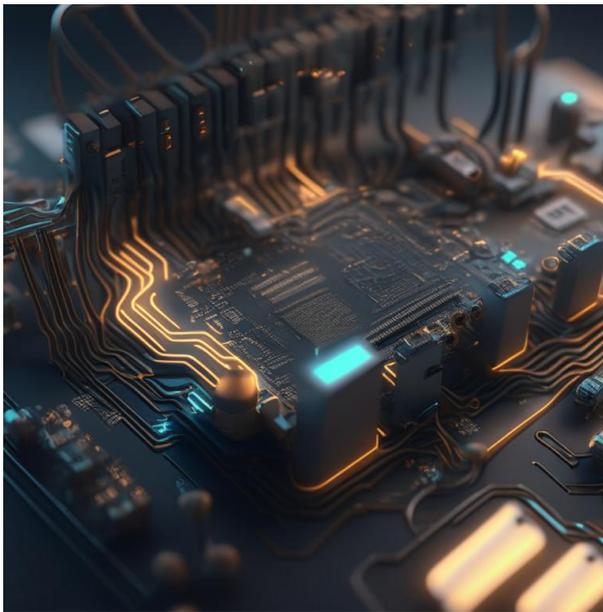


# ИТМО

С.М. Власов, В.А. Жданов, А.А. Маргун,  
К.А. Зименко, Л. Сулейман, Д.А. Галкина

## ПРОГРАММИРОВАНИЕ STM32 НА НАЛ. ЧАСТЬ 1



Санкт-Петербург

2025

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**УНИВЕРСИТЕТ ИТМО**

**С.М. Власов, В.А. Жданов, А.А. Маргун,  
К.А. Зименко, Л. Сулейман, Д.А. Галкина**

**ПРОГРАММИРОВАНИЕ STM32 НА  
НАЛ. ЧАСТЬ 1**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

**РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ  
В УНИВЕРСИТЕТЕ ИТМО**

по направлению подготовки 15.03.06 «Мехатроника и  
робототехника», 27.03.04 «Управление в технических системах» в  
качестве учебного пособия для реализации основных  
профессиональных образовательных программ высшего  
образования бакалавриата

**ИТМО**

**Санкт-Петербург**

**2025**

С.М. Власов, В.А. Жданов, А.А. Маргун, К.А. Зименко, Л. Сулейман, Д.А. Галкина. Программирование STM32 на HAL. Часть 1. — Санкт-Петербург: Университет ИТМО, 2025. — 38 с.

**Рецензент:**

И.А. Бжихатлов, к.т.н., факультет систем управления и робототехники, Университет ИТМО, Санкт-Петербург, Россия.

В лабораторном практикуме рассматриваются физические основы дискретных электронных компонентов, их принципы работы и принципы построения простейших электронных схем.

**ITMO**

**Университет ИТМО** (Санкт-Петербург) — национальный исследовательский университет, ведущий вуз России в области информационных, фотонных и биохимических технологий. Альма-матер победителей международных соревнований по программированию: ICPC (единственный в мире семикратный чемпион), GoogleCode Jam, Facebook Hacker Cup, Яндекс.Алгоритм, Russian Code Cup, Topcoder Open и др. Приоритетные направления: IT и искусственный интеллект, фотоника, робототехника, квантовые коммуникации, трансляционная медицина, Life Sciences, Art & Science, Science Communication. Входит в ТОП-100 по направлению «Автоматизация и управление» Шанхайского предметного рейтинга (ARWU) и занимает 74 место в мире в британском предметном рейтинге QS по компьютерным наукам (Computer Science and Information Systems). Представлен в мировом ТОП-200 по телекоммуникационным технологиям (Telecommunication engineering), а также в ТОП-300 по нанонаукам и нанотехнологиям (Nanoscience & Nanotechnology) ARWU. Входит в ТОП-200 по инженерным наукам (Engineering and Technology), в ТОП-300 по физике и астрономии (Physics & Astronomy), наукам о материалах (Materials Sciences), а также по машиностроению, аэрокосмической и промышленной инженерии (Mechanical, Aeronautical & Manufacturing Engineering) рейтинга QS. Лидер проекта «Приоритет – 2030».

© Университет ИТМО, 2025

© С.М. Власов, В.А. Жданов, А.А. Маргун, К.А. Зименко, Л. Сулейман, Д.А. Галкина, 2025

## СОДЕРЖАНИЕ

Введение . . . . .	4
Начало работы в среде разработки STM32CubeIDE . . . . .	5
Знакомство со средой разработки . . . . .	7
Лабораторная работа №1. . . . .	10
Цель работы . . . . .	10
Аппаратное обеспечение . . . . .	10
Программное обеспечение . . . . .	10
Общие сведения . . . . .	10
Порядок выполнения . . . . .	10
Контрольные вопросы: . . . . .	16
Лабораторная работа №2 . . . . .	17
Цель работы . . . . .	17
Аппаратное обеспечение . . . . .	17
Программное обеспечение . . . . .	17
Порядок выполнения . . . . .	17
Контрольные вопросы: . . . . .	19
Лабораторная работа №3. . . . .	20
Цель работы . . . . .	20
Аппаратное обеспечение . . . . .	20
Программное обеспечение . . . . .	20
Порядок выполнения . . . . .	20
Контрольные вопросы: . . . . .	21
Лабораторная работа №4. . . . .	22
Цель работы . . . . .	22
Аппаратное обеспечение . . . . .	22
Программное обеспечение . . . . .	22
Порядок выполнения . . . . .	22
Контрольные вопросы: . . . . .	23
Список литературы . . . . .	24
Приложение А. Электрические схемы лабораторного стенда	25

## Введение

Данный лабораторный практикум предназначен для освоения дисциплин «Программирование микроконтроллеров», «Цифровая и микроконтроллерная техника» и «Цифровая техника систем управления» в рамках образовательных программ направлений подготовки 15.03.06 «Мехатроника и Робототехника» и 27.03.04 «Управление в технических системах».

Основной целью пособия является формирование у обучающихся знаний и практических навыков программирования микроконтроллеров STM32 с использованием среды разработки STM32CubeIDE. В пособии содержатся пошаговые инструкции и методические рекомендации по выполнению лабораторных работ, направленных на изучение периферийных интерфейсов, включая GPIO, UART, таймеры, ЦАП и АЦП. В процессе выполнения лабораторных работ обучающиеся приобретают умения по настройке и конфигурированию периферийных модулей микроконтроллеров STM32, работе с отладочной платой Nucleo-F446RE и встроенным программатором-дебаггером ST-LINK/V2, программированию на языке C с использованием библиотеки HAL, работе с интегрированной средой разработки STM32CubeIDE.

Выполнение лабораторных работ способствует формированию у студентов компетенций, включающих способность разрабатывать программное обеспечение для встроенных систем, навыки работы с периферийными устройствами микроконтроллеров и их конфигурации для решения прикладных задач, применение методов диагностики и отладки программного обеспечения встроенных систем, а также способность к анализу и проектированию архитектуры программно-аппаратных комплексов на основе микроконтроллеров.

Для успешного выполнения лабораторных работ рекомендуется ознакомиться со следующими источниками: **Reference Manual** для STM32F446, содержащий подробное описание всех периферийных модулей и их регистров; **Datasheet** на микроконтроллер STM32F446RE, включающий информацию о характеристиках устройства, параметрах работы и функциональности; **Description of HAL and low-layer drivers**, предоставляющее информацию о библиотеке аппаратной абстракции HAL и низкоуровневых драйверах LL, которые упрощают программирование периферий.

