

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

ПУЛЬТ УПРАВЛЕНИЯ ДЛЯ ТАНКА НА ВСЕНАПРАВЛЕННЫХ  
КОЛЕСАХ

БГУИР КП 1-40 02 01 411 ПЗ

Студент: группы 150504,  
Губаревич А. Ю.

Руководитель:  
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 Губаревич Агате Юрьевне  
(фамилия, имя, отчество)

1. Тема проекта Пульт управления для танка 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. Кнопки – коммутируемое напряжение 12В, количество циклов коммутации 50000.
5. Джойстики – напряжение 5 В, поворот ручки 360.
6. ЖК-дисплей – напряжение 5 В, тип дисплея LCD.
7. Светодиоды – напряжение питания 5 В

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 пульта управления для танка на всенаправленных колесах (формат А4)

2. Функциональная электрическая схема пульта управления для танка на всенаправленных колесах (формат А3)

3. Принципиальная электрическая схема пульта управления для танка на всенаправленных колесах (формат А3)

3. Блок - схема пульта управления для танка на всенаправленных колесах (формат А2)

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

7. Дата выдачи задания 10.09.2023

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

разделы 1,2 к 24.09 – 20 %;

раздел 3 к 15.10 – 20 %;

раздел 4 к 05.11 – 25 %;

раздел 5 к 19.11 – 20 %;

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

защита курсового проекта с 07.12 по 14.12.

РУКОВОДИТЕЛЬ \_\_\_\_\_ Калютчик А. А.  
(подпись)

Задание принял к исполнению 10.09.2023 Губаревич А. Ю.  
(дата и подпись студента)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 ОБЗОР ЛИТЕРАТУРЫ .....	6
1.1 Состав устройства.....	6
1.2 Микроконтроллеры .....	6
1.3 Модуль радиопередачи .....	7
1.4 Источник питания.....	<b>Ошибка! Закладка не определена.</b>
1.5 Кнопки .....	8
1.6 Джойстики .....	<b>Ошибка! Закладка не определена.</b>
1.7 ЖК-дисплей.....	9
1.8 Светодиоды .....	9
2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА .....	11
2.1 Постановка задачи.....	11
2.2 Определение компонентов структуры устройства.....	11
2.3 Взаимодействие компонентов устройства .....	12
3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА.....	13
3.1 Обоснование выбора микроконтроллеров .....	13
3.2 Обоснование выбора модулей радиопередачи .....	13
3.3 Обоснование выбора источника питания.....	13
3.4 Обоснование выбора кнопок .....	14
3.5 Обоснование выбора джойстиков .....	134
3.6 Обоснование выбора ЖК-дисплея .....	<b>Ошибка! Закладка не определена.</b>
3.7 Обоснование выбора светодиодов .....	14
4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА ....	17
4.1 Расчёт мощности элементов схемы .....	17
4.2 Расчёт нагрузки светодиодов.....	17
4.3 Микроконтроллеры .....	186
4.4 Модуль радиопередачи .....	<b>Ошибка! Закладка не определена.</b>
4.5 Кнопки .....	17
4.6 Джойстики.....	17
4.7 ЖК-дисплей.....	19
4.8 Светодиоды .....	<b>Ошибка! Закладка не определена.</b>
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
ЗАКЛЮЧЕНИЕ.....	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	23
ПРИЛОЖЕНИЕ А .....	25
ПРИЛОЖЕНИЕ Б .....	26
ПРИЛОЖЕНИЕ В.....	27
ПРИЛОЖЕНИЕ Г .....	28
ПРИЛОЖЕНИЕ Д.....	29
ПРИЛОЖЕНИЕ Е.....	34
ПРИЛОЖЕНИЕ Ж.....	37

## **ВВЕДЕНИЕ**

В рамках данной курсовой работы представляется проект разработки пульта управления для танка на всенаправленных колесах. Тема данного проекта охватывает создание микропроцессорного устройства, предназначенного для управления передвижением танка с использованием колес, способных двигаться в любом направлении. Этот пульт будет оснащен современными элементами управления, такими как ЖК-дисплей, джойстик и кнопки, для обеспечения комфортного и эффективного взаимодействия с устройством.

