5	1	62.5
Датчик освещенности LM393	5	15	1	75
Датчик расстояния HC-SR04	5	15	1	75
Драйвер моторов L298N	5	36	2	360
Мотор-редуктор 1:48 3-8В	5	600	4	12 000
Светодиод	5	20	4	400
Суммарная мощность, мВт				13082

Таким образом потребляемая мощность будет равна:

$$P = 5 \cdot 22 + 5 \cdot 6 + 5 \cdot 15 + 5 \cdot 15 + 5 \cdot 36 \cdot 2 + 5 \cdot 600 \cdot 4 + 400 = 13082 \text{ Вт.}$$

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 15698 мВт.

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{15.698}{5} = 3.14 \text{ А.}$$

Данный теоретический расчёт мощности был верифицирован на практике путём измерения токов в различных режимах работы с помощью мультиметра. Реальные значения подтвердили правильность расчётов и адекватность выбранных компонентов.

## 4.2 Расчёт нагрузки светодиодов

Каждый светодиод подключен по схеме:

– +5В;

- анод светодиода;
- катод светодиода;
- резистор;
- цифровой вывод Arduino.

При установке вывода в состояние логического «0» он замыкает цепь на землю, и через светодиод протекает расчётный ток  $\sim 20$  мА. В данном курсовом проекте используется пять светодиодов различных цветов, подключенные к соответствующим пинам микроконтроллера (см. рисунок 4.1).

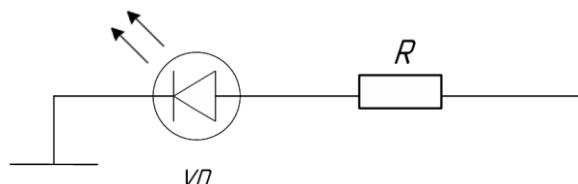


Рисунок 4.1 – Схема подключения светодиодов

Для ограничения тока светодиода используется резистор номиналом, рассчитываемым по следующей формуле:

$$R = \frac{U_{\Pi} - U_{\text{д}}}{I_{\text{пр}}},$$

где  $U_{\Pi}$  – напряжение питания,  $U_{\text{д}}$  – напряжение, падающее на светодиоде,  $I_{\text{пр}}$  – прямой ток светодиода.

В проекте используются светодиоды красного, желтого и зеленого цветов, со следующими параметрами:  $I_{\text{пр}} = 20$  мА.  $U_{\text{д}} = 1$  В. Красный стоит на пульте управления, желтый и зеленый на машинке, причем зеленый является передними фарами.

Получаем:

$$R = \frac{5 - 1}{20 \cdot 10^{-3}} = 200 \text{ Ом.}$$

Следовательно, для того, чтобы светодиод не перегорел он должен быть подключён через резистор с сопротивлением не менее 200 Ом, однако если взять слишком большое сопротивление, то светодиод будет гореть очень слабо, поэтому в данном проекте светодиоды подключаются через резисторы сопротивлением 220 Ом.

### 4.3 Микроконтроллеры

Микроконтроллер соединен со всеми модулями схемы через аналоговые или цифровые входы и выходы. Подробнее можно ознакомиться в приложении Г.

#### **4.4 Датчик освещенности**

Датчик освещенности на базе модуля с компаратором LM393 подключен к аналоговому входу микроконтроллера для реализации функции автоматического включения фар. Его интеграция в схему требует учета особенностей выходного сигнала и обеспечения стабильного питания для точных измерений.

Схема подключения к Arduino Uno:

1 VCC. Подключен к стабилизированному +5В цепи питания логической части платформы. Рекомендуется установить блокировочный керамический конденсатор 100 нФ непосредственно у выводов питания модуля для фильтрации высокочастотных помех.

2 GND/ Подключен к общему проводу (земле) системы.

3 A0 (Аналоговый выход). Ключевой вывод для работы в проекте. Подключен к аналоговому входу микроконтроллера, например, A1. На этом выводе формируется напряжение, пропорциональное уровню освещенности: чем темнее, тем выше сопротивление фотодиода и, соответственно, выше напряжение на выходе A0.

