Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Схемотехника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

АНАЛИЗАТОР ЗВУКОВОГО СИГНАЛА

БГУИР КП 1-40 02 01 104 ПЗ

Студент: группы 150504,   
Горбачевский К. В.

Руководитель:   
Калютчик А. А.

Минск 2023

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

**З А Д А Н И Е**

**по курсовому проектированию**

Студенту Горбачевскому Кириллу Витальевичу\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

1. Тема проекта Анализатор звукового сигнала

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Срок сдачи студентом законченного проекта с 06.12.2021 по 09.12.2021\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к проекту:

1. Микроконтроллер – тактовая частота не менее 20 кГц, не менее 10 входов/выходов.\_\_\_\_\_\_\_\_\_\_

2. Источник питания – напряжение 9 В, максимальный выходной ток не менее 3 А.\_\_\_\_\_\_\_\_\_\_\_

3. Модуль усиления – напряжение питания 7 – 36 В, коэфициент усиления не меньше 10. \_

4. Дисплей – напряжение питания 3.3-5 В, разрешение не менее 128x64.

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

Введение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.Обзор литературы.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.Разработка структуры микропроцессорного радиоуправляемого передвижного устройства.\_\_\_\_\_

3.Обоснование выбора узлов, элементов функциональной схемы устройства.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Разработка принципиальной электрической схемы устройства.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Разработка программного обеспечения. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заключение.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Литература.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перечень графического материала (с точным указанием обязательных чертежей)

1. Структурная схема анлизатора звукового сигнала устройства (формат А4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Функциональная электрическая схема анлизатора звукового сигнала (формат А3)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Принципиальная электрическая схема анлизатора звукового сигнала (формат А3)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Консультант по проекту (с назначением разделов проекта)\_A.A. Калютчик \_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Дата выдачи задания 10.09.2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Календарный график работы над проектом на весь период проектирования (с назначением сроков исполнения и трудоемкости отдельных этапов):

разделы 1,2 к 24.09 – 20 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел 3 к 15.10 – 20 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел 4 к 05.11 – 25 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел 5 к 19.11 – 20 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оформление пояснительной записки и графического материала к 06.12 – 15 %;\_\_\_\_\_\_

защита курсового проекта с 07.12 по 14.12.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РУКОВОДИТЕЛЬ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_асистент каф. ЭВМ Калютчик А.А.

(подпись)

Задание принял к исполнению 10.09.2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_К. В. Горбачевский

(дата и подпись студента)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc149809648)

[1 ОБЗОР КОМПОНЕНТОВ 5](#_Toc149809649)

[1.1 Состав устройства 5](#_Toc149809650)

[1.2 Микроконтроллеры 5](#_Toc149809651)

[1.3 Операционные усилители 6](#_Toc149809652)

[1.4 Гнездо и кабель 6](#_Toc149809653)

[1.5 Дисплей 6](#_Toc149809654)

[1.6 Питание 7](#_Toc149809655)

[2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА 8](#_Toc149809656)

[2.1 Постановка задачи 8](#_Toc149809657)

[2.2 Определение компонентов структуры устройства 8](#_Toc149809658)

[2.3 Взаимодействие компонентов устройства 8](#_Toc149809659)

[3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА 9](#_Toc149809660)

[3.1 Обоснование выбора микроконтроллеров 9](#_Toc149809661)

[3.2 Обоснование выбора операционного усилителя 9](#_Toc149809662)

[3.3 Обоснование выбора кабеля и гнезда 9](#_Toc149809663)

[3.4 Обоснование выбора дисплея 10](#_Toc149809664)

[3.5 Обоснование выбора питания 10](#_Toc149809665)

[4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА 11](#_Toc149809666)

[4.1 Расчёт мощности элементов схемы 11](#_Toc149809667)

[4.2 Расчёт нагрузки светодиодов 12](#_Toc149809668)

[4.3 Микроконтроллеры 13](#_Toc149809669)

[4.4 Датчик горючих газов 13](#_Toc149809670)

[4.5 Датчик освещенности 13](#_Toc149809671)

[4.6 Ультразвуковой датчик расстояния 14](#_Toc149809672)

[4.7 Модуль радиопередачи 14](#_Toc149809673)

[4.8 Драйвер моторов 14](#_Toc149809674)

[4.9 Мотор-редукторы 14](#_Toc149809675)

[4.10 Пьезодинамик 15](#_Toc149809676)

[4.11 Джойстики 15](#_Toc149809677)

[5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 16](#_Toc149809678)

[5.1 Требования к разработке программного обеспечения 16](#_Toc149809679)

[5.2 Блок-схема алгоритма 16](#_Toc149809680)

[5.3 Исходный код программы для устройства управления 17](#_Toc149809681)