Мобильные устройства, способные перемещаться в разные стороны, находят применение в различных областях, включая робототехнику, автоматизированные системы и развлекательные устройства. В данном случае, танк с колесами, способными двигаться во всех направлениях, предоставляет пользователю возможность управлять им с помощью удобного и интуитивно понятного пульта.

Основными элементами управления нашего пульта будут, джойстик, позволяющий точно и маневренно управлять движением, кнопки, предоставляющие дополнительные функциональные возможности, а также ЖК-дисплей для отображения информации о состоянии устройства.

Процесс разработки данного проекта будет включать в себя выбор и интеграцию компонентов управления, разработку программного обеспечения для обеспечения корректного управления танком, а также тестирование устройства с целью обеспечения его надежной и эффективной работы.

Таким образом, разработка пульта управления для танка на всенаправленных колесах представляет собой интересное и актуальное направление в области робототехники и автоматизации, и данная курсовая работа будет посвящена этому процессу.

# 1 ОБЗОР ЛИТЕРАТУРЫ

## 1.1 Состав устройства

Как сказано ранее, разрабатываемое микропроцессорное устройство выполняет функции управления посредством радиосигнала, информирует пользователя о состоянии танка и самого пульта. Для решения этих задач в состав устройства должны входить:

- микроконтроллер
- модуль радиопередачи
- источник питания
- кнопки
- джойстики
- ЖК-дисплей
- светодиоды для индикации

## 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 4 Model B	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
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 Модуль 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. 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 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 батарейка Mirex 6F22, литий-ионные (Li-ion) аккумуляторы и литий-полимерные (Li-Po) аккумуляторы. В таблице 1.3 приведены их сравнительные характеристики.

Таблица 1.3 — 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%

Продолжение таблицы 1.3

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

## 1.5 Кнопки

Кнопки — это простые, но важные компоненты в электронике, предоставляющие удобный и мгновенный способ взаимодействия с устройствами. Они могут выполнять различные функции, включая включение и выключение, выбор опций, отправку сигналов и многое другое.

Наиболее распространенными кнопками являются KAN0611-0510B, IT-0300 и KAN1211-0801B. Они работают по одному принципу, а их детальное сравнение приведено в таблице 1.4.

Таблица 1.4 — Сравнение кнопок

Параметры сравнения	KAN0611-0510B	IT-0300	KAN1211-0801B
Коммутируемое напряжение	12 В	12 В	12 В
Номинальный ток	50 мА	50 мА	50 мА
Контактное сопротивление (не более)	100 Ом	100 Ом	100 Ом
Диэлектрическая прочность	250 В	100 В	250 В
Количество циклов коммутации	50000	50000	10000
Рабочая температура	от -20 до +70 °C	от -25 до +80 °C	от -20 до +70 °C

## 1.6 Джойстики

Джойстик — это важный компонент в электронике, который предоставляет удобное управление различными устройствами. Он обеспечивает точное и многоазовое управление, позволяя перемещаться, вращаться и выполнять другие действия.

Для разработки микропроцессорного устройства был выбран джойстик модели КУ-032, который не имеет доступных аналогов. В таблице 1.5 приведены его основные характеристики.



Таблица 1.5 — Характеристики джойстика

Параметры сравнения	KY-032
Рабочее напряжение	3,3 – 5 В
Сопротивление потенциометров оси	10 кОМ
Рабочая температура	от -10 до +60 °С
Поворот ручки	360°

## 1.7 ЖК-дисплей

Жидкокристаллические дисплеи предоставляют наглядное и информативное визуальное отображение информации. ЖК-дисплей необходим для информирования пользователя о состоянии отдельных параметров танка.

Для сравнения были выбраны различные дисплеи на основе контроллеров HD44780, SSD1306 и ST7735. Результаты сравнения представлены в таблице 1.6.

Таблица 1.6 — Сравнение жк-дисплеев

Параметры сравнения	HD44780	SSD1306	ST7735
Рабочее напряжение	5 В	3-5 В	3,3 В
Рабочий ток	26 мА	20 мА	26 мА
Тип дисплея	LCD	OLED	IPC TFT
Рабочая температура	от -20 до +70 °С	от -40 до +70 °С	от -10 до +70 °С
Размер экрана	80 x 36 мм	27.3 x 27.8 мм	28 x 35 мм

## 1.8 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 и Super Bright LEDs. Их сравнительные характеристики приведены в таблице 1.7.

