

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

ТАНК НА ВСЕНАПРАВЛЕННЫХ КОЛЕСАХ

БГУИР КП 1-40 02 01 428 ПЗ

Студент: группы 150504,
Шалаш А. С.

Руководитель: ассистент каф. ЭВМ,
Калютчик А. А.

Минск 2023

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ

(подпись)

« _____ » _____ 2023 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту Шалаш Анне Сергеевне
(фамилия, имя, отчество)

1. Тема проекта Танк на всенаправленных колесах

2. Срок сдачи студентом законченного проекта с 06.12.2023 по 09.12.2023

3. Исходные данные к проекту:

1. Микроконтроллер – тактовая частота не менее 16 МГц, не менее 14 входов/выходов.
2. Источник питания – напряжение 9 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. Драйвер двигателей – напряжение питания 5 В, возможность подключения до 2х двигателей.
5. Двигатель постоянного тока – напряжение питания 3.3 – 5 В, передаточное число 1:48.
6. Контроллер с камерой – напряжение питания 3.3 – 5 В
7. Светодиоды – напряжение питания 2 – 3.4 В
8. Шаговый двигатель – напряжение питания 5 В
9. Драйвер шагового двигателя – напряжение питания 5 – 12 В
10. ШИМ контроллер – I2C интерфейс

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

Введение

1. Обзор литературы.
2. Разработка структуры танка на всенаправленных колесах.
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 танка на всенаправленных колесах (формат А4)

2. Функциональная электрическая схема танка на всенаправленных колесах (формат А4)

3. Принципиальная электрическая схема танка на всенаправленных колесах (формат А5)

6. Консультант по проекту (с назначением разделов проекта) А.А. Калютчик

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 А. А. Калютчик
(подпись)

Задание принял к исполнению 10.09.2021 А. С. Шалаш
(дата и подпись студента)

Оглавление

ВВЕДЕНИЕ	6
1 ОБЗОР ЛИТЕРАТУРЫ.....	8
1.1 Состав устройства.....	8
1.2 Микроконтроллеры.....	8
1.3 Модули радиопередачи	9
1.4 Драйверы двигателей.....	9
1.5 Двигатели постоянного тока.....	10
1.6 Камера.....	11
1.7 Светодиоды	11
1.8 Источник питания.....	12
1.9 Шаговый двигатель	13
1.10 Драйвер шагового двигателя	13
1.11 ШИМ контроллер	14
2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА.....	15
2.1 Постановка задачи.....	15
2.2 Определение компонентов структуры устройства.....	15
2.3 Взаимодействие компонентов устройства	16
3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА.....	17
3.1 Обоснование выбора микроконтроллеров	17
3.2 Обоснование выбора модулей радиопередачи	17
3.3 Обоснование выбора источника питания	18
3.4 Обоснование выбора светодиодов	18
3.5 Обоснование выбора камеры.....	20
3.6 Обоснование выбора драйвера двигателей	20
3.7 Обоснование выбора двигателей постоянного тока.....	21
3.8 Обоснование выбора шагового двигателя.....	22
3.9 Обоснование выбора драйвера шагового двигателя	22
3.10 Обоснование выбора ШИМ контроллера	23
4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА ..	24
4.1 Расчёт мощности элементов схемы	24
4.2 Расчёт нагрузки светодиодов.....	25
4.3 Микроконтроллеры	25
4.4 Камера.....	26
4.5 Радиомодуль.....	26
4.6 Шаговый двигатель	26
4.7 Драйвер шагового двигателя	27
4.8 Драйвер двигателей	27
4.9 Расчет нагрузки двигателей постоянного тока.....	27
4.10 ШИМ контроллер	28
5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	29
5.1 Требования к разработке программного обеспечения	29
5.2 Блок-схема алгоритма	29
5.3 Исходный код программы для танка на всенаправленных колесах	30
5.4 Исходный код программы для камеры.....	31
ЗАКЛЮЧЕНИЕ.....	32
ПРИЛОЖЕНИЕ А	34
ПРИЛОЖЕНИЕ Б	35
ПРИЛОЖЕНИЕ В.....	36
ПРИЛОЖЕНИЕ Г	37

ПРИЛОЖЕНИЕ Д.....	38
ПРИЛОЖЕНИЕ Е.....	46
ПРИЛОЖЕНИЕ Ж.....	47

ВВЕДЕНИЕ

Данный курсовой проект посвящен разработке танка на всенаправленных колесах, что представляет собой исследование в области мобильной робототехники и схемотехники. Основываясь на этой обширной теме, ставится задача сделать проект более конкретным и целенаправленным.

Одним из первостепенных шагов в этом процессе является установление спецификаций для нашего мобильного устройства. Для эффективного управления танком, будет использовано радиоуправление. Это позволит удаленно контролировать движение танка и его башню, делая его гибким средством для различных задач.

С учетом темы проекта, особое внимание будет уделено башне танка. Она будет оснащена механизмом, обеспечивающим ее подвижность, что позволит управлять направлением обзора и цели танка. Это важное дополнение к функциональности устройства, особенно в контексте различных приложений, включая исследовательские миссии и развлечения.

Таким образом, проект объединяет современные технологии, инженерное искусство и креативный дизайн, создавая возможность для разработки уникального и многофункционального мобильного танка на всенаправленных колесах.

Стоит отметить особенность проекта — передвижение танка на всенаправленных колесах. Всенаправленные колеса — это инновационная концепция колесного движения, которая позволяет объекту перемещаться в любом направлении без необходимости поворота. Они состоят из нескольких роликов, установленных под углами к основной оси колеса, и позволяют объекту двигаться вперед, назад, вбок и даже поворачивать на месте. Всенаправленные колеса нашли широкое применение в различных областях, включая мобильную робототехнику, транспортные системы и материально-техническое обеспечение. Их уникальная способность обеспечивать максимальную маневренность и точное управление делает их незаменимыми для современных автономных и управляемых транспортных средств, а также в роботах и механизированных системах.

В рамках проекта будет предусмотрена интеграция дополнительных компонентов для дополнительных функциональных возможностей танка на всенаправленных колесах. Одним из ключевых элементов будет контроллер, оборудованный камерой, способной передавать видеоизображение по Wi-Fi и Bluetooth. Это позволит не только контролировать движение танка, но и получать в реальном времени видеопоток с его "глазами". Такая система станет незаменимой для наблюдения в труднодоступных местах или для дистанционного исследования окружающей среды.

Дополнительно, также планируется использовать светодиоды, которые будут предназначены для обеспечения работы камеры в условиях пониженной видимости, таких как ночное время или места с недостаточной освещенностью. Эти светодиоды помогут улучшить качество видеозаписи и

расширить спектр применения танка, делая его более универсальным средством для наблюдения и исследования.

Кроме того, следует отметить, что камера будет размещена на специальной подвижной башне танка вместо боевого механизма. Это дизайнерское решение позволит максимально обеспечить обзорность и маневренность танка, что особенно важно для эффективного использования машины в различных ситуациях, включая исследования, мониторинг и развлечения.

В качестве основного вычислительного элемента устройства будет использована плата Arduino UNO на базе микроконтроллера ATmega328, также для увеличения недостающего количества входов будет подключен ШИМ контроллер. 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 загрузкой 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

Проект представляет собой мобильное устройство в форме танка, оснащенное всенаправленными колесами и разнообразными функциями. Это включает в себя передвижение, управление которым осуществляется посредством радиосигнала, видеозапись с использованием встроенной камеры, а также обеспечение подсветки для работы в условиях пониженной видимости.