## Начало работы в среде разработки STM32CubeIDE

Создайте рабочую папку для проекта. Запустите STM32CubeIDE, в открывшемся окне задайте путь к вашей рабочей папке (Рисунок 1). В пути к рабочей папке и названии проекта не должно быть русских букв. В этой папке будут храниться все лабораторные работы.

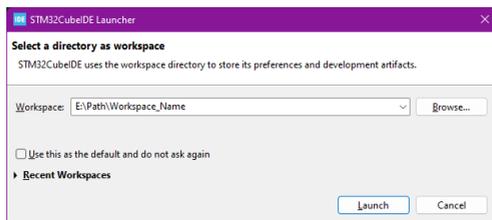


Рисунок 1. Окно выбора рабочей папки

В появившемся окне (Рисунок 2) необходимо создать проект. Либо нажмите «Start new STM32 project», либо «File->New->STM32 Project».

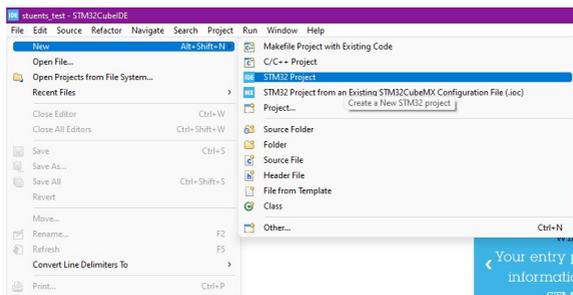


Рисунок 2. Стартовое окно

После открытия окна выбора конечного устройства (Рисунок 3) в поле «Commercial Part Number» необходимо ввести наименование микроконтроллера. Наименование контроллера можно найти на отладочной плате. Возможные варианты – STM32F446RET.

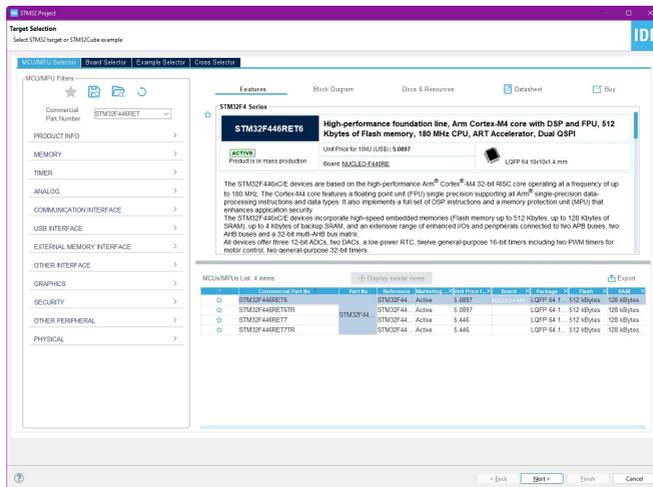


Рисунок 3. Окно выбора конечного устройства

В следующем окне (Рисунок 4) необходимо ввести название проекта, выберите язык – C, типы выходного файла – Executable, тип проекта – STM32Cube; нажмите Finish. На сообщение необходимо ответить Yes.

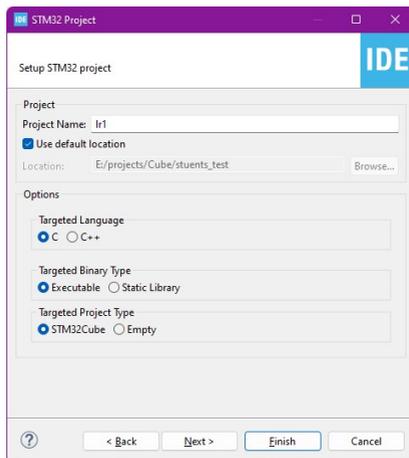


Рисунок 4. Окно настроек проекта

## Знакомство со средой разработки

На рисунке 5 изображена среда разработки STM32CubeIDE, которая используется для создания проектов, конфигурации и отладки микроконтроллеров STM32. В рамках лабораторных работ по программированию микроконтроллеров эта среда предоставляет удобный графический интерфейс, систематизированный набор инструментов и подробные средства отладки, что упрощает обучение и ускоряет разработку встроенных систем. Область «Project Explorer» (слева) отображает структуру текущего проекта, включая папки с исходными файлами, заголовочными файлами, конфигурационными файлами и т. д., что позволяет быстро переключаться между файлами, открывать их в редакторе, а также создавать новые или удалять ненужные.

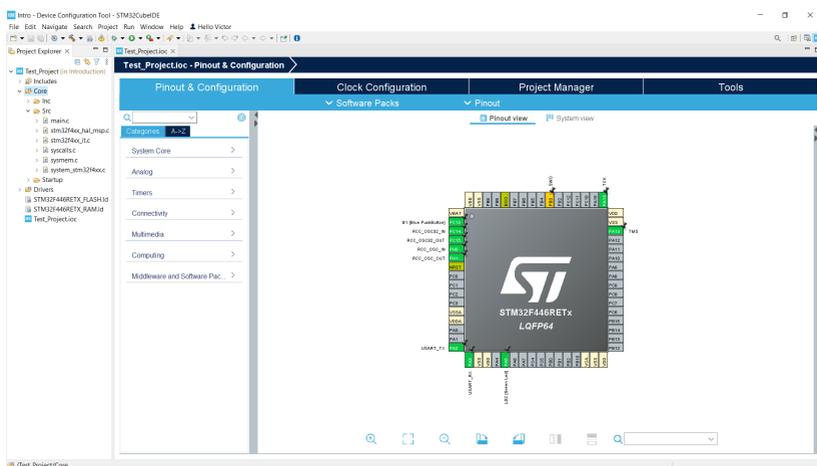


Рисунок 5. Окно визуальной конфигурации микроконтроллера

Центральная вкладка «Pinout & Configuration» содержит схематическое представление микроконтроллера (Рисунок 6), где каждая ножка (пин) показана отдельно и может быть настроена (GPIO, альтернативные функции, периферийные интерфейсы и т. д.).

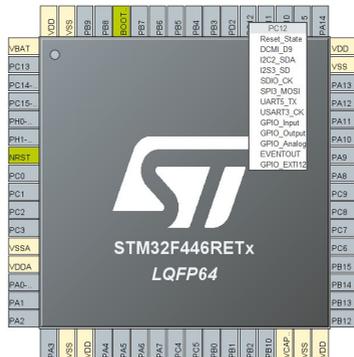


Рисунок 6. Графическое изображение микроконтроллера

В левой части окна расположены категории (System Core, Analog, Timers, Connectivity и др.), позволяющие выбрать нужный периферийный блок для конфигурации (Рисунок 7).

