

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

МИКРОПРОЦЕССОРНОЕ РАДИОУПРАВЛЯЕМОЕ ПЕРЕДВИЖНОЕ
УСТРОЙСТВО

БГУИР КП 1-40 02 01 104 ПЗ

Студент: группы 950501,
Деркач А. В.

Руководитель: доцент каф. ЭВМ
Селезнев И. Л.

Минск 2021

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 Деркач Анжелике Валерьевне
(фамилия, имя, отчество)

1. Тема проекта Микропроцессорное радиоуправляемое передвижное устройство

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 16 МГц, не менее 14 входов/выходов.
2. Источник питания – напряжение 5 В, максимальный выходной ток не менее 3 А.
3. Модуль радиопередачи – напряжение питания 3.3 – 5 В, скорость передачи не менее 9 Кб/с.
4. Датчик горючих газов – напряжение питания 5 В, диапазон измерений 300 – 10000 ppm.
5. Датчик расстояния – напряжение питания 5 В, точность до 0.3 см.
6. Датчик освещенности – напряжение питания 3.3 – 5 В, измеряемые значения от 0 до 65535 лк.
7. Драйвер моторов – напряжение питания 5 В, возможность подключения до 4х моторов.
8. Мотор-редукторы – напряжение питания 3.3 – 5 В, передаточное число 1:48.

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 микропроцессорного радиуправляемого передвижного устройства (формат А4)

2. Функциональная электрическая схема микропроцессорного радиуправляемого передвижного устройства (формат А3)

3. Принципиальная электрическая схема микропроцессорного радиуправляемого передвижного устройства (формат А3)

6. Консультант по проекту (с назначением разделов проекта) И.Л. Селезнёв

7. Дата выдачи задания 10.09.2021

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 _____ доцент каф. ЭВМ Селезнёв И.Л.
(подпись)

Задание принял к исполнению 10.09.2021

А. В. Деркач
(дата и подпись студента)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 ОБЗОР ЛИТЕРАТУРЫ	5
1.1 Состав устройства.....	5
1.2 Микроконтроллеры	5
1.3 Датчики горючих газов	6
1.4 Датчики освещенности.....	6
1.5 Ультразвуковые датчики расстояния.....	7
1.6 Модули радиопередачи	7
1.7 Драйверы моторов	8
1.8 Мотор-редукторы.....	9
1.9 Модули воспроизведения звука	9
2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА	10
2.1 Постановка задачи	10
2.2 Определение компонентов структуры устройства.....	10
2.3 Взаимодействие компонентов устройства	11
3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА.....	12
3.1 Обоснование выбора микроконтроллеров	12
3.2 Обоснование выбора датчика горючих газов	12
3.3 Обоснование выбора датчика освещенности.....	12
3.4 Обоснование выбора ультразвукового датчика расстояния.....	13
3.5 Обоснование выбора модулей радиопередачи	13
3.6 Обоснование выбора драйвера моторов	13
3.7 Обоснование выбора мотор-редукторов	14
3.8 Обоснование выбора модуля воспроизведения звука.....	14
4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА	15
4.1 Расчёт мощности элементов схемы	15
4.2 Расчёт нагрузки светодиодов.....	16
4.3 Микроконтроллеры	17
4.4 Датчик горючих газов	17
4.5 Датчик освещенности.....	17
4.6 Ультразвуковой датчик расстояния	18
4.7 Модуль радиопередачи	18
4.8 Драйвер моторов.....	18
4.9 Мотор-редукторы	18
4.10 Пьезодинамик	19
4.11 Джойстики.....	19
5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	20
5.1 Требования к разработке программного обеспечения	20
5.2 Блок-схема алгоритма	20
5.3 Исходный код программы для устройства управления	21
5.4 Исходный код программы для передвижного устройства	21
ЗАКЛЮЧЕНИЕ.....	23
ПРИЛОЖЕНИЕ А	26
ПРИЛОЖЕНИЕ Б	27
ПРИЛОЖЕНИЕ В.....	28
ПРИЛОЖЕНИЕ Г	29
ПРИЛОЖЕНИЕ Д.....	30
ПРИЛОЖЕНИЕ Е.....	35
ПРИЛОЖЕНИЕ Ж.....	36

ВВЕДЕНИЕ

Темой данного курсового проекта является разработка микропроцессорного устройства на базе микроконтроллера. Для реализации было выбрано радиоуправляемое передвижное устройство.

