

# СОДЕРЖАНИЕ

Введение.....	6
1 Обзор литературы .....	7
1.1 Микроконтроллерные платформы .....	7
1.2 Радиомодули для беспроводной связи.....	7
1.3 Системы привода и драйверы моторов.....	8
1.4 Датчики для мобильных роботов .....	8
1.5 Индикация и пользовательский интерфейс.....	8
1.6 Датчики освещенности .....	9
2 Структурное проектирование .....	10
2.1 Общая структура системы.....	10
2.2 Структура пульта управления.....	10
2.3 Структура мобильного робота .....	10
2.4 Взаимодействие модулей .....	10
2.5 Преимущества выбранной структуры.....	11
2.6 Перспективы развития структуры .....	11
3 Функциональное проектирование .....	12
3.1 Обоснование выбора микроконтроллеров.....	12
3.2 Обоснование выбора датчика освещенности .....	12
3.3 Обоснование выбора ультразвукового датчика расстояния .....	12
3.4 Обоснование выбора модулей радиопередачи.....	12
3.5 Обоснование выбора драйвера моторов .....	12
3.6 Обоснование выбора мотор-редукторов.....	13
3.7 Система освещения .....	13
3.8 Обработка прерываний .....	13
4 Разработка принципиальной электрической схемы устройства .....	14
4.1 Расчёт мощности элементов схемы.....	14
4.2 Расчёт нагрузки светодиодов .....	15
4.3 Микроконтроллеры .....	16
4.4 Датчик освещенности .....	16
4.5 Ультразвуковой датчик расстояния .....	17
4.6 Модуль радиопередачи .....	17
4.7 Драйвер моторов .....	17
4.8 Мотор-редукторы .....	17
5 Разработка программного обеспечения .....	18
5.1 Требования к разработке программного обеспечения .....	18
5.2 Блок-схема алгоритма.....	18
5.3 Исходный код программы для устройства управления .....	19
5.4 Исходный код программы для передвижного устройства .....	20
6 Разработка корпуса устройства .....	22
6.1 Концепция и материалы .....	22
6.2 Конструкция передвижного устройства .....	22
6.3 Конструкция пульта управления .....	23
7 Руководство пользователя .....	24

7.1 Назначение устройства .....	24
7.2 Комплектация .....	24
7.3 Подготовка к работе.....	24
7.4 Порядок работы .....	24
7.5 Меры предосторожности.....	25
7.6 Уход и техническое обслуживание .....	25
Заключение .....	26
Список использованных источников .....	27
Приложение А .....	28
Приложение Б .....	33
Приложение В.....	34
Приложение Г .....	35
Приложение Д.....	36
Приложение Е .....	37
Приложение Ж.....	38

## 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 робота 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.

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].

## 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.

### **1.3 Системы привода и драйверы моторов**

Мобильные роботы 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.

### **1.4 Датчики для мобильных роботов**

Классификация датчиков:

- тактильные (концевики);
- оптические (ИК, лидары);
- акустические (ультразвуковые);
- инерциальные (акселерометры).

Характеристики датчиков линии:

1 TCRT5000. Дальность 0.2-1.5 см, аналоговый выход.

2 QTR-8A. 8 сенсоров, цифровой выход, калибровка.

Для ориентации в пространстве и выполнения задач робототехники используют различные сенсоры:

- датчики линии (ИК-датчики отражения);
- ультразвуковые дальномеры (HC-SR04);
- энкодеры для измерения скорости вращения колес;
- датчики освещенности.

В данном проекте применяются датчики линии, при необходимости система может быть дополнена ультразвуковым датчиком HC-SR04 для измерения расстояния.

## 1.5 Индикация и пользовательский интерфейс

Типы дисплеев:

- 1 LCD 1602. Дешевый, требует много пинов.
- 2 OLED SSD1306. Яркий, I2C интерфейс.
- 3 TFT дисплеи. Цветное изображение, высокая цена.

Анализ джойстиков:

- 1 Потенциометрические. Аналоговый выход, износ.
- 2 Магниторезистивные. Высокая точность, цена.

Для взаимодействия с пользователем применяются:

- светодиоды – простейший способ отображения состояния;
- джойстики и кнопки – средства управления роботом.

## 1.6 Датчики освещенности

Наиболее распространёнными моделями датчиков освещенности являются фоторезистивный датчик освещенности на чипе LM393, датчик освещенности CJMCU-TEMT6000 и модуль освещения GY-302 на чипе BH1750FVI. В таблице 1.2 приведены их сравнительные характеристики.

