



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

*к курсовой работе*  
*по дисциплине «Микропроцессорные системы»*  
*на тему:*

**Устройство для измерения скорости чтения**

Студент

\_\_\_\_\_  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Т.А. Валиуллин

\_\_\_\_\_  
(И.О. Фамилия)

Руководитель

\_\_\_\_\_  
(Подпись, дата)

И.Б. Трамов

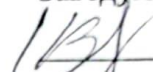
\_\_\_\_\_  
(И.О. Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6

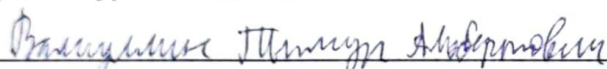
 А.В. Пролетарский

« 2 » сентября 2023 г.

## ЗАДАНИЕ на выполнение курсовой работы

по дисциплине Микропроцессорные системы

Студент группы ИУ6-71Б



**Тема курсовой работы:** Устройство для измерения скорости чтения

Направленность курсовой работы: учебная

Источник тематики (кафедра, предприятие, НИР): кафедра

График выполнения работы: 25% – 4 нед., 50% – 8 нед., 75% – 12 нед., 100% – 16 нед.

### Техническое задание:

Разработать МК-систему для измерения скорости чтения человека. Необходимо отобразить таблицу Шульте на ЖК-дисплее. Предусмотреть отображение нескольких вариантов таблиц (текущий вариант выбирает испытуемый). Пользователь фиксирует время старта и окончания теста при помощи управляющих кнопок. Предоставить возможность сохранять время испытания в памяти. При установлении нового рекорда, посылать сигнал на SPEAKER.

По результатам нескольких испытаний определить среднее время чтения испытуемого и выводить его на индикатор. Результаты испытаний передать по последовательному каналу в ПЭВМ.

Выбрать наиболее оптимальный вариант МК. Выбор обосновать.

Разработать схему, алгоритмы и программу. Отладить проект в симуляторе или на макете. Оценить потребляемую мощность. Описать принципы и технологию программирования используемого микроконтроллера.

### Оформление курсовой работы:

1. Расчетно-пояснительная записка на 30-35 листах формата А4.

2. Перечень графического материала:

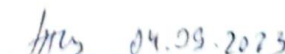
а) схема электрическая функциональная;

б) схема электрическая принципиальная.

Дата выдачи задания: «4» сентября 2023 г.

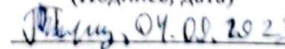
Дата защиты: «20» декабря 2023 г.

Руководитель курсовой работы

  
(Подпись, дата)

Н.Б. Трамов  
(И.О.Фамилия)

Студент

  
(Подпись, дата)

П.А. Власов  
(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах; один выдается студенту, второй хранится на кафедре

## РЕФЕРАТ

РПЗ ... страниц, ... рисунков, ... таблиц, ... источников, ... приложения.

МИКРОКОНТРОЛЛЕР STM32, СИСТЕМА, СКОРОСТЬ ЧТЕНИЯ, ИЗМЕРЕНИЕ, UART, SPI, ЖИДКОКРИСТАЛЛИЧЕСКИЙ ДИСПЛЕЙ.

Объектом разработки является устройство для измерения скорости чтения.

Цель работы – разработка устройства ограниченной сложности, создание модели этого устройства и необходимой технической документации к нему.

Результатом работы является комплект конструкторской документации для изготовления устройства, исходные коды программ для программирования памяти микроконтроллера и модель устройства в среде разработки Proteus 8.

Устройство должно обладать следующими техническими характеристиками:

- Ввод размера таблицы Шульте;
- Отображение таблицы Шульте;
- Подсчет времени чтения таблицы Шульте;
- Сохранение времени чтения в память;
- Подсчет среднего времени чтения;
- Сохранение рекордного времени чтения;
- Вывод сигнала на SPEAKER;
- Отправка времени чтения и среднего времени в ПЭВМ.

## **ОБОЗНАЧЕНИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ**

МПС – Микропроцессорные системы.

ЖК-дисплей – жидкокристаллический дисплей.

МК – микроконтроллер.

Proteus 8 - пакет программ для автоматизированного проектирования (САПР) электронных схем.

ТЗ – техническое задание.

TFT (Thin-Film Transistor) – Тонко-Пленочный Транзистор.

UART (Universal asynchronous receiver/transmitter) – последовательный универсальный синхронный/асинхронный приемопередатчик.

SPI (Serial Peripheral Interface) – интерфейс для взаимодействия МК с внешними устройствами.

УГО – условное графическое обозначение.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	6
1 Конструкторская часть .....	7
1.1 Анализ требований технического задания для выбора МК.....	7
1.2 Проектирование функциональной схемы.....	11
1.2.1 Микроконтроллер STM32F103C8T6 .....	11
1.2.2 Используемые элементы и интерфейсы .....	13
1.2.3 Используемые порты .....	15
1.2.4 Организация памяти МК .....	16
1.2.5 Расчет экрана и выбор дисплея.....	17
1.2.6 Использование интерфейса SPI для ЖК-дисплея.....	19
1.2.7 Прием данных от ПЭВМ .....	26
1.2.8 Использование интерфейса USART для работы с ПЭВМ .....	27
1.2.9 Использование таймера для генерации сигнала для динамика.....	35
1.2.10 Построение функциональной схемы.....	38
1.3 Построение принципиальной схемы.....	39
1.3.1 Работа с устройством через дисплеи.....	39
1.3.2 Описание и расчет алгоритма генерации таблицы .....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	43

## **ВВЕДЕНИЕ**

В данной работе производится разработка измерения скорости чтения. В процессе выполнения работы проведён анализ технического задания, создана концепция устройства, разработаны электрические схемы, построен алгоритм и управляющая программа для МК, выполнено интерактивное моделирование устройства.

Система состоит из МК, четырех кнопок для управления процессом измерения скорости чтения, кнопки RESET, ЖК-дисплея для отображения таблицы и информации для пользователя, динамика для вывода сигнала о рекордном времени чтения, виртуального терминала для симуляции ввода/вывода с ПЭВМ.

Актуальность разрабатываемого устройства измерения скорости чтения заключается в том, что с его метод измерения скорости чтения с использованием таблиц Шульте позволяет оценить не столько навыки чтения, сколько оценить концентрацию, скорость чтения и память пользователя.

Для образовательных учреждений это может быть инструментом для оптимизации методов обучения, в то время как в медицинской сфере - средством для диагностики некоторых расстройств чтения. Такой подход также применим для оценки эффективности чтения пользователя и его умственных способностей.

## **1 Конструкторская часть**

### **1.1 Анализ требований технического задания для выбора МК**

Поставленная задача требует более сложного вывода на дисплей, чем просто вывод текста. После анализа требований было решено позволить пользователю выбрать размер таблицы от 3 до 7 включительно. Таким образом отображаться должно было от 9 до 49 чисел, большинство из которых двузначные. Более того простого вывода чисел на ЖК-дисплей недостаточно. Необходим специальный вывод в виде таблицы.

Так как устройство создано для измерения скорости чтения пользователя, выводимые числа должны быть просто считывать, нельзя допускать случаи, когда числа налезают друг на друга, или стоят слишком близко. В связи с этим, было решено отделить их при помощи линий так, чтобы получалась полноценная таблица.

Таким образом вывод на ЖК-дисплей необходим достаточно сложный и специфицированный. Для выбора размера таблицы должен присутствовать вывод текста для создания небольшого меню. Должна быть также реализована возможность заполнения ЖК-дисплея цветом целиком для завершения игры и вывода линий для четкости таблицы.

В связи с этим решено использовать TFT-дисплей. Это экран, созданный на основе жидких кристаллов и под управлением тонкопленочных транзисторов. Подобные дисплеи задействуются в самых разных устройствах для отображения текста и изображений. Их можно встретить в компьютерах, ноутбуках, смартфонах, электронных книгах и других девайсах [1].

Конструкция TFT ЖК-дисплея представлена следующим образом. Слои накладываются друг на друга, поэтому его часто сравнивают с сэндвичем. Его основная часть состоит из: тонкопленочного транзистора, цветного фильтра и слоя жидких кристаллов. Дисплей обеспечивает более высокую яркость и четкость изображения по сравнению с более старыми типами LCD.

TFT ЖК-дисплей, будучи достаточно бюджетным вариантом четкого и специфического вывода, является идеальным вариантом для поставленной задачи.

Скорость вывода данных на дисплей также важна. Это связано с тем, что по ТЗ программа задумана для считывания времени и вычисления среднего времени, а также подсчета среднего времени. При таких условиях работы медленный вывод таблицы на экран не подходит, по следующим причинам:

- Точность измерения времени – если отображение таблицы занимает слишком много времени, измерение времени чтения пользователя может быть неточным. Быстрое обновление экрана позволяет точнее отслеживать, сколько времени пользователь проводит при чтении таблицы.

- Сохранение пользовательского внимания – пользовательское внимание может снижаться, если таблица Шульте отображается медленно. Быстрое обновление экрана помогает поддерживать интерес пользователя и предотвращать отвлечение в процессе выполнения задания.

- Психологический аспект – медленное отображение таблицы может вызвать чувство неудовлетворенности у пользователя. Быстрое и плавное обновление, наоборот, может улучшить восприятие программы и создать положительный опыт использования.

Весь проведенный анализ свидетельствует о том, что для задачи необходим микроконтроллер с высокой производительностью, скоростью работы с дисплеем и возможностью работы с TFT-дисплеем.

Таким образом, основными критериями выбора МК после анализа технического задания были выбраны поддержка более точной работы с ЖК-дисплеем и высокая производительность МК для наиболее быстрого отображения картинки на ЖК-дисплее. Для поставленной задачи МК должен быть способен так же поддерживать интерфейсы USART для взаимодействия с ПЭВМ, а именно отправки по результатам нескольких испытаний среднего



времени чтения испытуемого на ПЭВМ, и SPI для взаимодействия с ЖК-дисплеем.

