



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 «Радиоэлектроника и лазерная техника (РЛ)»

---

КАФЕДРА «Технологии приборостроения (РЛ6)»

---

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Способы проекции изображений на графический экран»*

Студент РЛ6-71  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

С. В. Филимонов  
(И. О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Д. А. Семеренко  
(И. О. Фамилия)

*2023 г.*

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 Принцип работы дисплеев</b>	<b>4</b>
<b>2 Функциональная схема</b>	<b>6</b>
<b>3 Алгоритм работы микроконтроллера</b>	<b>7</b>
<b>4 Исследовательский раздел</b>	<b>9</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>13</b>
<b>СПИСОК ЛИТЕРАТУРЫ</b>	<b>14</b>

## ВВЕДЕНИЕ

В данной работе будут продемонстрированы основные способы вывода и настройки популярных графических дисплеев серии «STxxxx» и «ILIxxxx».

Главной проблемой при проектировании является: интерфейс обмена данными между компьютером и дисплеем, так как они имеют разные скорости работы. Будут происходить долгие обновления кадров, вызванные задержками синхронизации. Решением является повышение скорости работы микроконтроллера-посредника участвующего в передаче данных.

# 1 Принцип работы дисплеев

Кадр дисплея представляет из себя трехмерный массив  $X$  и  $Y$ , в котором координаты пикселя на плоскости, а  $Z$  цвет каждого пикселя.

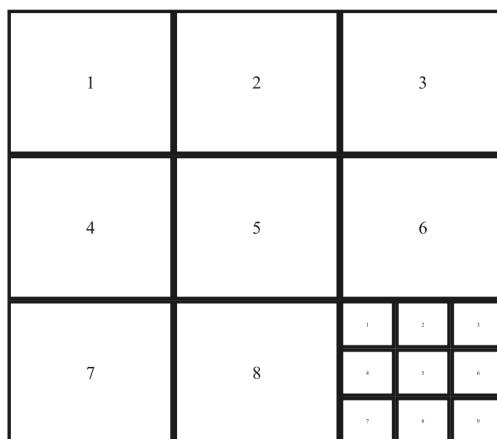


Рисунок 1.1 – Схема разбиения больших экранов на экран по меньше.

Современные контроллеры обладают достаточным количеством памяти, чтобы хранить небольшие разрешения дисплеев, но с увеличением разрешения дисплея, память микроконтроллера не увеличивается. Решением данной проблемы служит разбиение дисплея на секторы меньшего разрешения. Если разрешение дисплея особо большое, например речь идет о 4К экране для компьютера, секторы такого дисплея дробят еще на более мелкие секторы. Готовые небольшие секторы уже хранятся в памяти парой, и пока происходит вывод одного сектора, для другого происходит процесс отрисовки.

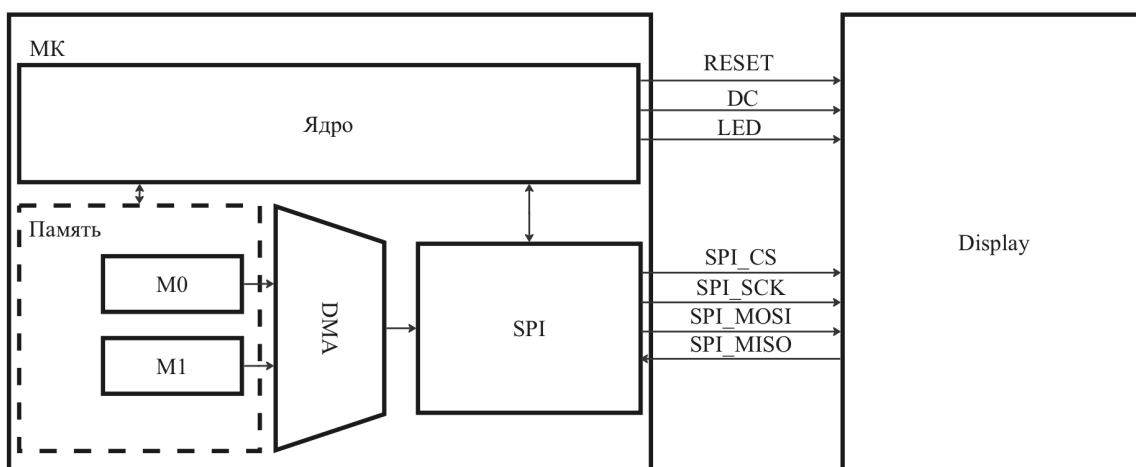


Рисунок 1.2 – Принципиальная схема подключения дисплея.