Таблица 1.7 — Сравнение типов светодиодов

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

## **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 пульт управления для радиоуправляемого передвижного устройства. Для реализации было выбрано устройство, управляющее не только движением передвижного устройства, но и его световыми индикаторами, а также вращением камеры. Исходя из этого, были выделены следующие функции, которые должно выполнять данное устройство:

- управление направлением движения путем регулирования тяговых усилий колес
- вывод информации о состоянии передвижного устройства
- управление световой индикацией передвижного устройства
- управление направлением камеры на передвижном устройстве
- световая индикация о состоянии передвижного устройства

### **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 — источник питания схемы.

3) Модуль управления — два одноосевых джойстика и семь кнопок, которые управляют направлением, движением и подсветкой передвижного устройства, а также направлением камеры.

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

5) Модуль индикации — светодиод, который выполняет функцию индикации о состоянии устройства.

6) Информационный модуль — ЖК-дисплей, который отображает информацию о состоянии компонентов передвижного устройства.

## **2.3 Взаимодействие компонентов устройства**

Устройство управления оснащено двумя двухосевыми джойстиками, предназначенными для управления движением передвижного устройства. При перемещении соответствующих джойстиков в нужном направлении микроконтроллер, установленный на устройстве управления, считывает эти сигналы и передает информацию на радиопередатчик. Радиопередатчик направляет эту информацию на радиоприемник, подключенный к микроконтроллеру, установленному на передвижном устройстве. После анализа полученных данных контроллер запускает двигатели передвижного устройства, обеспечивая движение в соответствии с направлением, указанным на устройстве управления.

Кроме того, на устройстве управления установлены кнопки, которые управляют дополнительными функциями передвижного устройства. Одна из кнопок включает основные фары на передвижном устройстве, а другая активирует RGB-подсветку. Третья кнопка позволяет изменять цвет RGB-подсветки. Путем считывания сигналов от кнопок, микроконтроллер на устройстве управления анализирует их состояние и отправляет соответствующие сигналы на микроконтроллер, установленный на передвижном устройстве.

Дополнительно, на устройстве управления присутствуют две кнопки для управления боковым движением передвижного устройства, так как оно создано на всенаправленных колесах.

Также имеются две кнопки для поворота камеры на 180 градусов.

Все эти функции позволяют пользователю удаленно управлять движением передвижного устройства и осуществлять дополнительные интерактивные действия через радиоуправление с помощью микроконтроллера.

## **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.



Рисунок 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.

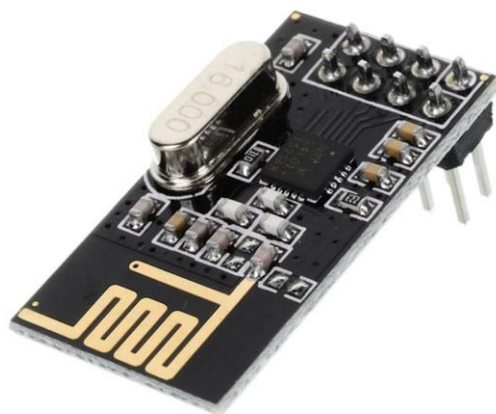


Рисунок 3.2 – модуль радиопередачи NRF24L01

### **3.3 Обоснование выбора источника питания**

В таблице 1.3 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 кнопок**

Рассмотрев сравнительные характеристики кнопок, представленных в таблице 1.4 обзора литературы, видно, что все кнопки имеют одинаковое напряжение питания, потребляемый ток и контактное сопротивление. Было принято решение выбрать кнопки KAN0611-0510B так как они наиболее распространены и имеют самую доступную цену.

### **3.5 Обоснование выбора джойстиков**

В данном устройстве будут использованы одноосевые джойстики модели KY-032. Их характеристики приведены в таблице 1.5 обзора литературы. Данные джойстики были выбраны по причине полного соответствия требованиям, а также отсутствия аналогов на рынке.