Семейство микроконтроллеров STM32 отлично взаимодействует с TFT ЖК-дисплеями благодаря своей мощности, богатым периферийным возможностям и широким набором интерфейсов и библиотек взаимодействия с различными представителями данного вида дисплеем. Микроконтроллеры STM32 поддерживают высокие частоты тактирования, что важно для быстрого обновления графического содержимого на дисплее. Богатство периферийных блоков, таких как таймеры и SPI/I2C, обеспечивает удобные средства для управления TFT-дисплеями, а наличие встроенной графической библиотеки или поддержки сторонних библиотек упрощает программирование и визуализацию данных.

В то же время в связи с низкими функциональными требованиями устройства были рассмотрены в первую очередь бюджетные варианты микроконтроллеров.

Серии микроконтроллеров семейства STM32 делятся на 4 типа:

1. Высокопроизводительные – F2, F4, F7, H7
2. Широкого использования – F0, G0, F1, F3, G4
3. Сверхнизкого потребления – L0, L1, L4+, L5
4. Беспроводные – WB, WL

Из всех типов наиболее подходящий – тип широкого использования, так как в задаче используется несколько внешних устройств, использующих различные интерфейсы. МК широкого использования обладают достаточной производительностью и являются достаточно экономным вариантом, что делает их лучшим вариантом, а использование высокопроизводительных или сверхнизкого потребления скорее является неоправданным излишком. Беспроводные МК в данной задаче рассматривать не целесообразно.

Среди серий МК широкого использования выбрана золотая середина – серия F1. МК данной серии обладают достаточными характеристиками, необходимыми для решения поставленной задачи, и в то же время не

являются столь дорогими, как серии F3, G4, цены на которые выше примерно в два раза.

Среди МК серии STM32F1 были выделены четыре наиболее популярные модели: STM32F103C8T6, STM32F103RB, STM32F103VET6, STM32F103ZET6. Их основные характеристики приведены в таблице 1.

Таблица 1 – Сравнительная таблица микроконтроллеров серии STM32F1

<b>Микроконтроллер</b>	<b>Flash (Кб)</b>	<b>RAM (Кб)</b>	<b>Частота (МГц)</b>	<b>Число пинов ввода-вывода</b>
STM32F103C8T6	64	20	72	37
STM32F103RB	128	20	72	50
STM32F103VET6	512	64	72	80
STM32F103ZET6	512	64	72	112

Указанные в таблице микроконтроллеры отсортированы сверху вниз в порядке роста памятей и количества пинов и, соответственно, цены. Слишком большое количество памяти или количества пинов МК для данного устройства не нужны, 64 Кбайт Flash памяти и 20 Кбайт RAM памяти будет более чем достаточно. По этой причине выбирать что-то кроме STM32F103C8T6 не выгодно, потому что это будет более затратно.

Еще одной важной причиной для выбора МК STM32F103C8T6 является факт того, что данный МК был изучен во время курса МПС в 6 семестре. Более того мною была проведена физическая лабораторная работа с отладочной платой с этим МК.

Таким образом, выбранный для устройства МК – STM32F103C8T6. У меня есть реальный опыт работы с ним, он обладает более чем достаточными характеристиками для решения поставленной задачи и в то же время является бюджетным вариантом.

## **1.2 Проектирование функциональной схемы**

### **1.2.1 Микроконтроллер STM32F103C8T6**

В настоящей работе выбран микроконтроллер STM32F103C8T6. STM32 – семейство 32-битных микроконтроллеров, производимых фирмой STMicroelectronics. Контроллеры этого семейства основаны на различных ядрах ARM, в зависимости от серии. Микроконтроллеры серии STM32F1 имеют ядро ARM Cortex-M3, сопряженное системой внутренних шин с периферийными устройствами [2].

В составе МК имеются:

- процессорное ядро ARM Cortex-M3 с максимальной тактовой частотой 72 МГц и контроллером прерываний NVIC (Nested vectored interrupt controller);
- Flash-память объемом 64 Кбайт;
- SRAM объемом 20 Кбайт;
- 16-разрядные порты ввода-вывода (GPIO – General-purpose inputs/outputs);
- 16-разрядные таймеры (TIM);
- аналого-цифровые преобразователи (ADC);
- контроллер прямого доступа к памяти (DMA);
- контроллер внешних прерываний (EXTI – External interrupt/event controller);
- интерфейсы для связи с другими устройствами: UART, SPI, I2C, CAN, USB.

Применительно к поставленной задаче семейство микроконтроллеров STM32 предоставляет высокую производительность и широкие возможности в области взаимодействия с различными видами дисплеев.

Структурная схема микроконтроллера представлена на рисунке 1, а его УГО представлено на рисунке 2.