4 D0 (Цифровой выход). В данной реализации может быть не задействован. При необходимости его можно подключить к любому цифровому входу Arduino для получения дискретного сигнала «светло/темно», порог срабатывания которого регулируется встроенным потенциометром на модуле.

#### **4.5 Ультразвуковой датчик расстояния**

Схема подключения к Arduino Uno:

1 VCC. Подключен к стабилизированному 5 В.

2 Trig (Запуск). Подключен к цифровому выходу Arduino. Для формирования чёткого запускающего импульса длиной 10 мкс.

3 Echo (Ответ). Подключен к цифровому входу Arduino. Напряжение на выводе Echo от датчика составляет 5 В, что соответствует логическим уровням Arduino и не требует согласования.

4 GND. Подключение к общей земле.

#### **4.6 Модуль радиопередачи**

В проекте использованы две различные версии модуля NRF24L01, что накладывает особенности на их интеграцию в схему:

Базовая версия (на пульте управления): Модуль подключается к Arduino Uno через интерфейс SPI (выводы MOSI, MISO, SCK) и два управляющих цифровых вывода (CE, CSN). Для борьбы с самовозбуждением и обеспечения стабильности работы, на шину питания модуля (VCC и GND) в непосредственной близости от разъёма установлен керамический конденсатор 0,1 мКФ, который установлен параллельно. Это подавляет высокочастотные

помехи, возникающие при переключениях передатчика.

Версия с усилителем NRF24L01 + PA + LNA. Помимо вышеуказанного, данный модуль имеет повышенное энергопотребление в моменты передачи (до 120 мА). Поэтому его питание (контакт VCC) заведено не от вывода 3.3V Arduino, а от отдельного стабилизатора напряжения (например, AMS1117-3.3) или напрямую от стабильного источника 3.3V на плате, способного отдавать ток не менее 200 мА. Питание модуля от нестабильного или слабого источника – наиболее частая причина сбоев в работе.

## 4.7 Драйвер моторов

Ключевым потребителем являются моторы. Их питание осуществляется напрямую от аккумулятора 7.4 В (VS драйверов). Аккумуляторная батарея платформы должна иметь ёмкость, достаточную для желаемого времени работы, и быть способной отдавать ток не менее 2-3 А (с учётом пусковых токов). Источник 5В (стабилизатор) должен обеспечивать ток не менее 0.5 А. Направление вращения задаётся комбинацией сигналов INx на драйвере. При расчёте времени работы аккумулятора необходимо учитывать, что ток потребления мотора резко возрастает в момент старта (пусковой ток) и при увеличении механической нагрузки (подъём, препятствие).

## 4.8 Мотор-редукторы

Все они связаны с микроконтроллером через драйвер двигателей L298N. Мотор-редукторы закреплены на нижней платформе шасси. Каждый мотор подключён к соответствующим выходам (OUT1-OUT4) драйвера L298N через парные провода. При поступлении питания на соответствующие входы моторов они начинают вращаться в заданную сторону.

## 4.9 Схема расширения ввода-вывода на регистре сдвига

Для управления многочисленными устройствами индикации (светодиодами статуса, фарами) в условиях ограниченного количества цифровых выводов Arduino Uno была применена схема на основе регистра сдвига 74HC595. Данная микросхема позволяет последовательно загружать данные (по одному биту за такт) по линии данных (DS), используя сигнал тактовой частоты (SHCP). После загрузжения полного байта (8 бит) подачей сигнала на вход защелки (STCP) данные параллельно появляются на ее 8 выходных линиях (Q0...Q7). Это позволило организовать управление до 8 нагрузками, используя всего 3 вывода микроконтроллера, которые были задействованы для реализации интерфейса SPI в программном режиме.

## **5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

### **5.1 Требования к разработке программного обеспечения**

Разработанное микропроцессорное устройство представляет собой автономную машинку, которая собирает информацию о трех параметрах: расстояние до ближайшего препятствия, уровень освещенности и команды управления с пульта.