Рисунок 3.3 – джойстик KY-032

### 3.6 Обоснование выбора ЖК-дисплея

В таблице 1.6 обзора литературы приведены сравнения ЖК-дисплеев. HD44780, SSD1306 и ST7735 — это различные типы дисплеев, каждый со своими особенностями и возможностями. Не смотря на то, что HD44780 имеет ограниченные графические возможности, предлагая только символьное отображение, этого достаточно для функций нашего устройства. Данный дисплей был выбран в силу своей универсальности и доступности.

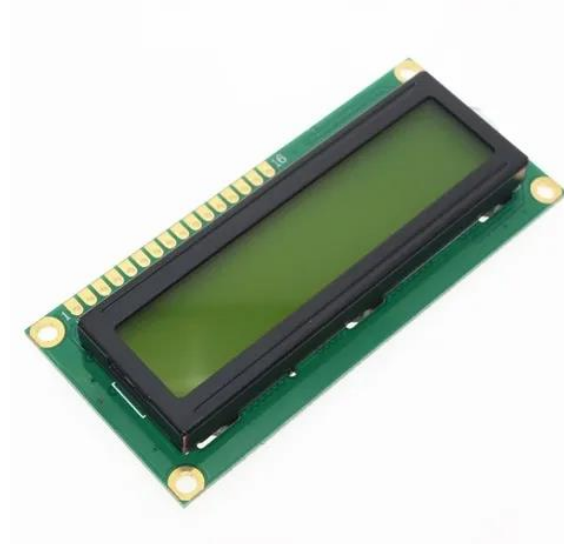


Рисунок 3.4 – ЖК-дисплей HD44780

### 3.7 Обоснование выбора светодиодов

В качестве модуля индикации в данном курсовом проекте будет использован светодиод RGB LED. Его характеристики представлены в таблице 1.7 обзора литературы. Данный модуль был выбран по причине того,

что различными цветами светодиода можно будет сигнализировать о различных состояниях устройства. Также данный светодиод отличается своей эффективностью и доступностью.



## 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 представлен в таблице 4.1.

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 NRF24L01	3,3	14	1	46,2
Джойстик	5	8	2	80
Кнопка	5	50	7	700
ЖК-дисплей HD44780	5	26	1	130
Светодиод	5	20	1	100
Суммарная мощность, мВт				1166,2

В реализованной схеме используются микроконтроллер Arduino UNO R3, модуль радиопередачи NRF24L01, 2 джойстика, 7 кнопок, ЖК-дисплей HD44780 и светодиод.

Таким образом потребляемая мощность будет равна:

$$P = 5 \cdot 22 + 3,3 \cdot 14 + 5 \cdot 8 \cdot 2 + 5 \cdot 50 \cdot 7 + 5 \cdot 26 + 5 \cdot 20 = 1166,2 \text{ мВт.}$$

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 1399,4 мВт.

Рассчитаем потребляемый ток:

$$I = \frac{P}{U} = \frac{1.399}{5} = 0.2798 \approx 0.28 \text{ А}$$

### 4.2 Расчёт нагрузки светодиодов

В данном курсовом проекте используется один красный светодиод, подключенный к соответствующим пинам микроконтроллера (см. рисунок 4.2).

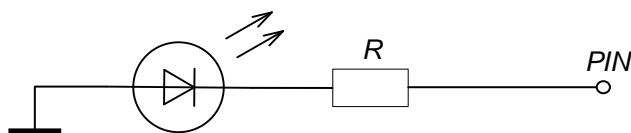


Рисунок 4.2 — Схема подключения светодиодов

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

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

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

В проекте используются светодиоды RGB LED, 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. На входы A4 и A5 подключен ЖК-дисплей.

На цифровые выходы D9-D13 подключен радио модуль. На выход D8 подключен красный светодиод, а на выходы D1-D7 подключено 7 кнопок.

Данный микроконтроллер питается от напряжения 9 В.

#### **4.4 Модуль радиопередачи**

Информация о выбранных модулях радиопередачи представлена в пункте 3.2 раздела 3.

На устройстве управления используется модуль NRF24L01, который питается от напряжения 3,3 В. Также на модуле есть выходы MOSI (Master Out Slave In) для передачи данных от микроконтроллера к модулю и MISO (Master In Slave Out) для передачи данных от модуля к микроконтроллеру. Они подключены к цифровым входам D11 и D12 микроконтроллера соответственно. Выход SCK (Serial Clock) это тактирующий сигнал, используемый для синхронизации передачи данных, он подключен к цифровому входу D13. Выход CE (Chip Enable) — сигнал для включения или выключения модуля, подключен к входу D9, а выход CSN (Chip Select Not) отвечает за выбор устройства для обмена данными и подключен ко входу D10.