Для успешной реализации этих функций, устройство должно включать в себя ряд ключевых компонентов:

- микроконтроллер
- модуль радиопередачи
- драйвер двигателей с двигателями
- контроллер с камерой
- светодиоды
- источник питания
- драйвер шагового двигателя с шаговым двигателем
- ШИМ контроллер

1.2 Микроконтроллеры

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

Таблица 1.1 — Сравнение микроконтроллеров

Параметры сравнения	Arduino UNO	Raspberry PI 2	OLIMEXINO-STM32
Микроконтроллер	ATmega328	Broadcom BCM2711	stm32f103rbt6
Входное напряжение	6 – 20 В	5 В	9 – 30 В
Флэш-память	32 Кб	2/4/8 гб	128 Кб
ОЗУ	2 Кб	2/4/8 гб	20 Кб
Тактовая частота	16 МГц	1.5 ГГц	72 МГц
Разрядность	8 бит	64 бит	12 бит
Цифровые входы/выходы	14 шт	40 шт	15 шт
Аналоговые входы/выходы	6 шт	0 шт	6 шт

Продолжение таблицы 1.1

Выходное напряжение	3.3В, 5 В	5 В	3.3В, 5 В
Рабочая температура	от -25 до +85 °С	от -40 до +85 °С	от -25 до +85 °С
Встроенный видеочип	нету	есть	нету
Размеры	69 мм × 53 мм	85.6 мм × 56.5 мм	101.6 мм × 86 мм

1.3 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 1.2, приведены их сравнительные характеристики.

Таблица 1.2 — 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

1.4 Драйверы двигателей

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

Для сравнения были выбраны модули L9110S, L298P и более простой и новый модуль MX1508, спроектированный на базе модуля L298N. Результаты сравнения представлены в таблице 1.3.

Таблица 1.3 — Сравнение драйверов моторов

Параметры сравнения	L9110S	L298P	MX1508
Напряжение питания логики	2.5 – 12 В	5 В	5 В

Продолжение таблицы 1.3

Потребляемый ток логики	до 36 мА	до 36 мА	до 36 мА
Напряжение питания приводов	2.5 – 12 В	5 – 25 В	2 – 10 В
Потребляемый ток приводов	до 0.8 А	до 2 А	до 2 А
Рабочая температура	от -25 до +125 °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

1.5 Двигатели постоянного тока

При выборе двигателя для проекта на Arduino важно учесть его характеристики, такие как точность позиционирования, скорость вращения, управляемость и применимость к конкретным задачам. Рынок предлагает разнообразие моторов, каждый из которых обладает уникальными особенностями, подходящими для различных целей.

Были рассмотрены различные типы двигателей постоянного тока, такие как DC Brush Motors, Stepper Motors, Servo Motors и BLDC Motors, их основные характеристики: способ управления, точность позиционирования, скорость, область применения, сложность управления и другие параметры. Их сравнительные характеристики приведены в таблице 1.4.

Таблица 1.4 — Сравнение прямых двухосевых мотор-редукторов

Параметры сравнения	DC Brush Motors	Stepper Motors	Servo Motors	BLDC Motors
Управление	Простое	Точное	Угол поворота	Электронное
Точность позиционирования	Обычно низкая	Высокая	Высокая	Высокая
Скорость	Средняя до высокой	Обычно средняя	Ограниченная	Высокая
Применение	Широкий спектр	3D-принтеры, ЧПУ-станки	Радиоуправляемые модели	Автомобильная промышленность, электроника
Сложность управления	Простое	Среднее	Простое	Среднее
Цена	Низкая	Средняя	Низкая	Высокая
Эффективность	Обычно низкая	Обычно средняя	Средняя	Высокая

1.6 Камера

Камеры открывают перед разработчиками и инженерами множество возможностей для создания интересных и инновационных проектов. С их помощью можно осуществлять видеозапись, проводить анализ окружающей среды, создавать автономные системы и реализовывать различные видео- и фото-приложения.

Для сравнения были выбраны следующие модели: ESP32-CAM, OV7670, ArduCam Mini и Raspberry Pi Camera Module. Их сравнительные характеристики приведены в таблице 1.5.

Таблица 1.5 — Сравнение моделей камер

Параметры сравнения	ESP32-CAM	OV7670	ArduCam Mini	Raspberry Pi Camera Module
Разрешение	VGA (640x480), QVGA (320x240), QQVGA (160x120)	640x480	До 5 МП	До 12 МП
Интерфейс	SPI для подключения к плате ESP32	Вывод видеосигнала по параллельному интерфейсу	интерфейс I2C или SPI	интерфейс CSI
Скорость съемки	60 кадров/с	до 30 кадров/с	Зависит от модели, но обычно высокая	Зависит от модели, но обычно высокая

1.7 Светодиоды

Светодиоды являются одними из наиболее популярных и важных компонентов в мире электроники и микроконтроллеров. Они предоставляют простой и эффективный способ создания визуальных индикаторов, сигнализации и декоративных эффектов. Светодиоды доступны в разнообразных моделях, цветах и конфигурациях, что делает их идеальным выбором для широкого спектра приложений.

Для сравнения были выбраны следующие типы: One-Color LEDs, RGB LEDs, Super Bright LEDs и PWM LEDs. Их сравнительные характеристики приведены в таблице 1.6.

Таблица 1.6 — Сравнение типов светодиодов

Параметры сравнения	One-Color LEDs	RGB LEDs	Super Bright LEDs	PWM LEDs
Цвет	в одном цвете	красного, зеленого и синего элементов	одноцветные или RGB	одноцветные или RGB
Яркость	Стандарт	Стандартные и сверхяркие версии	высокая яркость и видимость на дистанции	стандартные или сверхяркие
Напряжение	2-3 В	2-3 В	более высокое напряжение и ток по сравнению с обычными светодиодами	разные значения в зависимости от модели
Размер и форма	5 мм и 3 мм, в миниатюрных и SMD-корпусах	Доступны в различных размерах	стандарт	Варьируется в зависимости от типа

1.8 Источник питания

Источники питания играют важную роль в обеспечении электропитания электронных устройств. Важность надежного источника питания не может быть недооценена, так как он обеспечивает нормальное функционирование и долгий срок службы многих современных технологий.

Наиболее распространёнными и доступными источниками питания являются батарейки LR-6 (тип AA), литий-ионные (Li-ion) аккумуляторы и литий-полимерные (Li-Po) аккумуляторы. В таблице 1.7 приведены их сравнительные характеристики.

Таблица 1.7 — Сравнение источников питания

Параметры сравнения	Mirex 6F22	Li-ion	Li-Po
Тип химии	Щелочная	Литий-ион	Литий-полимер
Емкость	1000-2980 мАч	2000 - 3600 мАч	До 4200 мАч
Циклы зарядки и разрядки	Одноразовая	300-500 циклов	300-500 циклов
Саморазряд (ежемесячно)	1-2%	2-3%	2-3%
Рабочая температура	от -20 до +60 °C	от -20 до +60 °C	от -20 до +60 °C
Номинальное электрическое напряжение	9 В	3,6 – 3,8 В	3,7 В

1.9 Шаговый двигатель

Шаговые двигатели обеспечивают точное перемещение объектов на определенные расстояния, деля их на шаги. Используются в робототехнике, производстве и 3D-принтерах для точного управления движением..

Для сравнения были выбраны следующие модели: 28YBJ-48, 28BYJ-45 и 28BYJ-50. Их сравнительные характеристики приведены в таблице 1.8.