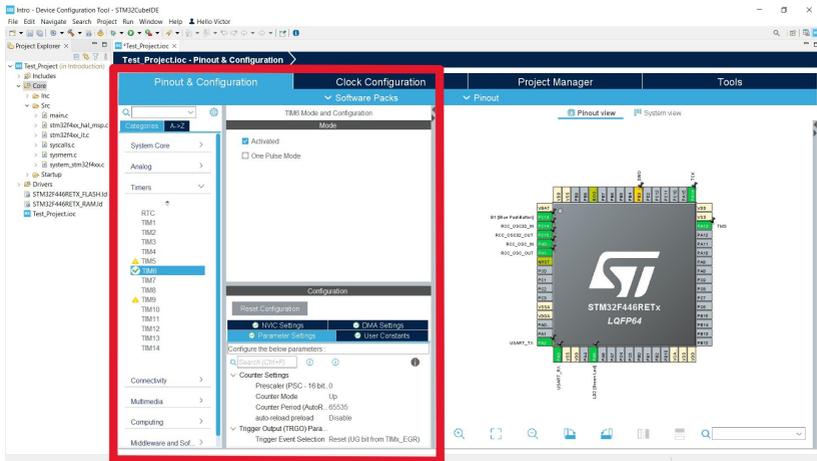


Рисунок 7. Область настроек периферийных блоков

Вкладка «Clock Configuration» (Рисунок 8) служит для настройки тактирования системы: выбора внутреннего или внешнего генератора, задания делителей тактовых сигналов и частот шин (AHB, APB) и т. д., при этом графический интерфейс

позволяет наглядно проследить цепочку генераторов и вычислить ИТОГОВЫЕ ЧАСТОТЫ

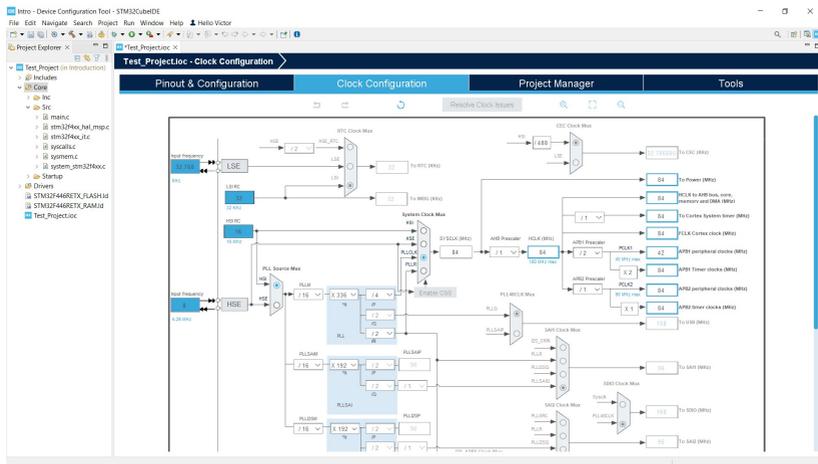


Рисунок 8. Окно настроек тактирования системы

Программирование микроконтроллера в STM32CubeIDE выполняется по следующему базовому алгоритму:

- При необходимости задайте параметры тактирования системы во вкладке Clock Configuration.
- В разделе Pinout Configuration настройте необходимые пины.
- Включите и настройте необходимые периферийные модули.
- Сохраните проект и нажмите GENERATE CODE. STM32CubeIDE автоматически сгенерирует базовый программный код, соответствующий параметрам, заданным в графическом конфигураторе.
- Напишите программу и скомпилируйте проект, нажав Build. Убедитесь, что процесс завершился без ошибок.
- Подключите микроконтроллер к компьютеру через отладчик ST-LINK и нажмите Run. Это загрузит скомпилированный код в память микроконтроллера.
- Если процесс загрузки выполнен успешно, в окне Console появится сообщение: Download verified successfully.

# Лабораторная работа №1

## Цель работы

- Получение навыков работы с периферией GPIO микроконтроллера серии F446RE.

## Аппаратное обеспечение

- Лабораторный стенд.

## Программное обеспечение

- STM32CubeIDE.

## Общие сведения

- Порты ввода-вывода (англ. General-Purpose Input/ Output, GPIO)

## Порядок выполнения

- На основе схемы лабораторного стенда определить пины, к которым подключены две кнопки (синяя и красная) и 8 светодиодов.
- В схеме лабораторного стенда изучить схему подключения кнопок.
- Разработать программу в соответствии с вариантом.
- Дополнительное задание: нажатие на кнопку должно отображаться включением подсветки кнопки (для этого нужно также определить пины, к которым подключены светодиоды кнопок).

Таблица 1. Исходные данные для работы

№ Варианта	Задание
1	<p>«Счётчик с границами»: Отображать на 8 светодиодах двоичную запись переменной (например, var), изменяемой кнопками.</p> <ul style="list-style-type: none"> <li>• Начальное значение var = 0. Диапазон var: от 0 до максимально представимого числа для 8 бит (0 до 255).</li> <li>• При нажатии красной кнопки var увеличивается на 1, если не достигнут максимум. Если var = 255, при нажатии красной кнопки значение не меняется.</li> <li>• При нажатии синей кнопки var уменьшается на 1, если var &gt; 0. Если var = 0, при нажатии синей кнопки значение не меняется.</li> </ul>
2	<p>«Счётчик с циклическим переполнением»: Отображать на 8 светодиодах двоичное число (var), аналогично Варианту 1, но с другой логикой на границах:</p> <ul style="list-style-type: none"> <li>• Начальное значение var = 0, диапазон значений от 0 до 255.</li> <li>• При нажатии красной кнопки, если var &lt; 255, увеличить его на 1. Если var = 255, при нажатии красной кнопки установить var = 0 (циклическое переполнение).</li> <li>• При нажатии синей кнопки, если var &gt; 0, уменьшить его на 1. Если var = 0, при нажатии синей кнопки установить var = 255 (циклическое задание максимума).</li> </ul>
Продолжение на следующей странице...	

Таблица 1. Исходные данные для работы

№ Варианта	Задание
3	<p>«Пошаговое включение-выключение с реверсом»:</p> <ul style="list-style-type: none"> <li>● Начальное состояние: все светодиоды выключены.</li> <li>● Красная кнопка: при каждом нажатии загорается один светодиод, двигаясь от младшего (LED0) к старшему (LED7). Таким образом, красная кнопка «перемещает» включённый светодиод вперёд по порядку. Если включённым был самый старший светодиод (LED7), следующее нажатие красной кнопки возвращает систему в состояние без горящих светодиодов.</li> <li>● Синяя кнопка: делает то же самое, но в обратном направлении – если светодиод «двигается» вперёд по красной, то синяя кнопка движет его назад.</li> <li>● Пример последовательности (кр = красная, син = синяя): <ul style="list-style-type: none"> <li>– кр: 0b00000001</li> <li>– кр: 0b00000010</li> <li>– кр: 0b00000100</li> <li>– син: 0b00000010</li> <li>– син: 0b00000001</li> <li>– син: 0b00000000</li> </ul> </li> </ul>
Продолжение на следующей странице...	

Таблица 1. Исходные данные для работы

№ Варианта	Задание
4	<p>«Нарастающее и убывающее заполнение»:</p> <ul style="list-style-type: none"> <li>● Начальное состояние: все светодиоды выключены.</li> <li>● Красная кнопка: при первом нажатии загорается LED0. При следующем нажатии дополнительно к уже горящим загорается следующий светодиод (LED1), и так далее, формируя «нарастающую» цепочку. Например: <ul style="list-style-type: none"> <li>– кр: 0b00000001</li> <li>– кр: 0b00000011</li> <li>– кр: 0b00000111</li> <li>– ... до тех пор, пока не загорятся все светодиоды (11111111).</li> </ul> </li> <li>● Синяя кнопка: работает в обратном порядке <ul style="list-style-type: none"> <li>– при нажатии синей кнопки цепочка укорачивается на один светодиод справа. Например: <ul style="list-style-type: none"> <li>– син: 0b00000011</li> <li>– син: 0b00000001</li> <li>– син: 0b00000000</li> </ul> </li> </ul> </li> </ul> <p>Таким образом, красная кнопка расширяет область включённых светодиодов, синяя – сужает.</p>
Продолжение на следующей странице...	