[5.4 Исходный код программы для передвижного устройства 17](#_Toc149809682)

[ЗАКЛЮЧЕНИЕ 19](#_Toc149809683)

[ПРИЛОЖЕНИЕ А 22](#_Toc149809684)

[ПРИЛОЖЕНИЕ Б 23](#_Toc149809685)

[ПРИЛОЖЕНИЕ В 24](#_Toc149809686)

[ПРИЛОЖЕНИЕ Г 25](#_Toc149809687)

[ПРИЛОЖЕНИЕ Д 26](#_Toc149809688)

[ПРИЛОЖЕНИЕ Е 31](#_Toc149809689)

[ПРИЛОЖЕНИЕ Ж 32](#_Toc149809690)

# ВВЕДЕНИЕ

Темой данного курсового проекта является разработка устройства, анализирующего аналоговый сигнал на базе микроконтроллера.

Современная технологическая эпоха обогатила мир музыки и музыкантов инновационными средствами и инструментами для творчества. Однако, несмотря на множество цифровых достижений в области музыкального оборудования, аналоговые инструменты, такие как гитары, остаются популярными среди музыкантов разного уровня мастерства.

Важным аспектом игры на аналоговых инструментах, в том числе на гитаре, является настройка для достижения чистых и гармоничных звуков. В данном контексте представляется актуальной тема создания устройств, позволяющих музыкантам легко и точно настраивать свои инструменты.

Целью данного курсового проекта является разработка и реализация анализатора аналогового сигнала на базе микроконтроллера, предназначенного для настройки гитары.

Данное устройство может быть использовано не только для анализа сигнала исходящего от электрогитары. Прибор сможет определять частоту (тон) любого сигнала, передаваемого по кабелю Jack 6.3

В данной работе будет рассмотрен весь процесс разработки и реализации анализатора аналогового сигнала, начиная с выбора аппаратной платформы и заканчивая созданием программного обеспечения для обработки сигнала. Также будут рассмотрены различные методы анализа и алгоритмы, применяемые для определения настроенности струн инструмента.

В качестве основного вычислительного элемента устройства будет использована плата Arduino UNO. Выбор данной платы мотивирован тем, что она уже у меня была. Разработка программного обеспечения будет происходить в интегрированной среде разработки для Windows Arduino IDE 1.8.16. В данной среде есть все необходимое для написания программного обеспечения с последующей загрузки на плату.

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.

# ОБЗОР КОМПОНЕНТОВ

## Состав устройства

Как сказано ранее, разрабатываемое микропроцессорное устройство выполняет функции анализа аналогового сигнала, а если быть точнее - измерения частоты аналогового сигнала и определения тона. Для решения этих задач в состав устройства должны входить:

- микроконтроллер

- усилитель

- гнездо для кабеля

- кабель

- дисплей для вывода информации

- питание

## Микроконтроллеры

Существует огромное разнообразие плат с разными микроконтроллерами. Все они отличаются размерами, параметрами, предустановленными интерфейсами и выполняемыми задачами. Для сравнения была выбрана плата Arduino UNO и аналоги других производителей. Результаты сравнения приведены в таблице 1.1.

Таблица 1.1 — Сравнение микроконтроллеров

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры сравнения** | **Arduino UNO** | **Raspberry Pi Zero** | **NodeMcu Lua** |
| Микроконтроллер | ATmega328 | ARM Cortex-A53 | ESP8266 |
| Входное напряжение | 6 – 20 В | 6 – 28 В | 3.6 – 20 В |
| Флэш-память | 32 Кб | порт для microCD | 4 мб |
| ОЗУ | 2 Кб | 512 Мб | 20 Кб |
| Тактовая частота | 16 МГц | 1 ГГц | 80-160 МГц |
| Разрядность | 8 бит | 32 бит | 32 бит |
| Цифровые  входы/выходы | 14 шт | 26 шт | 11 шт |
| Аналоговые  входы/выходы | 6 шт | 0 шт | 1 шт |
| Выходное напряжение | 3.3В, 5 В | 3.3В, 5 В | 3.3В, 5 В |
| Рабочая температура | от -25 до +85 ºС | от -40 до +85 ºС | от -25 до +85 ºС |

Для получения более подробной информации о рассмотренных микроконтроллерах использовались источники [4, 5, 6].

## Операционные усилители

Сам по себе сигнал, исходящий из гитары, очень слабый и для анализа требует усиления.

Для данной схемы выбран операционный усилитель TL082, но существует куда больше усилителей, выполняющих разные задачи. Результаты сравнения приведены в таблице 1.2.