Данная тема достаточно широка, поэтому первым делом необходимо ограничить ее. Под устройством понимается малогабаритное устройство, осуществляющее передвижение по суше посредством регулирования тяговых усилий колес. Так как данное устройство само по себе не несет за собой никаких полезных действий кроме передвижения, ему необходимы датчики, которые и будут осуществлять необходимые функции.

В настоящее время имеется много опасных профессий, среда работы в которых может серьезно навредить здоровью человека. Причиной может служить наличие горючих газов в среде, о которых человек может даже не догадываться. Исходя из этого, в данном устройстве будет установлен датчик горючих газов, который будет сигнализировать об опасности на безопасном расстоянии.

Для более эффективной и удобной работы на устройство также будут установлены датчик освещенности и ультразвуковой датчик расстояния. Датчик освещенности позволит работать в условиях пониженной видимости, а ультразвуковой датчик расстояния снизит вероятность повреждения устройства во время эксплуатации.

В качестве основного вычислительного элемента устройства будет использована плата Arduino UNO на базе микроконтроллера ATmega328. Выбор в сторону данного контроллера был произведен в связи с наличием таких плат и основных элементов для их работы, а также опыта разработки программного обеспечения для данного типа микроконтроллеров. 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. В конце устройство подлежит тестированию, чтобы проверить правильность сборки и исключить сбои при эксплуатации.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Состав устройства

Как сказано ранее, разрабатываемое микропроцессорное устройство выполняет функции передвижения, которое управляется посредством радиосигнала, измерения концентрации горючих газов в среде, определения степени освещенности рабочей зоны, измерения расстояния до объектов, а также информирует пользователя о всех измеряемых параметрах. Для решения этих задач в состав устройства должны входить:

- микроконтроллер
- датчик горючих газов
- датчик освещенности
- ультразвуковой датчик расстояния
- модуль радиопередачи
- модуль воспроизведения звука
- драйвер моторов с моторами
- светодиоды для индикации
- устройства управления

1.2 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 2	OLIMEXINO-STM32
Микроконтроллер	ATmega328	ARM Cortex-A7	stm32f103rht6
Входное напряжение	6 – 20 В	6 – 28 В	9 – 30 В
Флэш-память	32 Кб	порт для microCD	128 Кб
ОЗУ	2 Кб	1024 Мб	20 Кб
Тактовая частота	16 МГц	900 МГц	72 МГц
Разрядность	8 бит	32 бит	12 бит
Цифровые входы/выходы	14 шт	26 шт	15 шт
Аналоговые входы/выходы	6 шт	0 шт	6 шт
Выходное напряжение	3.3В, 5 В	3.3В, 5 В	3.3В, 5 В

Продолжение таблицы 1.1

Рабочая температура	от -25 до +85 °C	от -40 до +85 °C	от -25 до +85 °C
Встроенный видеочип	нету	есть	нету
Размеры	69 мм × 53 мм	85.6 мм × 56.5 мм	101.6 мм × 86 мм

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

1.3 Датчики горючих газов

На сегодняшний день можно найти большое количество различных датчиков газов. Из всего множества для сравнения были выбраны три наиболее подходящих (MQ-5, MQ-9 и MQ-2). Результаты полученного сравнения приведены в таблице 1.2.

Таблица 1.2 — Сравнение датчиков горючих газов

Параметры сравнения	MQ-5	MQ-9	MQ-2
Детектируемые газы	метан, пропан, бутан, H ₂ , коксовый газ	угарный газ, метан, пропан, бутан	метан, пропан, н-бутан, H ₂ , дым
Напряжение питания	5 В	5 В	5 В
Потребляемый ток	до 150 мА	до 150 мА	до 150 мА
Рабочая температура	от -20 до +50 °C	от -20 до +50 °C	от -20 до +50 °C
Диапазон измерений	300 – 10000 ppm	500 – 10000 ppm	300 – 5000 ppm

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

1.4 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 1.3 приведены их сравнительные характеристики.