Таблица 1. Исходные данные для работы

№ Варианта	Задание
5	<p>«Поэтапное симметричное заполнение и очистка»:</p> <ul style="list-style-type: none"> <li>● Начальное состояние: все светодиоды выключены.</li> <li>● Красная кнопка: <ul style="list-style-type: none"> <li>– При первом нажатии загораются крайние светодиоды (LED0 и LED7): 0b10000001</li> <li>– При втором – к уже горящим добавляются ещё два крайних с каждой стороны к центру: 0b11000011</li> <li>– При следующем – 0b11100111</li> <li>– Ещё одно нажатие – 0b11111111 (все горят).</li> </ul> </li> <li>● Когда все горят, следующее нажатие красной кнопки начинает обратный процесс – гаснут крайние пары: <ul style="list-style-type: none"> <li>– кр: 0b01111110</li> <li>– кр: 0b00111100</li> <li>– и т.д., пока не погаснут все.</li> </ul> </li> <li>● Синяя кнопка выполняет тот же процесс, но в обратную сторону: если красная «расширяет», синяя «сужает», или наоборот, в зависимости от текущего состояния. То есть синяя кнопка меняет направление изменения паттерна.</li> </ul>
Продолжение на следующей странице...	

Таблица 1. Исходные данные для работы

№ Варианта	Задание
6	<p>«Комбинация»:</p> <ul style="list-style-type: none"> <li>• Начальное состояние: все светодиоды выключены.</li> <li>• При нажатии какой-либо из кнопок последовательно, начиная с LED0, набирается сообщение в двоичной форме, где красная кнопка – «единица», светодиод горит, синяя – «ноль», светодиод не горит. <ul style="list-style-type: none"> <li>– кр: 0b00000001</li> <li>– син: 0b00000010</li> <li>– син: 0b00000100</li> <li>– кр: 0b00001001</li> </ul> </li> <li>• При одновременном нажатии красной и синей кнопок сообщение сбрасывается (все светодиоды гаснут).</li> </ul>
7	<p>«Тетрис»:</p> <ul style="list-style-type: none"> <li>• Начальное состояние: все светодиоды выключены.</li> <li>• Реализовать одномерный восьмибитный тетрис, где каждая «фигура» это один горящий светодиод.</li> <li>• Когда все светодиоды заполняются «фигурами», система переходит в начальное состояние (все светодиоды гаснут).</li> </ul>
8	<p>«Код Грея»: Реализовать задание из Варианта №1, но при этом, число отображается в форме кода Грея.</p>

## Контрольные вопросы:

- Что такое GPIO (General-Purpose Input/Output) и для чего он используется в микроконтроллере STM32?
- Какие существуют основные режимы работы GPIO?
- Какой смысл имеют режимы “Push-Pull” и “Open-Drain”?
- В каком случае на выходе стоит применять встроенные подтяжки (pull-up/pull-down), а когда – внешние резисторы?
- Как правильно считывать состояние входа и как избежать “дребезга контактов” (contact bounce)?
- Какое напряжение может выдавать и принимать пины в цифровых режимах?

## Лабораторная работа №2

### Цель работы

- Получение навыков работы с периферией USART микроконтроллера серии F446RE.

### Аппаратное обеспечение

- Лабораторный стенд,
- Логический анализатор.

### Программное обеспечение

- STM32CubeIDE,
- Terminal.

### Порядок выполнения

- Инициализировать пины кнопок и светодиодов из предыдущей лабораторной работы.
- На основе схемы стенда определить, какой USART используется для передачи данных на компьютер.
- Инициализировать пины для линий USART (TX и RX) и настроить сам USART (основные настройки: задать асинхронный режим работы, установить скорость передачи данных, включить глобальные прерывания от USART).
- Разработать протокол обмена данными с компьютером. Компьютер выступает главным устройством, МК отвечает на запросы. Пакет данных от компьютера должен содержать байты, идентификаторы начала и конца посылки.

Должны быть реализованы следующие команды:

Команда	Ответ
Проверка связи (ring)	Повторяет принятую команду
Включить все светодиоды	Повторяет принятую команду
Выключить все светодиоды	Повторяет принятую команду
Включение определенного набора светодиодов в соответствии с вариантом	Повторяет принятую команду

Результат работы программы необходимо продемонстрировать при помощи утилиты Terminal и логического анализатора.

Таблица 2. Исходные данные для работы

№ Варианта	Задание
1	Скорость передачи данных: 9600 бит/с. Отправляется десятичное число (0–8). В соответствии с этим числом загорается нужное количество светодиодов.
2	Скорость передачи данных: 14400 бит/с. Светодиоды управляются по отдельности, отправляется команда формата: № светодиода, состояние.
3	Скорость передачи данных: 19200 бит/с. Состояние светодиодов отправляется в виде двоичного числа, например, 0b11100011.
4	Скорость передачи данных: 28800 бит/с. Отправляется число десятичное (0–255). Светодиоды отображают запись этого числа в двоичной форме.
5	Скорость передачи данных: 38400 бит/с. Отправляется десятичное число (0–8). После этого загораются все светодиоды, кроме светодиода под этим номером.
6	Скорость передачи данных: 56000 бит/с. Отправляется 0 или 1. Состояние светодиодов отображает последовательную отправку нулей и единиц.
Продолжение на следующей странице...	

Таблица 2. Исходные данные для работы

№ Варианта	Задание
7	Скорость передачи данных: 57600 бит/с. Отправляется десятичное число (0–7), которое соответствует длине шага (количество светодиодов), который должен сделать горящий светодиод.
8	Скорость передачи данных: 115200 бит/с. Отправляется десятичное число (0–255). Светодиоды отображают запись этого числа в форме кода Грея.

### Контрольные вопросы:

- Что такое UART и для чего он предназначен?
- Как формируется и интерпретируется кадр данных UART (start bit, data bits, parity bit, stop bit)?
- Какие основные параметры конфигурации UART необходимо настроить?
- Зачем использовать прерывания при обмене данными по UART, если можно работать в режиме опроса (Polling)?
- Какие типы прерываний обычно генерируются периферией UART (прерывание по приёму байта, по окончанию передачи, по ошибкам)?
- Как отличить прерывание по приёму данных от прерывания по ошибке или по окончанию передачи?

## Лабораторная работа №3

### Цель работы

- Получение навыков работы с перифериями Timer и DAC микроконтроллера серии F446RE.

### Аппаратное обеспечение

- Лабораторный стенд,
- Осциллограф.

### Программное обеспечение

- STM32CubeIDE,
- Terminal,
- MultiVirAnalyzer.

### Порядок выполнения

- В окне графической инициализации контроллера, используя внешний источник тактовых импульсов, во вкладке Clock configuration настроить частоту процессора в соответствии с вариантом.
- На основе схемы лабораторного стенда определить пины, к которым подключен один из каналов DAC.
- В окне графической инициализации контроллера включить нужный канал DAC.
- Выбрать таймер для формирования прерываний и определить его входную частоту. Рассчитать значения Prescaler и Counter Period.
- Включить глобальные прерывания от выбранного таймера.
- В main файле добавить callback-функцию таймера.
- В main функции запустить таймер и DAC.

- Разработать программу формирования синусоидального сигнала заданной частоты на выходе ЦАП. Начальные частоту и амплитуду сигнала выбрать в соответствии с вариантом.
- Разработать функцию для изменения частоты и амплитуды сигнала по команде через UART.

Результат работы программы необходимо продемонстрировать при помощи осциллографа.

Таблица 3. Исходные данные для работы

№ Варианта	Частота CPU, МГц	Начальная частота синусоиды, Гц	Начальная амплитуда синусоиды, В
1	100	5	0.4
2	110	10	0.8
3	120	15	1.2
4	130	20	1.6
5	140	25	2
6	150	30	2.4
7	160	35	2.8
8	170	40	3.2

### Контрольные вопросы:

- Для чего в микроконтроллере используются таймеры и чем отличаются базовые, универсальные (general-purpose) и передовые (advanced) таймеры?
- Как связаны тактовая частота ядра микроконтроллера, шина и входная частота таймера?
- Какие основные параметры настраиваются при работе с таймером?
- Каков общий принцип работы цифро-аналогового преобразователя?
- Что такое разрешение ЦАП (например, 8, 12 бит) и как оно влияет на качество выходного сигнала?