Таблица 1.8 — Сравнение шаговых двигателей

Параметры сравнения	28YBJ-48	28BYJ-45	28BYJ-50
Шаговый угол	5.625 градусов на шаг	5.625 градусов на шаг	5.625 градусов на шаг
Напряжение питания	5 – 12 В	5 В	5 В
Ток обмотки	низкий	низкий	низкий
Крутящий момент	невысокий	невысокий	невысокий
Размеры и фактор	компактные размеры	компактные размеры	компактные размеры

1.10 Драйвер шагового двигателя

Драйверы шаговых двигателей — ULN2003, A4988 и L298N являются ключевыми компонентами в системах управления двигателями, предоставляя возможность точного управления и контроля за шаговыми моторами. Каждый из них обладает своими уникальными характеристиками, которые определяют их эффективность, надежность и область применения.

Теперь проведем сравнение этих драйверов, выявив их особенности, преимущества и возможные сферы применения для более точного выбора в таблице 1.9.

Таблица 1.9 — Сравнение источников питания

Параметры сравнения	ULN2003	A4988	L298N
Напряжение питания	5 – 12 В	8 – 35 В	5 – 35 В
Способность управления	управление до 7 выводов	управление микрошагами	управление двумя двигателями, направлением и скоростью
Ток усилителя	до 500 мА	до 2А	до 2А

Продолжение таблицы 1.9

Защитные функции	защита от обратной полярности и помех	защита от тепловых перегрузок и коротких замыканий	встроенные диоды для защиты от обратной полярности, имеет тепловую защиту
------------------	---------------------------------------	--	---

1.11 ШИМ контроллер

Контроллеры ШИМ (PWM) являются важным элементом для управления сервоприводами, светодиодами и другими устройствами, требующими точного и гибкого управления сигналами. В контексте курсового проекта необходимость восполнения недостатка ШИМ входов и управления светодиодами, 16-канальный ШИМ контроллер, например как PCA9685, представляет собой мощное и эффективное решение.

16-канальный ШИМ контроллер позволяет индивидуальное управление до 16 сервомоторами (или другими устройствами) с использованием протокола связи I2C. Он предоставляет широкие возможности настройки для каждого канала, включая регулировку яркости, скорости и положения сервомоторов, что делает его идеальным инструментом для реализации сложных механических конструкций, робототехники, умного дома и других проектов, где требуется точное и гибкое управление множеством устройств.

Далее представлено сравнение 16-канального ШИМ контроллера PCA9685 с аналогами Adafruit 16-Ch PWM/Servo HAT for Raspberry Pi и TowerPro SG90 в таблице 1.10.

Таблица 1.10 — Сравнение ШИМ контроллеров

Параметры сравнения	PCA9685	Adafruit 16-Ch PWM/Servo HAT for Raspberry Pi	TowerPro SG90
Количество каналов	16	16	Разное количество в зависимости от модели
Интерфейс связи	I2C	I2C	Прямое управление через цифровые пины микроконтроллера
Дополнительные особенности	Возможность управления яркостью, скоростью и положением сервомоторов и других устройств с использованием ШИМ. Допускает каскадное подключение.	Специально разработан для Raspberry Pi, высокая совместимость с платформой. Поддержка соответствующих библиотек.	Небольшие сервомоторы, управляемые напрямую через цифровые пины микроконтроллера. Ограничены по количеству каналов и точности управления.

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 на всенаправленных колесах с функцией видеозаписи. 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
- поворот модуля камеры на 180 градусов
- осуществление видеозаписи
- световая подсветка

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) Блок питания — источник питания схемы.
- 3) Модуль видеозаписи — камера и контроллер для нее.
- 4) Модуль освещения — 2 одноцветных светодиода, 2 RGB светодиода.
- 5) Модуль радиоприёма — приёмник, который получает информацию с радиопередатчика.
- 6) Модуль передвижения — двигатели постоянного тока и драйвер двигателей, которые реализуют передвижение по суше.
- 7) Модуль поворота камеры — шаговый двигатель и драйвер шагового двигателя.
- 8) ШИМ контроллер — компонент, используемый для увеличения ШИМ входов.

2.3 Взаимодействие компонентов устройства

Устройство на базе микроконтроллера Arduino Uno представляет собой комплексную систему, включающую в себя множество компонентов и модулей для обеспечения разнообразных функций.

Разработка радиоуправляемого передвижного устройства начинается с четкого определения его функций и компонентов, а также понимания взаимосвязи между ними для выполнения необходимых задач. Основные функции этого устройства включают в себя передвижение по суше и управление направлением движения, дистанционное управление через радиосигнал, видеозапись с возможностью поворота камеры на 180 градусов и световую подсветку.

Чтобы реализовать эти функции, необходимы определенные компоненты. Ключевым элементом является микроконтроллер, выполняющий обработку информации и выдачу управляющих сигналов. Для питания всей системы используется блок питания, который соединен с микроконтроллером. Модуль видеозаписи, включающий камеру и ее контроллер, обеспечивает функцию записи видео. Для освещения включены различные светодиоды, такие как одноцветные и RGB. Модуль радиоприема получает информацию с радиопередатчика для дистанционного управления. Для передвижения устройства по суше используются двигатели постоянного тока и их драйвер. Поворот камеры на 180 градусов обеспечивается шаговым двигателем и соответствующим драйвером. Для увеличения количества входов используется ШИМ контроллер.

Эти компоненты взаимодействуют для обеспечения выполнения каждой функции устройства. Их связь и включение в общую структуру устройства представлены на структурной схеме в приложении А. Управление устройством осуществляется через пульт радиоуправления. Контроллер на устройстве интерпретирует полученные сигналы и реализует управление двигателями, направляя устройство в соответствии с указанным направлением.

Модуль питания является ключевым элементом, обеспечивающим энергией все компоненты системы, обеспечивая бесперебойную работу всех узлов устройства.

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 4 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 4 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, его изображение представлено на рис. 3.1.



Рис. 3.1 — Микроконтроллер Arduino UNO

3.2 Обоснование выбора модулей радиопередачи

После анализа модулей радиопередачи, представленных в таблице 1.2 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 использован модуль NRF24L01. Данный модуль выбран по причине меньшего потребления тока, а также из-за значительно большей скорости передачи данных. Кроме того модуль NRF24L01 имеет более надежный протокол связи, который обеспечивает надежную передачу данных на дальние расстояния.

Изображение выбранного радиомодуля представлено на рис. 3.2.

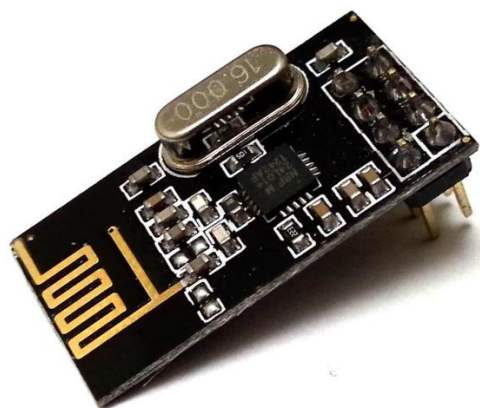


Рис. 3.2 — Радиомодуль NRF24L01

3.3 Обоснование выбора источника питания

В таблице 1.7 обзора литературы приведены сравнения источников питания. В данном курсовом проекте будет использована модель Mirex 6F22 в связи со своей ценовой доступностью, меньшей склонностью к саморазряду и удовлетворения всех необходимых параметров для проектируемого устройства. Данный источник питания также является более безопасным по сравнению с Li-ion и Li-Po батареями и менее подверженными риску перегрева, взрыва или возгорания.

3.4 Обоснование выбора светодиодов

В качестве модуля освещения в данном курсовом проекте будут использованы светодиоды RGB LED (рис. 3.3) и One-Color LED (рис. 3.4). Их характеристики представлены в таблице 1.6 обзора литературы. Данный модуль был выбран по причине того, что различные цвета светодиодов могут быть необходимы по личным нуждам пользователя, а свет одного цвета используется в качестве стандартной подсветки.