Таблица 1.3 — Сравнение датчиков освещенности

Параметры сравнения	Фоторезистивный датчик освещенности	Датчик освещенности CJMCU-TEMT6000	Модуль освещения GY-302
Угол чувствительности	±60°	±60°	±60°

Продолжение таблицы 1.3

Напряжение питания	3.3 – 5 В	3.3 – 5 В	3 – 5 В
Рабочая температура	от 0 до +70 °С	от -40 до +85 °С	от -25 до +85 °С
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 характеристик о данных датчиках использовалась источники [11, 12, 13].

1.5 Ультразвуковые датчики расстояния

Наиболее распространенными ультразвуковыми датчиками расстояния являются модели HC-SR04, DYP-ME007 и RKP-KS103. Они работают по одному принципу, а их детальное сравнение приведено в таблице 1.4.

Таблица 1.4 — Сравнение ультразвуковых датчиков расстояния

Параметры сравнения	HC-SR04	DYP-ME007	RKP-KS103
Частота звукового излучения	40 кГц	40 кГц	40 кГц
Напряжение питания	5 В	5 В	3 – 5.5 В
Потребляемый ток	до 15 мА	до 15 мА	до 10 мА
Рабочая температура	от -30 до +80 °С	от -15 до +60 °С	от -30 до +85 °С
Диапазон измеряемого расстояния	от 2 до 450 см	от 3 до 400 см	от 1 до 550 см
Рабочий угол наблюдения	30°	30°	30°
Точность	0.3 см	0.3 см	1 см

Для получения точной информации о данных датчиках использовалась техническая спецификация [14, 15] и источник [16].

1.6 Модули радиопередачи

Существуют модули с однонаправленной и двунаправленной передачей радиосигнала. Для сравнения был выбран радио модуль NRF24L01 с двунаправленной передачей и передатчик MX-05V с приёмником XD-RF-5V, связь между которыми проходит от передатчика к приёмнику соответственно. Ниже, в таблице 1.5, приведены их сравнительные характеристики.

Таблица 1.5 — Сравнение модулей радиопередачи

Параметры сравнения	NRF24L01	передатчик MX-05V	приёмник XD-RF-5V
Частота радиосигнала	2.4 ГГц	433 МГц	433 МГц
Напряжение питания	1.9 – 3.6 В	3.5 – 12 В	5 В
Потребляемый ток	до 14 мА	до 28 мА	до 6 мА
Рабочая температура	от -40 до +85 °С	от -20 до +85 °С	от -20 до +85 °С
Скорость приёма/передачи данных	до 2 Мб/с	до 10 Кб/с	9.6 Кб/с
Расстояние приёма/передачи	до 100 м	до 200 м	до 200 м

Для получения информации о данных модулях использовалась техническая спецификация [17] и источники [18, 19].

1.7 Драйверы моторов

Драйвера моторов являются неотъемлемой частью при проектировании передвижных устройств. Благодаря драйверам регулируется питание, поступающее на моторы, скорость и направление их вращения.

Для сравнения были выбраны модули L298N, L298P и более простой и новый модуль MX1508, спроектированный на базе модуля L298N. Результаты сравнения представлены в таблице 1.6.

Таблица 1.6 — Сравнение драйверов моторов

Параметры сравнения	L298N	L298P	MX1508
Напряжение питания логики	5 В	5 В	5 В
Потребляемый ток логики	до 36 мА	до 36 мА	до 36 мА
Напряжение питания приводов	5 – 35 В	5 – 25 В	2 – 10 В
Потребляемый ток приводов	до 3 А	до 2 А	до 2 А
Рабочая температура	от -25 до +135 °С	от -25 до +130 °С	от -40 до +80 °С
Размер	43.5 × 43.2 × 29.4 мм	58 × 56 × 19 мм	24.7 × 21 × 5 мм

Для получения информации о данных драйверах использовались источники [20, 21, 22].

1.8 Мотор-редукторы

Существует множество видов мотор-редукторов, которые отличаются по форме и характеристикам. По форме они разделяются на 4 типа: прямой одноосевой, прямой двухосевой, угловой одноосевой и угловой двухосевой.

Для сборки проекта будут использоваться прямые двухосевые мотор-редукторы с передаточным числом 1:48, которых на рынке представлено только два. Их сравнительные характеристики приведены в таблице 1.7.