## Лабораторная работа №4

### Цель работы

- Получение навыков работы с периферией ADC микроконтроллера серии F446RE.

### Аппаратное обеспечение

- Лабораторный стенд
- Осциллограф.

### Программное обеспечение

- STM32CubeIDE,
- MultiVirAnalyzer,
- Matlab Simulink.

### Порядок выполнения

- Повторить шаги 1–8 из лабораторной работы №3.
- Изменить скорость передачи данных USART на 1000000 бит/с.
- На основе схемы лабораторного стенда определить пины, к которым подключен один из каналов ADC.
- В окне графической инициализации контроллера включить нужный канал ADC.
- Выбрать таймер для формирования прерываний для отправки данных через USART в Simulink и определить его входную частоту. Рассчитать значения Prescaler и Counter Period. Частота прерывания для отправки данных должна быть равна 1кГц.
- Включить глобальные прерывания от выбранного таймера.
- В main функции запустить выбранный таймер.

- В обработчике прерывания таймера необходимо сформировать пакет данных для отправки в matlab, отправить пакет и запустить преобразования на АЦП.

Результат работы программы необходимо продемонстрировать при помощи осциллографа.

Таблица 4. Исходные данные для работы

№ Варианта	Частота CPU, МГц	Начальная частота синусоиды, Гц	Начальная амплитуда синусоиды, В
1	100	5	0.4
2	110	10	0.8
3	120	15	1.2
4	130	20	1.6
5	140	25	2
6	150	30	2.4
7	160	35	2.8
8	170	40	3.2

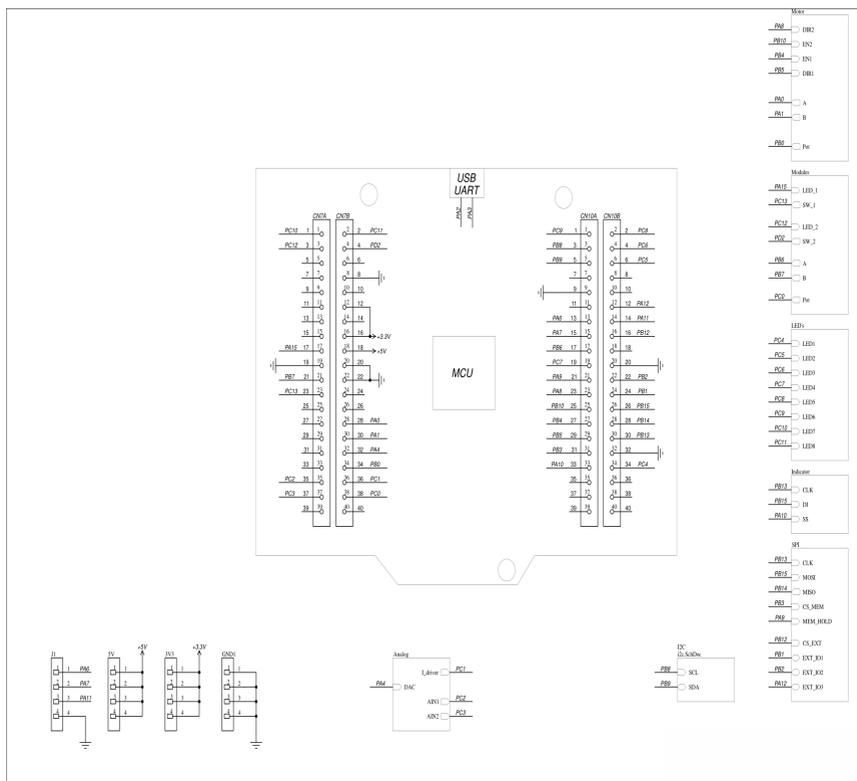
### Контрольные вопросы:

- Каков общий принцип преобразования аналогового сигнала в цифровое значение?
- Что такое разрядность (разрешение) АЦП (например, 12, 16 бит) и как она влияет на точность измерений?
- Как связаны опорное напряжение ( $V_{ref}$ ) и максимальное входное напряжение, измеряемое АЦП?
- В чём разница между одиночным (Single) и непрерывным (Continuous) режимами?
- Как помехи (шумы) могут искажать показания АЦП и каким образом можно снизить их влияние?

## СПИСОК ЛИТЕРАТУРЫ

- 1 STM32CubeIDE. Integrated Development Environment for STM32. URL: <https://www.st.com/en/development-tools/stm32cubeide.html> (дата обращения: 13.01.2025).
- 2 NUCLEO-F446RE. STM32 Nucleo-64 development board with STM32F446RE MCU. URL: <https://www.st.com/en/evaluation-tools/nucleo-f446re.html> (дата обращения: 13.01.2025).
- 3 STM32F446xC/E. Datasheet – production data // STMicroelectronics. 2021. 189 p.
- 4 RM0390. Reference manual // STMicroelectronics. 2021. 1347 p.
- 5 UM1725. Description of STM32F4 HAL and low-layer drivers UM1725 User manual // STMicroelectronics. 2023. 2227 p.
- 6 Ключарёв, А. А. Программирование микроконтроллеров STM32 : учебное пособие / А. А. Ключарёв, К. А. Кочин, А. А. Фоменкова. — Санкт-Петербург : ГУАП, 2023. — 196 с.