Таблица 1.2 — Сравнение усилителей

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры сравнения** | **TL082** | **OPA2134** | **LT1803** |
| Коэффициент усиления | 86 | 104-120 | 50 |
| Полоса пропускания | 5 MHz | 8 MHz | 85 MHz |
| Рабочая температура | от -40 до +125 ºС | от -55 до +125 ºС | от -40 до +85 ºС |
| Кол-во каналов | 2 | 2 | 1 |

Для получения точной информации о данных датчиках использовалась техническая спецификация [8, 9, 10].

## Гнездо и кабель

Почти все профессиональное музыкальное оборудование взаимодействует между собой посредством кабелей Jack 6.3мм. Для устройства представленного в данной курсовой работе был выбран именно такой формат кабеля и гнезда. Существует два типа кабелей Jack 6.3: mono и stereo. В данной курсовой работе этот параметр не имеет значения так как анализируется один канал сигнала.

## Дисплей

Для того чтобы устройством было комфортно пользоваться и снимать данные в схеме предусмотрен небольшой OLED дисплей.

Сигнал, обработанный программой в символы/числовые значения, будет передан на дисплей с выходов микроконтроллера.

Дисплей имеет разрешение 128x64, поддерживает желтый и синий цвета.

## Питание

Для работы устройства требуется питание. В данной схеме использовались две батарейки 6F22 (“Крона”) отдельно для микроконтроллера и схемы с усилителем.

# РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

## 2.1 Постановка задачи

Для того, чтобы составить структуру разрабатываемого устройства, необходимо выделить функции, которые будет выполнять устройство, затем определить компоненты и связь между ними исходя из данных функций. Результаты можно посмотреть на структурной схеме, представленной в приложении А.

В рамках данного курсового проекта необходимо разработать анализатор звукового сигнала. Для реализации было выбрано устройство, определяющее частоту сигнала (тон), исходящего от электрогитары.

## 2.2 Определение компонентов структуры устройства

Компоненты структуры устройства выбираются исходя из функций, определенных в постановке задачи. Проанализировав выделенные функции, были определены следующие компоненты, представленные ниже.

1) Микроконтроллер — ключевой компонент всей схемы. Выполняет функцию обработки поступающей информации и выдает значение.

2) Питание — источние постоянного напряжения 9V.

3) Модуль усиления сигнала – компонент схемы, усиливающий слабый аналаговый сигнал для дальнейшего анализа микроконтроллером.

2.3 Взаимодействие компонентов устройства

При колебании струн электрогитары, электромагнитный сигнал поступает на вход модуля усиления по кабелю Jack 6.3. Усиленный сигнал с выхода модуля усиления анализируется микроконтроллером и значение частоты исходного сигнала выводятся на OLED дисплей

Модуль питания взаимодействует со всеми элементами схемы напрямую или через модуль усиления, благодаря ему осуществляется питание всех необходимых элементов.

# ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

## Обоснование выбора микроконтроллеров

В данной курсовом проекте в качестве контроллера могла быть использована любая плата из представленных в таблице 1.1 так как устройство не требует больших затрат в памяти и мощности для корректной работы.

Контроллер Raspberry PI Zero является чем-то большим чем просто микроконтроллером. Данный “мини-компьютер” имеет процессор на 1ГГц и 512мб оперативной памяти что в сотни раз больше чем аналогичные характеристики других микроконтроллеров. И именно по этой причине Raspberry PI Zero не подходит для решения относительно простой задачи анализа звукового сигнала и его пераметры будут излишни.

NodeMcu Lua с другой стороны самый близкий к Arduino контроллер, со встроенным WI-FI модулем. Он превосходит Arduino Uno по основным характеристикам и в разы меньше по размерам. Но для разработки полу-аналаговой схемы работать с таким контроллем не комфортно из-за малых габаритов и отсутствия отдельного входа питания, который в Arduino Uno присутсвует.

В связи со всеми вышеперечисленными причинами, был выбран контроллер Arduino Uno который уже имелся в наличии.

## Обоснование выбора операционного усилителя

Операционные усилители семейства TL являются самыми популярными и доступными операционными усилителями. Стоимость TL082 в несколько раз меньше стоимости OPA2134 при почти одинаковых характеристиках, а усилитель LT1803 хоть и обладает более лучшими характеристиками, тем не менее почти не встречается в продаже.

## Обоснование выбора кабеля и гнезда

Для данных компонентов почти не существует альтернатива. Электрогитара имеет выход Jack 6.3-mono. Кабель расчитан на прямое подключение к гитаре без использования переходников которые сильно влияют на шум сигнала. По таким соображение в качестве входной точки схемы было выбрано гнездно польностью сочетающиеся с кабелем – Jack 6.3-mono. Это стандартное гнездо, используемое во всех музыкальных приборах начиная от тюнеров и комбо-усилителей, заканчивая профессиональным студийным оборудованием, колонками и наушниками.