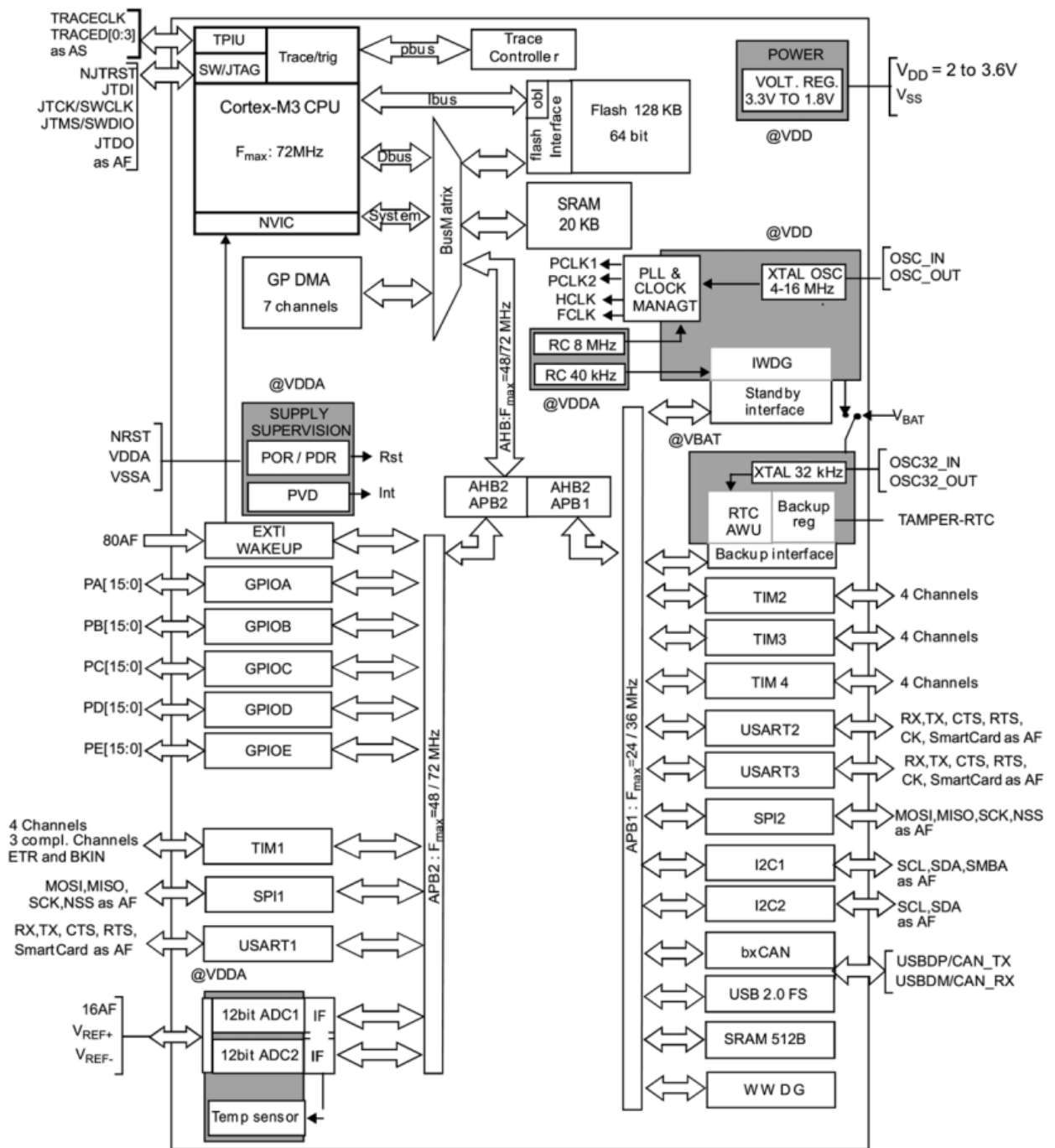


Рисунок 1 – Структурная схема микроконтроллера STM32F103C8T6

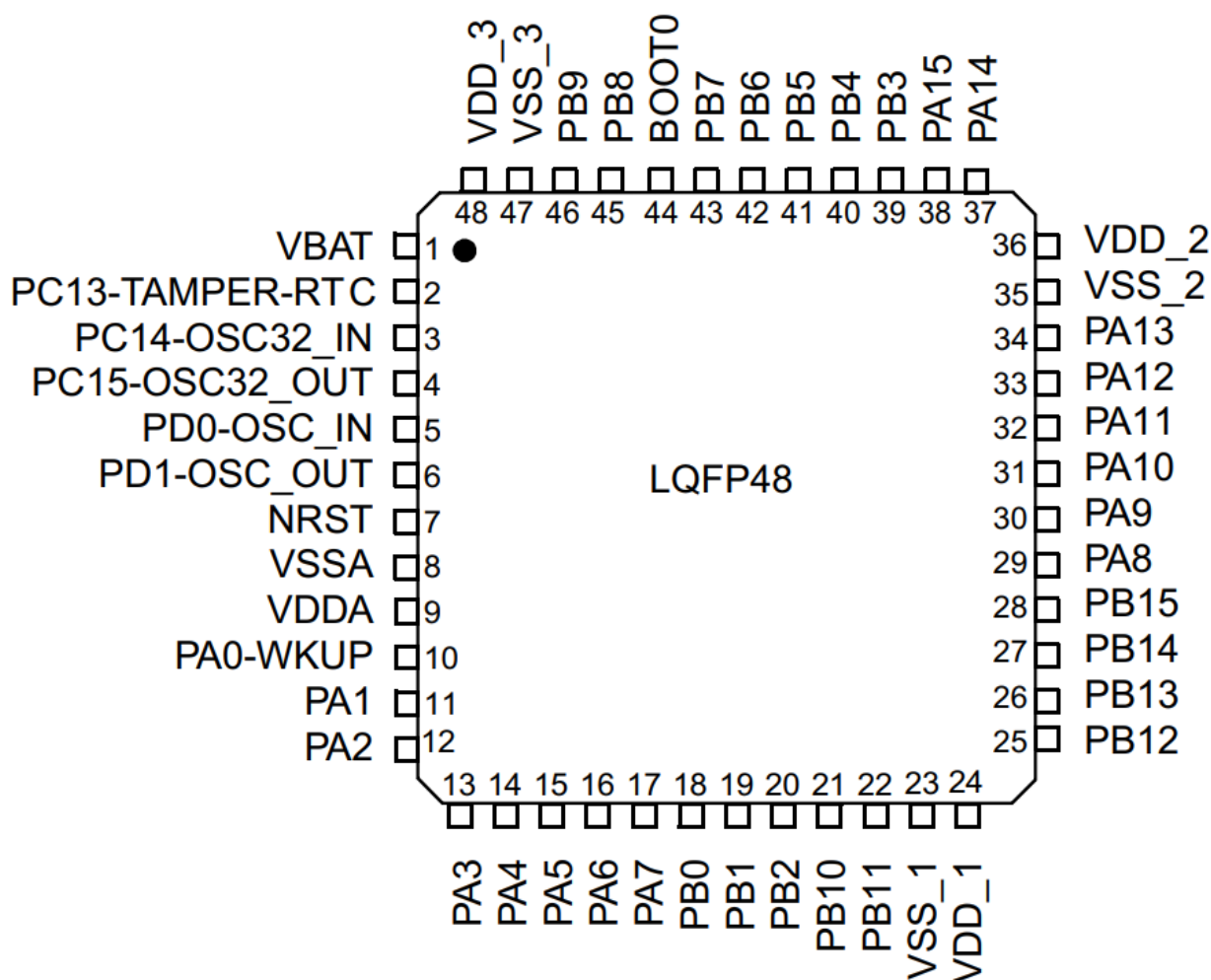


Рисунок 2 – УГО микроконтроллера STM32F103C8T6

### 1.2.2 Используемые элементы и интерфейсы

Для работы устройства измерения скорости чтения использованы не все элементы архитектуры МК STM32F103C8T6. Среди использованных элементов и интерфейсов:

- Порты А, В – использованные пины и их назначение описано в пункте 1.2.3.
- Указатель стека – играет важную роль в организации стека, используемого для управления вызовами подпрограмм. Указатель стека используется для сохранения адреса возврата и регистров при вызове функций. Это обеспечивает корректный возврат из функций и поддерживает структуру вызовов функций.

– Регистры общего назначения – предназначены для хранения операндов арифметико-логических операций, а также адресов или отдельных компонентов адресов ячеек памяти.

– АЛУ – выполняет арифметические и логические операции, обеспечивает выполнение базовых математических операций и манипуляций с битами.

– SREG – регистр состояния, содержит набор флагов, показывающих текущее состояние работы микроконтроллера.

– Память SRAM – статическая память МК, хранящая объявленные переменные.

– Память Flash – память МК, хранящая загруженную в него программу.

– Программный счетчик – указывает на следующую по исполнению команду.

– Регистры команд – содержит исполняемую в настоящий момент команду(или следующую), то есть команду, адресуемую счетчиком команд.

– Декодер – выделяет код операции и операнды команды и далее вызывает микропрограмму, исполняющую данную команду.

– Сигналы управления – нужны для синхронизации обработки данных.

– Логика программирования – устанавливает логику того, как будет вшита программа в МК.

– Генератор – генератор тактовых импульсов. Необходим для синхронизации работы МК.

– Управление синхронизацией и сбросом – обрабатывает тактовые сигналы и отвечает за сброс состояния МК.

– Прерывания – обрабатывает внешние прерывания и прерывания периферийных устройств МК (таймеров, портов и т.д.). В устройстве используются прерывания с портов для обработки нажатия кнопок и прерывания UART.

– UART (Universal Asynchronous Receiver Transmitter) – интерфейс, при помощи которого происходит передача данных в МК из ПЭВМ.

– SPI (Serial Peripheral Interface) – интерфейс для связи МК с другими внешними устройствами. В устройстве используется для прошивки МК и вывода данных на жидкокристаллический дисплей.

– Таймеры – МК содержит в себе четыре 16-ти разрядных таймеров (TIM1, TIM2, TIM3, TIM4). В устройстве используется только один канал таймера TIM2 для генерации ШИМ сигнала для динамика.

### **1.2.3 Используемые порты**

МК STM32F103C8T6 содержит в себе пять портов – A, B, C, D, E. В устройстве используются порты A и B.

Порт A используется для:

- PA1 – отправка ШИМ сигнала с таймера TIM2 на динамик;
- PA3 – отправка данных или команд на дисплей;
- PA5 – тактовый сигнал SCK для интерфейса SPI для LCD-дисплея;
- PA7 – MOSI-пин для интерфейса SPI для LCD-дисплея;
- PA9 – отправка данных по UART на ПЭВМ;
- PA10 – прием данных по UART от ПЭВМ;
- PA13 – SWDIO-пин для программатора ST-LINK V2;
- PA14 – SWCLK-пин для программатора ST-LINK V2.

Порт B используется для:

– PB6 – пин с внешним прерыванием по нажатию кнопки увеличения размера таблицы;

– PB7 – пин с внешним прерыванием по нажатию кнопки уменьшения размера таблицы;

– PB8 – пин с внешним прерыванием по нажатию кнопки начала измерения;

– PB9 – пин с внешним прерыванием по нажатию кнопки завершения измерения.

## 1.2.4 Организация памяти МК

На рисунке 3 представлена карта памяти МК STM32F103C8T6 [3].

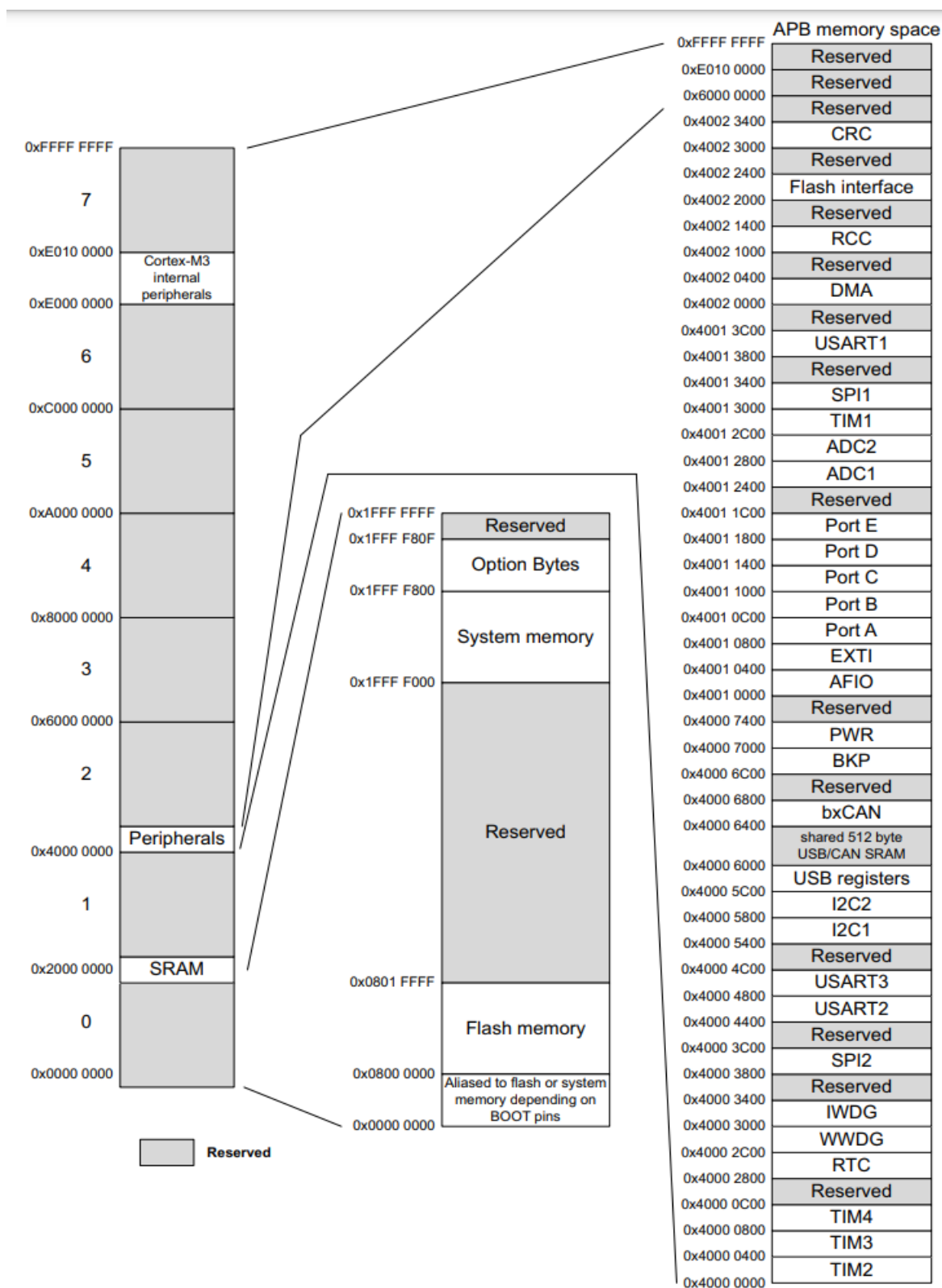


Рисунок 3 – Организация памяти МК STM32F103C8T6



### 1.2.5 Расчет экрана и выбор дисплея

Для выбора дисплея в первую очередь необходимо рассчитать достаточный размер экрана. Так как был выбран TFT ЖК-дисплей то его размер (или разрешение) измеряется в пикселях. Так как текст должен быть достаточно читаем, ибо это важно при чтении таблицы на скорость, размер шрифта 5x3 пикселей является не подходящим вариантом – цифры не всегда разборчивы и тяжело ищутся.