# Приложение А. Электрические схемы лабораторного стенда



Общая схема подключений

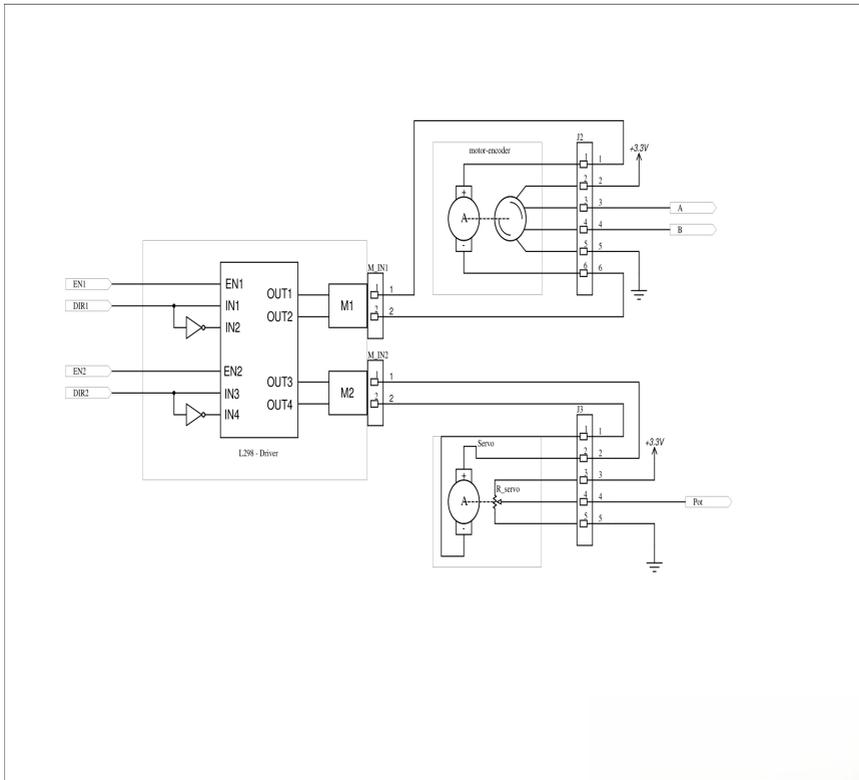


Схема подключения ДПТ

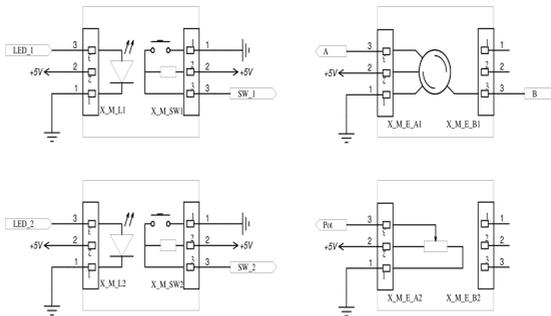
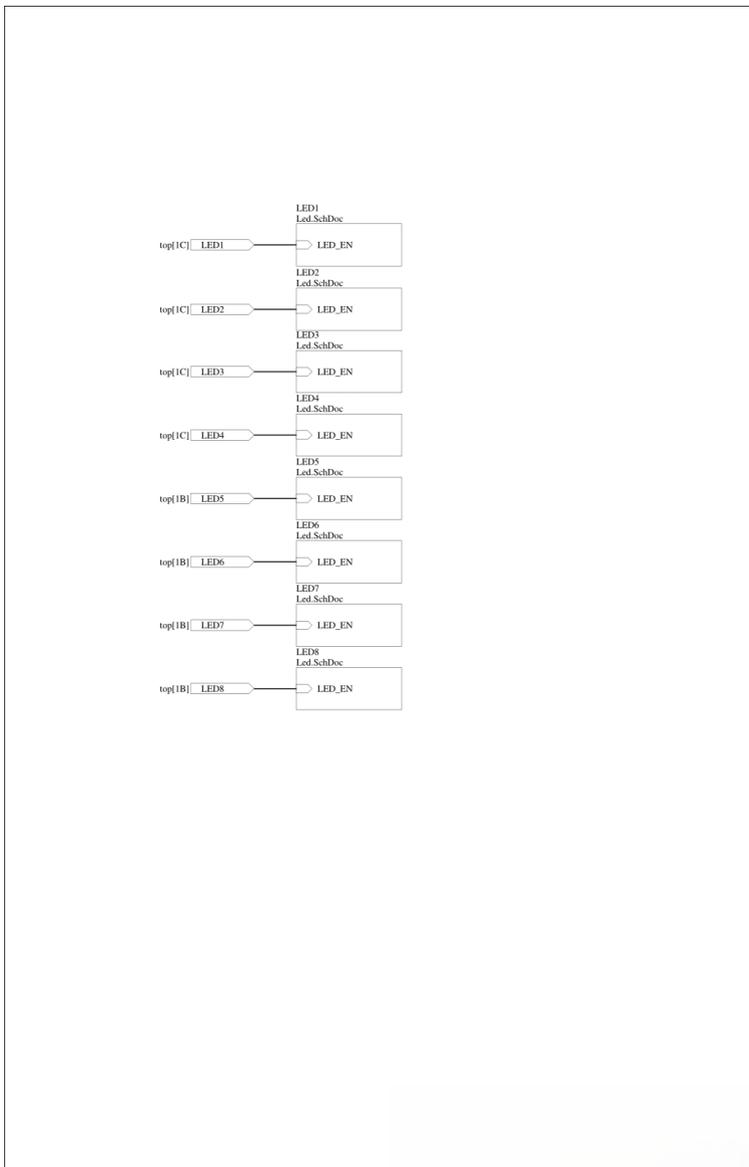