Устройство работает следующим образом. При включении машинки происходит инициализация всех систем: настраиваются пины управления моторами, датчики расстояния и освещенности, а также последовательный порт для связи с пультом управления

В течение всего времени работы устройство непрерывно анализирует показания с датчиков. При получении информации о недостаточной освещенности автоматически включаются фары, что обеспечивает безопасность работы в темное время суток. Когда ультразвуковой датчик обнаруживает препятствие на расстоянии менее 15 см, система безопасности блокирует движение вперед и выполняет экстренную остановку, сопровождая это световой индикацией.

Управление машинкой осуществляется с пульта дистанционного управления через последовательный интерфейс. При изменении положения джойстика микроконтроллер пульта анализирует полученные данные и отправляет соответствующую команду на машинку.

### **5.2 Блок-схема алгоритма**

Блок-схема алгоритма представляет собой графическое представление логики работы системы управления автономной машинкой. Алгоритм разделен на две основные части: алгоритм работы самой машинки и алгоритм работы пульта дистанционного управления.

Алгоритм работы машинки начинается с этапа инициализации системы. На этом этапе выполняются начальные настройки: конфигурирование пинов управления моторами, настройка датчиков расстояния и освещенности, инициализация последовательного порта для связи с пультом управления. После успешной инициализации система переходит в основной рабочий цикл.

В основном цикле последовательно выполняются три ключевые операции. Первая операция - проверка расстояния до препятствий с помощью ультразвукового датчика. Если обнаруживается препятствие на расстоянии менее установленного порога, система безопасности активирует экстренную остановку моторов и включает световую индикацию. Вторая операция - контроль уровня освещенности. Датчик освещенности постоянно отслеживает условия окружающей среды, и при недостаточной освещенности автоматически включаются фары. Третья операция - обработка команд от пульта управления. Система проверяет наличие входящих команд и при их получении выполняет соответствующие действия: изменение направления

движения, остановку или управление светом.

Алгоритм работы пульта управления также начинается с этапа инициализации, в ходе которого настраиваются элементы управления (джойстик, кнопки), инициализируется дисплей и последовательный порт. Основной цикл пульта включает непрерывный опрос положения джойстика и состояния кнопок. При изменении положения джойстика система определяет направление движения и формирует соответствующую команду. Отдельно обрабатываются нажатия кнопок: кнопка экстренной остановки немедленно передает команду остановки, а кнопка управления светом переключает состояние фар.

Оба алгоритма работают в непрерывном цикле, обеспечивая надежное и безопасное управление машинкой с возможностью автоматической адаптации к изменяющимся условиям окружающей среды.

### **5.3 Исходный код программы для устройства управления**

Пульт дистанционного управления считывает показания с двухосевого джойстика и состояния кнопок, а затем преобразует данные для удобной отправки по последовательному интерфейсу. Исходный код программного обеспечения для данного устройства можно найти в приложении А.

Функция `void setup()` (строки 152-165) необходима для начальной настройки контроллера. Здесь задаются входные и выходные пины для элементов управления: джойстика (строки 153-154), кнопок (строки 155-156) и светодиода индикации передачи (строка 156).

Функция `void loop()` (строки 167-172) является главной циклической функцией и работает на протяжении всей работы микроконтроллера. Здесь реализована основная логика работы пульта: опрос джойстика, проверка состояния кнопок и обновление информации на дисплее.

Функция `void checkJoystick()` (строки 174-200) обрабатывает данные с джойстика. В строках 175-177 считаются текущие значения осей X и Y, а также состояние кнопки джойстика. В блоках 179-193 анализируются положения джойстика и формируются соответствующие команды управления движением. При обнаружении значительного изменения положения джойстика отправляется команда движения вперед (строка 181), назад (строка 183), влево (строка 185), вправо (строка 187) или остановки (строка 189).

Функция `void checkButtons()` (строки 202-219) обрабатывает нажатия отдельных кнопок. При нажатии кнопки экстренной остановки (строки 203-208) отправляется команда 'S' и выводится соответствующее сообщение на дисплей. Кнопка управления светом (строки 210-218) переключает состояние фар и отправляет команды '1' или '0'.

Функция `void sendCommand(char command)` (строки 221-226) отправляет одиночную команду через последовательный порт с визуальной индикацией передачи данных.

## 5.4 Исходный код программы для передвижного устройства

Данное устройство является основным в разрабатываемом курсовом проекте, здесь реализована большая часть логики автономного поведения и управления. Исходный код программного обеспечения можно найти в приложении А (строки 000-132).