Размер шрифта 10x6 пикселей на символ является подходящим по читаемости и занимает достаточно места чтобы уместить таблицу в 49 двузначных цифр.

Для выбора размера экрана проведем расчет в пикселях.

В ширину необходимо: 3 пикселя для отступа слева +  $6 * 2 * 7$  пикселей для цифр +  $6 * 6$  пикселей для пробелов + 3 пикселя для отступа справа = 126 пикселей в ширину экрана.

В высоту необходимо: 3 пикселя для отступа сверху +  $10 * 7$  пикселей для цифр +  $5 * 7$  пикселей для межстрочного интервала + 20 пикселей для строки ошибок + 3 пикселя для отступа снизу = 131 пиксель в высоту экрана.

Получилось, что достаточный размер экрана 131x126 пикселей. Вариант поменьше взять нельзя, из вариантов побольше наиболее подходящий – 160x128 пикселей. Наиболее популярные контроллеры дисплеев такого разрешения, поддерживающие работу с МК семейства STM32 – ILI9163, ST7735. Сравнительный анализ дисплеев приведен в таблице 2.

Таблица 2 – Сравнительный анализ контроллеров дисплеев

Критерий	ST7735	ILI9163
Цветовая глубина	16 бит	16 бит
Интерфейсы	SPI, I2C	SPI
Цена	Ниже ILI9163	Выше ST7735
Размер	1,8 Дюймовый	1,8 Дюймовый

Как можно увидеть, контроллеры и дисплеи достаточно схожи между собой. В силу меньше стоимости ST7735 был выбран он. ST7735 – это однокристалльный контроллер/драйвер для графического TFT ЖК-дисплея. Он может выполнять операции чтения/записи данных в оперативной памяти дисплея без внешнего тактового сигнала для минимизации энергопотребления [4].

Основные пины взаимодействия дисплея:

– IM2 – выбор шины параллельного и последовательного интерфейса – при установке в 1 параллельный, при 0 – последовательный.

– IM1, IM0 – выбор типа параллельного интерфейса. В таблице 3 представлены возможные значения.

Таблица 3 – Типы параллельного интерфейса

IM1	IM0	Параллельный интерфейс
0	0	8 бит
0	1	16 бит
1	0	9 бит
1	1	18 бит

– SPI4W – 0 при трех линиях SPI, 1 при четырех линиях.

– RESX – сигнал перезапустит устройство и нужно его использовать для правильной инициализации устройства.

– CSX – пин выбора микроконтроллера устройства, работает по низкому сигналу.

– D/CX – пин выбора данных или команды на интерфейсе микроконтроллера дисплея. При 1 – данные или параметры, при 0 – команды. При SPI используется как SCL.

– RDX – дает возможность считать при включенном параллельном интерфейсе в микроконтроллере.

– WRX(D/CX) – дает возможность писать при включенном параллельном интерфейсе в микроконтроллере. При 4 линейном SPI используется как D/CX.

– D[17:0] – используются как шины отправки данных параллельного интерфейса микроконтроллера. D0 это сигнал входа/выхода при последовательном интерфейсе. При последовательном интерфейсе сигналы D[17:1] не используются.

– TE – пин вывода для синхронизации микроконтроллера с частотой устройства, активируемый программно командой перезапуска.

– OSC – контролирующий пин вывода внутреннего тактового генератора, активируемый программно командой перезапуска.

Пины выбора режима дисплея:

– EXTС – использование режима расширенных команд. При 0 используются обычные команды, при 1 расширенный набор команд NVM.

– GM1, GM0 – пины выбора разрешения. При обоих пинах в состоянии 1 – разрешение 132x162, при обоих 0 – 128x160.

– SRGB – пин настройки порядка фильтров цветов RGB. В устройстве не важен.

– SMX/SMY – пины, отвечающие за направление вывода на дисплей.

По умолчанию началом экрана считается левый верхний угол.

– LCM – пин выбора типа кристалла, белый при 0 и черный при 1.

– GS – пин изменения гаммы. Оставлен по умолчанию.

– TESEL – пин используется для изменения вывода TE сигнала.

Работает только при GM[1:0] = 00 и при 0 выводит номер строки из 162, при 1 номер строки из 160.

### **1.2.6 Использование интерфейса SPI для ЖК-дисплея**

Интерфейс SPI (Serial Peripheral Interface – последовательный периферийный интерфейс) является высокоскоростным синхронным последовательным интерфейсом. Он обеспечивает обмен данными между

микроконтроллером и различными периферийными устройствами, такими как АЦП, ЦАП, цифровые потенциометры, карты памяти, другие микросхемы и микроконтроллеры [5].

МК STM32F103C8T6 содержит два интерфейса SPI, которые обеспечивают передачу данных на частотах до 18 МГц. Один интерфейс SPI расположен на низкоскоростной шине APB1, работающей на тактовой частоте до 36 МГц, а другой – на высокоскоростной шине периферийных устройств APB2, которая работает на тактовой частоте до 72 МГц. Для увеличения эффективности передачи данных в микроконтроллере выделено два канала DMA. По интерфейсу SPI можно связать ведущий микроконтроллер с одним или несколькими ведомыми устройствами.

Одно из устройств должно быть определено ведущим (мастер), а остальные – ведомыми (подчинённые). Связь между устройствами осуществляется с помощью следующих линий связи:

- MOSI – выход данных для ведущего или вход данных для ведомого устройства;
- MISO – вход данных для ведущего или выход данных для ведомого устройства;
- SCK – сигнал общей синхронизации интерфейса.

Существует четыре режима передачи данных по SPI, которые определяются полярностью и фазой тактового сигнала. Отличие режимов заключается в том, что активным уровнем сигнала синхронизации может быть единичный или нулевой потенциал, а запись данных может производиться по фронту или спаду импульса данного синхросигнала. Эти режимы интерфейса обозначаются цифрами 0, 1, 2 и 3. На рисунке 4 представлена диаграмма всех перечисленных режимов работы интерфейса SPI.

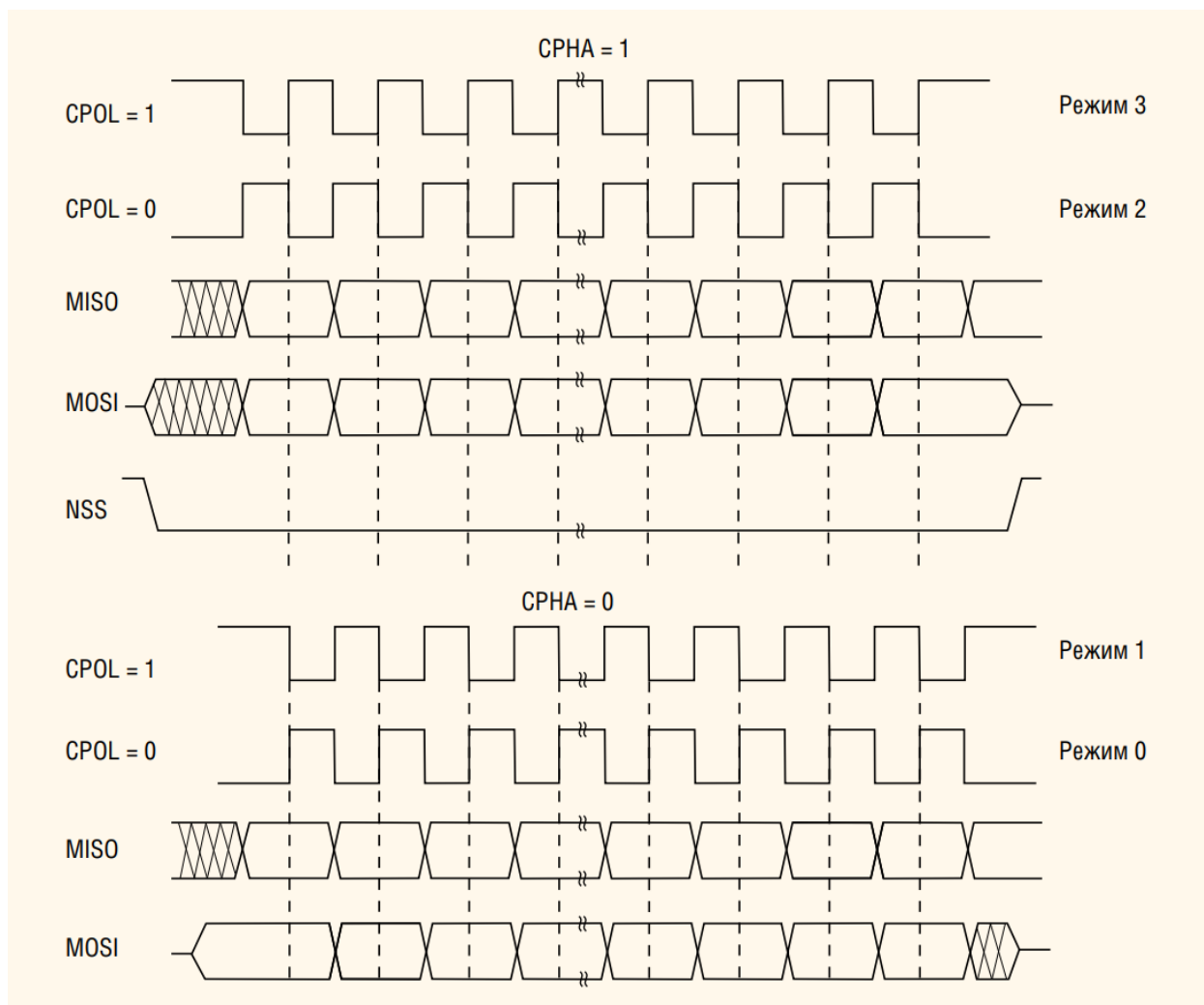


Рисунок 4 – Диаграмма режимов работы интерфейса SPI

Микроконтроллер позволяет для каждого интерфейса SPI задать полярность и фазу тактового сигнала, определяя тем самым режим его работы. Кроме того, для микроконтроллера можно установить формат передачи данных 8-разрядными или 16-разрядными словами и определить порядок передачи данных – старшим или младшим битом вперёд. Это позволяет микроконтроллеру с помощью обоих интерфейсов SPI обмениваться информацией с любыми другими SPI-устройствами.