Таблица 1.7 — Сравнение прямых двухосевых мотор-редукторов

Параметры сравнения	Мотор-редуктор 1:48 3-8V	Мотор-редуктор 1:48 6V
Напряжение питания	3 – 8 В	6 В
Потребляемый ток	до 700 мА	до 670 мА
Передаточное число	1:48	1:48
Скорость вращения без нагрузки	до 200 об/мин	до 180 об/мин
Крутящий момент	2 кг/см	2 кг/см

Для получения информации о мотор-редукторах использовался источник [23].

1.9 Модули воспроизведения звука

Для сравнения был выбран модуль DFPlayer MP3 mini, который представляет собой проигрыватель музыки с возможностью воспроизведения записанных мелодий на microSD, и обычный пьезодинамик. Результаты сравнения приведены в таблице 1.8.

Таблица 1.8 — Сравнение модулей воспроизведения звука

Параметры сравнения	Пьезодинамик	DFPlayer MP3 mini
Напряжение питания	3.3 – 5 В	3.3 – 5 В
Потребляемый ток	до 10 мА	до 250 мА
Рабочая температура	от -20 до +70 °C	от -40 до +80 °C
Поддерживаемые файловые системы	-	FAT16, FAT32
Форматы аудиофайлов	-	MP3, WAV, WMA

Для получения информации о данных модулях использовались источники [24, 25].

2 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 радиоуправляемое передвижное устройство. Для реализации было выбрано устройство, осуществляющее передвижение по суше и оповещающее об опасности при утечке газа. Исходя из этого, были выделены следующие функции, которые должно выполнять данное устройство:

- передвижение по суше и изменение направления движения путем регулирования тяговых усилий колес
- управление с помощью дистанционного пульта посредством радиосигнала
- получение информации о наличии горючих газов в окружающей среде
- звуковое оповещение об опасности
- получение информации о степени освещенности рабочей зоны и включение собственного освещения при недостаточном количестве первого
- считывание расстояния до ближайшего объекта и остановка, если расстояние довольно мало
- световая индикация о состоянии

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 управляющие сигналы.

2) Модуль питания — стабилизатор напряжения и источник питания схемы.

3) Модуль управления — два двухосевых джойстика, которые управляют направлением и движением устройства.

4) Модуль радиопередачи — передатчик, который передает информацию на радиоприёмник.

5) Модуль радиоприёма — приёмник, который получает информацию с радиопередатчика.

6) Модуль передвижения — моторы-редукторы и драйвер моторов для данных редукторов, которые реализуют передвижение по суше.

7) Модуль оповещения — модуль, который оповещает об опасности посредством звукового сигнала.

8) Модуль индикации — светодиоды, которые выполняют функцию индикации состояния устройства.

9) Датчик горючих газов — датчик, считывающий информацию о концентрации горючих газов в окружающей среде.

10) Датчик освещенности — датчик, считывающий информацию о степени освещенности рабочего пространства устройства.

11) Датчик расстояния — датчик, анализирующий расстояние до ближайшего объекта.

2.3 Взаимодействие компонентов устройства

Устройство последовательно считывает информацию со всех датчиков, затем эти данные передаются на контроллер, который их анализирует.

Контроллер сравнивает допустимые значения и полученные значения с датчика горючих газов и, при превышении, подает сигнал на модуль оповещения, который включает звуковой сигнал.

Затем контроллер анализирует данные с датчика освещенности. При недостаточном освещении подается сигнал на модуль индикации, который включает светодиоды, реализующие собственное освещение устройства.

Следующим анализируется расстояние до объекта, информация о котором поступает с датчика расстояния. При недостаточной дальности контроллер передает сигнал драйверу моторов, который выполняет остановку устройства, даже, если подан активный сигнал на разрешение движения, а также сигнал модулю индикации, который включает соответствующие светодиоды.

Пользователь дистанционно взаимодействует с устройством посредством пульта, на котором установлены двухосевые джойстики, используя модуль радиопередачи. При перемещении соответствующих джойстиков в сторону, в которую будет осуществляться движение, контроллер, установленный на пульте, считывает выставленное направление и передает информацию на радиопередатчик. Радиопередатчик, в свою очередь, отправляет информацию на радиоприёмник, после чего контроллер, установленный на устройстве, анализирует полученную информацию и, при определенных значениях, отправляет сигнал драйверу моторов, который запускает вращение двигателей в нужную сторону.