Схема подключения модулей



Блок светодиодов

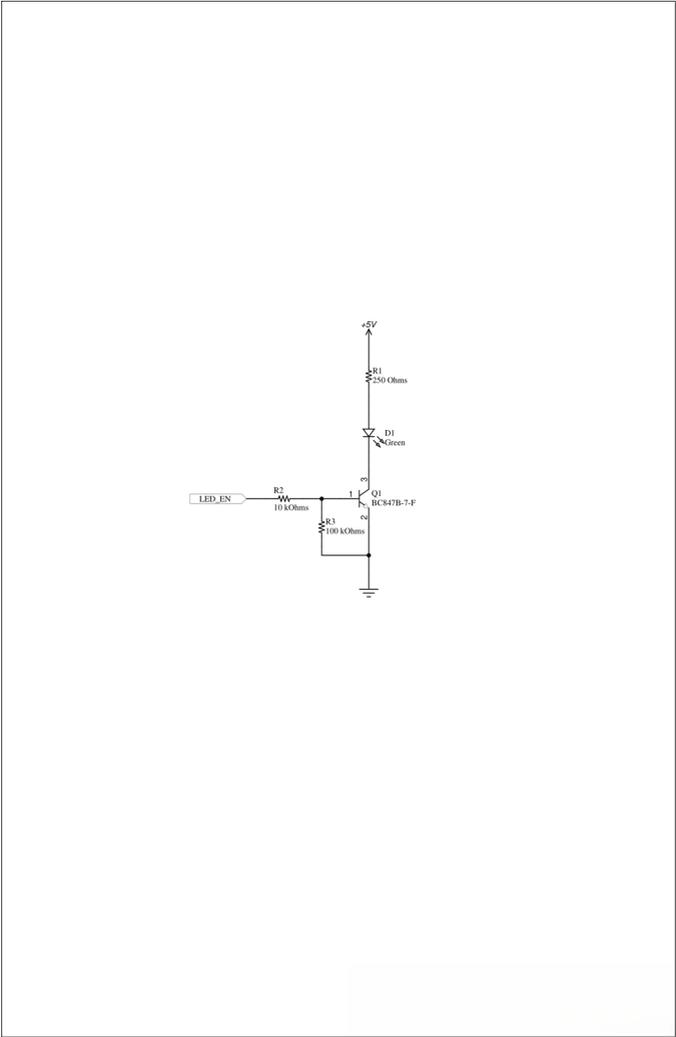


Схема подключения одного светодиода в блоке

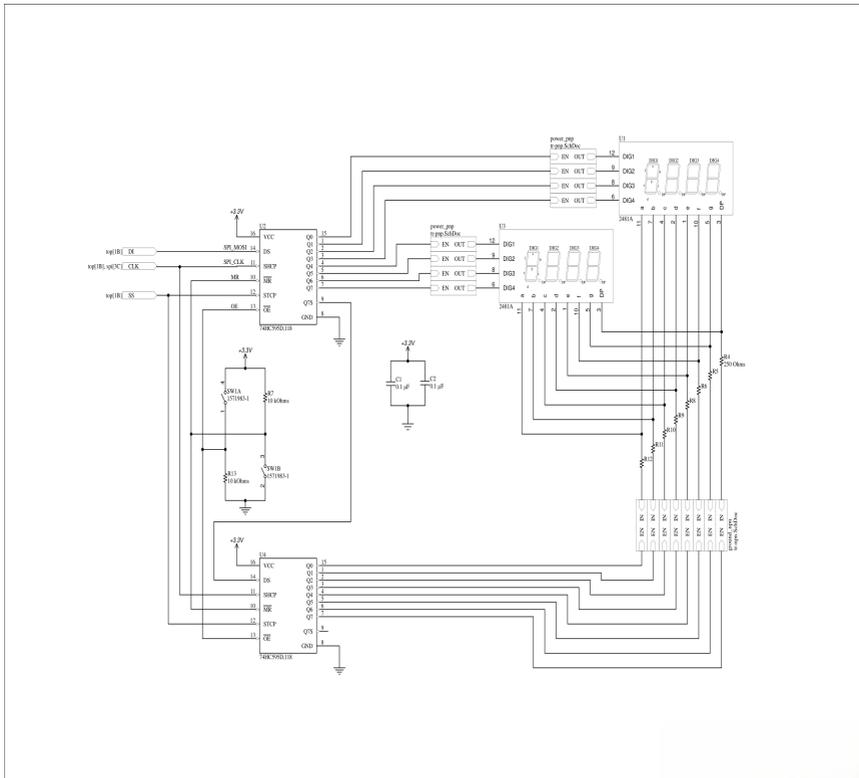
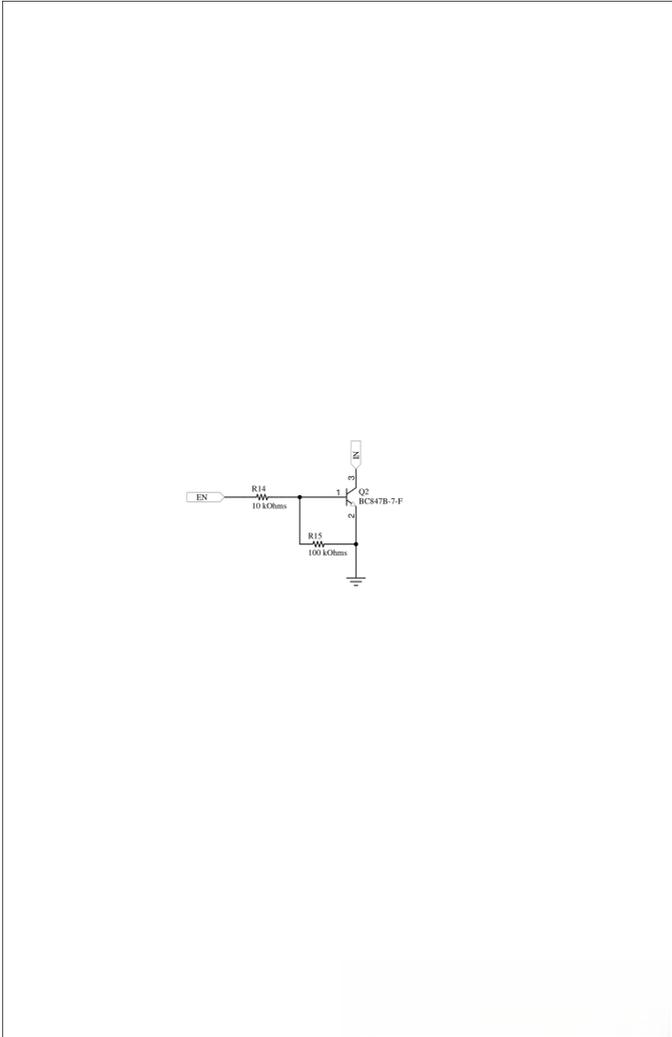
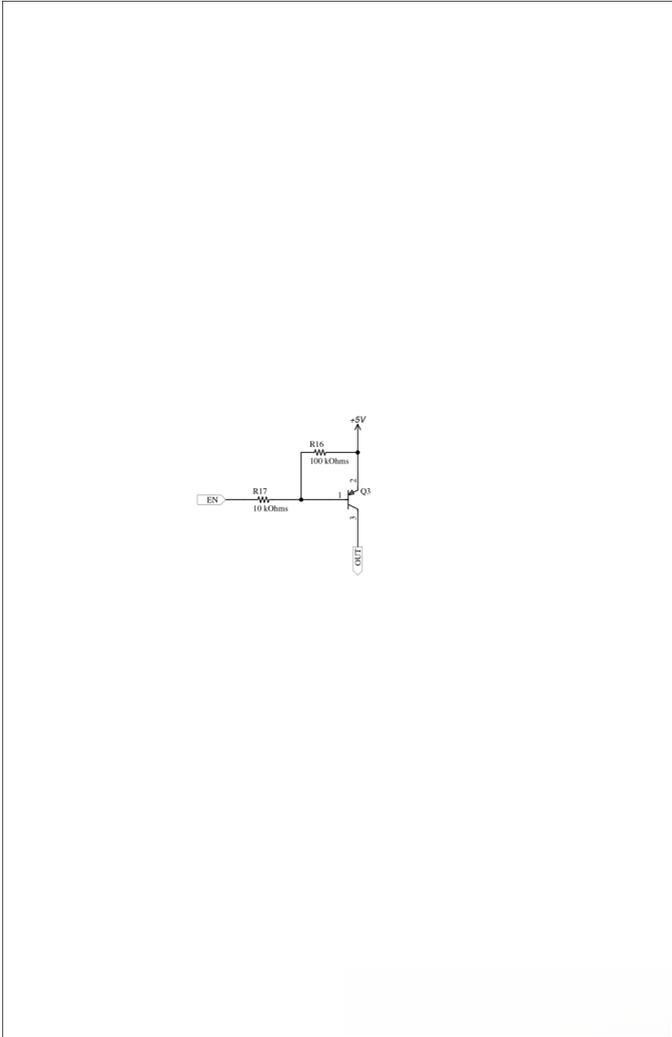


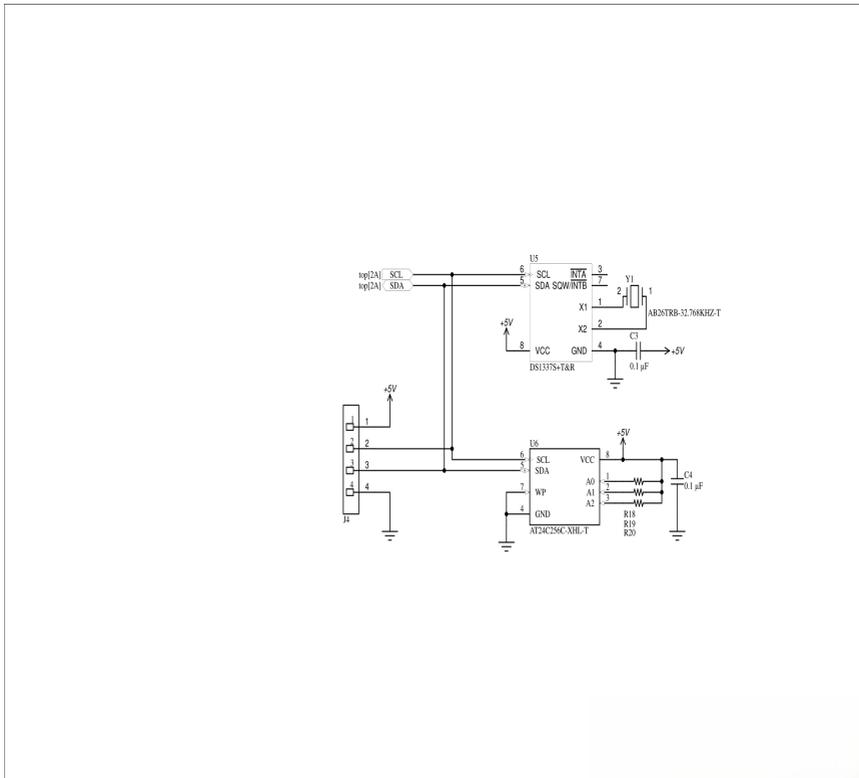
Схема подключения 7-сегментных индикаторов