Рис. 3.3 — RGB светодиод



Рис. 3.4 — One-Color светодиод

3.5 Обоснование выбора камеры

В таблице 1.5 обзора литературы приведены сравнения камер. ESP32-CAM представляет собой уникальное устройство, сочетающее в себе ESP32 и камеру в компактном форм-факторе. Его выбор обосновывается рядом ключевых преимуществ:

Интеграция Wi-Fi и Bluetooth обеспечивает беспроводное подключение и передачу видеопотоков, что увеличивает возможности управления камерой. Это дает гибкость в управлении и использовании устройства в различных проектах.

Маленький размер и простота интеграции делают ESP32-CAM удобным для встраивания в разнообразные проекты. Программирование через Arduino IDE облегчает разработку и упрощает использование.

Высокое качество изображения позволяет модулю ESP32-CAM работать с высоким разрешением видео и изображений, обеспечивая качественный видеопоток.

Низкое энергопотребление делает ESP32-CAM эффективным в использовании с батарейными источниками питания, что важно для проектов, где долговечность работы играет ключевую роль.

Доступность и относительно низкая стоимость делают модуль ESP32-CAM привлекательным решением для различных проектов, где требуется камера в комбинации с Arduino или другими микроконтроллерами. Изображение модуля представлено на рис. 3.5.



Рис. 3.5 — ESP32-CAM

3.6 Обоснование выбора драйвера двигателей

В таблице 1.3 обзора литературы приведены сравнения драйверов двигателей. В данном курсовом проекте будет использована модель L9110S (рис. 3.6) исходя из наличия данного драйвера, возможности подключения

одновременно двух двигателей и удовлетворения всех необходимых параметров для проектируемого устройства. Данный драйвер имеет достаточный диапазон напряжения питания и доступен в цене.

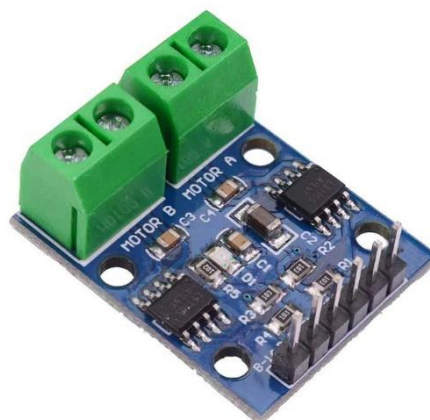


Рис. 3.6 — Драйвер двигателей L9110S

3.7 Обоснование выбора двигателей постоянного тока

DC Brush Motors — это надежные, доступные и управляемые двигатели, идеально подходящие для широкого спектра проектов, особенно для использования с Arduino. Их преимущества включают простоту управления без сложной электроники, возможность работать с различными скоростями, доступность на рынке по низкой цене и легкую интеграцию с Arduino. Они универсальны и могут быть использованы в робототехнике, механизмах передвижения и других приложениях, где требуется надежность и гибкость. Изображение двигателей представлено на рис. 3.7.



Рис. 3.7 — Двигатели DC Brush Motors

3.8 Обоснование выбора шагового двигателя

Выбор шагового двигателя 28YBJ-48 (рис. 3.8) обоснован несколькими ключевыми причинами. Характеристики сравниваемых шаговых двигателей приведены в таблице 1.8 обзора литературы. Во-первых, его универсальность: этот вид двигателя широко используется в DIY-проектах, предоставляя достаточную функциональность для различных областей, будь то робототехника, автоматизация или 3D-принтеры. Далее, доступность и стоимость этого типа двигателя — он часто доступен на рынке и имеет относительно невысокую цену, что делает его привлекательным вариантом для проектов с ограниченным бюджетом.

Этот вид двигателя также эффективен для устройства танка на всенаправленных колесах, обладая достаточной мощностью и точностью для множества задач. Его выбор можно обосновать сочетанием характеристик, доступностью и соответствием требованиям проекта, что делает его удобным и эффективным решением для множества сценариев использования.

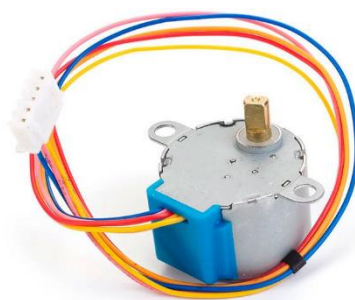


Рис. 3.8 — Шаговый двигатель 28YBJ-48

3.9 Обоснование выбора драйвера шагового двигателя

Выбор драйвера шагового двигателя ULN2003 (рис. 3.9) обусловлен несколькими ключевыми причинами. Характеристики сравниваемых драйверов шагового двигателя приведены в таблице 1.9 обзора литературы. Во-первых, его простота использования — ULN2003 предоставляет простой и удобный способ управления униполярными шаговыми двигателями, что делает его привлекательным для простых проектов. Кроме того, он обладает достаточным функционалом для основных проектов, предоставляя возможность управления до 7 выводами, что позволяет контролировать не только шаговые двигатели, но и другие нагрузки с помощью микроконтроллеров.

Его способность усиливать ток до 500 мА на канал обеспечивает достаточную мощность для управления шаговыми двигателями, а также

защитные функции, такие как защита от обратной полярности и помех, обеспечивают безопасность в работе с электроникой.

Также стоит учесть, что ULN2003 является широко распространенным и доступным на рынке компонентом, что упрощает его получение для множества проектов. В целом, выбор ULN2003 обоснован его простотой использования, функциональностью, надежностью и широким применением в различных электронных проектах.



Рис. 3.9 — Драйвер шагового двигателя ULN2003

3.10 Обоснование выбора ШИМ контроллера

В таблице 1.11 обзора литературы приведены сравнения ШИМ контроллеров. PCA9685 (рис. 3.10) обладает несколькими преимуществами по сравнению со своими аналогами. Например, он обеспечивает расширенный функционал для управления яркостью, скоростью и положением сервомоторов через ШИМ. Его 12-битное разрешение гарантирует высокую точность управления, что особенно ценно для проектов, требующих точности. Контроллер предоставляет 16 каналов управления и поддерживает каскадное подключение до 16 устройств на шине I2C, что позволяет эффективно управлять большим количеством устройств через единый интерфейс. Это делает PCA9685 гибким и мощным инструментом для различных проектов.

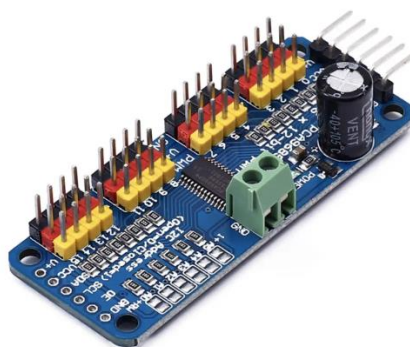


Рис. 3.10 — ШИМ контроллер PCA9685

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 представлен в таблице 4.1.

Таблица 4.1 — Расчет мощности элементов схемы устройства

Блок	U, В	I, мА	Кол-во	P, мВт
Микроконтроллер Arduino UNO R3	5	22	1	110
Модуль радиоприема NRF24L01	3,3	14	1	70
Драйвер двигателей L9110S	5	36	2	360
Двигатель постоянного тока 3-8V	5	600	4	12 000
Камера ESP32-CAM	5	180	1	900
Шаговый двигатель 28YBJ-48	5	200	1	1000
Драйвер шагового двигателя ULN2003	5	36	1	180
Светодиод	5	20	4	400
ШИМ контроллер PCA9685	5	12	1	60
Суммарная мощность, мВт				16676,2

В реализованной схеме используются микроконтроллер Arduino UNO R3, модуль радиоприема NRF24L01, драйвер двигателей L9110S, двигатели постоянного тока 3-8V, камера ESP32-CAM, шаговый двигатель 28YBJ-48, драйвер шагового двигателя ULN2003, модуль расширения GPIO портов, ШИМ контроллер и 4 светодиода.