## Обоснование выбора дисплея

В данной курсовой работе предусмотрен вывод результата анализа звукового сигнала на дисплей в качестве цифр, буква и символов. Для данной задачи был выбран OLED дисплей небольшого разрешения 128x64 пикселя. Стоимость OLED дисплея в несколько раз превышает стоимость ЖК-дисплея с аналогичным разрешением из-за различных типов матриц и способа передачи изображения. Но OLED дисплей обладает лучшими характеристиками цветопередачи и отзывчивости, а также прост в использовании.

## Обоснование выбора питания

Для данной схемы предусмотрено питание от двух батареек 6F22 (“Крона”) отдельно для микроконтроллера и схемы с усилителем. При использовании батареек прибор становиться мобильным и появляется возможность его использования без подключения к сети постоянному напряжения. Микроконтроллер Arduino UNO требует для себя питания 9V. Поэтому были выбраны батарейки типа 6F22 (“Крона”).

# РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА

## 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 MX-05V | 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:

Р = 5 ∙ 22 + 5 ∙ 28 + 5 ∙ 8 ∙ 2 + 5 ∙ 20 = 430 мВт.

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

Рассчитаем потребляемый ток:

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 XD-RF-5V | 5 | 6 | 1 | 30 |
| Датчик горючих газов MQ-5 | 5 | 150 | 1 | 750 |
| Датчик освещенности LM393 | 5 | 15 | 1 | 75 |
| Датчик расстояния HC-SR04 | 5 | 15 | 1 | 75 |
| Драйвер моторов L298N | 5 | 36 | 1 | 180 |
| Мотор-редуктор 1:48 3-8V | 5 | 600 | 4 | 12 000 |

Продолжение таблицы 4.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Пьезодинамик | 5 | 10 | 1 | 50 |
| Светодиод | 5 | 20 | 4 | 400 |
| Суммарная мощность, мВт | | | | 13670 |

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

Таким образом потребляемая мощность будет равна:

Р = 5 ∙ 22 + 5 ∙ 6 + 5 ∙ 150 + 5 ∙ 15 + 5 ∙ 15 + 5 ∙ 36 + 5 ∙ 600 ∙ 4 + 5 ∙ 10 + 5 ∙ 20 ∙ 4 = 13670 мВт.

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 16404 мВт.

Рассчитаем потребляемый ток:

## Расчёт нагрузки светодиодов

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



Рисунок 4.2 — Схема подключения светодиодов

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

где *U*П – напряжения питания, *U*Д – напряжение, падающее на светодиоде, *I*ПР – прямой ток светодиода.

В проекте используются светодиоды белого, желтого и зеленого цветов, со следующими параметрами: *I*ПР = 20 мА. *U*Д = 1 В.

Получаем:



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

Информация о выбранном микроконтроллере Arduino UNO представлена в пункте 3.1 раздела 3.

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 B.

## 4.4 Датчик горючих газов

Информация о выбранном датчике горючих газов MQ-5 представлена в пункте 3.2 раздела 3. Данный датчик подключен как к аналоговому входу микроконтроллера A0, так и к цифровому D2, это обусловлено тем, что через цифровой вход передается информация о присутствии газа, а через аналоговый о его концентрации. Питается датчик от напряжения 5 В.

## 4.5 Датчик освещенности

Информация о выбранном датчике освещенности LM393 представлена в пункте 3.3 раздела 3. 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.

## 4.6 Ультразвуковой датчик расстояния

Информация о выбранном ультразвуковом датчике расстояния HC-SR04 представлена в пункте 3.4 раздела 3. 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

Информация о выбранных модулях радиопередачи представлена в пункте 3.5 раздела 3.

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 микроконтроллер. Через данный выход микроконтроллер получает информацию с радиоприемника.

Оба модуля подключены к антенне для улучшения радиосигнала.

## 4.8 Драйвер моторов

Информация о выбранном драйвере моторов L298N представлена в пункте 3.6 раздела 3. 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.

## 4.10 Пьезодинамик

Для получения звукового сигнала используется пьезодинамик, который подключается к цифровому выходу Arduino D10. Микроконтроллер включает и отключает питание на данном динамике из-за чего включается и отключается звук.

## 4.11 Джойстики

В качестве управления в данном курсовом проекте используются два джойстика подключенных в схеме устройства управления к аналоговым входам Arduino A1 и A2. На вход VCC подается напряжение 5 В.

# РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## Требования к разработке программного обеспечения

Разработанное микропроцессорное устройство собирает информацию о трех параметрах: концентрация горючих газов в среде, уровень освещенности и расстояние до ближайшего объекта.

Устройство работает следующим образом. При включении пульта дистанционного управления на нем загорается светодиод, который сигнализирует о том, что питание подключено. Когда пользователь меняет направление на джойстиках, микроконтроллер анализирует полученную информацию и отправляет код состояния на радиоприемник устройства через радиопередатчик, установленный на пульте. Затем полученная информация анализируется на устройстве и в зависимости от полученного кода состояния запускаются или приостанавливаются двигатели.