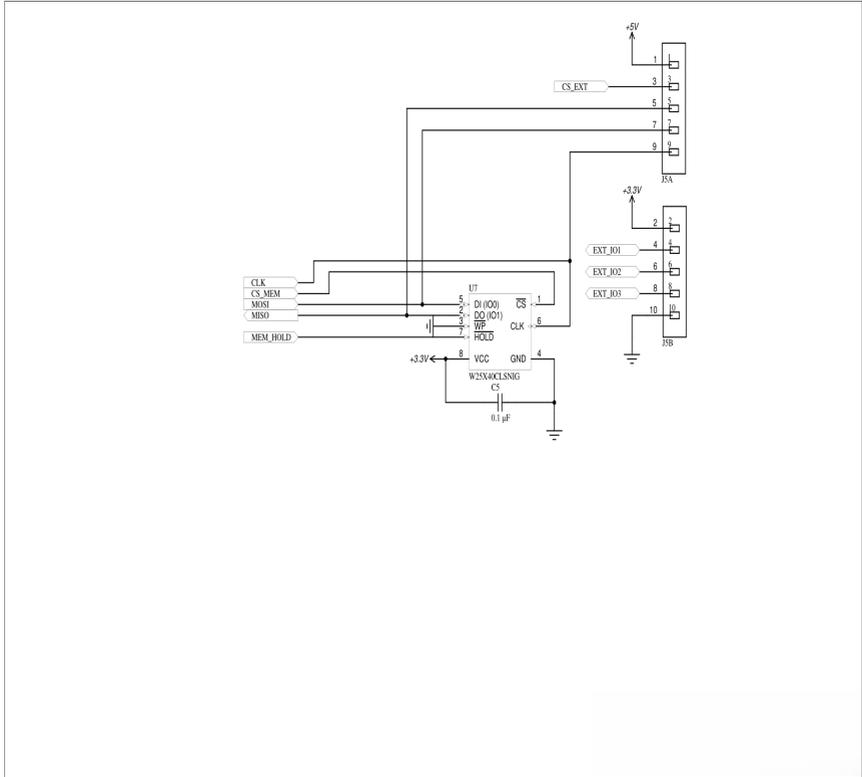
Драйвер 1 7-сегментного индикатора



Драйвер 2 7-сегментного индикатора



Подключение устройств по шине i2c



Подключение устройств по шине SPI

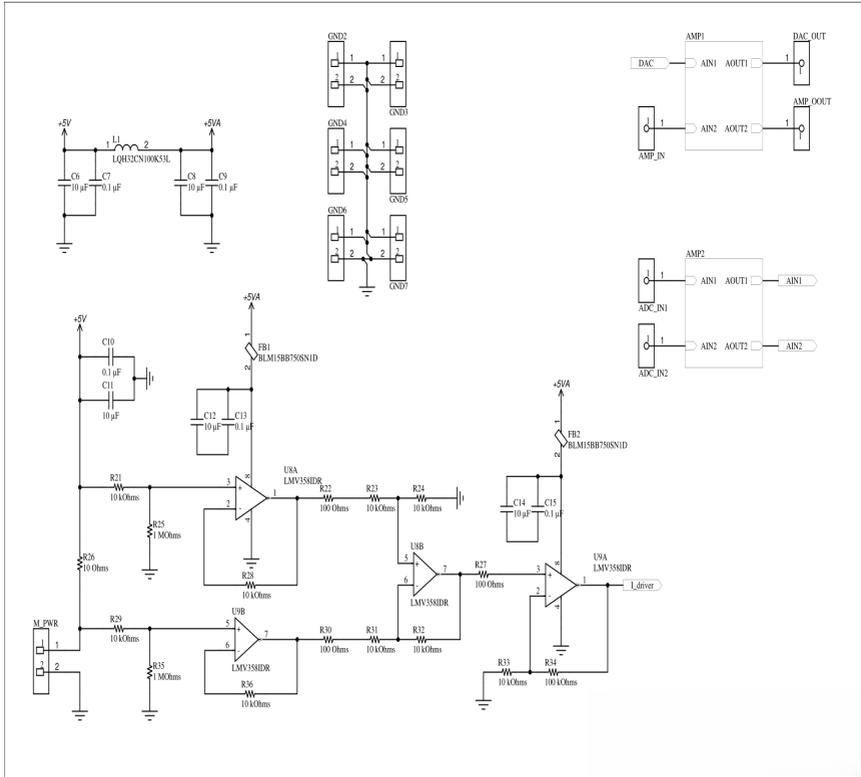
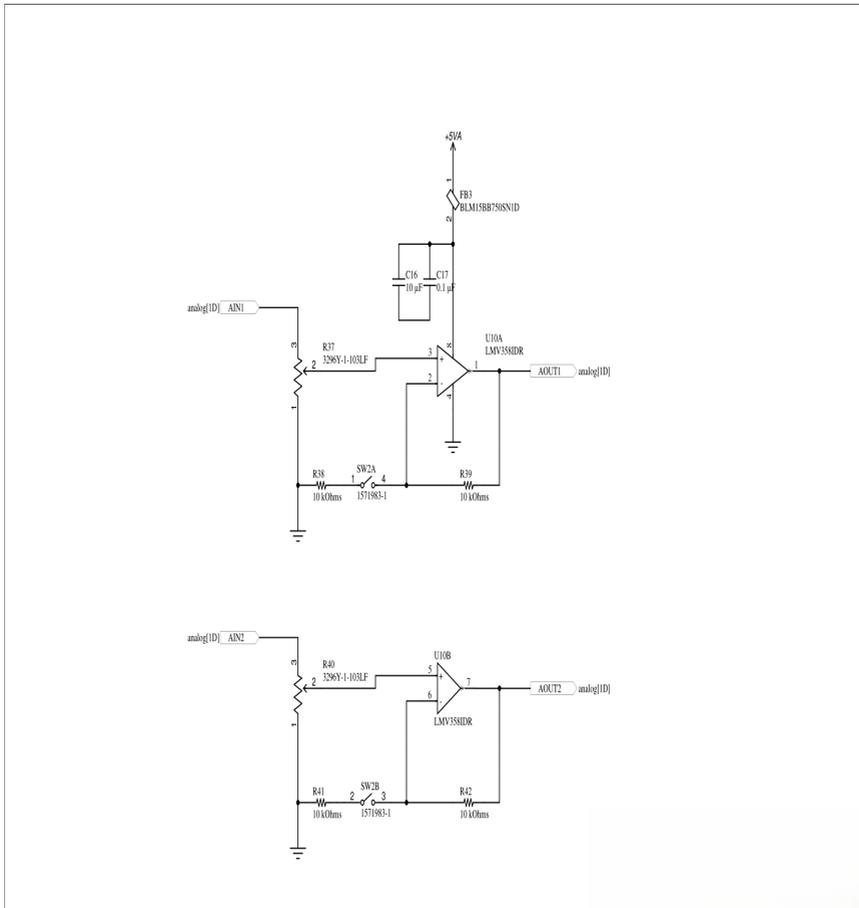


Схема работы с аналоговыми сигналами



Блок ОУ

Власов Сергей Михайлович  
Жданов Виктор Андреевич  
Маргун Алексей Анатольевич  
Зименко Константин Александрович  
Сулейман Лейла  
Галкина Дарья Алексеевна

**Программирование STM32 на HAL. Часть 1**  
**Лабораторный практикум**

В авторской редакции  
Редакционно-издательский отдел Университета ИТМО  
Зав. РИО Н.Ф. Гусарова  
Подписано к печати  
Заказ №  
Тираж  
Отпечатано на ризографе

**Редакционно-издательский отдел  
Университета ИТМО**

197101, Санкт-Петербург, Кронверкский пр., 49, литер А