Порт «RESET» является выводом аппаратного сброса настроек. Порт

«LED» является выводом анода подсветки дисплея. Порт «DC» служит для информирования дисплея о том какой тип данных получает дисплей: «DC = 1» дисплей на вход получает команду, «DC = 0» дисплей на входе получает данные. Сами данные дисплей получает по последовательному периферийному интерфейсу.

Обновление кадра сектора осуществляется путем разделения внутренней памяти на две ячейки, по ячейки на сектор. Ядром напрямую через последовательный интерфейс передаем координаты выводимого сектора. После начинается DMA передача данных сектора. Пока идет передача сектора, другой сектор готовится к передаче. Данные процессы должны быть синхронизированы, чтобы пока готовился один кадр, выводился другой. Соответственно после выводятся та область памяти, которая готовилась к передаче, а так которая уже вывелась меняться с ней местами.

## 2 Функциональная схема

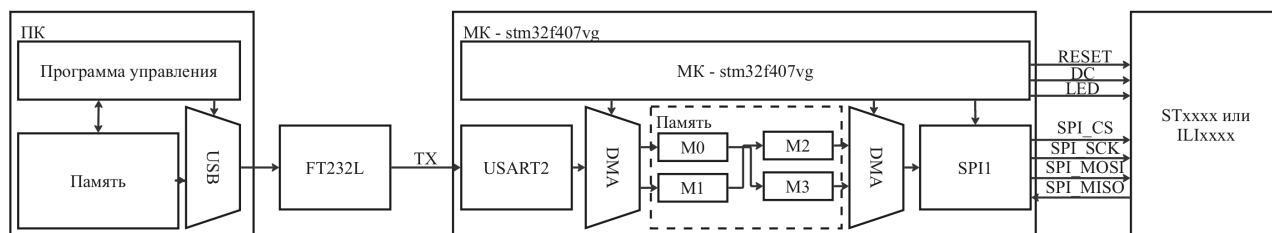


Рисунок 2.1 – Функциональная схема устройства.

Идея работы устройства заключается в том, что с компьютера поступают данные о выводимом кадре, через декодер «FT232L» они передаются в микроконтроллер по универсальному синхронному приёмопередатчик. В Микроконтроллере DMA модуль записывает в одну из ячеек памяти считанный сектор. После для дисплея происходит настройка на обновление кадра. На порт «DC» поступает сигнал равные логической единице, по последовательному периферийному интерфейсу поступает команда об обновлении кадра. И следом приходит команда о выводе координаты по оси OX. На порт DC поступает сигнал равные логическому нулю, По последовательному периферийному интерфейсу поступает левая и правая координата для оси OX. Аналогичным образом происходит настройка координат оси OY. После происходит подтверждение начала передачи путем отправки соответствующей команды, при отправлении команды на выводе «DC» логическая единица. После на порт «DC» подается логический ноль и начинается передача сектора из памяти.

Пока происходит передача сектора, то в это время в другую ячейку памяти происходит запись следующего сектора. Эти операции происходят в цикле пока не будет вывод весь кадр.

Так как дисплеи «ILI» и «ST» имеют одинаковый интерфейс взаимодействия, то не важно какой дисплей будет использоваться.

### 3 Алгоритм работы микроконтроллера

В предыдущей главе была описана общая схема работы вывода кадров на секторы дисплея. В данной главе в начале будет описана схема инициализации микроконтроллера для работы, после «тонкости» работы вывода кадров.