В течении всего времени устройство анализирует показания с датчиков. При получении информации о высокой концентрации горючих газов в среде включается пьезодинамик, который начинает издавать звук. Если в рабочей зоне недостаточно освещения, тогда включаются два светодиода, закрепленных на передней стороне устройства. Когда поступает информация о том, что перед устройством находится какой-то объект, то движение вперед останавливается и загораются светодиоды, установленные по бокам устройства. Если устройству не поступает никакая из вышеперечисленных информаций, то оно работает в обычном режиме без индикации.

## Блок-схема алгоритма

Блок-схема — это схематичное представление процесса, системы или компьютерного алгоритма. Блок-схемы часто применяются в разных сферах деятельности, чтобы документировать, изучать, планировать, совершенствовать и объяснять сложные процессы с помощью простых логичных диаграмм.

Рассмотрим блок-схему алгоритма программного обеспечения данного курсового проекта, представленную в приложении Г.

На первом листе приложения представлена блок схема алгоритма самого устройства. Блоки 2 – 3 представляют собой подготовку программы для дальнейшей работы (инициализация переменных и определение модулей, подключенных к микроконтроллеру). Ключевыми являются блоки 5 – 29, которые реализуют саму логику программы в бесконечном цикле. В блоках 5, 9, 15 происходит получение данных с датчиков освещенности, расстояния и газа соответственно. В блоках 6 – 8, 11 – 14, 16 – 18 данная информация анализируется и, при необходимости, включается соответствующая индикация. В 20 блоке проверяется наличие входящего сообщения от радиопередатчика, полученного в 19 блоке. Если данные присутствуют, то, в зависимости от полученного кода (блоки 21 – 29) происходит движение, заданное пользователем.

На втором листе приложения представлена блок схема пульта дистанционного управления устройством. Блоки 2 – 3 аналогичны блокам устройства. Блоки 5 – 6 реализуют получение информации о действиях пользователя, а блоки 7 – 15 анализируют полученные данные и выставляют соответствующий код на радиопередатчик. В блоке 16 информация отправляется устройству.

## Исходный код программы для устройства управления

Пульт дистанционного управления считывает показания с двухосевых джойстиков, а затем преобразует данные для быстрой и удобной отправки по радиоканалу. Исходный код программного обеспечение под данное устройство можно найти в приложении Д (строки 1 – 53).

Функция void setup() (строки 22 – 27) необходима для начальной настройки контроллера, здесь задаются входные и выходные пины, а так же другие настройки. В данном случае vw\_setup(2000) (строка 23) настраивает наш передатчик, а pinMode(pinX, INPUT) и pinMode(pinY, INPUT) (строки 25 и 26 соответственно) задают пины джойстика, как входные значения на плату.

Функция void loop() (строки 29 – 48) является главной, циклической функцией и работает на протяжении всей работы микроконтроллера. Здесь задается основная логика работы микроконтроллера. На начальном этапе работы функции обнуляются входные значения с джойстиков, а также очищается строка для отправки информации на радиоприёмник. Затем считываются новые данные с джойстиков, на основе полученных значений формируется новое сообщение, которое хранит номер операции для взаимодействия с устройством, подключенному к приёмнику. После этого сообщение конвертируется в массив char и передается на радиопередатчик для отправки.

Функция void send (char \*message) (строки 50 – 53) получает в качестве параметра массив символов, которые необходимо отправить на радиоприёмник. Полученный массив отправляется и затем происходит ожидание полной отправки сообщения.

## Исходный код программы для передвижного устройства

Данное устройство является основным в разрабатываемом курсовом проекте и здесь реализована большая часть логики. Исходный код программного обеспечение можно найти в приложении Д (строки 58 – 235).

Функция void setup() (строки 97 – 115) выполняет такие же задачи, как и в устройстве управления. В строках 98 – 99 настраивается радиоприемник, затем в строках 101 – 114 задаются входные и выходные пины устройства.

Функция void loop() (строки 117 – 124) является главной, здесь вызываются все методы для выполнения поставленных задач устройства. Она также работает на протяжении всей работы микроконтроллера.

Функция void checkIlumination() (строки 127 – 136) реализует логику взаимодействия с датчиком освещенности, здесь происходит считывание информации (строка 128) с последующей проверкой минимального допустимого значения. Строки 131 и 134 включают и выключают собственное освещение устройства соответственно.