#### **4.5 Кнопки**

В данном курсовом проекте используются семь кнопок KAN0611-0510B подключенных к цифровым входам D1-D7. На вход VCC подается напряжение 5 В.

#### **4.6 Джойстики**

В качестве управления в данном курсовом проекте используются два джойстика KY-032 подключенных в схеме устройства управления к аналоговым входам Arduino A1 и A2. На вход VCC подается напряжение 5 В.

#### **4.7 ЖК-дисплей**

Информация о выбранном ЖК-дисплее HD44780 представлена в пункте 3.6 раздела 3. На вход VCC подается напряжение 5 В. Входы SDA (Serial Data) — линия для передачи данных между микроконтроллером и дисплеем и SCL (Serial Clock) — линия тактового сигнала, который управляет временной последовательностью передачи данных подключаются к микроконтроллеру через аналоговые входы A4 и A5 соответственно.

#### **4.8 Светодиод**

В качестве индикатора работы устройства в данном курсовом проекте используется красный светодиод. Он подключен к цифровому входу D8 Arduino Uno. На вход VCC подается напряжение 5 В.

## **5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

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

Разработанное микропроцессорное устройство управляет движением устройства передвижения, а также некоторыми его компонентами.

Устройство работает следующим образом. При включении 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 передвижения через радиопередатчик, установленный на пульте. Затем полученная информация анализируется на устройстве передвижения и в зависимости от полученного кода состояния запускаются или приостанавливаются двигатели.

При нажатии первых двух кнопок на устройство передвижения посылается сигнал о езде боком в соответствующем направлении. При нажатии следующих двух кнопок на устройство передвижения посылается сигнал о вращении шагового двигателя вправо или влево. При нажатии пятой кнопки на устройство передвижения посылается сигнал о включении белых светодиодов. При нажатии шестой кнопки на устройство передвижения посылается сигнал о включении rgb-светодиодов. При нажатии седьмой кнопки на устройство передвижения посылается сигнал о изменении цвета rgb-светодиодов.

От устройства передвижения на микроконтроллер поступает информация о состоянии его светодиодов. Далее данная информация отображается на ЖК-дисплее.

### **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 логичных диаграмм.

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

Блоки 2 – 3 представляют собой подготовку программы для дальнейшей работы (инициализация переменных и определение модулей, подключенных к микроконтроллеру). Ключевыми являются блоки 5 – 48, которые реализуют саму логику программы в бесконечном цикле. Блоки 17 и 40 реализуют получение информации о действиях пользователя, а блоки 18 – 39 и 41 – 46 анализируют полученные данные и выставляют соответствующий код на радиопередатчик. В блоке 47 информация отправляется устройству. В блоке 5 происходит сброс устройства вывода информации, в блоках 6 и 12

анализируется какая информация должна быть выведена на экран, в блоках 7-8, 11, 13 – 16 информация выводится на экран.

### 5.3 Исходный код программы

Пульт дистанционного управления считывает показания с одноосевых джойстиков и кнопок, а затем преобразует данные для быстрой и удобной отправки по радиоканалу. Также по радиоканалу пульт получает информацию о состоянии компонентов передвижного устройства и выводит их на ЖК-дисплей. Исходный код программного обеспечения под данное устройство можно найти в приложении Д (строки 1 – 244).

Функция `void setup()` (строки 44 – 79) необходима для начальной настройки контроллера, здесь задаются входные и выходные пины, а так же другие настройки. В данном случае `radio.openWritingPipe(address)` (строка 46) настраивает наш радио модуль. `pinMode(buttonLeftPin, INPUT_PULLUP)`  
`pinMode(buttonRightPin, INPUT_PULLUP)`  
`pinMode(buttonTowerLeft, INPUT_PULLUP)`  
`pinMode(buttonTowerRight, INPUT_PULLUP)`  
`pinMode(buttonWhiteLed, INPUT_PULLUP)`  
`pinMode(buttonRGBLed, INPUT_PULLUP)`  
`pinMode(buttonRGBLedChange, INPUT_PULLUP)` (строки 50 – 57) задают пины кнопок и светодиода, как входные значения на плату. `digitalWrite(RED_LED_PIN, HIGH)` (строка 59) отвечает за включение индикатора при включении пульта. `lcd.begin(16, 2)` (строка 61) инициализирует ЖК-дисплей.

