

# SSD1306: контроллер/драйвер для дисплеев OLED/PLED | hardware

*microsin*

**SSD1306** это однокристалльный CMOS драйвер матричных индикаторов OLED/PLED, совмещенный с графическим контроллером. Он обрабатывает матрицу из **128** сегментов и **64** общих сигналов. Эта микросхема разработана для **OLED**-панелей с общим катодом (**Common Cathode type OLED**). На основе **SSD1306** наибольшее распространение получили монохромные индикаторы **96x16**, **128x32** и **128x64** точек. Такие индикаторы можно купить на **AliExpress** и **Ebay** в виде готового модуля, снабженного интерфейсом **I2C** или **SPI**.

**SSD1306** имеет встроенное управление контрастностью, ОЗУ экрана и генератор. Все это уменьшает количество необходимых внешних компонентов и общее энергопотребление. Управление яркостью имеет **256** градаций. Данные и команды отправляются от **MCU** через аппаратно выбираемый интерфейс. Это может быть параллельный интерфейс, совместимый **6800/8080 MCU**, **I2C** или **SPI**. Контроллер **SSD1306** подходит для многих приложений портативных устройств, такие как дополнительный индикатор для телефона, **MP3** плеер, калькулятор и т. п.

Примечание: этот перевод даташита [1] был сделан для уточнения доступа к индикатору через последовательный интерфейс **I2C**. Поэтому некоторые разделы даташита, не представляющие интерес, не переведены. Это касается таблиц **10-1** (пример установки смещения экрана и начальной строки экрана без **Remap**), **10-2** (пример установки смещения экрана и начальной строки экрана с функцией **Remap**), **12-1** (DC-характеристики) и другие таблицы и рисунки, которые интересны только разработчикам **OLED**-модулей. Информацию по этим моментам см. в оригинальном даташите.

Основные функции:

- Матричная панель **128 x 64** точек
- Питание:
  - $V_{DD} = 1.65V \dots 3.3V$  для логики микросхемы
  - $V_{CC} = 7V \dots 15V$  для питания сегментов панели
- Параметры матричного дисплея:
  - Управляющее выходное напряжение для **OLED 15V** максимум
  - Максимальный ток сегмента **100 мкА**
  - Общий вытекающий ток **15 мА**
  - **256** ступеней управления током контрастности и яркости
- Встроенная буферная память (**SRAM**) экрана **128 x 64** бит
- Выбираемый уровнями ножек интерфейс подключения к **MCU**:
  - **8-битный 6800/8080** параллельный интерфейс
  - **3/4** проводный **SPI**
  - **I2C**
- Непрерывная функция прокрутки графики экрана в горизонтальном и вертикальном направлениях, которая может использоваться для равномерного износа сегментов
- Сигнал синхронизации записи в **RAM**
- Программируемая частота фреймов (**Frame Rate**) и коэффициент мультиплексирования (**Multiplexing Ratio**)
- Переназначение строк (**Row Re-mapping**) и столбцов (**Column Re-mapping**)
- Встроенный генератор
- Разводка чипа для **COG (Chip On Glass, чип на стекле)** и **COF (Chip On Film, чип на гибком шлейфе)**.
- Широкий диапазон рабочих температур: **-40 .. 85°C**