Таблица 1.2 – Сравнение датчиков освещенности

Параметры сравнения	Фоторезистивный датчик освещенности	Датчик освещенности CJMCU-TEMT6000	Модуль освещения GY-302
Угол чувствительности	$\pm 60^\circ$	$\pm 60^\circ$	$\pm 60^\circ$
Напряжение питания	3.3 – 5 В	3.3 – 5 В	3 – 5 В
Рабочая температура	от 0 до +70 °C	от -40 до +85 °C	от -25 до +85 °C
Потребляемый ток	до 15 мА	до 20 мА	до 120 мкА
Измеряемые значения	от 0 до 65535 лк	от 0 до 65535 лк	от 0 до 65535 лк
Тип выходного канала	аналоговый и цифровой	аналоговый	цифровой

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

## **2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

### **2.1 Общая структура системы**

Система состоит из двух взаимосвязанных частей:

- 1 Пульт управления – обеспечивает ввод управляющих команд (джойстик), их обработку микроконтроллером и передачу по радиоканалу.
- 2 Мобильный робот – принимает команды, обрабатывает их, управляет моторами, анализирует данные датчиков и передаёт обратную связь на пульт.
- 3 Основой системы является радиоканал NRF24L01, обеспечивающий двустороннюю связь между пультом и роботом.

### **2.2 Структура пульта управления**

Пульт управления включает в себя:

- Джойстик – формирует управляющие сигналы;
- Arduino UNO – выполняет обработку сигналов и формирование пакетов для передачи;
- NRF24L01 – передаёт данные на робота;
- источник питания (аккумулятор 18650 или powerbank) – обеспечивает работу пульта.

### **2.3 Структура мобильного робота**

Мобильный робот 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**

Пульт передаёт управляющие команды по радиоканалу.

Робот принимает команды, обрабатывает их и изменяет направление/скорость движения.

Датчики передают информацию в микроконтроллер, который может использовать её для коррекции движения или передачи данных обратно на пульт.

Светодиоды и OLED-дисплей (если установлен) обеспечивают визуальную обратную связь пользователю.

## **2.5 Преимущества выбранной структуры**

Модульность – каждый компонент можно заменить или модернизировать.

Простота реализации – используется доступная элементная база.

Расширяемость – возможна установка дополнительных датчиков или переход на более мощные драйверы моторов.

Доступность компонентов – все модули широко распространены и имеют низкую стоимость.

## **2.6. Перспективы развития структуры**

Переход на STM32.

Добавление резервирования каналов связи.

Система самодиагностики.

## **3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

### **3.1 Обоснование выбора микроконтроллеров**

В данной курсовом проекте используется плата Arduino UNO так как она полностью подходит под поставленные задачи. Данный контроллер был выбран также по причине доступности и наличия опыта разработки под данный тип микроконтроллеров.

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

В данном проекте используется фоторезистивный датчик освещенности на чипе LM393 так как он имеет как аналоговый, так и цифровой выходы, что может быть полезным при работе с данным модулем. Также данная модель датчика была выбрана по причине потребления меньшего тока, что значительно уменьшит затраты на при подключении к общей схеме устройства.

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

Было принято решение выбрать датчик HC-SR04 из-за большего диапазона измеряемого расстояния и рабочей температуры. По остальным параметрам, таким как частота звукового излучения, напряжение питания, потребляемый ток и рабочий угол наблюдения данные датчики имеют одинаковые характеристики.

### **3.4 Обоснование выбора модулей радиопередачи**

После анализа модулей радиопередачи, представленных в таблице 1.5 обзора литературы видно, что модуль NRF24L01 имеет лучшие характеристики, по сравнению с передатчиком MX-05V и приемником XD-RF-5V, но рассчитан на меньшее расстояние передачи радиосигнала.

В данном устройстве будут использованы радиопередатчик NRF24L01 и приемник NRF24L01 + PA + LNA. Хоть организация радиопередачи между этими двумя модулями является односторонней, в данном курсовом проекте это не существенно так как передача всегда будет происходить от пульта управления к устройству передвижения. Также данные модули были выбраны по причине большего расстояния передачи и существенно меньшей стоимости.

### **3.5 Обоснование выбора драйвера моторов**

В данном курсовом проекте будет использована модель L298N исходя из наличия данного драйвера, возможности подключения одновременно четырех моторов и удовлетворения всех необходимых параметров для

проектируемого устройства. Данный драйвер имеет довольно большой диапазон напряжения питания и может быть использован для стабилизации питания других датчиков.

### **3.6 Обоснование выбора мотор-редукторов**

В данном устройстве будут использованы прямые двухосевые мотор-редукторы с передаточным числом 1:48. Они имеют более подходящий диапазон напряжения питания и более высокую скорость вращения.

### **3.7 Система освещения**

Режимы работы фар:

- постоянное освещение;
- мигающий режим;
- автоматическое включение при низкой освещенности.

### **3.8 Обработка прерываний**

Приоритеты прерываний:

- аварийная остановка;
- данные с радиомодуля;
- обработка датчиков.