Функция void checkDistanceToObject() (строки 138 – 159) реализует логику взаимодействия с датчиком расстояния. В строках 140 – 145 посылается два ультразвуковых сигнала равные 2 и 10 миллисекунд для получения информации о расстоянии до ближайшего объекта. В строке 149 полученное значение переводится в сантиметры, после чего (строки 151 – 158) сравнивается с минимально допустимым и, в зависимости от результата включаются и выключаются боковые светодиоды устройства (строки 153 и 157 соответственно), а также происходит остановка устройства, если значение довольно мало.

Функция void checkOfGas() (строки 161 – 171) реализует логику взаимодействия с датчиком горючих газов. Изначально происходит считывание информации с данного датчика (строки 162 – 163), а затем проверка допустимых значений (строки 165 – 169), если значение выше допустимого, то вызывается метод turnOnSound(), в ином случае метод turnOffSound() о которых будет сказано ниже.

Функция void turnOnSound() (строки 173 – 175) включает пьезодинамик, а функция void turnOffSound() (строки 177 – 179) его выключает.

Функция void selectStateOfMotors(char state) (строки 181 – 198) получает в качестве параметра состояние движения, которое отправил пользователь. В теле метода происходит сравнение со всеми возможными состояниями и, в зависимости от результатов сравнения, вызываются методы для работы с моторами.

Функции void mooveForward() (строки 200 – 205) , void mooveBack() (строки 207 – 212), void turnLeft() (строки 214 – 220) , void turnRight() (строки 222 – 228) и void stand() (строки 230 – 235) реализуют логику взаимодействия с моторами: движение прямо, назад, влево, вправо и бездействие соответственно.

# ЗАКЛЮЧЕНИЕ

В результате работы над данным курсовым проектом было разработано работоспособное микропроцессорное радиоуправляемое передвижное устройство со своим программным обеспечением. Устройство отслеживает показания концентрации горючих газов в среде, уровень освещенности и расстояние до ближайшего объекта. Помимо этого, осуществляется анализ полученных значений и соответствующая индикация показателей. Данный проект был спроектирован в соответствии с поставленными задачами, весь функционал был реализован в полном объеме.

Разработанное микропроцессорное устройство обладает следующими достоинствами: относительно низкая стоимость, простота реализации и сборки. Однако существенным недостатком является необходимость в написании собственного программного обеспечения для взаимодействия со всеми подключенными датчиками и анализа полученных данных.

В дальнейшем планируется усовершенствование данного курсового проекта. Одним из таких улучшений является оптимизация алгоритма анализа полученных данных, улучшение питания, а также создание более дружественного интерфейса.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1]. Вычислительные машины, системы и сети: дипломное проектирование (методическое пособие) [Электронный ресурс] : Минск БГУИР 2019. – Электронные данные. – Режим доступа: <https://www.bsuir.by/m/12_100229_1_136308.pdf>

[2]. Документация Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.arduino.cc/>

[3]. Геддес, М. 25 крутых проектов с Arduino / М. Геддес ; [пер. с англ. М. А. Райтмана]. ­– Москва : Эксмо, 2019. – 272 c.

[4]. Arduino UNO [Электронный ресурс]. – Электронные данные. – Режим доступа: http://arduino.ru/Hardware/ArduinoBoardUno – Дата доступа: 11.09.2021

[5]. 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

[6]. 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

[7]. Статья – Датчики [Электронный ресурс] – Электронные данные. – Режим доступа: <http://www.electrolibrary.info/subscribe/sub_16_datchiki.htm> – Дата доступа: 18.09.2021

[8]. MQ-5 [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://files.seeedstudio.com/wiki/Grove-Gas\_Sensor-MQ5/res/MQ-5.pdf](https://files.seeedstudio.com/wiki/Grove-Gas_Sensor-MQ5/res/MQ-5.pdf%20) – Дата доступа: 22.09.2021

[9]. MQ-2 new [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.pololu.com/file/0J309/MQ2.pdf> – Дата доступа: 22.09.2021

[10]. Microsoft Word - MQ-9 New [Электронный ресурс]. – Электронные данные. – Режим доступа: https://files.amperka.ru/datasheets/MQ-9.pdf – Дата доступа: 22.09.2021

[11]. Подключение фоторезистора к ардуино и работа с датчиком освещенности [Электронный ресурс]. – Электронные данные. – Режим доступа: https://arduinomaster.ru/datchiki-arduino/photorezistor-arduino-datchik-sveta/ – Дата доступа: 24.09.2021

[12]. TEMP6000 — аналоговый датчик освещенности [Электронный ресурс]. – Электронные данные. – Режим доступа: https://voltiq.ru/shop/temt6000-light-sensor/ – Дата доступа: 25.09.2021

[13]. GY-302 BH1750 модуль освещения, Light Sensor для Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.avrobot.ru/product\_info.php?products\_id=4033 – Дата доступа: 25.09.2021

[14]. 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

[15]. 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

[16]. Ультразвуковой датчик расстояния, температуры и освещенности [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robot-kit.ru/3086/ – Дата доступа: 28.09.2021