#### Инициализация микроконтроллера

Производится настройка частоты работы. Ядро настраивается на частоту 168 МГц, шина АHB1/APB1 на частоту 82 МГц, АHB2/APB2 42 МГц. После с помощью системной функции «SystemCoreClock\_Update()» происходит обновление переменной «SystemCoreClock», которая хранит в себе значение текущей частоты работы ядра.

Производится настройка периферии. Инициализируются порты входа и выхода. Порты, которые будут работать по стандартам универсального синхронного приёмопередатчика, в микроконтроллере USART2 и последовательному периферийному интерфейсу, в микроконтроллере SPI1 настраиваются на альтернативные функции, соответствующих режимов работы. Модуль универсального синхронного приёмопередатчика настраивается на чтение и запись, возможность считывать и передавать данные через DMA канал. Задается его частота работы по формуле:

$$\text{Baud} = \frac{\text{SystemCoreClock}}{2 \cdot 460900}$$

Происходит включение данного модуля. Следом настраивается DMA канал на прослушивание в не циклическом режиме, со сбрасыванием прерывания по приему данных.

Модуль последовательного периферийного интерфейса настраивается на режим записи по DMA, в режиме «мастера», деление частоты скорости остаются по умолчанию равной одной второй. Произведя настройку, описанную выше, программа входит в бесконечный цикл, в состояние ожидания команды на обновление экрана дисплея.

#### Особенность работы алгоритма обновления кадра.

Алгоритм, описанный в предыдущей главе работает аналогичным образом в данном проекте. Но так как поддерживается большой набор дисплеев, то у них всех есть отличия в разрешении формата цвета. То есть некоторые

дисплеи работают с цветом в формате RGB565, а некоторые только с цветом в формате RGB666. Из-за этого возникают разногласия, если делать универсальный вывод. Для того, чтобы решить такую проблему было решено указывать при компиляции проекта какой дисплей будет подключаться, чтобы исключить возможность вывода искаженных кадров на дисплей.

Готовый код проекта выложен в открытом доступе на сайте github в отдельном репозитории для физического устройства([https://github.com/khosta77/stm32f4\\_ILI9488](https://github.com/khosta77/stm32f4_ILI9488)). Для драйвера([https://github.com/khosta77/ft2xx\\_fast\\_start](https://github.com/khosta77/ft2xx_fast_start)) на компьютер.



## 4 Исследовательский раздел

Есть группа дисплеев: 16-битные ST7735 и ILI9348, 18-битный ILI9488. Предлагается проверить качество вывода цвета у данных дисплеев.

Так как данная проверка и оценка не является стандартизированной. То методикой проверки качества дисплея будет фотографирование дисплеев по очередно в объектив камеры «Apple iphone 12» так, чтобы в центре снимка был дисплей. Камера была по центру дисплея, на расстоянии примерно 10 сантиметров. Уровень освещенности исходил только от дисплеев, эксперимент проходил без внешних источников света. Снимки будут перемещены на компьютер и по пиксельно проверены в программе «Paint» с помощью инструмента «Пипетка».

Эталонными значениями для камеры будет снимок экрана компьютера macbook. Тестовыми данными будет снимок с 6-ю цветами: красным(0xFF0000), зеленым(0x00FF00), синим(0x0000FF), желтым(0xFFFF00), фиолетовым(0xFF00FF), бирюзовым(0x00FFFF).

Важным будет уточнить, что существуют разные способы проверки качества дисплеев. Использование специальной аппаратуры, устройство, которое прилоняется к дисплею и измеряет цветовой спектр. Специальные откалиброванные камеры, для 100% передачи цвета. Ввиду отсутствия данной аппаратуры был предложен способ описанный выше. Не достаточно его может служить камера «Apple iphone 12» так как с завода она может иметь отклонения, которые добавляют насыщенности снимку, что вносит искажение в полученный результат.