Функция `void loop()` (строки 81 – 92) является главной, циклической функцией и работает на протяжении всей работы микроконтроллера. Здесь вызываются все основные функции нашей программы.

Функция `void send()` (строки 94 – 98) формирует массив чисел, обозначающих определенные команды для устройства передвижения. Полученный массив отправляется и затем происходит ожидание полной отправки сообщения.

Функция `void drive()` (строки 101 – 149) отвечает за управление движением, основываясь на считанных значениях с джойстика и состоянии кнопок `buttonLeftPin` и `buttonRightPin`. Существует 10 шаблонов движения, текущий шаблон движения записывается в переменную `driveNum`, которая позже будет добавлена в массив в функции `send()`.

Функция `void displayInfoOnLCD()` (строки 152 – 185) предназначена для отображения информации на символьном LCD-дисплее. Основываясь на информации о нажатии кнопок она выводит на дисплей информацию о состоянии фар и подсветки нашего устройства передвижения.

В функции `void checkButtons()` (строки 187 – 244) проверяется состояние различных кнопок и устанавливают соответствующие значения переменных (`towerRotate`, `whiteLedOn`, `rgbLedOn`, `rgbColor`) в зависимости от состояния этих кнопок. Далее эти значения будут добавлены в массив в функции `send()`.

## ЗАКЛЮЧЕНИЕ

В результате работы над данным курсовым проектом было разработано работоспособное микропроцессорное устройство управления со своим программным обеспечением. Устройство управляет колесами передвижного устройства, вращает шаговый двигатель с прикрепленной камерой, а также включает светодиоды на нем. Помимо этого данное устройство получает информацию о состоянии составляющих компонентов передвижного устройства и вывод ее на ЖК-дисплей. Данный проект был спроектирован в соответствии с поставленными задачами, весь функционал был реализован в полном объеме.

Разработанное микропроцессорное устройство обладает следующими достоинствами: простота реализации и сборки. Однако существенным недостатком является необходимость в написании собственного программного обеспечения для взаимодействия со всеми подключенными компонентами, а также относительно высокая стоимость.

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

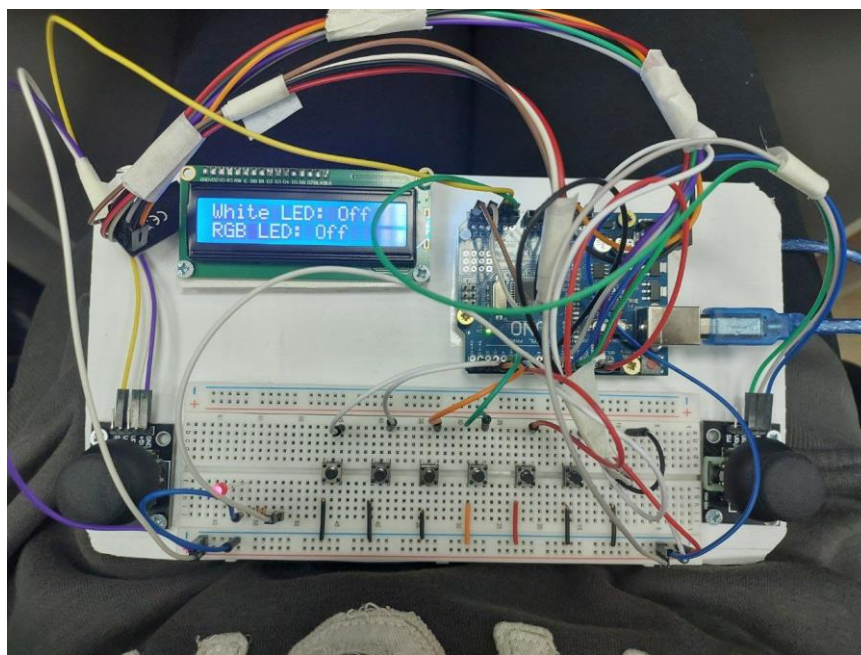


Рисунок 6.1 – Готовое устройство «пульт управления танком на всенаправленных колесах»