## 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

Потребляемая мощность разрабатываемого устройства равна сумме мощностей, потребляемых его элементами. Расчет мощности элементов схемы устройства управления и самого устройства представлены в таблице 4.1 и 4.2 соответственно.

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	5	28	1	140
Джойстик	5	8	2	80
Светодиод	5	20	1	100
Суммарная мощность, мВт				430

В реализованной схеме используются микроконтроллер Arduino UNO R3, модуль радиопередачи MX-05V, 2 джойстика и светодиод.

Таким образом потребляемая мощность будет равна:

$$P = 5 \cdot 22 + 5 \cdot 28 + 5 \cdot 8 \cdot 2 + 5 \cdot 20 = 430 \text{ мВт.}$$

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 516 мВт.

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{0.430}{5} = 0.086 \approx 0.09 \text{ A.}$$

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 NRF24L01	5	6	1	30
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 3-8V	5	600	4	12 000
Светодиод	5	20	4	400
Суммарная мощность, мВт				13000

В реализованной схеме используются микроконтроллер Arduino UNO R3, модуль радиоприема XD-RF-5V, датчик освещенности LM393, датчик расстояния HC-SR04, драйвер моторов L298N, 4 мотор-редуктора 1:48 3-8V и 4 светодиода.

Таким образом потребляемая мощность будет равна:

$$P = 5 \cdot 22 + 5 \cdot 6 + 5 \cdot 15 + 5 \cdot 15 + 5 \cdot 36 \cdot 2 + 5 \cdot 600 \cdot 4 + 400 = 13 \text{ Вт.}$$

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 15600 мВт.

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{15.600}{5} = 3.12 \text{ А.}$$

## 4.2 Расчёт нагрузки светодиодов

В данном курсовом проекте используется пять светодиодов различных цветов, подключенные к соответствующим pinам микроконтроллера (см. рисунок 4.2).

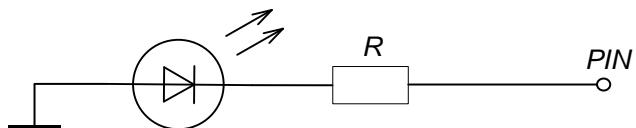


Рисунок 4.1 — Схема подключения светодиодов

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

$$R = \frac{U_{\Pi} - U_D}{I_{\text{ПР}}},$$

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

В проекте используются светодиоды белого, желтого и зеленого цветов, со следующими параметрами:  $I_{\text{ПР}} = 20 \text{ мА}$ .  $U_D = 1 \text{ В}$ .

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.

В схеме с устройством управления к аналоговым входам A1 и A2 подключены джойстики, а на цифровой выход D12 подключен радиопередатчик.

В схеме передвижного устройства модуль освещенности подключен к аналоговому входу A1, датчик газов к аналоговому входу A0 и цифровому входу D2, датчик расстояния подключен к цифровым входам D9 и D8, радиоприемник к цифровому входу D11, пьезодинамик к цифровому выходу D10. Драйвер моторов подключен к Arduino через цифровые выходы D4 – D7, а светодиоды через цифровые выходы D12 и D13. Данный микроконтроллер на обоих схемах питается от напряжения 5 В.

### 4.4 Датчик освещенности

Данный датчик в схеме питается от напряжения 5 В. К аналоговому входу A1 микроконтроллера подключен выход датчика A0, через который поступает информация о степени освещенности.

## **4.5 Ультразвуковой датчик расстояния**

Данный датчик подключается к контроллеру с помощью двух своих выходов Echo и Trig, в данной схеме они подключены к цифровым входам микроконтроллера D8 и D9. Питается датчик от напряжения 5 В.

## **4.7 Модуль радиопередачи**

На устройстве управления используется радиопередатчик FS1000A/XD-FST, который питается от напряжения 5 В. Также на модуле есть вход Data, он подключен к цифровому выходу D12 микроконтроллера, через него осуществляется передача данных с Arduino на модуль.

На передвижном устройстве установлен радиоприемник XD-RF-5V, который также питается от напряжения 5 В. На данном модуле есть единственный выход Data, который подключен к цифровому входу D11 микроконтроллера. Через данный выход микроконтроллер получает информацию с радиоприемника.

Оба модуля подключены к антенне для улучшения радиосигнала.

## **4.8 Драйвер моторов**

На вход VCC подается напряжение 5 В. Входы IN1, IN2, IN3, IN4 подключаются к микроконтроллеру, который выставляет направление вращения моторов. Драйвер анализирует полученные значения и запускает моторы через выходы OUT1, OUT2, OUT3 и OUT4.

## **4.9 Мотор-редукторы**

Информация о выбранных мотор-редукторах представлена в пункте 3.7 раздела 3. Все они связаны с микроконтроллером через драйвер двигателей L298N. При поступлении питания на соответствующие входы моторов они начинают вращаться в заданную сторону.

## **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.

## **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**