Рассмотрим внутреннюю архитектуру SPI микроконтроллера STM32, которая представлена на рисунке 5.



CR1. Три разряда предполагают наличие восьми значений скорости. Таким образом, скорость обмена данными по интерфейсу SPI для микроконтроллера STM32 с тактовой частотой 24 МГц может изменяться от  $24 \text{ МГц}/2=12 \text{ Мбод}$  до  $24 \text{ МГц}/8=3 \text{ Мбод}$ .

Для работы с интерфейсом SPI в микроконтроллере STM32 имеются специальные регистры. Формат этих регистров с названием входящих в них разрядов представлен на рисунке 6.

Сдвиг	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 × 00	SPI_CR1	Резерв																BIDMODE	BIDOE	CRCEN	CRCNEXT	DFE	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 × 04	SPI_CR2	Резерв																								TXEE	RXNEIE	ERRIE	Резерв	SSDE	TXDMAEN	RXDMAEN	
	Исх. значение																									0	0	0		0	0	0	
0 × 08	SPI_SR	Резерв																								BSY	OVR	MODE	CRCERR	Резерв	TXE	RXNE	
	Исх. значение																									0	0	0	0		1	0	
0 × 0C	SPI_DR	Резерв																DR[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 × 10	SPI_CRCPR	Резерв																CRCPOLY[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0 × 14	SPI_RXCRCR	Резерв																RXCRC[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 × 18	SPI_TXCRRCR	Резерв																TXCRC[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 6 – Формат регистров SPI

Регистры:

- SPI\_CR1 – первый управляющий регистр;
- SPI\_CR2 – второй управляющий регистр;
- SPI\_SR – регистр статуса;
- SPI\_DR – регистр данных;
- SPI\_CRCPR – регистр, содержащий полином для вычисления CRC;
- SPI\_RXCRCR – регистр, содержащий CRC принятых данных;

– SPI\_TXCRCR – регистр, содержащий CRC передаваемых данных.

Некоторые из этих регистров используются для работы в режиме I2S.

Регистр SPI\_CR1 является первым управляющим регистром интерфейса SPI. Он имеет следующие управляющие разряды:

0. CPHA задаёт фазу тактового сигнала;
1. CPOL устанавливает полярность тактового сигнала;
2. MSTR назначает режим работы интерфейса (0 – ведомый, 1 – ведущий);
- 5...3. 0BR [2:0] задают скорость обмена (000 –  $f_{PCLK}/2$ , 001 –  $f_{PCLK}/4$ , 010 –  $f_{PCLK}/8$ , 011 –  $f_{PCLK}/16$ , 100 –  $f_{PCLK}/32$ , 101 –  $f_{PCLK}/64$ , 110 –  $f_{PCLK}/128$ , 111 –  $f_{PCLK}/256$ );
6. SPE управляет интерфейсом (0 – отключает, 1 – включает);
7. LSBFIRST задаёт направление передачи (0 – младшим разрядом вперёд, 1 – старшим разрядом вперёд);
8. SSI определяет значение NSS при SSM=1;
9. SSM выбирает источник сигнала NSS (0 – с внешнего вывода, 1 – программно от разряда SSI);
10. RX ONLY совместно с битом BIDIMODE определяет направление передачи в однонаправленном режиме;
11. DFF определяет формат данных (0–8 бит, 1–16 бит);
12. CRCNEXT управляет передачей кода CRC (0 – данные, 1 – CRC);
13. CRCEN регулирует аппаратное вычисление CRC (0 – запрещено, 1 – разрешено). Для корректной операции этот бит должен записываться только при отключённом интерфейсе SPI, когда SPE = 0;
14. BIDIOE совместно с битом BIDIMODE управляет двунаправленным режимом работы интерфейса (0 – приём, 1 – передача);
15. BIDIMODE управляет двунаправленным режимом работы интерфейса (0 – двухпроводный однонаправленный режим, 1 – однопроводной двунаправленный режим).



SPI\_CR2 является вторым управляющим регистром интерфейса SPI и имеет следующие разряды, которые управляют:

0. RXDMAEN – запросом DMA для приёмника (0 – запрещает, 1 – разрешает);

1. TXDMAEN – запросом DMA для передатчика (0 – запрещает, 1 – разрешает);

2. SSOE – сигналом NSS в режиме мастера (0 – запрещает, 1 – разрешает);

4...3. – зарезервированы;

5. ERRIE – прерыванием в случае ошибки (0–запрещает 1–разрешает);

6. RXNEIE – прерыванием приёма данных (0–запрещает 1–разрешает);

7. TXEIE – управляет прерыванием передачи данных (0–запрещает 1–разрешает);

15...8. – зарезервированы.

Регистр статуса SPI\_SR включает в себя следующие разряды:

0. RXNE устанавливается, если в буфере приёмника есть принятые данные;

1. TXE – устанавливается, если буфер передатчика пуст и готов принять новые данные;

2, 3. – зарезервированы;

4. CRCERR устанавливается при ошибке CRC при приёме данных;

5. MODF устанавливается, когда в режиме мастера к сигналу NSS прикладывается низкий потенциал;

6. OVR – флаг переполнения, устанавливается при приёме новых данных, если предыдущие не были прочитаны;

7. BSY – флаг занятости, устанавливается, если интерфейс занят обменом данными или буфер данных передатчика не пустой.

Регистр данных SPI\_DR состоит из 16 разрядов данных. В этот регистр данные записываются для передачи и читаются из него при приёме.

### 1.2.7 Прием данных от ПЭВМ

Приём данных от ПЭВМ происходит через драйвер MAX232. MAX232 – интегральная схема, преобразующая сигналы последовательного порта RS-232 в цифровые сигналы.

RS-232 – стандарт физического уровня для синхронного и асинхронного интерфейса (USART и UART). Обеспечивает передачу данных и некоторых специальных сигналов между терминалом и устройством приема. Сигнал, поступающий от интерфейса RS-232, через преобразователь передается в микроконтроллер на вход RxD.

К внешнему устройству MAX232 подключен через разъем DB-9. На схеме условное обозначение – XP2.

УГО MAX232 показано на рисунке 7. Назначение пинов описано в таблице 4.

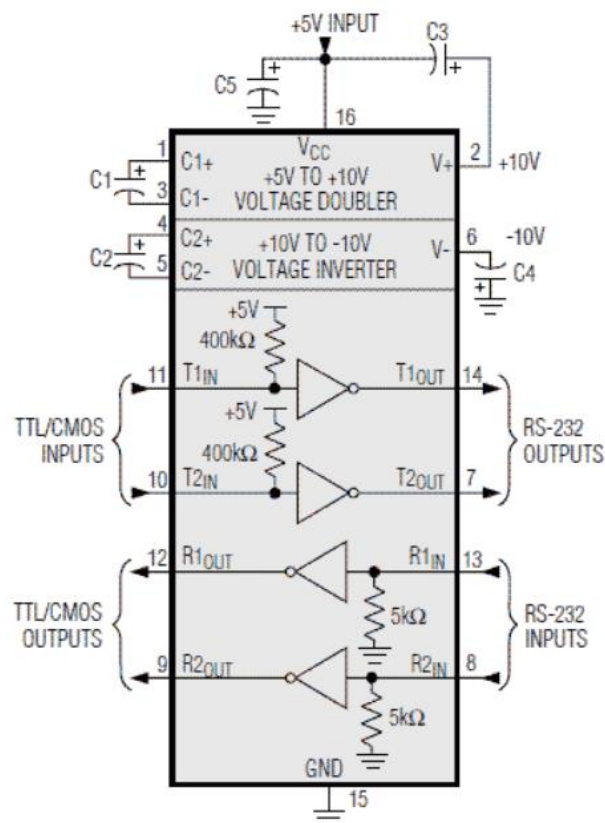


Рисунок 7 – УГО MAX232

Таблица 4 – Назначение пинов MAX232

Номер	Имя	Тип	Описание
1	C1+	–	Положительный вывод C1 для подключения конденсатора
2	VS+	O	Выход положительного заряда для накопительного конденсатора
3	C1-	–	Отрицательный вывод C1 для подключения конденсатора
4	C2+	–	Положительный вывод C2 для подключения конденсатора
5	C2-	–	Отрицательный вывод C2 для подключения конденсатора
6	VS-	O	Выход отрицательного заряда для накопительного конденсатора
7, 14	T2OUT, T1OUT	O	Вывод данных по линии RS232
8, 13	R2IN, R1IN	I	Ввод данных по линии RS232
9, 12	R2OUT, R1OUT	O	Вывод логических данных
10, 11	T2IN, T1IN	I	Ввод логических данных
15	GND	–	Земля
16	Vcc	–	Напряжение питания, подключение к внешнему источнику питания 5 В

Когда микросхема MAX232 получает на вход логический "0" от внешнего устройства, она преобразует его в напряжение от +5 до +15В, а когда получает логическую "1" - преобразует её в напряжение от -5 до -15В, и по тому же принципу выполняет обратные преобразования от RS-232 к внешнему устройству.