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

3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

3.1 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. Контроллеры Raspberry PI 2 и OLIMEXINO-STM32 превосходят Arduino UNO по памяти и мощности, но требуют для работы больше входного напряжения и имеют большую стоимость. Так же у Raspberry PI 2 нет аналоговых входов/выходов, которые могут пригодиться при сборке проекта.

Исходя из вышеперечисленного в данном 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 горючих газов

Рассмотрев сравнительные характеристики датчиков горючих газов, представленных в таблице 1.2 обзора литературы, видно, что все датчики имеют одинаковое напряжение питания, потребляемый ток и рабочую температуру. Было принято решение выбрать датчик MQ-5 так как он имеет больший диапазон измерения и наиболее подходящие в данном проекте детектируемые газы.

3.3 Обоснование выбора датчика освещенности

В таблице 1.3 обзора литературы приведены сравнения наиболее распространенных датчиков освещенности. Как видно из сравнения, все они имеют схожие характеристики и отличаются типом выходного сигнала и потребляемым током, а также рабочей температурой. Для работы устройства подойдут диапазоны рабочей температуры всех датчиков так как само устройство не проектируется для работы в экстремальных условиях, поэтому данный параметр при выборе датчиков можно упустить. 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.

3.4 Обоснование выбора ультразвукового датчика расстояния

После сравнения ультразвуковых датчиков расстояния (см. таблицу 1.4) при выборе можно исключить датчик RKP-KS103 так как он имеет более низкую точность по сравнению с остальными моделями, что является очень важным показателем.

Проанализировав оставшиеся два датчика (HC-SR04 и DYP-ME007), 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.

3.5 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 MX-05V и приемник XD-RF-5V. Хотя организация радиопередачи между этими двумя модулями является однонаправленной, 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.

3.6 Обоснование выбора драйвера моторов

В таблице 1.6 обзора литературы приведены сравнения драйверов 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.

3.7 Обоснование выбора мотор-редукторов

В данном устройстве будут использованы прямые двухосевые мотор-редукторы с передаточным числом 1:48. Характеристики сравниваемых редукторов приведены в таблице 1.7 обзора литературы. Исходя из данного сравнения были выбраны мотор-редукторы 1:48 3-8V так как они имеют более подходящий диапазон напряжения питания и более высокую скорость вращения.

3.8 Обоснование выбора модуля воспроизведения звука

В качестве модуля воспроизведения звука в данном курсовом проекте будет использован пьезодинамик. Его характеристики, а также характеристики DFPlayer MP3 mini представлены в таблице 1.8 обзора литературы. Данный модуль был выбран по причине потребления меньшего тока и простоте.

Модуль DFPlayer MP3 mini определенно имеет больший функционал, но в данном проекте его наличие не обязательно так как модуль воспроизведения звука нужен только для оповещения о наличии вредных газов в среде, с чем хорошо справляется пьезодинамик.

4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА

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 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:

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

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

Рассчитаем потребляемый ток:

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

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 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:

$$P = 5 \cdot 22 + 5 \cdot 6 + 5 \cdot 150 + 5 \cdot 15 + 5 \cdot 15 + 5 \cdot 36 + 5 \cdot 600 \cdot 4 + 5 \cdot 10 + 5 \cdot 20 \cdot 4 = 13670 \text{ мВт.}$$

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

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{16.404}{5} = 3.2808 \approx 3.28 \text{ А}$$

4.2 Расчёт нагрузки светодиодов

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

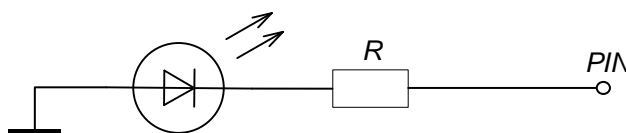


Рисунок 4.2 — Схема подключения светодиодов

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

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

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

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

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

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 обеих схемах питается от напряжения 5 В.

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 микроконтроллера. 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.

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 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 информации, то оно работает в обычном режиме без индикации.

5.2 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 логических 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.

5.3 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 обеспечения 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.

5.4 Исходный код программы для передвижного устройства

Данное устройство является основным в разрабатываемом курсовом проекте и здесь реализована большая часть логики. Исходный код программного обеспечения 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 checkIllumination()` (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 посылаются 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 с.
- [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 – 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://radioprogram.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-pezdinamik-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 со-
общений
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 присут-
ствии 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