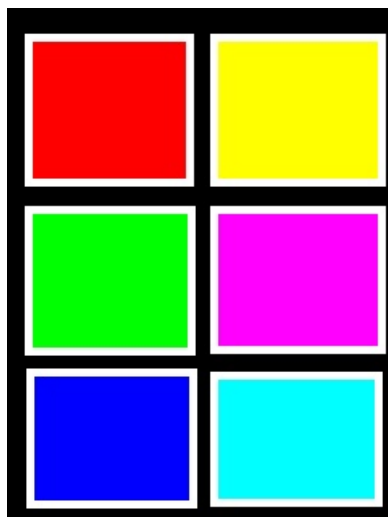


Рисунок 4.1 – Пример тестового изображения.

Проведем измерения, результат занесем в таблицу:



Рисунок 4.2 – Все измерения цвета. Слева на право: Экран Macbook, ILI9488, ILI9341, ST7735.

	Red	Green	Blue
Эталон, экран macbook			
Зона красного цвета	255	0	0
Зона зеленого цвета	63	243	0
Зона синего цвета	0	0	251
Зона желтого цвета	245	250	0
Зона фиолетового цвета	250	0	247
Зона бирюзового цвета	48	218	217
ST7735			
Зона красного цвета	245	141	168
Зона зеленого цвета	156	255	211
Зона синего цвета	0	42	255
Зона желтого цвета	223	233	194
Зона фиолетового цвета	180	131	255
Зона бирюзового цвета	157	255	255
ILI9341			
Зона красного цвета	158	55	63
Зона зеленого цвета	0	156	90
Зона синего цвета	0	28	255
Зона желтого цвета	134	201	142
Зона фиолетового цвета	104	53	255
Зона бирюзового цвета	0	161	255
ILI9488			
Зона красного цвета	139	67	90
Зона зеленого цвета	10	229	87
Зона синего цвета	15	18	255
Зона желтого цвета	178	233	176
Зона фиолетового цвета	160	117	255
Зона бирюзового цвета	0	186	250

По результатам измерений можем сделать промежуточные вывод.

Эталонные значения близки к заданным. Отклонения могут быть связаны с неидеальными условиями эксперимента: Внешние источники света, не идеально чисты экран, колебания расстояний. Из этого следует, что камера давала в дальнейших тестах не значительные отклонения.

ST7735: зеленый и красный были переданы дисплеем слишком яркими, с наполненностью всех цветов, в отличии от синего, который единственный близок к эталонным значениям. Желтый цвет близок к белому цвету, не корректная передача. Фиолетовый с более глубоким синим цветом, другие компоненты не полные из-за этого фиолетовый выходит затемненным. Бирюзовые цветы слишком близок белому цвету.

II9341 и II9488: Результаты дисплеев схожи между собой. Корректно передается синий цвет. На II9488 красный выходит чуть тусклее, чем на II9341. Остальные цвета схожи между собой, разница возникает из погрешности измерений.

Для глаза 3 протестированных дисплея выходят по качеству одинаковыми.

## ЗАКЛЮЧЕНИЕ

В данной работе я привел основной способ вывода изображений на графический дисплей. Полученное устройство позволяет тестировать различные дисплеи на качество выводимого ими изображения.

В ходе работы трудность, с которой я предполагал столкнуться оказалась не значительной. Главной проблемой оказались разные цветовые форматы различных дисплеев, из-за этого оказалось не возможным разработать универсальное устройство способное работать с любыми дисплеями, без предварительной настройки.

В будущей версии устройства я планирую решить эту проблему, чтобы можно было подключать любой дисплей и он начинал работать без предварительной программной настройки. Так же изготовление печатной платы для дисплея будет способствовать исключению дребезга контактов. В заказанной печатной плате будет расположен модуль для внешней памяти, который позволит расширить поддерживаемое разрешение.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Компьютерное зрение. Теория и алгоритмы - автор Р. Клетте
- [2] Цифровая обработка изображений - авторы Р. Гонсалес, Р. Вудс, С. Эддинс
- [3] МИР цифровой обработки. Цифровая обработка изображений - автор Б. Яне
- [4] Практические рекомендации по разработке печатных плат - автор С. Тютюков
- [5] Reference manual 0090 STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs
- [6] Полупроводниковая схемотехника. Том 1 и 2 - авторы У. Титце, К. Шенк