### 1.2.8 Использование интерфейса USART для работы с ПЭВМ

Интерфейс USART (Universal Synchronous Asynchronous Receiver Transmitter) в микроконтроллерах STM32 представляет собой универсальный последовательный интерфейс, который может работать в режиме синхронной или асинхронной передачи данных. Он обеспечивает возможность обмена данными между микроконтроллером и другими устройствами, такими как датчики, модули связи и периферийные устройства.

USART в STM32 поддерживает передачу данных через одну линию для приема (RX) и одну для передачи (TX). Он также может работать в

полудуплексном режиме, когда одна линия используется для передачи и приема данных.

USART может настраиваться на разные скорости передачи данных (бодрейты), количество бит данных, контроль четности, стоповые биты и другие параметры через специальные регистры микроконтроллера. Это обеспечивает гибкость в настройке передачи данных в соответствии с требованиями конкретного приложения.

В разрабатываемой системе USART используется в асинхронном режиме для вывода текста на виртуальный, который выступает в роли ПЭВМ. Рассмотрим настройку USART для устройства.

В устройстве USART используется в асинхронном режиме, контроль сигнала CTS/RTS отключен, baud rate – 115200 бит/с, длина каждой посылки – 8 бит, включая бит четности, контроль четности отключен, используется один стоп-бит, оверсемплинг в режиме 16-семплирования. Кроме того, включены прерывания для USART.

Оверсемплинг в USART относится к технике, используемой для приема данных в асинхронном режиме. Эта техника помогает улучшить точность синхронизации битов данных, особенно при работе с высокими скоростями передачи данных.

Оверсемплинг подразумевает выбор частоты сэмплирования (число раз, которое система измеряет состояние входного сигнала за определенный промежуток времени) значительно выше, чем минимально необходимая частота для корректного считывания данных.

В USART для асинхронной передачи, оверсемплинг обычно используется для более точного определения момента прихода каждого бита данных. К примеру, в режиме 16-семплирования (16x oversampling), каждый бит данных будет сэмплироваться 16 раз за период передачи, что улучшает точность считывания данных и помогает бороться с потерей или искажением сигнала в условиях шумов или неполадок в канале связи.

Эта техника позволяет повысить устойчивость и надежность приема данных по USART, особенно при работе на высоких скоростях передачи данных или в условиях, где возможны помехи или искажения сигнала.

Всего существует 7 регистров, связанных с настройкой и работой USART: USART\_SR (Status register), USART\_DR (Data register), USART\_BRR (Baud rate register), USART\_CR1 (Control register 1), USART\_CR2 (Control register 2), USART\_CR3 (Control register 3), USART\_GTPR (Guard time and prescaler register). Ниже будут описаны все регистры кроме неиспользованных регистров для настройки [7]. Формат регистров представлен на рисунке 8.

**Table 182. USART register map and reset values**

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	USART_SR	Reserved																						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE	
	Reset value																							0	0	1	1	0	0	0	0	0	0	
0x04	USART_DR	Reserved																						DR[8:0]										
	Reset value																							0	0	0	0	0	0	0	0	0		
0x08	USART_BRR	Reserved													DIV_Mantissa[15:4]										DIV_Fraction [3:0]									
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0C	USART_CR1	Reserved													UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK						
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x10	USART_CR2	Reserved													LINEN	STOP [1:0]	CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]									
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0							
0x14	USART_CR3	Reserved																						CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	Reset value																							0	0	0	0	0	0	0	0	0	0	0
0x18	USART_GTPR	Reserved													GT[7:0]							PSC[7:0]												
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

**Рисунок 8 – Формат регистров USART**

Начнем с настройки USART. Для этого используются control-регистры и регистр управления скоростью передачи.

Регистр USART\_CR1:

– UE: USART enable - включить USART (включается установкой бита в состояние 1).

– M: Word length - длина слова, задаёт количество бит данных в одном фрейме. Бит не должен модифицироваться в процессе обмена данными (это касается как передачи, так и приёма). 0 - 1 старт-бит, 8 бит данных, n стоп-битов; 1 - 1 старт-бит, 9 бит данных, n стоп-битов. Примечание. Бит чётности считается битом данных.

– WAKE: Wakeup method - метод пробуждения USART. 0 - "линия свободна" (Idle line); 1- адресная метка.

– PCE: Parity control enable - включить аппаратный контроль чётности (генерация бита чётности при передаче данных и проверка в принимаемых данных).

– PS: Parity selection - выбор метода контроля чётности. Выбор происходит после завершения передачи/приёма текущего байта. 0 - контроль на чётность; 1 - контроль на нечётность.

– PEIE: PE interrupt enable - разрешение прерывания от PE. 0 – прерывание запрещено; 1 – генерируется прерывание от USART, когда USART\_SR.PE==1.

– TXEIE: TXE interrupt enable - разрешение прерывания от TXE. 0 – прерывание запрещено; 1 – генерируется прерывание от USART, когда USART\_SR.TXE==1.

– TCIE: Transmission complete interrupt enable - разрешение прерывания после завершения передачи. 0 – прерывание запрещено; 1 – генерируется прерывание от USART, когда USART\_SR.TC==1.

– RXNEIE: RXNE interrupt enable - разрешение прерывания от RXNE. 0: прерывание запрещено; 1: генерируется прерывание от USART, когда USART\_SR.ORE==1 или USART\_SR.RXNE==1.

– IDLEIE: IDLE interrupt enable - разрешение прерывания при обнаружении, что "линия свободна" (Idle line). 0: прерывание запрещено; 1: генерируется прерывание от USART, когда USART\_SR.IDLE==1.

– TE: Transmitter enable - включить передатчик USART (включается установкой бита в 1).

– RE: Receiver enable - включить приёмник USART (включается установкой бита в 1). После установки бита, приёмник начинает поиск стартового бита во входном сигнале.

– RWU: Receiver wakeup - переводит USART в тихий режим. Этот бит устанавливается и сбрасывается программно, а также может сбрасываться аппаратно при обнаружении пробуждающей последовательности.

– SBK: Send break - отправить Break посылку. Бит может быть установлен и сброшен программно. Его необходимо программно установить в 1 для формирования Break посылки, он будет сброшен аппаратно во время формирования stop-бита в Break фрейме. 0: Break-символ не передаётся; 1: Break-символ будет передан.

Теперь опишем регистр BRR, с помощью которого контролируется скорость передачи данных через USART.

– DIV\_Mantissa[11:0]: mantissa of USARTDIV - целая часть коэффициента деления делителя частоты.

– DIV\_Fraction[3:0]: fraction of USARTDIV - дробная часть коэффициента деления. В режиме с OVER8==1 в битовом поле DIV\_Fraction[3:0] старший бит [3] не используется и должен быть сброшен.

С помощью регистра USART\_BRR задаётся скорость передачи - одновременно как для приёмника USART, так и для передатчика. На рисунке 8 представлена схема, показывающая, как именно высчитывается скорость передачи.

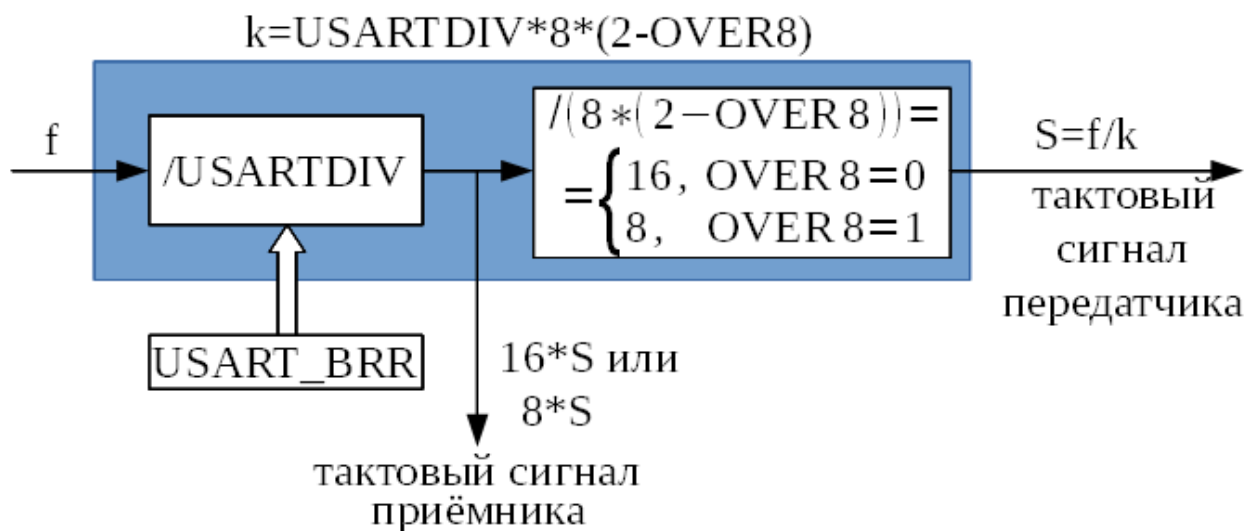


Рисунок 8 – Вычисление скорости приема и передачи

В данной системе было принято решение использовать baud rate = 115200, поэтому был выставлен USART\_BRR = 69. Проверим:  $8000000 / 69 = 115942 \sim 115200$ .