## 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](https://www.bsuir.by/m/12_100229_1_136308.pdf)

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

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

[4]. 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



**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 #include <SPI.h>
002 #include <nRF24L01.h>
003 #include <RF24.h>
004 #include <LiquidCrystal_I2C.h>
005
006 LiquidCrystal_I2C lcd(0x27, 16, 2);
007
008 #define RED_LED_PIN 8
009
010 RF24 radio(9, 10);
011
012 const byte address[6] = "00101";
013
014 int xAxisPin = A1;
015 int yAxisPin = A2;
016 int buttonLeftPin = 1;
017 int buttonRightPin = 2;
018 int buttonTowerLeft = 3;
019 int buttonTowerRight = 4;
020 int buttonWhiteLed = 5;
021 int buttonRGBLed = 6;
022 int buttonRGBLedChange = 7;
023
024 int prevButtonStateLeft = HIGH;
025 int prevButtonStateRight = HIGH;
026 int prevButtonStateTowerLeft = HIGH;
027 int prevButtonStateTowerRight = HIGH;
028 int prevButtonStateWhiteLed = HIGH;
029 int prevButtonStateRgbLed = HIGH;
030 int prevButtonStateRgbLedChange = HIGH;
```

```

031
032 int xAxisValue, yAxisValue;
033 bool moveLeft = false;
034 bool moveRight = false;
035 int driveNum = 0;
036 int towerRotate = 0;
037 int whiteLedOn = 0;
038 int rgbLedOn = 0;
039 int rgbColor = 1;
040
041 void setup() {
042     radio.begin();
043     radio.openWritingPipe(address);
044     radio.setPALevel(RF24_PA_LOW);
045     radio.stopListening();
046
047     pinMode(buttonLeftPin, INPUT_PULLUP);
048     pinMode(buttonRightPin, INPUT_PULLUP);
049     pinMode(buttonTowerLeft, INPUT_PULLUP);
050     pinMode(buttonTowerRight, INPUT_PULLUP);
051     pinMode(buttonWhiteLed, INPUT_PULLUP);
052     pinMode(buttonRGBLed, INPUT_PULLUP);
053     pinMode(buttonRGBLedChange, INPUT_PULLUP);
054     pinMode(RED_LED_PIN, OUTPUT);
055
056     digitalWrite(RED_LED_PIN, HIGH);
057
058     lcd.begin(16, 2);
059
060     lcd.init();
061     lcd.backlight();
062
063     lcd.blink();
064     delay(500);

```

```

065    lcd.noBlink();
066
067    lcd.setCursor(1, 0);
068    lcd.print("DEVICE");
069    lcd.setCursor(8, 1);
070    lcd.print("IS ON");
071    delay(500);
072
073    lcd.cursor();
074    delay(500);
075    lcd.noCursor();
076 }
077
078 void loop() {
079
080    displayInfoOnLCD();
081
082    checkButtons();
083    drive();
084    send();
085
086    delay(50);
087 }
088
089 void send() {
090    int dataToSend[5] = {driveNum, towerRotate, whiteLedOn,
    rgbLedOn, rgbColor};
091    radio.write(&dataToSend, sizeof(dataToSend));
092 }
093
094 void drive() {
095
096    xAxisValue = analogRead(xAxisPin);
097    yAxisValue = analogRead(yAxisPin);

```

```

098
099  if (digitalRead(buttonLeftPin) == LOW) {
100      moveLeft = true;
101      moveRight = false;
102  } else if (digitalRead(buttonRightPin) == LOW) {
103      moveRight = true;
104      moveLeft = false;
105  } else {
106      moveLeft = false;
107      moveRight = false;
108  }
109
110  int x = map(xAxisValue, 0, 1023, -255, 255);
111  int y = map(yAxisValue, 0, 1023, -255, 255);
112
113  if (y < 50 && y > -50 && x < 50 && x > -50 && moveLeft &&
!moveRight) {
114      driveNum = 9;
115  } else if (y < 50 && y > -50 && x < 50 && x > -50 && moveRight
&& !moveLeft) {
116      driveNum = 10;
117  } else if (y >= 50 && x < 50 && x > -50) {
118      driveNum = 1;
119  } else if (y <= -50 && x < 50 && x > -50) {
120      driveNum = 2;
121  } else if (y >= 50 && x <= -50) {
122      driveNum = 3;
123  } else if (y >= 50 && x >= 50) {
124      driveNum = 4;
125  } else if (y <= -50 && x <= -50) {
126      driveNum = 5;
127  } else if (y <= -50 && x >= 50) {
128      driveNum = 6;
129  } else if (y > -50 && y < 50 && x <= -50) {

```