[17]. NRF24L01 Datasheet (PDF) – List of Unclassified Manufacturers [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1243924/ETC1/NRF24L01.html> – Дата доступа: 01.10.2021

[18]. Беспроводной передатчик на 433 МГц [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/product/besprovodnoj-peredatchik-na-433-mgc – Дата доступа: 01.10.2021

[19]. Беспроводной приемник на 433 МГц [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/product/besprovodnoj-priemnik-na-433-mgc – Дата доступа: 01.10.2021

[20]. Драйвер двигателя L298N [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/wiki/arduino-moduli/drayver-dvigatelya-l298n/ – Дата доступа: 02.10.2021

[21]. Плата расширения L298P Motor Shield (для Arduino) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://radioprog.ru/shop/merch/41 – Дата доступа: 02.10.2021

[22]. Обзор драйвера мотора MX1508 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robotchip.ru/obzor-drayvera-motora-mx1508/ – Дата доступа: 02.10.2021

[23]. Мотор постоянного тока с редуктором 1:48 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/wiki/arduino-mechanics/motor-postoyannogo-toka-reduktorom-1-48/ – Дата доступа: 03.10.2021

[24]. Пищалка – пьезодинамик Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: https://arduinomaster.ru/uroki-arduino/pishhalka-pezodinamik-arduino/ – Дата доступа: 03.10.2021

[25]. Microsoft Word – MP3 – TF – 16P V1.0 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://picaxe.com/docs/spe033.pdf – Дата доступа: 03.10.2021

# ПРИЛОЖЕНИЕ А

(обязательное)

**Схема структурная**

# ПРИЛОЖЕНИЕ Б

(обязательное)

**Схема функциональная**

# ПРИЛОЖЕНИЕ В

(обязательное)

**Схема принципиальная**

# ПРИЛОЖЕНИЕ Г

(обязательное)

**Схема программы**

# ПРИЛОЖЕНИЕ Д

(обязательное)

**Листинг кода**

001. /\*

002. Скетч для передатчика

003. Подключите передатчик к 12 контакту Arduino

004. \*/

005.

006. #include <VirtualWire.h>

007.

008. #define pinX A2 // ось X джойстика

009. #define pinY A1 // ось Y джойстика

010.

011. const int MAX\_AXIS\_VALUE = 700;

012. const int MIN\_AXIS\_VALUE = 200;

013.

014. char FORWARD = '1';

015. char BACK = '2';

016. char RIGHT = '3';

017. char LEFT = '4';

018. char STOP = '5'

019.

020. char outputMessage[1];

021.

022. void setup() {

023. vw\_setup(2000);

024.

025. pinMode(pinX, INPUT);

026. pinMode(pinY, INPUT);

027. }

028.

029. void loop() {

030. int X = analogRead(pinX);

031. int Y = analogRead(pinY);

032. String output;

033.

034. if (X > MAX\_AXIS\_VALUE) {

035. output = FORWARD;

036. } else if (X < MIN\_AXIS\_VALUE) {

037. output = BACK;

038. } else if (Y > MAX\_AXIS\_VALUE) {

039. output = RIGHT;

040. } else if (Y < MIN\_AXIS\_VALUE) {

041. output = LEFT;

042. } else {

043. output = STOP;

044. }

045.

046. output.toCharArray(outputMessage, 2);

047. send(outputMessage);

048. }

049.

050. void send (char \*message) {

051. vw\_send((uint8\_t \*)message, strlen(message)); //отправка сообщения

052. vw\_wait\_tx(); // Ожидание полной отправки сообщения

053. }

054.

055.

056.

057. /\*

058. Скетч для приемника

059. Подключите приемник к 11 контакту Arduino

060. \*/

061.

062. #include <VirtualWire.h>

063. #include <NewPing.h>

064.

065. #define motor\_IN1 4

066. #define motor\_IN2 5

067. #define motor\_IN3 6

068. #define motor\_IN4 7

069.

070. #define headlights 13

071. #define side\_lights 12

072.

073. #define light\_sensor\_D0 A1

074.

075. #define trigPin 8

076. #define echoPin 9

077.

078. #define MQ5\_digitalSignal 2

079. #define MQ5\_analogSignal A0

080. #define soundPin 10

081.

082. byte message[VW\_MAX\_MESSAGE\_LEN]; // Массив входящих сообщений

083. byte messageLength = VW\_MAX\_MESSAGE\_LEN; // Размер массива входящих сообщений

084.

085. int stops;

086. boolean isGas; //переменная для хранения значения о присутствии газа

087. int gasValue = 0; //переменная для хранения количества газа

088.

089. int PERMISSIBLE\_GAS\_VALUE = 300;

090. int MIN\_DISTANCE\_BEFORE\_STOP = 12;

091.