Далее рассмотрим USART\_DR – регистр, через который передаются непосредственно данные.

DR[8:0]: Data value - регистр данных. Содержит полученный или передаваемый символ, в зависимости от того, производится чтение из него или запись в регистр. Регистр выполняет двойную функцию за счёт того, что он является составным, он объединяет в себе два регистра: один для передачи (TDR) и один для приёма (RDR). TDR обеспечивает загрузку данных в выходной сдвигающий регистр, сдвигающий регистр преобразует загруженное в него слово в последовательную форму. Получаемые в последовательной форме данные накапливаются в приёмном сдвигающем регистре, когда фрейм получен полностью, данные из сдвигающего регистра передаются в регистр RDR, который реализует параллельный интерфейс между внутренней шиной микроконтроллера и входным сдвигающим регистром.

Когда осуществляется передача данных с включённым контролем чётности (USART\_CR1.PCE==1), старший бит, записываемый в регистр USART\_DR (бит [7] или [8], в зависимости от выбранной длины слова, см.



USART\_CR1.M), не учитывается. Он замещается вычисленным битом чётности.

При получении данных с включённым контролем чётности, при чтении из USART\_DR будем получать значение, содержащее полученный бит чётности.

Последний рассматриваемый регистр в USART – USART\_SR (status register).

– CTS: CTS flag - флаг изменения состояния nCTS. Устанавливается аппаратно, когда происходит переключение сигнала на входе nCTS. Если установлен бит CTSIE (USART\_CR3.CTSIE==1), то при установке флага генерируется прерывание. Флаг сбрасывается программно записью 0.

– LBD: LIN break detection flag - флаг приёма посылки Break. Устанавливается аппаратно при обнаружении посылки Break на входе; если установлен бит LBDIE (USART\_CR3.LBDIE==1), то генерируется прерывание. Флаг сбрасывается программно записью 0.

– TXE: Transmit data register empty - флаг устанавливается аппаратно, когда содержимое регистра передаваемых данных TDR пересылается в сдвигающий регистр (доступ к TDR осуществляется путём записи в регистр USART\_DR). Если установлен бит TXEIE (USART\_CR1.TXEIE==1), генерируется прерывание. Флаг сбрасывается путём записи в регистр USART\_DR.

– TC: Transmission complete - флаг завершения передачи, устанавливается аппаратно, если передача фрейма завершена, и флаг TXE установлен (т.е. регистр передаваемых данных пуст, больше нет данных для передачи). Если USART\_CR1.TCIE==1, то при установке флага генерируется прерывание. Флаг сбрасывается программно последовательностью действий: чтение регистра USART\_SR, затем запись в USART\_DR. Также бит может быть сброшен записью в него 0. Примечание. После сброса этот бит установлен.

– RXNE: Read data register not empty - регистр данных для чтения не пуст. Флаг устанавливается аппаратно, когда содержимое принимающего сдвигающего регистра передаётся в регистр принимаемых данных RDR. Если USART\_CR1.RXNEIE==1, при этом генерируется прерывание. Флаг сбрасывается чтением из регистра USART\_DR. Также бит может быть сброшен записью в него 0.

– IDLE: IDLE line detected - линия свободна. Флаг устанавливается аппаратно, если обнаружено что линия свободна. Это происходит, если получен целый фрейм единиц. При этом генерируется прерывание, если USART\_CR1.IDLEIE==1. Флаг сбрасывается программно последовательностью действий: чтение регистра USART\_SR с последующим чтением из регистра USART\_DR.

– ORE: Overrun error - ошибка переполнения. Флаг устанавливается аппаратно, когда слово, полученное в сдвигающей регистр готово к перемещению в регистр принимаемых данных RDR, но RXNE==1 (регистр RDR не пуст, содержит ещё не прочитанные из него принятые USART данные). Если USART\_CR1.RXNEIE==1, то при установке флага генерируется исключение. Флаг сбрасывается программно последовательностью действий: чтение из регистра USART\_SR с последующим чтением из USART\_DR.

– NF: Noise detected flag - флаг устанавливается аппаратно при обнаружении шума в полученном фрейме. Сбрасывается программно последовательностью действий: чтение из регистра USART\_SR, затем чтение из регистра USART\_DR.

– FE: Framing error - ошибка фрейма. Флаг устанавливается аппаратно в случае нарушения синхронизации, чрезмерного шума в линии, при обнаружении символа Break. Флаг сбрасывается программно последовательностью действий: чтение из регистра USART\_SR, затем чтение из регистра USART\_DR. Примечание. В отношении генерации прерывания этот флаг полностью аналогичен флагу NF.

– PE: Parity error - ошибка чётности. Флаг устанавливается аппаратно, когда в принятом фрейме обнаружена ошибка чётности (если контроль чётности включён). Если USART\_CR1.PEIE==1, то генерируется прерывание. Флаг сбрасывается программно последовательностью действий: чтение из регистра USART\_SR, затем чтение либо запись регистра USART\_DR. Перед сбросом флага, программа должна дождаться установки флага RXNE (регистр данных для чтения не пуст).

### **1.2.9 Использование таймера для генерации сигнала для динамика**

Микроконтроллеры STM32F103xx имеют в своём составе множество таймеров с большим количеством поддерживаемых функций. С помощью любого таймера можно формировать интервалы времени с требуемой длительностью с генерацией прерывания или DMA запроса по окончании интервала.

Кроме того, можно формировать одиночные импульсы заданной длительности или периодические импульсы с заданной длительностью и частотой повторения; подсчитывать количество импульсов внешнего сигнала (счётчик может работать в режиме сложения или вычитания); поддерживается режим широтно-импульсной модуляции.

В данной системе используется только один из каналов одного таймера общего назначения. Он используется для генерации ШИМ-сигнала для динамика в случае, если был поставлен новый рекорд по скорости чтения таблицы какого-либо размера. При этом рекорды считаются для таблиц разного размера отдельно. Используется второй канал второго таймера. Конфигурация таймера представлена на рисунке 9.

Slave Mode

Disable

Trigger Source

Disable

Clock Source

Internal Clock

Channel1

Disable

Channel2

PWM Generation CH2

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

⏪

⏩

ⓘ

Counter Settings

Prescaler (PSC - 16 bits value)

127

Counter Mode

Up

Counter Period (AutoReload Register - 127)

Internal Clock Division (CKD)

No Division

auto-reload preload

Disable

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Trigger Event Selection

Reset (UG bit from TIMx\_EGR)

PWM Generation Channel 2

Mode

PWM mode 1

Pulse (16 bits value)

63

Output compare preload

Enable

Fast Mode

Disable

Рисунок 9 – Конфигурация таймера TIM2 для работы в режиме ШИП

Таймер непосредственно работает через регистры TIM2\_CNT, TIM2\_PSC и TIM2\_ARR. Они полностью содержат в себе значения текущего счетчика таймера, значение делителя и значение автоматической перезагрузки соответственно, и более подробно их рассматривать смысла нет. делителя и автоматическая перезагрузка были выставлены как 127. Также таймер имеет множество регистров для настройки режима работы. Рассмотрим только те, которые необходимо было настроить вручную.

Регистр TIM2\_CR1 представлен на рисунке 10.

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMx_CR1	Reserved																						CKD [1:0]	ARPE	CMS [1:0]	DIR	OPM	URS	UDIS	CEN		
Reset value																							0	0	0	0	0	0	0	0	0	

Рисунок 10 – Регистр TIM2\_CR1

- CKD: Clock division;
- ARPE: Auto-reload preload enable. Бит для включения режима предзагрузки регистра TIMx\_ARR: 0: TIMx\_ARR не буферизируется; 1: используется буферизация регистра TIMx\_ARR. Когда буферизация включена, новое значение, записанное в регистр, начинает использоваться после очередного события обновления.
- CMS: Center-aligned mode selection;
- DIR: Direction;
- OPM: One-pulse mode;
- URS: Update request source;
- UDIS: Update disable;
- CEN: Counter enable.

Далее рассмотрим TIM2\_CCMR1(Compare and capture mode register), который представлен на рисунке 11. Каналы могут быть использованы в режимы захвата(input) и в режиме сравнения(output). При этом одни биты могут иметь разный смысл в зависимости от режима. Для режима ШИМ необходим режим сравнения, поэтому нужно смотреть на верхний ряд битов.

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
TIMx_CR2	Reserved																		OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]		CCDS	CCUS	Reserved		CCPC											
Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 11 – Регистр TIM2\_CR2

- OC2CE: Output compare 2 clear enable;
- OC2M[2:0]: Output compare 2 mode;
- OC2PE: Output compare 2 preload enable;
- OC2FE: Output compare 2 fast enable;

- CC2S[1:0]: Capture/Compare 2 selection;
- OC1CE: Output compare 1 clear enable;
- OC1M: Output compare 1 mode. Эти биты определяют поведение выходного сигнала. Для режима ШИМ существует два вариант – 110 (PWM Mode 1) и 111 (PWM Mode 2). Был выбран PWM Mode 1, который выдает на пин единицу при возрастающем счетчике и 0 при убывающем (PWM Mode 2 действует наоборот).

- OC1PE: Output compare 1 preload enable. 0 – предзагрузка отключена; 1 – предзагрузка включена. Для режима ШИМ предзагрузка должна быть включена.

- OC1FE: Output compare 1 fast enable;

- CC1S: Capture/Compare 1 selection.

Последний используемый регистр - TIM2\_CCR1 (Compare and Capture register). Он содержит в себе 16 бит данных.