Функция `void setup()` (строки 019-028) выполняет начальную настройку системы, подготавливая все компоненты к корректной работе. В строках 020-024 настраиваются пины управления моторами и светодиодами, определяя их режимы работы. Строки 025-026 обеспечивают начальное, безопасное состояние системы – остановку моторов и выключение света, предотвращая неожиданные действия при старте. В строке 027 инициализируется последовательный порт для связи с пультом управления, устанавливая канал для получения команд.

Функция `void loop()` (строки 030-35) является главным рабочим циклом системы, постоянно выполняющимся для обеспечения ее активности и отзывчивости. Здесь последовательно и циклически вызываются все основные функции: проверка препятствий, управление освещением и обработка команд с пульта управления, что создает непрерывный процесс управления.

Функция `void checkObstacle()` (строки 037-047) реализует логику взаимодействия с ультразвуковым датчиком расстояния, обеспечивая базовую безопасность устройства. В строке 038 получается текущее расстояние до ближайшего препятствия, что является исходными данными для анализа. В строках 039-046 выполняется анализ полученных данных - если расстояние меньше минимально допустимого и препятствие еще не было обнаружено, система выполняет экстренную остановку (строка 041) и устанавливает флаг обнаружения препятствия (строка 042), предотвращая столкновение.

Функция `void checkLightSensor()` (строки 049-058) управляет автоматическим включением фар, адаптируя работу устройства к условиям окружающей среды. В строке 050 считывается текущий уровень освещенности, оценивая необходимость в дополнительном свете. В строках 051-057 анализируется полученное значение - если освещенность ниже порогового значения и фары выключены, они включаются (строки 052-053), и наоборот, при достаточной освещенности фары выключаются (строки 054-056), реализуя энергоэффективность.

Функция `void checkControlCommands()` (строки 060-065) проверяет наличие входящих команд от пульта управления, обеспечивая связь с оператором. При обнаружении данных в последовательном порту функция считывает команду (строка 062) и передает ее на выполнение (строка 063), выступая в роли диспетчера.

Функция `void executeCommand(char cmd)` (строки 067-097) обрабатывает полученные команды управления, преобразуя символьные инструкции в

конкретные действия. В строках 068-069 выполняется проверка возможности движения вперед (отсутствие препятствий), добавляя уровень безопасности при ручном управлении. В строках 070-096 реализован выбор действий в зависимости от полученной команды: движение вперед (строки 071-075), назад (строки 076-078), влево (строки 079-081), вправо (строки 082-084), остановка (строки 085-087), принудительное включение (строки 088-091) и выключение (строки 092-095) фар, что покрывает весь спектр управления.

Функции управления моторами `void moveForward()` (строки 099-104), `void moveBackward()` (строки 106-111), `void turnLeft()` (строки 113-118), `void turnRight()` (строки 120-125) и `void stopMotors()` (строки 127-132) реализуют непосредственное управление силовой электроникой, формируя необходимые сигналы на пинах драйверов моторов для обеспечения требуемого направления движения, выступая в качестве низкоуровневых исполнительных модулей.

## **6 РАЗРАБОТКА КОРПУСА УСТРОЙСТВА**

Разработка корпусов является одним из ключевых этапов проектирования любого радиоэлектронного устройства, поскольку она в значительной степени определяет не только его эстетическое восприятие, но и такие критически важные параметры, как функциональность, надежность, ремонтопригодность и удобство сборки. В рамках данного проекта были спроектированы два корпусных изделия: корпус передвижного устройства (роботизированной платформы) и корпус пульта дистанционного управления. Основными задачами при их проектировании являлись:

- 1 Обеспечение необходимой жесткости и механической прочности конструкции.
- 2 Предоставление легкого и интуитивно понятного доступа к внутренним компонентам для проведения монтажа, отладки и последующего обслуживания.
- 3 Защита электронных компонентов от потенциальных механических повреждений, статического электричества и пыли.
- 4 Создание эргономичного и визуально привлекательного продукта.