Таким образом потребляемая мощность будет равна:

$$P = 5 \cdot 22 + 3.3 \cdot 14 + 5 \cdot 36 \cdot 2 + 5 \cdot 600 \cdot 4 + 5 \cdot 180 + 5 \cdot 200 + 5 \cdot 360 + 5 \cdot 20 \cdot 4 + 5 \cdot 12 = 16676,2 \text{ мВт.}$$

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 20012,04 мВт.

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{20.012}{5} = 4.0024 \approx 4 \text{ А}$$

4.2 Расчёт нагрузки светодиодов

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

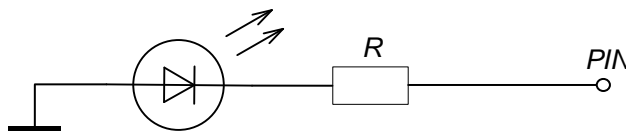


Рисунок 4.2 — Схема подключения светодиодов

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 цвета, а также RGB LED светодиоды, со следующими параметрами: $I_{\text{ПР}} = 20$ мА. $U_{\text{Д}} = 1$ 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 передвижного устройства камера подключена к выходам 5 В и GND, радиомодуль к цифровым входам D9-D13, драйвер шагового двигателя подключен к цифровым входам D3, D5, D6 и D8, ШИМ контроллер к аналоговым входам A4 и A5.

4.4 Камера

Информация о выбранной камере ESP32-CAM представлена в пункте 3.3 раздела 3.

Данная камера питается от напряжения 5 В платы Arduino UNO. На этапе прошивки микроконтроллера ESP32-CAM ее выходы подключены к выходам Arduino Uno согласно таблице 4.1, в последней строке таблицы указано подключение выходов обеих плат к собственным входам. Через это подключение используются для отправки и приема информации.

Таблица 4.2 — Подключение ESP32-CAM к Arduino UNO

ESP32CAM	Arduino UNO
U0T	TX
U0R	RX
5V	5V
GND	GND
GPIO-0 – GND	Reset – GND

После этапа прошивки достаточно оставить подключенными только выходы питания — 5 В и GND.

4.5 Радиомодуль

Информация о выбранном радиомодуле NRF24L01 представлена в пункте 3.2 раздела 3. Данный радиомодуль в схеме питается от напряжения 3,3 В. К цифровым входам D19-D23 микроконтроллера подключены выходы радиомодуля SCK, MISO, MOSI, CSN и CE соответственно, через которые поступают сигналы.

4.6 Шаговый двигатель

Информация о выбранном шаговом двигателе 28YBJ-48 представлена в пункте 3.8 раздела 3. Данный двигатель подключается к драйверу шагового двигателя ULN2003 четырех проводов от шагового двигателя и подключаются к выходам ULN2003, провода 1, 2, 3 и 4 в соответствии с распиновкой шагового двигателя. Питается двигатель от напряжения 5 В.

4.7 Драйвер шагового двигателя

Информация о выбранном драйвере шагового двигателя ULN2003 представлена в пункте 3.9 раздела 3.

Драйвер подключен своими выходами IN1-IN4 к выходам микроконтроллера D8, D6, D5 и D3 соответственно. Питается драйвер шагового двигателя от напряжения 5 В.

4.8 Драйвер двигателей

Информация о выбранном драйвере двигателей L9110S представлена в пункте 3.6 раздела 3. На вход VCC подается напряжение 5 В. Входы двух драйверов двигателей AI-A, BI-A, AI-B, BI-B подключаются к ШИМ выходам ШИМ контроллера PWM0, PWM1, PWM2, PWM3, PWM4, PWM5, PWM6, PWM7, PWM8, соответственно. Драйверы анализируют полученные значения и запускает моторы через выходы OUT1, OUT2, OUT3 и OUT4.

4.9 Расчет нагрузки двигателей постоянного тока

Информация о выбранных двигателях постоянного тока представлена в пункте 3.7 раздела 3. Все они связаны с микроконтроллером через драйвер двигателей L9110S. При поступлении питания на соответствующие входы моторов они начинают вращаться в заданную сторону.

Однако при подключении двигателя, для избегания помех следует подключить конденсатор, как показано на примере на рис. 4.3

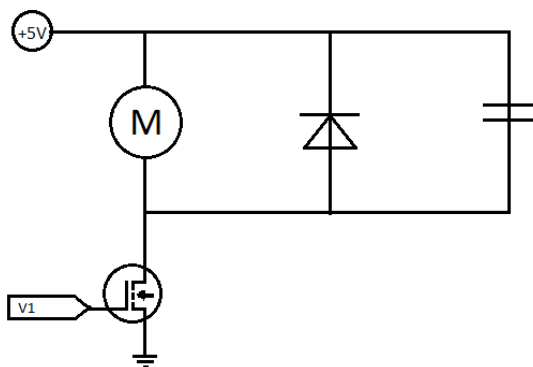


Рисунок 4.3 — Схема подключения двигателя

В отличие от схемы на рисунке, в рамках этого проекта необходим только конденсатор, поскольку функции остальных элементов компенсирует драйвер двигателей.

Чтобы рассчитать емкость конденсатора используем формулу:

$$I = C \times \frac{dV}{dt}$$

Где I — ток, C — емкость конденсатора, dV — изменение напряжения, dt — время изменения.

Подставим известные значения:

$$0.6A = C \times \frac{1V}{8,33 \times 10^{-6}c}$$

Решим уравнение относительно емкости C :

$$C = \frac{0.6A \times 8,33 \times 10^{-6} c}{5V} = 0.9996 \times 10^{-6} \Phi \approx 1 \text{ мк}\Phi$$

Следовательно, для того чтобы двигатели не мешали помехами они должны быть подключены через конденсатор емкостью примерно 1 мкФ.

4.10 ШИМ контроллер

Информация о выбранных ШИМ контроллерах представлена в пункте 3.11 раздела 3. Для увеличения количества ШИМ портов используется ШИМ контроллер, который подключается к аналоговым выходам Arduino A4 и A5 через собственные выходы SDA и SCL соответственно. Питается ШИМ контроллер от 5 В.

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Требования к разработке программного обеспечения

Разработанный танк на всенаправленных колесах оснащен камерой, которая расположена на башне. Башня имеет функцию поворота на 180 градусов. В качестве освещения подключено 4 светодиода с возможностью управления ими.

Устройство работает следующим образом. Через радиомодуль танк получает сигналы с управляющего устройства. Когда пользователь меняет направление на джойстиках или нажимает кнопки, микроконтроллер анализирует полученную информацию и отправляет код состояния на радиомодуль. Затем полученная информация анализируется на устройстве и в зависимости от полученного кода состояния запускаются или приостанавливаются двигатели.

В течении всего времени устройство транслирует запись с камеры по Wi-Fi. По нажатию кнопок, относящихся к управлению светодиодами, пользователь может включать или выключать их и менять цвет света RGB светодиодов.

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 танка на всенаправленных колесах

Код предназначен для управления танком, объединяя различные компоненты — моторы, шаговый двигатель, светодиоды, радиомодуль и камеру ESP32-CAM. Исходный код программного обеспечения под данное устройство можно найти в приложении Д (строки 1 – 53).

Функция `void handleFirstElement(value)` управляет состоянием светодиодов в зависимости от переданного значения `value`. При `value` равном 1, включаются оба светодиода; в противном случае оба светодиода выключаются.