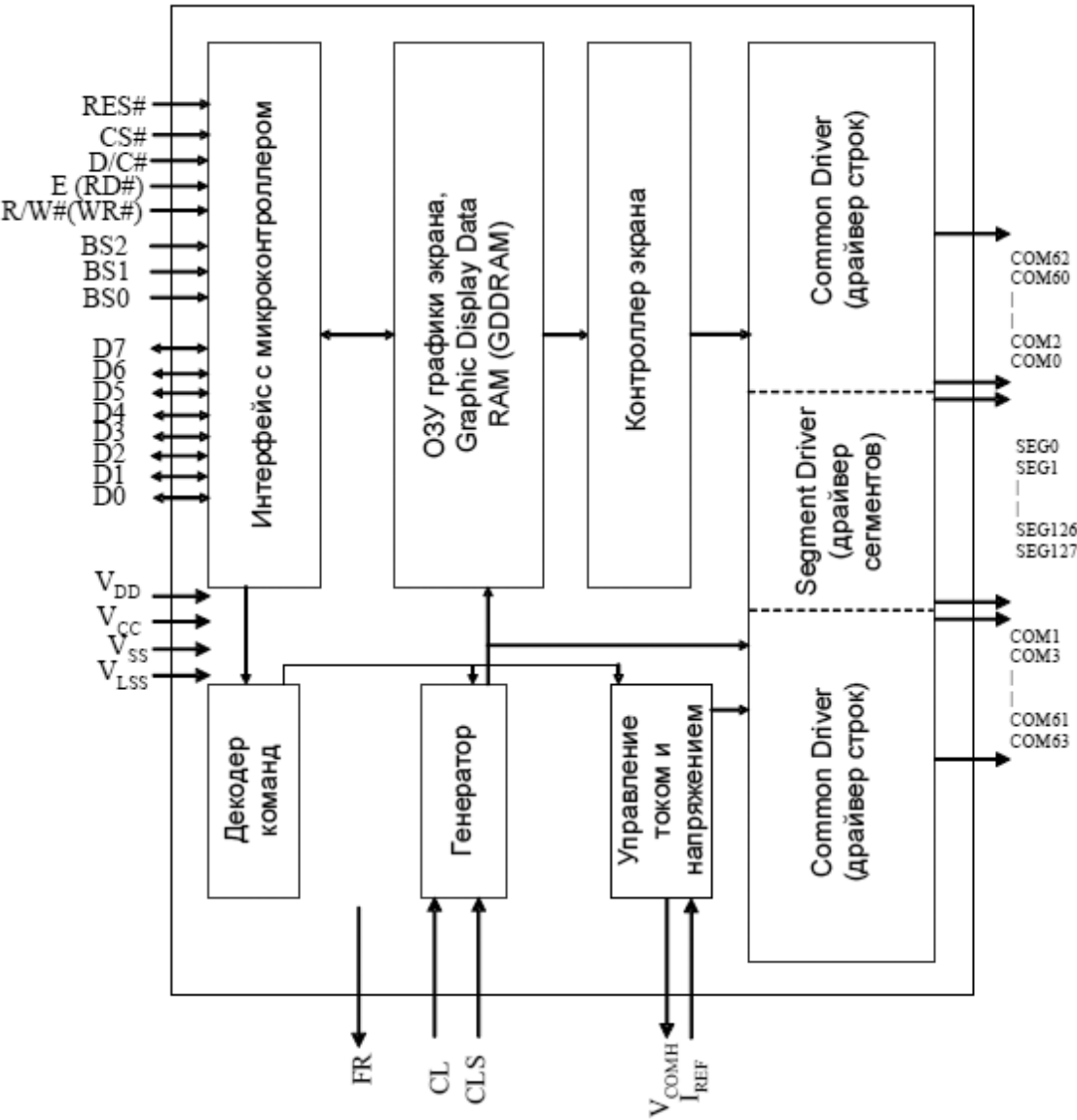


Рис. 4-1. Блок-схема SSD1306.

Таблица 6-1. Описание выводов.

Система обозначений типа выводов для таблицы 6-1 (колонка "Тип"):

**I** вход

**O** выход

**IO** двунаправленный сигнал (input/output)

**P** вывод для питания

**NC** не подключено

Мнем.	Тип	Описание
<b>V<sub>DD</sub></b>	<b>P</b>	Вывод питания для функционирования логики ядра.
<b>V<sub>CC</sub></b>	<b>P</b>	Вывод питания (вход) для силовой логики панели. Это самое высокое положительное напряжение, которое подается на контроллер.
<b>V<sub>SS</sub></b>	<b>P</b>	Общий провод (земля, <b>GND</b> ).
<b>V<sub>LSS</sub></b>	<b>P</b>	Общий провод аналоговых схем. Должен быть внешней цепью накоротко соединен с <b>V<sub>SS</sub></b> .
<b>V<sub>COMH</sub></b>	<b>O</b>	Уровень напряжения для не выбранного состояния <b>COM</b> . Между <b>V<sub>COMH</sub></b> и <b>V<sub>SS</sub></b> должен быть подключен развязывающий конденсатор.
<b>V<sub>BAT</sub></b>	<b>P</b>	Зарезервированный вывод, должен быть подключен к <b>V<sub>DD</sub></b> .
<b>BGGND</b>	<b>P</b>	Зарезервированный вывод, должен быть соединен с общим проводом.
<b>C1P/C1N</b> <b>C2P/C2N</b>	<b>I</b>	Зарезервировано для будущего использования. Эти выводы не должны быть подключены.
<b>V<sub>BREF</sub></b>	<b>P</b>	Зарезервированный вывод, не должен быть никуда подключен.