### **6.1 Концепция и материалы**

Было принято решение изготовить корпус машинки методом сборки из переплетного картона толщиной 3 мм. Данный подход обладает рядом преимуществ:

- 1 Технологичность. Не самый трудный способ разрезания картона с помощью ножа для бумаги.
  - 2 Ремонтопригодность. В случае повреждения одного из элементов его легко заменить.
  - 3 Жесткость. Правильно спроектированная коробчатая конструкция обладает высокой прочностью.
  - 4 Внешний вид. Позволяет создать стильный, геометричный дизайн.
- Для крепления пластин клей и скотч. Это обеспечит разборность и прочность соединения.

### **6.2 Конструкция передвижного устройства**

Корпус машинки спроектирован как двухуровневая конструкция:

1 Нижний уровень. На этой пластине крепятся 4 мотор-редуктора с колесами, драйвер моторов L298N и аккумуляторная батарея. Данная пластина является силовой и должна иметь наибольшую толщину (4-5 мм).

2 Верхний уровень. На него устанавливаются все датчики и системы индикации. Плата Arduino UNO также может быть размещена здесь.

Оба уровня соединяются между собой с помощью стоек, образуя прочную конструкцию.

### **6.3 Конструкция пульта управления**

Пульт управления выполнен в виде компактной коробки, эргономично лежащей в руках. Его основу составляет прямоугольная пластина, на которой крепятся все компоненты. Arduino UNO и радиопередатчик крепятся внутри корпуса. Светодиод индикации питания выводится на лицевую панель. Верхняя крышка корпуса пульта является съемной для обеспечения доступа к кнопке питания и разъему для зарядки/питания.

## **7 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Настоящее руководство описывает порядок работы с радиоуправляемой машинкой, оснащенной датчиками.

### **7.1 Назначение устройства**

Радиоуправляемая машинка предназначена для движения в помещении и на открытом воздухе, а также для демонстрации работы различных датчиков: освещенности, расстояния. Управление осуществляется с помощью портативного пульта по радиоканалу.

### **7.2 Комплектация**

Комплектация состоит из:

- 1 Передвижное устройство (машинка) в сборе – 1 шт.
- 2 Пульт дистанционного управления – 1 шт.
- 3 Кабель micro-USB для программирования и питания пульта – 1 шт.
- 4 Кабель для зарядки/питания машинки (зависит от выбранного аккумулятора) – 1 шт.

### **7.3 Подготовка к работе**

Зарядка аккумуляторов. Убедитесь, что аккумуляторы машинки и пульта управления полностью заряжены.

Включение пульта. Подайте питание на пульт управления. О готовности к работе сигнализирует светодиод на его лицевой панели.

Включение машинки. Включите питание на машинке. После включения машинка готова к приему команд.

### **7.4 Порядок работы**

Управление движением:

1 Вперед/Назад. Наклоньте джойстик от себя для движения вперед. Наклоньте джойстик на себя для движения назад.

2 Влево/Вправо. Наклоньте джойстик влево для поворота налево. Наклоньте джойстик вправо для поворота направо.

Остановка. Отпустите джойстик в нейтральное положение, и машинка остановится.

Работа систем автоматики:

1 Автоматическое освещение ("Фары"). При въезде в темное помещение или в условиях недостаточной освещенности белые светодиоды на машинке включаются автоматически.

2 Предупреждение о препятствиях. При приближении к препятствию на расстояние менее 15 см машинка автоматически остановится, а боковые

желтые светодиоды начнут мигать. Для возобновления движения необходимо отдать команду назад или в сторону.

## **7.5 Меры предосторожности**

Не подвергайте устройство воздействию влаги и прямых струй воды.

Избегайте сильных ударов и падений устройства.

Не блокируйте колеса машинки во время движения во избежание перегрева двигателей и драйвера.

Используйте для питания только рекомендованные источники.

Не разбирайте устройство при включенном питании.

## **7.6 Уход и техническое обслуживание**

Регулярно очищайте корпус машинки и пульта от пыли сухой мягкой тканью.

Следите за чистотой колес для обеспечения лучшего сцепления с поверхностью.

Периодически проверяйте надежность всех механических соединений и креплений. Особенно важно, так как устройство обладает необычайной хрупкостью.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсового проекта было успешно разработано и реализовано функциональное микропроцессорное устройство на радиоуправлении. Все поставленные перед проектом задачи были выполнены в полном объеме.