Функция `void handleSecondElement(value)` регулирует цвет светодиодов в соответствии с переданным значением `value`. При `value` равном 1, включаются красные светодиоды; при любом другом значении все светодиоды выключаются.

Функция `void handleThirdElement(value)` устанавливает цвета светодиодов в зависимости от значения `value`. При:

- 1: оба светодиода светятся красным,
- 2: оба светодиода светятся зеленым,
- 3: оба светодиода светятся синим.

Функция `void tower(receivedMessage)` управляет шаговым двигателем в зависимости от принятого сообщения. Если сообщение равно 1, двигатель вращается влево; если равно 2, двигатель вращается вправо.

Функция `void performAction(action)` выбирает действие для управления моторами в соответствии с переданным значением `action`. Действия:

- 0: остановить все моторы,
- 1: движение вперед,
- 2: движение назад,
- 3-10: различные комбинации движения (подробнее в комментариях к коду).

Функции управления двигателями — группа функций, таких как `void forward()`, `void backward()`, `void forwardLeft()`, `void forwardRight()`, `void backwardLeft()`, `void backwardRight()`, `void spinLeft()`, `void spinRight()`, `void sidewaysLeft()` и

`void sidewaysRight()` устанавливают направление вращения каждого из четырех двигателей в зависимости от вызванной команды.

Функция `void stopMotors()` останавливает все двигатели, устанавливая сигналы на них в LOW.

Таким образом, код позволяет интегрировать данные от пульта управления через радиомодуль, преобразовывать их в управление моторами, шаговым двигателем и светодиодами для управления машинкой в реальном времени.

5.4 Исходный код программы для камеры

Код инициализирует камеру и подключается к Wi-Fi сети для создания веб-сервера, через который можно управлять камерой

Функция `void startCameraServer()` инициализирует веб-сервер для камеры, позволяя получить доступ к изображениям с камеры через веб-интерфейс.

Функция `void setupLedFlash(int pin)` настраивает вспышку камеры, используя указанный пин.

Функция `void setup()` запускается при старте устройства. В ней происходит настройка последовательного порта для связи с компьютером, инициализация конфигурации камеры (расположение пинов, разрешение, формат изображения), проверка доступности PSRAM (память высокой скорости), настройка параметров изображения (насыщенность, яркость) и подключение к Wi-Fi сети.

Функция `void loop()` выполняется в бесконечном цикле после `setup()`. В данном случае, она ничего не делает, так как все операции обработки запросов происходят в отдельном задании, выполняемом веб-сервером. Она просто ждет 10 секунд перед повторным выполнением цикла.

Основная задача этого кода — настройка камеры, установка веб-сервера для обработки запросов и предоставление интерфейса для управления и получения изображений с камеры через браузер

ЗАКЛЮЧЕНИЕ

В результате работы над данным курсовым проектом был разработан танк на всенаправленных колесах со своим программным обеспечением. Устройство за счет наличия всенаправленных колес имеет особенности передвижения, например езда боком. Помимо этого, осуществляется трансляция с камеры, установленной на устройстве и ее поворот по желанию пользователя. При необходимости можно воспользоваться подсветкой в виде четырех светодиодов. Данный проект был спроектирован в соответствии с поставленными задачами, весь функционал был реализован в полном объеме. Ниже представлены изображение устройства (рис. 6.1) и изображение с трансляции камеры (рис. 6.2).

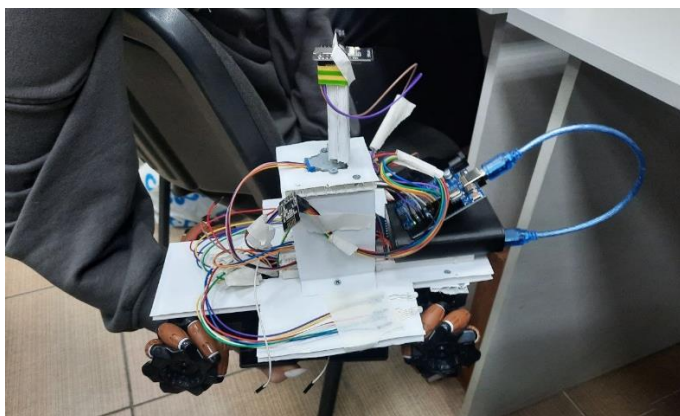


Рис. 6.1 — Готовое устройство танка на всенаправленных колесах

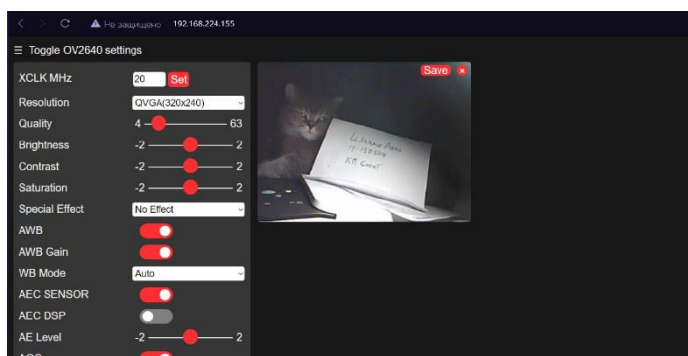


Рис. 6.2 — Трансляция с камеры танка

Разработанное 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 компонентами устройства.

В дальнейшем планируется усовершенствование данного курсового проекта. Одними из таких улучшений являются улучшение питания, добавление большего количества функций, а также создание более дружелюбного 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: 11.09.2023
- [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: 12.09.2023
- [6]. OLIMEXINO-STN32 development board [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://data.electronshik.ru/pdf/pdf/o/olimexino-stm32.pdf> — Дата доступа: 12.09.2023
- [7]. Беспроводной передатчик на 433 МГц [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://3d-diy.ru/product/besprovodnoj-peredatchik-na-433-mgc> — Дата доступа: 01.10.2023
- [8]. Беспроводной приемник на 433 МГц [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://3d-diy.ru/product/besprovodnoj-priemnik-na-433-mgc> — Дата доступа: 01.10.2023
- [9]. Драйвер двигателя L298N [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://3d-diy.ru/wiki/arduino-moduli/drayver-dvigatelya-l298n/> — Дата доступа: 02.10.2023
- [10]. Обзор драйвера мотора MX1508 [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://robotchip.ru/obzor-drayvera-motora-mx1508/> — Дата доступа: 02.10.2023
- [11]. Мотор постоянного тока [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://3d-diy.ru/wiki/arduino-mechanics/motor-postoyannogo-toka-reduktorom-1-48/> — Дата доступа: 03.10.2023
- [12]. Microsoft Word — MP3 — TF — 16P V1.0 [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://picaxe.com/docs/spe033.pdf> — Дата доступа: 03.10.2023