092. char BACK = '1';

093. char FORWARD = '2';

094. char LEFT = '3';

095. char RIGHT = '4';

096.

097. void setup() {

098. vw\_setup(2000);

099. vw\_rx\_start(); // Активировать процесс приемника.

100.

101. pinMode(motor\_IN1, OUTPUT);

102. pinMode(motor\_IN2, OUTPUT);

103. pinMode(motor\_IN3, OUTPUT);

104. pinMode(motor\_IN4, OUTPUT);

105.

106. pinMode (light\_sensor\_D0, INPUT);

107. pinMode(headlights, OUTPUT);

108.

109. pinMode(trigPin, OUTPUT);

110. pinMode(echoPin, INPUT);

111. pinMode(side\_lights, OUTPUT);

112.

113. pinMode(MQ5\_digitalSignal, INPUT);

114. pinMode(soundPin, OUTPUT);

115. }

116.

117. void loop() {

118. checkIllumination();

119. checkDistanceToObject();

120. checkOfGas();

121.

122. if (vw\_get\_message(message, &messageLength)) {

123. selectStateOfMotors(message[0]);

124. }

125. }

126.

127. void checkIllumination() {

128. int xD0 = digitalRead (light\_sensor\_D0);

129.

130. if (xD0 == HIGH) {

131. digitalWrite (headlights, HIGH);

132. }

133. else {

134. digitalWrite (headlights, LOW);

135. }

136. }

137.

138. void checkDistanceToObject() {

139. int duration, cm;

140. digitalWrite(trigPin, LOW);

141. delayMicroseconds(2);

142.

143. digitalWrite(trigPin, HIGH);

144. delayMicroseconds(10);

145. digitalWrite(trigPin, LOW);

146.

147. duration = pulseIn(echoPin, HIGH);

148.

149. cm = duration / 58; // вычисляем расстояние в сантиметрах

150.

151. if (cm < MIN\_DISTANCE\_BEFORE\_STOP && cm > 0) {

152. stops = 1;

153. digitalWrite (side\_lights, HIGH);

154.

155. } else {

156. stops = 0;

157. digitalWrite (side\_lights, LOW);

158. }

159. }

160.

161. void checkOfGas() {

162. isGas = digitalRead(MQ5\_digitalSignal); //считываем значение о присутствии газа

163. gasValue = analogRead(MQ5\_analogSignal); // и о его количестве

164.

165. if (gasValue < PERMISSIBLE\_GAS\_VALUE) {

166. turnOffSound();

167. }

168. else {

169. turnOnSound();

170. }

171. }

172.

173. void turnOnSound() {

174. digitalWrite(soundPin, 50); // включаем пьезодинамик

175. }

176.

177. void turnOffSound() {

178. digitalWrite(soundPin, 0); // отключаем пьезодинамик

179. }

180.

181. void selectStateOfMotors(char state) {

182.

183. if (state == BACK) {

184. mooveBack();

185.

186. } else if (state == FORWARD && stops == 0) {

187. mooveForward();

188.

189. } else if (state == LEFT) {

190. turnLeft();

191.

192. } else if (state == RIGHT) {

193. turnRight();

194.

195. } else {

196. stand();

197. }

198. }

199.

200. void mooveForward() {

201. digitalWrite(motor\_IN1, LOW);

202. digitalWrite(motor\_IN2, HIGH);

203. digitalWrite(motor\_IN3, LOW);

204. digitalWrite(motor\_IN4, HIGH);

205. }

206.

207. void mooveBack() {

208. digitalWrite(motor\_IN1, HIGH);

209. digitalWrite(motor\_IN2, LOW);

210. digitalWrite(motor\_IN3, HIGH);

211. digitalWrite(motor\_IN4, LOW);

212. }

213.

214. void turnLeft() {

215. digitalWrite(motor\_IN1, LOW);

216. digitalWrite(motor\_IN2, HIGH);

217.

218. digitalWrite(motor\_IN3, HIGH);

219. digitalWrite(motor\_IN4, LOW);

220. }

221.

222. void turnRight() {

223. digitalWrite(motor\_IN1, HIGH);

224. digitalWrite(motor\_IN2, LOW);

225.

226. digitalWrite(motor\_IN3, LOW);

227. digitalWrite(motor\_IN4, HIGH);

228. }

229.

230. void stand() {

231. digitalWrite(motor\_IN1, LOW);

232. digitalWrite(motor\_IN2, LOW);

233. digitalWrite(motor\_IN3, LOW);

234. digitalWrite(motor\_IN4, LOW);

235. }

# ПРИЛОЖЕНИЕ Е

(обязательное)

**Перечень элементов**

# ПРИЛОЖЕНИЕ Ж

(обязательное)

**Ведомость документов**