К основным достоинствам разработанного устройства можно отнести его относительно низкую себестоимость, а также простоту сборки и реализации. В то же время, к его недостаткам относится необходимость самостоятельной разработки программного обеспечения для управления подключенными компонентами и обработки поступающей с них информации.

Перспективы развития проекта связаны с дальнейшим совершенствованием конструкции и программной части. В их числе – внедрение голосового управления, создание графического интерфейса пользователя. Эти улучшения можно реализовывать поэтапно, превращая учебную машинку в сложный и многофункциональный проект.

Практическая значимость работы заключается в создании универсальной мобильной платформы, которая может быть адаптирована под различные прикладные задачи за счет замены или добавления датчиков и исполнительных механизмов

Проведенная работа позволила закрепить на практике теоретические знания по схемотехнике, программированию микроконтроллеров и проектированию электронных устройств, а также получить ценный опыт комплексного подхода к решению инженерной задачи.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Arduino UNO [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardUno> – Дата доступа: 11.09.2021
- [2] Raspberry PI 2 Model B — второе поколение Raspberry Pi [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://micro-pi.ru/raspberry-pi-2-model-b-rpi-bcm2836-bcm2837/> – Дата доступа: 12.09.2021
- [3] OLIMEXINO-STN32 development board [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://data.electronshik.ru/pdf/pdf/o/olimexino-stm32.pdf> – Дата доступа: 12.09.2021
- [4] Ultrasonic Sensor. HC-SR04 Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://datasheetspdf.com/pdf/1380136/ETC/HC-SR04/1> – Дата доступа: 28.09.2021
- [5] Detection Sensor. DYP-ME007 Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://datasheetspdf.com/pdf/1251016/ETC/DYP-ME007/1> – Дата доступа: 28.09.2021
- [6] Ультразвуковой датчик расстояния, температуры и освещенности [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://robot-kit.ru/3086/> – Дата доступа: 28.09.2021

## ПРИЛОЖЕНИЕ А

*(обязательное)*

### Код программы

```
000 #include <NewPing.h>
001
002 #define FRONT_IN1 4
003 #define FRONT_IN2 5
004 #define FRONT_IN3 6
005 #define FRONT_IN4 7
006 #define HEADLIGHTS 13
007 #define TRIG_PIN 2
008 #define ECHO_PIN 3
009 #define LIGHT_SENSOR A0
010
011 NewPing sonar(TRIG_PIN, ECHO_PIN, 200);
012
013 const int MIN_DISTANCE = 15;
014 const int LIGHT_THRESHOLD = 300;
015
016 bool lightsOn = false;
017 bool obstacleDetected = false;
018
019 void setup() {
020     pinMode(FRONT_IN1, OUTPUT);
021     pinMode(FRONT_IN2, OUTPUT);
022     pinMode(FRONT_IN3, OUTPUT);
023     pinMode(FRONT_IN4, OUTPUT);
024     pinMode(HEADLIGHTS, OUTPUT);
025     stopMotors();
026     digitalWrite(HEADLIGHTS, LOW);
027     Serial.begin(9600);
028 }
029
030 void loop() {
031     checkObstacle();
032     checkLightSensor();
033     checkControlCommands();
034     delay(100);
035 }
036
037 void checkObstacle() {
038     int distance = sonar.ping_cm();
039     if (distance > 0 && distance < MIN_DISTANCE) {
040         if (!obstacleDetected) {
041             stopMotors();
042             obstacleDetected = true;
043         }
044     } else {
045         obstacleDetected = false;
046     }
}
```

```

047 }
048
049 void checkLightSensor() {
050     int lightLevel = analogRead(LIGHT_SENSOR);
051     if (lightLevel < LIGHT_THRESHOLD && !lightsOn) {
052         digitalWrite(HEADLIGHTS, HIGH);
053         lightsOn = true;
054     } else if (lightLevel >= LIGHT_THRESHOLD && lightsOn) {
055         digitalWrite(HEADLIGHTS, LOW);
056         lightsOn = false;
057     }
058 }
059
060 void checkControlCommands() {
061     if (Serial.available()) {
062         char command = Serial.read();
063         executeCommand(command);
064     }
065 }
066
067 void executeCommand(char cmd) {
068     int distance = sonar.ping_cm();
069     bool canMoveForward = !(distance > 0 && distance <
MIN_DISTANCE);
070     switch(cmd) {
071         case 'F':
072             if (canMoveForward) {
073                 moveForward();
074             }
075             break;
076         case 'B':
077             moveBackward();
078             break;
079         case 'L':
080             turnLeft();
081             break;
082         case 'R':
083             turnRight();
084             break;
085         case 'S':
086             stopMotors();
087             break;
088         case '1':
089             digitalWrite(HEADLIGHTS, HIGH);
090             lightsOn = true;
091             break;
092         case '0':
093             digitalWrite(HEADLIGHTS, LOW);
094             lightsOn = false;
095             break;
096     }
097 }
098