Для режима ШИМ была выбрана частота 500Гц, так как это стандартная частота звукового генератора в Proteus 8. Частота работы микроконтроллера – 8МГц.  $f = \text{AutoReload} * \text{Prescaler} * \text{Freq}_{\text{PWM}}$ , тогда для того, чтобы получить  $\text{Freq}_{\text{PWM}}$  примерно равную 500Гц, можно выставить AutoReload и Prescaler как 127.  $8000000 / 127 / 127 \approx 500\text{Гц}$ . При этом используется коэффициент заполнения 50% - Для этого возьмем для Pulse примерно половину Prescaler – 63. Именно это значение будет храниться в TIM2\_CCR1.

#### **1.2.10 Построение функциональной схемы**

На основе пунктов 1.2.1 – 1.2.9 была построена функциональная схема устройства измерения скорости чтения. Она представлена на рисунке 12.

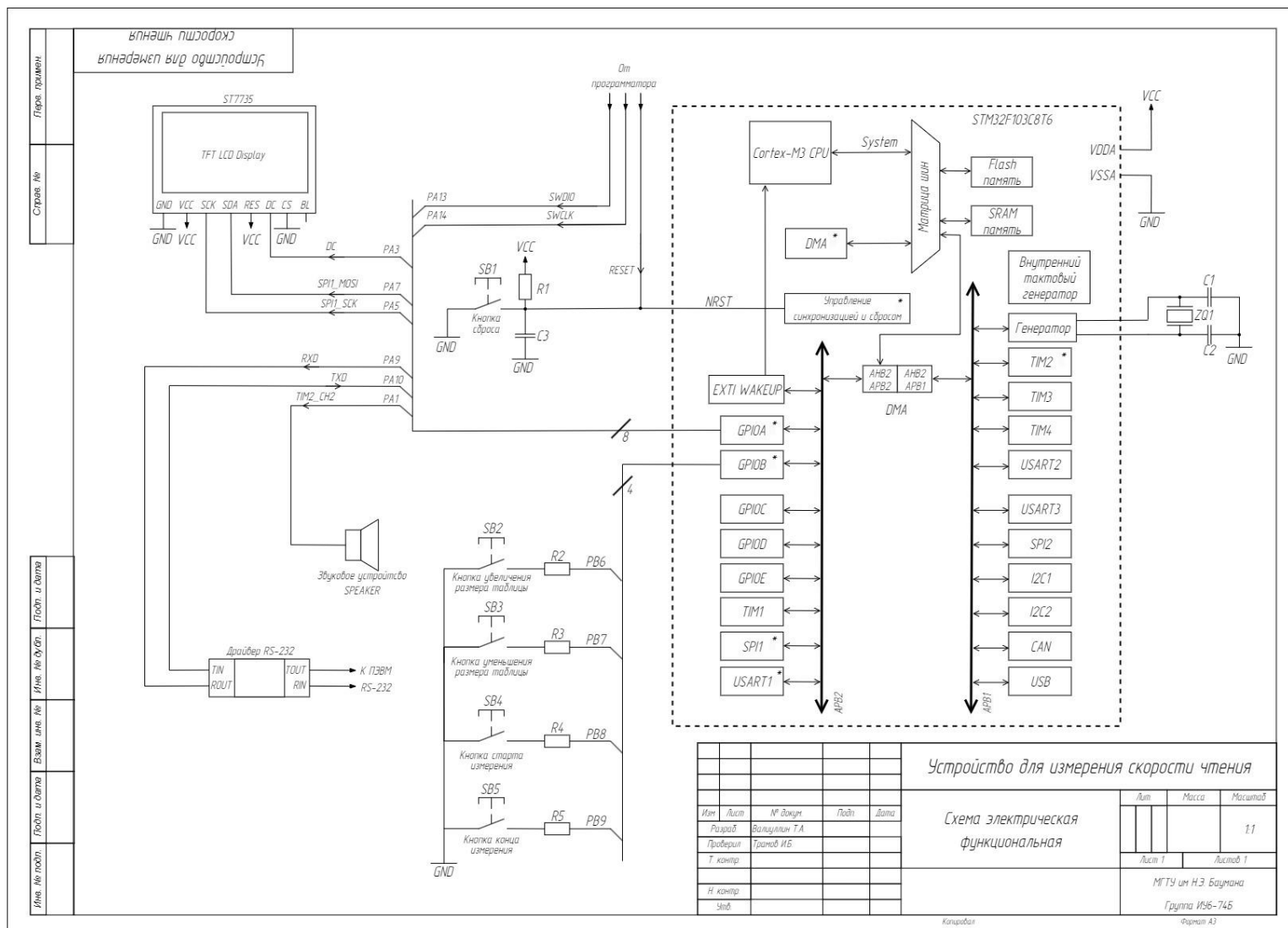


Рисунок 12 – Спроектированная функциональная схема устройства

### 1.3 Построение принципиальной схемы

#### 1.3.1 Работа с устройством через дисплеи

Взаимодействие с устройством происходит при помощи 4 кнопок – «Увеличение размера таблицы», «Уменьшение размера таблицы», «Начало измерения», «Конец измерения». Также присутствует кнопка сброса состояния МК для случая зависания работы программы.

Основной источник вывода информации – дисплей. Вне измерения скорости чтения таблицей Шульте на дисплее имеется надпись о выборе размера от 3 до 7. При нажатии кнопок увеличения и уменьшения размер соответственно меняется, о чем пишется на дисплее.

При нажатии кнопки старт на дисплей выводится сгенерированная таблица Шульте и начинается отсчет времени. При нажатии кнопки стоп измерение заканчивается, результаты измерения посылаются на ПЭВМ (и

индивидуальный и средний), а также при установлении рекорда на динамик отправляется секундный сигнал ШИМ.

### 1.3.2 Описание и расчет алгоритма генерации таблицы

При нажатии кнопки начала измерения генерируется таблица Шульте выбранного размера. Входным параметром алгоритма является размер таблицы. Алгоритм работает следующим образом: сначала генерируется таблица нужного размера, содержащая значения от 1 до размер \* размер включительно в последовательном порядке. Для того, чтобы перемешать таблицу, берутся 2 значения таблицы наугад и меняются местами. Такие случайные перестановки выполняются размер \* размер \* 10 раз, что в среднем приводит к 20 перестановкам для каждого элемента. Схема алгоритма представлена на рисунке 13.



Рисунок 13 – Алгоритм генерации таблицы



Реализованный на языке программирования Си [8] алгоритм требует  $1 + \text{размер} * \text{размер} + 7 * 10 * \text{размер} * \text{размер}$  действий =  $1 + 71 * \text{размер} * \text{размер}$  действий. Так как размер от 3 до 7 то получается алгоритм выполняет от 640 до 3480 действий. Для любого размера измерение времени показало, что генерация таблицы занимает менее 100 мс. При этом время измерения не учитывает это время. Поэтому полученное время генерации таблицы допустимо.

### **1.3.3 Описание алгоритма прерывания кнопок**

Кнопки работают на специально включенных для них прерываниях. При вызове функции прерываний проверяется какой из пинов подал сигнал, чтобы понять какая из кнопок была нажата и вызвать соответствующий алгоритм. При определении нажатой кнопки проверяются следующие ошибки: изменение размера таблицы вне границ [3;7] или нажатии кнопки изменения размера во время чтения таблицы. Описание алгоритма функции прерываний представлено на рисунке 14.

#### **РИСУНОК 14 СХЕМА АЛГОРИТМА ПРЕРЫВАНИЙ ОТ КНОПОК**

### **1.3.4 Описание и расчет алгоритма отображения таблицы**

При нажатии кнопки начала измерения после генерации таблицы выполняется алгоритм вывода таблицы на дисплей. Так как ЖК-дисплей ST7735 является TFT-дисплеем, то данные на нем отображаются попиксельно. Вывод цифр и символов в общем происходит так же. Для вывода таблицы выводятся строки чисел. В каждой строке столько двузначных чисел сколько выбран размер таблицы. При этом пробелов, стоящих между числами, в строке = размер – 1. Таким образом при максимальном размере 7 строка достигает длины  $7 * 2 + 6 = 20$  символов. В алгоритме берется пустая строка из 20 символов и в нее вставляются числа и пробелы на нужные места. Так делается с каждой строкой таблицы. После этого происходит вывод линий таблицы. В таблице линий: размер – 1 вертикальных и столько же горизонтальных. Горизонтальные линии

выводятся между строками чисел, а вертикальные – на месте пробелов между числами. Схема алгоритма представлена на рисунке 15.

#### **РИСУНОК 15 СХЕМА АЛГОРИТМА ОТОБРАЖЕНИЯ ТАБЛИЦЫ**

Расчет операций во время выполнения алгоритма:  $(\text{размер} - 1) * 13 + 5 + 10 * \text{размер} + 7 = 23 * \text{размер} - 1$ . При разных размерах это значение достигает от 68 до 160 операций, что выполняется очень быстро. Однако сама посылка данных дисплею занимает достаточно длительное время.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 <https://monitor4ik.com/stati/tft/>

2 [https://e-learning.bmstu.ru/ru6/pluginfile.php/19506/mod\\_resource/content/2/%D0%9B%D0%A0%20STM32%202023.pdf](https://e-learning.bmstu.ru/ru6/pluginfile.php/19506/mod_resource/content/2/%D0%9B%D0%A0%20STM32%202023.pdf)

3 <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>

4 <https://www.displayfuture.com/Display/datasheet/controller/ST7735.pdf>

5 <https://303421.selcdn.ru/soel-upload/clouds/1/iblock/e84/e84786cd2be538cfb9eb44a3938d1dcc/20140118.pdf>

6 [https://www.rotr.info/electronics/mcu/stm32f100/stm32\\_spi/spi\\_registers.htm](https://www.rotr.info/electronics/mcu/stm32f100/stm32_spi/spi_registers.htm)

7 [https://www.rotr.info/electronics/mcu/arm\\_usart\\_stm32\\_reg.htm](https://www.rotr.info/electronics/mcu/arm_usart_stm32_reg.htm)

8 книга по си какая нить