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

```
/*
  Скетч для танка
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Stepper_28BYJ.h>
#include <GyverMotor.h>
#include "PCF8574.h"

RF24 radio(9, 10); // Пины CE, CSN

const byte address[6] = "00101"; // Адрес канала для связи

//подключения шагового двигател
const int stepsPerRevolution = 4096;
Stepper_28BYJ myStepper(stepsPerRevolution, 8, 6, 5, 3);

bool button1Pressed = false;
bool button2Pressed = false;

int actions[5];

#define WHITE_LED_1_CHANNEL 8
#define WHITE_LED_2_CHANNEL 9
#define RGB_LED_1_RED_CHANNEL 12
#define RGB_LED_1_GREEN_CHANNEL 11
#define RGB_LED_1_BLUE_CHANNEL 10
#define RGB_LED_2_RED_CHANNEL 15
#define RGB_LED_2_GREEN_CHANNEL 14
#define RGB_LED_2_BLUE_CHANNEL 13

#define motor_FL_1 0
#define motor_FR_1 2
#define motor_BL_1 4
#define motor_BR_1 6

#define motor_FL_2 1
#define motor_FR_3 3
#define motor_BL_4 5
#define motor_BR_5 7

// Адрес устройства PCA9685
#define PCA9685_ADDRESS 0x40
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(PCA9685_ADDRESS);

void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
```

```

    radio.setPALevel(RF24_PA_LOW); // Выбор мощности передачи (можно изме-
нить)
    radio.startListening();

    myStepper.setSpeed(15); // скорость вращения (36 градусов в секунду)

    Wire.begin();
    pwm.begin();
    pwm.setPWMFreq(1600); // Установка частоты ШИМ
}

void loop() {
    if (radio.available()) {
        radio.read(&actions, sizeof(actions));

        //

        tower(actions[1]);

        handleFirstElement(actions[2]);
        handleSecondElement(actions[3]);
        if (actions[3] == 1)
        {
            handleThirdElement(actions[4]);
        }

        performAction(actions[0]);
    }
}

// Функция для обработки третьего элемента массива
void handleFirstElement(int value) {
    if (value == 1) {
        pwm.setPWM(WHITE_LED_1_CHANNEL, 0, 4095); // Включить белый светодиод
1 на максимальную яркость
        pwm.setPWM(WHITE_LED_2_CHANNEL, 0, 4095); // Включить белый светодиод
2 на максимальную яркость
    }
    else if (value == 0) {
        pwm.setPWM(WHITE_LED_1_CHANNEL, 0, 0); // Выключить белый светодиод 1
        pwm.setPWM(WHITE_LED_2_CHANNEL, 0, 0); // Выключить белый светодиод 2
    }
}

// Функция для обработки четвертого элемента массива
void handleSecondElement(int value) {
    if (value == 1) {
        pwm.setPWM(RGB_LED_1_RED_CHANNEL, 0, 4095); // Включаем красный
светодиод 1
        pwm.setPWM(RGB_LED_2_RED_CHANNEL, 0, 4095); // Включаем красный
светодиод 2
    }
    else if (value == 0) {
        pwm.setPWM(RGB_LED_1_RED_CHANNEL, 0, 0); // Выключить красный канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_1_GREEN_CHANNEL, 0, 0); // Выключить зеленый канал
RGB светодиода 1
    }
}

```

```

        pwm.setPWM(RGB_LED_1_BLUE_CHANNEL, 0, 0); // Выключить синий канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_RED_CHANNEL, 0, 0); // Выключить красный канал
RGB светодиода 2
        pwm.setPWM(RGB_LED_2_GREEN_CHANNEL, 0, 0); // Выключить зеленый канал
RGB светодиода 2
        pwm.setPWM(RGB_LED_2_BLUE_CHANNEL, 0, 0); // Выключить синий канал
RGB светодиода 2
    }
}

// Функция для обработки пятого элемента массива
void handleThirdElement(int value) {
    if (value == 1) {
        pwm.setPWM(RGB_LED_1_RED_CHANNEL, 0, 4095); // Включаем красный
светодиод 1
        pwm.setPWM(RGB_LED_2_RED_CHANNEL, 0, 4095); // Включаем красный
светодиод 2
        pwm.setPWM(RGB_LED_1_GREEN_CHANNEL, 0, 0); // Выключить зеленый канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_GREEN_CHANNEL, 0, 0); // Выключить зеленый канал
RGB светодиода 2
        pwm.setPWM(RGB_LED_1_BLUE_CHANNEL, 0, 0); // Выключить синий канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_BLUE_CHANNEL, 0, 0); // Выключить синий канал
RGB светодиода 2

    }
    else if (value == 2) {
        pwm.setPWM(RGB_LED_1_RED_CHANNEL, 0, 0); // Выключить красный канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_RED_CHANNEL, 0, 0); // Выключить красный канал
RGB светодиода 2
        pwm.setPWM(RGB_LED_1_GREEN_CHANNEL, 0, 4095); // Включить зеленый
канал RGB светодиода 1 на половинную яркость
        pwm.setPWM(RGB_LED_2_GREEN_CHANNEL, 0, 4095); // Включить зеленый
канал RGB светодиода 2 на половинную яркость
        pwm.setPWM(RGB_LED_1_BLUE_CHANNEL, 0, 0); // Выключить синий канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_BLUE_CHANNEL, 0, 0); // Выключить синий канал
RGB светодиода 2
    }
    else if (value == 3) {
        pwm.setPWM(RGB_LED_1_RED_CHANNEL, 0, 0); // Выключить красный канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_RED_CHANNEL, 0, 0); // Выключить красный канал
RGB светодиода 2
        pwm.setPWM(RGB_LED_1_GREEN_CHANNEL, 0, 0); // Выключить зеленый канал
RGB светодиода 1
        pwm.setPWM(RGB_LED_2_GREEN_CHANNEL, 0, 0); // Выключить зеленый канал
RGB светодиода 2
        pwm.setPWM(RGB_LED_1_BLUE_CHANNEL, 0, 4095); // Включить синий канал
RGB светодиода 1 на четверть яркости
        pwm.setPWM(RGB_LED_2_BLUE_CHANNEL, 0, 4095); // Включить синий канал
RGB светодиода 2 на четверть яркости
    }
}

```

```

void tower(int receivedMessage) {
    if (receivedMessage == 1 && !button1Pressed) {
        button1Pressed = true;
    }
    else if (receivedMessage == 2 && !button2Pressed) {
        button2Pressed = true;
    }

    if (button1Pressed) {
        myStepper.step(36);
        if (receivedMessage != 1) {
            myStepper.step(0); // остановка двигателя
            button1Pressed = false;
        }
    }
    else if (button2Pressed) {
        myStepper.step(-36);
        if (receivedMessage != 2) {
            myStepper.step(0); // остановка двигателя
            button2Pressed = false;
        }
    }
}

void performAction(int action) {
    switch (action) {
        case 0:
            // Ничего не делать
            stopMotors();
            break;
        case 1:
            // Ехать вперед
            forward();
            break;
        case 2:
            // Ехать назад
            backward();
            break;
        case 3:
            // Поворот влево при езде вперед
            forwardLeft();
            break;
        case 4:
            // Поворот вправо при езде вперед
            forwardRight();
            break;
        case 5:
            // Поворот влево при езде назад
            backwardLeft();
            break;
        case 6:
            // Поворот вправо при езде назад
            backwardRight();
            break;
        case 7:
            // Поворот влево на месте
            spinLeft();
            break;
    }
}

```



```

    case 8:
        // Поворот вправо на месте
        spinRight();
        break;
    case 9:
        // Езда боком влево
        sidewaysLeft();
        break;
    case 10:
        // Езда боком вправо
        sidewaysRight();
        break;
    default:
        stopMotors();
    }
}

// Функции управления двигателями в зависимости от вашей конфигурации
void forward() {
    pwm.setPWM(motor_BL_1, 0, 3200);
    pwm.setPWM(motor_BL_2, 0, 0);
    pwm.setPWM(motor_FL_1, 0, 3200);
    pwm.setPWM(motor_FL_2, 0, 0);
    pwm.setPWM(motor_FR_1, 0, 3200);
    pwm.setPWM(motor_FR_2, 0, 0);
    pwm.setPWM(motor_BR_1, 0, 3200);
    pwm.setPWM(motor_BR_2, 0, 0);
}

void backward() {
    pwm.setPWM(motor_BL_1, 0, 0);
    pwm.setPWM(motor_BL_2, 0, 3200);
    pwm.setPWM(motor_FL_1, 0, 0);
    pwm.setPWM(motor_FL_2, 0, 3200);
    pwm.setPWM(motor_FR_1, 0, 0);
    pwm.setPWM(motor_FR_2, 0, 3200);
    pwm.setPWM(motor_BR_1, 0, 0);
    pwm.setPWM(motor_BR_2, 0, 3200);
}

void forwardLeft() {
    forward();
    pwm.setPWM(motor_BL_1, 0, 0); // Уменьшаем скорость левой стороны
    pwm.setPWM(motor_BL_2, 0, 0); // Уменьшаем скорость левой стороны
}

void forwardRight() {
    forward();
    pwm.setPWM(motor_BR_1, 0, 0); // Уменьшаем скорость правой стороны
    pwm.setPWM(motor_BR_2, 0, 0); // Уменьшаем скорость правой стороны
}

void backwardLeft() {
    backward();
    pwm.setPWM(motor_FL_1, 0, 0); // Уменьшаем скорость левой стороны
    pwm.setPWM(motor_FL_2, 0, 0); // Уменьшаем скорость левой стороны
}