```

```

099 void moveForward() {
100   digitalWrite(FRONT_IN1, LOW);
101   digitalWrite(FRONT_IN2, HIGH);
102   digitalWrite(FRONT_IN3, LOW);
103   digitalWrite(FRONT_IN4, HIGH);
104 }
105
106 void moveBackward() {
107   digitalWrite(FRONT_IN1, HIGH);
108   digitalWrite(FRONT_IN2, LOW);
109   digitalWrite(FRONT_IN3, HIGH);
110   digitalWrite(FRONT_IN4, LOW);
111 }
112
113 void turnLeft() {
114   digitalWrite(FRONT_IN1, HIGH);
115   digitalWrite(FRONT_IN2, LOW);
116   digitalWrite(FRONT_IN3, LOW);
117   digitalWrite(FRONT_IN4, HIGH);
118 }
119
120 void turnRight() {
121   digitalWrite(FRONT_IN1, LOW);
122   digitalWrite(FRONT_IN2, HIGH);
123   digitalWrite(FRONT_IN3, HIGH);
124   digitalWrite(FRONT_IN4, LOW);
125 }
126
127 void stopMotors() {
128   digitalWrite(FRONT_IN1, LOW);
129   digitalWrite(FRONT_IN2, LOW);
130   digitalWrite(FRONT_IN3, LOW);
131   digitalWrite(FRONT_IN4, LOW);
132 }
133 // ПУЛЬТ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ
134 // ДЛЯ АВТОНОМНОЙ МАШИНКИ НА ARDUINO
135
136 #INCLUDE <LIQUIDCRYSTAL.H>
137
138 #DEFINE JOY_X A0      // ДЖОЙСТИК ОСЬ X
139 #DEFINE JOY_Y A1      // ДЖОЙСТИК ОСЬ Y
140 #DEFINE JOY_BTN 2     // КНОПКА ДЖОЙСТИКА
141 #DEFINE BTN_STOP 3    // КНОПКА ЭКСТРЕННОЙ ОСТАНОВКИ
142 #DEFINE BTN_LIGHT 4   // КНОПКА УПРАВЛЕНИЯ СВЕТОМ
143 #DEFINE LED_SEND 5    // СВЕТОДИОД ПЕРЕДАЧИ ДАННЫХ
144
145 LIQUIDCRYSTAL LCD(7, 8, 9, 10, 11, 12);
146
147 INT LASTJOYX = 0;
148 INT LASTJOYY = 0;
149 BOOL LASTBTNSTATE = HIGH;
150 BOOL LIGHTSTATE = FALSE;
151