```

130     driveNum = 7;
131 } else if (y > -50 && y < 50 && x >= 50) {
132     driveNum = 8;
133 } else {
134     driveNum = 0;
135 }
136 }
137
138 void displayInfoOnLCD() {
139     lcd.clear(); // Очистка дисплея
140
141     lcd.setCursor(0, 0);
142     lcd.print("White LED: ");
143     if (whiteLedOn == 0) {
144         lcd.print("Off");
145     } else {
146         lcd.print("On ");
147     }
148
149     lcd.setCursor(0, 1);
150     lcd.print("RGB LED: ");
151     if (rgbLedOn == 0) {
152         lcd.print("Off");
153     } else {
154         switch (rgbColor) {
155             case 1:
156                 lcd.print("Red");
157                 break;
158             case 2:
159                 lcd.print("Green");
160                 break;
161             case 3:
162                 lcd.print("Blue");
163                 break;

```

```

164         default:
165             lcd.print("Unknown");
166             break;
167     }
168 }
169 }
170
171 void checkButtons() {
172     int currentButtonStateLeft = digitalRead(buttonLeftPin);
173     int currentButtonStateRight = digitalRead(buttonRightPin);
174     int currentButtonStateTowerLeft = digitalRead(buttonTowerLeft);
175     int currentButtonStateTowerRight =
digitalRead(buttonTowerRight);
176     int currentButtonStateWhiteLed = digitalRead(buttonWhiteLed);
177     int currentButtonStateRgbLed = digitalRead(buttonRGBLed);
178     int currentButtonStateRgbLedChange =
digitalRead(buttonRGBLedChange);
179
180     if (currentButtonStateLeft == LOW && prevButtonStateLeft ==
HIGH) {
181         moveLeft = true;
182         moveRight = false;
183     } else if (currentButtonStateRight == LOW &&
prevButtonStateRight == HIGH) {
184         moveRight = true;
185         moveLeft = false;
186     } else {
187         moveLeft = false;
188         moveRight = false;
189     }
190
191     if (digitalRead(buttonTowerLeft) == LOW) {
192         towerRotate = 1;
193     } else if (digitalRead(buttonTowerRight) == LOW) {
194         towerRotate = 2;

```

```

195     } else {
196         towerRotate = 0;
197     }
198
199     if (currentButtonStateWhiteLed == LOW && prevButtonStateWhiteLed
== HIGH) {
200         whiteLedOn = (whiteLedOn == 0) ? 1 : 0;
201     }
202
203     if (currentButtonStateRgbLed == LOW && prevButtonStateRgbLed ==
HIGH) {
204         rgbLedOn = (rgbLedOn == 0) ? 1 : 0;
205     }
206
207     if (currentButtonStateRgbLedChange == LOW &&
prevButtonStateRgbLedChange == HIGH && rgbLedOn == 1) {
208         if (rgbColor == 1) {
209             rgbColor = 2;
210         } else if (rgbColor == 2) {
211             rgbColor = 3;
212         } else if (rgbColor == 3) {
213             rgbColor = 1;
214         }
215     }
216
217     if (rgbLedOn == 0) {
218         rgbColor = 1;
219     }
220
221     prevButtonStateLeft = currentButtonStateLeft;
222     prevButtonStateRight = currentButtonStateRight;
223     prevButtonStateWhiteLed = currentButtonStateWhiteLed;
224     prevButtonStateRgbLed = currentButtonStateRgbLed;
225     prevButtonStateRgbLedChange = currentButtonStateRgbLedChange;
226 }

```

**ПРИЛОЖЕНИЕ Е**  
(обязательное)

**Перечень элементов**

**ПРИЛОЖЕНИЕ Ж**  
(обязательное)

**Ведомость документов**