```

```

void backwardRight() {
    backward();
    pwm.setPWM(motor_FR_1, 0, 0); // Уменьшаем скорость правой стороны
    pwm.setPWM(motor_FR_2, 0, 0); // Уменьшаем скорость правой стороны
}

void spinLeft() {
    pwm.setPWM(motor_FL_1, 0, 3200);
    pwm.setPWM(motor_FL_2, 0, 0);
    pwm.setPWM(motor_FR_1, 0, 0);
    pwm.setPWM(motor_FR_2, 0, 3200);
    pwm.setPWM(motor_BL_1, 0, 3200);
    pwm.setPWM(motor_BL_2, 0, 0);
    pwm.setPWM(motor_BR_1, 0, 0);
    pwm.setPWM(motor_BR_2, 0, 3200);
}

void spinRight() {
    pwm.setPWM(motor_FL_1, 0, 0);
    pwm.setPWM(motor_FL_2, 0, 3200);
    pwm.setPWM(motor_FR_1, 0, 3200);
    pwm.setPWM(motor_FR_2, 0, 0);
    pwm.setPWM(motor_BL_1, 0, 0);
    pwm.setPWM(motor_BL_2, 0, 3200);
    pwm.setPWM(motor_BR_1, 0, 3200);
    pwm.setPWM(motor_BR_2, 0, 0);
}

void sidewaysLeft() {
    pwm.setPWM(motor_FL_1, 0, 0);
    pwm.setPWM(motor_FL_2, 0, 3200);
    pwm.setPWM(motor_FR_1, 0, 3200);
    pwm.setPWM(motor_FR_2, 0, 0);
    pwm.setPWM(motor_BL_1, 0, 3200);
    pwm.setPWM(motor_BL_2, 0, 0);
    pwm.setPWM(motor_BR_1, 0, 0);
    pwm.setPWM(motor_BR_2, 0, 3200);
}

void sidewaysRight() {
    pwm.setPWM(motor_FL_1, 0, 3200);
    pwm.setPWM(motor_FL_2, 0, 0);
    pwm.setPWM(motor_FR_1, 0, 0);
    pwm.setPWM(motor_FR_2, 0, 3200);
    pwm.setPWM(motor_BL_1, 0, 0);
    pwm.setPWM(motor_BL_2, 0, 3200);
    pwm.setPWM(motor_BR_1, 0, 3200);
    pwm.setPWM(motor_BR_2, 0, 0);
}

void stopMotors() {
    pwm.setPWM(motor_FL_1, 0, 0);
    pwm.setPWM(motor_FL_2, 0, 0);
    pwm.setPWM(motor_FR_1, 0, 0);
    pwm.setPWM(motor_FR_2, 0, 0);
    pwm.setPWM(motor_BL_1, 0, 0);
    pwm.setPWM(motor_BL_2, 0, 0);
    pwm.setPWM(motor_BR_1, 0, 0);
}

```

```

        pwm.setPWM(motor_BR_2, 0, 0);
    }
    /*
    Скетч для камеры
    */
    #include "esp_camera.h"
    #include <WiFi.h>

    #define CAMERA_MODEL_AI_THINKER

    #include "camera_pins.h"

    // =====
    // Enter your WiFi credentials
    // =====
    const char* ssid = "Крыса";
    const char* password = "26082004";

    void startCameraServer();
    void setupLedFlash(int pin);

    void setup() {
        Serial.begin(115200);
        Serial.setDebugOutput(true);
        Serial.println();

        camera_config_t config;
        config.ledc_channel = LEDC_CHANNEL_0;
        config.ledc_timer = LEDC_TIMER_0;
        config.pin_d0 = Y2_GPIO_NUM;
        config.pin_d1 = Y3_GPIO_NUM;
        config.pin_d2 = Y4_GPIO_NUM;
        config.pin_d3 = Y5_GPIO_NUM;
        config.pin_d4 = Y6_GPIO_NUM;
        config.pin_d5 = Y7_GPIO_NUM;
        config.pin_d6 = Y8_GPIO_NUM;
        config.pin_d7 = Y9_GPIO_NUM;
        config.pin_xclk = XCLK_GPIO_NUM;
        config.pin_pclk = PCLK_GPIO_NUM;
        config.pin_vsync = VSYNC_GPIO_NUM;
        config.pin_href = HREF_GPIO_NUM;
        config.pin_sccb_sda = SIOD_GPIO_NUM;
        config.pin_sccb_scl = SIOC_GPIO_NUM;
        config.pin_pwdn = PWDN_GPIO_NUM;
        config.pin_reset = RESET_GPIO_NUM;
        config.xclk_freq_hz = 20000000;
        config.frame_size = FRAMESIZE_UXGA;
        config.pixel_format = PIXFORMAT_JPEG; // for streaming
        //config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition
        config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
        config.fb_location = CAMERA_FB_IN_PSRAM;
        config.jpeg_quality = 12;
        config.fb_count = 1;

        // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
        // for larger pre-allocated frame buffer.
        if (config.pixel_format == PIXFORMAT_JPEG) {

```

```

    if (psramFound()) {
        config.jpeg_quality = 10;
        config.fb_count = 2;
        config.grab_mode = CAMERA_GRAB_LATEST;
    }
    else {
        // Limit the frame size when PSRAM is not available
        config.frame_size = FRAME_SIZE_SVGA;
        config.fb_location = CAMERA_FB_IN_DRAM;
    }
}
else {
    // Best option for face detection/recognition
    config.frame_size = FRAME_SIZE_240X240;

    // camera init
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {
        Serial.printf("Camera init failed with error 0x%x", err);
        return;
    }

    sensor_t* s = esp_camera_sensor_get();
    // initial sensors are flipped vertically and colors are a bit saturated
    if (s->id.PID == OV3660_PID) {
        s->set_vflip(s, 1); // flip it back
        s->set_brightness(s, 1); // up the brightness just a bit
        s->set_saturation(s, -2); // lower the saturation
    }
    // drop down frame size for higher initial frame rate
    if (config.pixel_format == PIXFORMAT_JPEG) {
        s->set_framesize(s, FRAME_SIZE_QVGA);
    }

    WiFi.begin(ssid, password);
    WiFi.setSleep(false);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    startCameraServer();

    Serial.print("Camera Ready! Use 'http://");
    Serial.print(WiFi.localIP());
    Serial.println("' to connect");
}

void loop() {
    // Do nothing. Everything is done in another task by the web server
    delay(10000);
}

```

ПРИЛОЖЕНИЕ Е
(обязательное)

Перечень элементов

ПРИЛОЖЕНИЕ Ж
(обязательное)

Ведомость документов