```

```

152 VOID SETUP() {
153     PINMODE(JOY_BTN, INPUT_PULLUP);
154     PINMODE(BTN_STOP, INPUT_PULLUP);
155     PINMODE(BTN_LIGHT, INPUT_PULLUP);
156     PINMODE(LED_SEND, OUTPUT);
157
158     LCD.BEGIN(16, 2);
159     LCD.PRINT("РУЛЬ   УПРАВ.");
160     LCD.SETCURSOR(0, 1);
161     LCD.PRINT("МАШИНКОУ");
162     SERIAL.BEGIN(9600);
163     DELAY(2000);
164     LCD.CLEAR();
165 }
166
167 VOID LOOP() {
168     CHECKJOYSTICK();
169     CHECKBUTTONS();
170     UPDATEDISPLAY();
171     DELAY(100);
172 }
173
174 VOID CHECKJOYSTICK() {
175     INT JOYX = ANALOGREAD(JOY_X);
176     INT JOYY = ANALOGREAD(JOY_Y);
177     BOOL BTNSTATE = DIGITALREAD(JOY_BTN);
178
179     IF (ABS(JOYX - LASTJOYX) > 50 || ABS(JOYY - LASTJOYY) >
50) {
180         IF (JOYY < 400) {
181             SENDCOMMAND('F'); // ВПЕРЕД
182         } ELSE IF (JOYY > 600) {
183             SENDCOMMAND('B'); // НАЗАД
184         } ELSE IF (JOYX < 400) {
185             SENDCOMMAND('L'); // ВЛЕВО
186         } ELSE IF (JOYX > 600) {
187             SENDCOMMAND('R'); // ВПРАВО
188         } ELSE {
189             SENDCOMMAND('S'); // СТОП
190         }
191         LASTJOYX = JOYX;
192         LASTJOYY = JOYY;
193     }
194
195     IF (BTNSTATE == LOW && LASTBTNSTATE == HIGH) {
196         SENDCOMMAND('D'); // ДАННЫЕ ДАТЧИКОВ
197         DELAY(200);
198     }
199     LASTBTNSTATE = BTNSTATE;
200 }
201
202 VOID CHECKBUTTONS() {
203     IF (DIGITALREAD(BTN_STOP) == LOW) {

```

```

204     SENDCOMMAND('S'); // ЭКСТРЕННАЯ ОСТАНОВКА
205     LCD.SETCURSOR(0, 1);
206     LCD.PRINT("СТОП!      ");
207     DELAY(500);
208 }
209
210 IF (DIGITALREAD(BTN_LIGHT) == LOW) {
211     LIGHTSTATE = !LIGHTSTATE;
212     IF (LIGHTSTATE) {
213         SENDCOMMAND('1'); // ФАРЫ ВКЛ
214     } ELSE {
215         SENDCOMMAND('0'); // ФАРЫ ВЫКЛ
216     }
217     DELAY(300);
218 }
219 }
220
221 VOID SENDCOMMAND(CHAR COMMAND) {
222     SERIAL.WRITE(COMMAND);
223     DIGITALWRITE(LED_SEND, HIGH);
224     DELAY(50);
225     DIGITALWRITE(LED_SEND, LOW);
226 }
227
228 VOID UPDATEDISPLAY() {
229     LCD.SETCURSOR(0, 0);
230     LCD.PRINT("COCT.: ");
231
232     INT JOYX = ANALOGREAD(JOY_X);
233     INT JOYY = ANALOGREAD(JOY_Y);
234
235     IF (JOYY < 400) {
236         LCD.PRINT("ВПЕРЕД ");
237     } ELSE IF (JOYY > 600) {
238         LCD.PRINT("НАЗАД ");
239     } ELSE IF (JOYX < 400) {
240         LCD.PRINT("ВЛЕВО ");
241     } ELSE IF (JOYX > 600) {
242         LCD.PRINT("ВПРАВО ");
243     } ELSE {
244         LCD.PRINT("СТОП      ");
245     }
246
247     LCD.SETCURSOR(0, 1);
248     LCD.PRINT("C BET:");
249     LCD.PRINT(LIGHTSTATE ? "ON " : "OFF");
250     LCD.PRINT("      ");
251 }

```

**ПРИЛОЖЕНИЕ Б**  
*(обязательное)*

**Схема электрическая структурная**

**ПРИЛОЖЕНИЕ В**  
*(обязательное)*

**Схема электрическая функциональная**

**ПРИЛОЖЕНИЕ Г**  
*(обязательное)*

**Схема электрическая принципиальная**

**ПРИЛОЖЕНИЕ Д**  
*(обязательное)*

**Схема программы**

**ПРИЛОЖЕНИЕ Е**  
*(обязательное)*

**Перечень элементов**

**ПРИЛОЖЕНИЕ Ж**  
*(обязательное)*

**Ведомость документов**