<b>BS[2:0]</b>	<b>I</b>	Выводы для выбора режима работы интерфейса с микроконтроллером, подробнее см. таблицу 7.1.
<b>I<sub>REF</sub></b>	<b>I</b>	Этот вывод задает опорный ток для выходного тока <b>SEG</b> . Между <b>I<sub>REF</sub></b> и <b>V<sub>SS</sub></b> должен быть подключен резистор (см. рис. 8-15), через который течет ток порядка 12.5μA. Для вычисления номинала резистора см. формулу в разделе "Блок управления <b>SEG/COM</b> ".
<b>FR</b>	<b>O</b>	Это выход сигнала синхронизации фрейма. Выбор правильного момента записи <b>MCU</b> данных в <b>RAM</b> относительно этого сигнала позволит избежать искажения изображения при выводе. Если этот сигнал не используется, то он должен оставаться не подключенным. Подробнее про использование сигнала <b>FR</b> см. секцию "Синхронизация <b>FR</b> ".
<b>CL</b>	<b>I</b>	Вывод для подачи сигнала внешнего тактирования. Когда разрешено тактирование от внутреннего генератора ( <b>CLS</b> = лог. 1), этот вывод не используется и должен быть подключен к <b>V<sub>SS</sub></b> . Когда внутреннее тактирование запрещено ( <b>CLS</b> = лог. 0), вывод <b>CL</b> работает как вход для подачи внешнего сигнала тактирования.
<b>CLS</b>	<b>I</b>	Вывод для разрешения внутреннего тактирования. Когда на этот вывод подан уровень лог. 1 ( <b>CLS</b> соединен с <b>V<sub>DD</sub></b> ), разрешена работа внутреннего генератора тактов. Когда на вывод <b>CLS</b> подан лог. 0 ( <b>CLS</b> соединен с <b>V<sub>SS</sub></b> ), работа внутреннего генератора запрещена и на вход <b>CL</b> должен быть сигнал внешнего тактирования.
<b>CS#</b>	<b>I</b>	Это вывод для выборки внешнего интерфейса ( <b>chip select</b> ), который подключается к микроконтроллеру. Обмен данными с микроконтроллером возможен только тогда, когда на <b>CS#</b> подан лог. 0.

<b>RES#</b>	<b>I</b>	Вывод для сигнала сброса. Когда на него подан лог. 0, то выполняется инициализация контроллера. Во время нормальной работы на вводе <b>RES#</b> должен удерживаться уровень лог. 1.
<b>D/C#</b>	<b>I</b>	Это вывод для управления подачей данных (когда лог. 1) или команды (когда лог. 0). В режиме интерфейса <b>I2C</b> этот вывод работает как <b>SA0</b> для выбора адреса подчиненного устройства <b>I2C</b> . Когда выбран 3-проводный интерфейс <b>SPI</b> , этот вывод должен быть подключен к <b>V<sub>SS</sub></b> . Для подробной информации по взаимодействию с <b>MCU</b> см. временные диаграммы на рисунках от 13-1 до 13-5.
<b>E (RD#)</b>	<b>I</b>	Когда выбран параллельный интерфейс стиля <b>6800</b> , этот вывод будет использоваться как сигнал разрешения ( <b>Enable, E</b> ). Операция чтения/записи инициируется, когда этот вывод подтянут к лог. 1, и чип контроллера выбран сигналом <b>CS#</b> . Когда выбран параллельный интерфейс стиля <b>8080</b> , этот вывод принимает сигнал чтения ( <b>Read, RD#</b> ). Операция чтения инициируется, когда этот вывод подтянут к лог. 0, и чип контроллера выбран сигналом <b>CS#</b> . Когда выбран последовательный интерфейс <b>I2C</b> или <b>SPI</b> , этот вывод должен быть соединен с <b>V<sub>SS</sub></b> .
<b>R/W# (WR#)</b>	<b>I</b>	Это вход для управления чтением/записью интерфейса с микроконтроллером. Когда выбран режим интерфейса <b>6800</b> , этот вывод работает как выбор режима чтения или записи ( <b>R/W#</b> ). Режим чтения выбран, когда на этом входе лог. 1, и режим записи, когда лог. 0. В режиме интерфейса <b>8080</b> этот вход работает как сигнал разрешения записи ( <b>WR#</b> ). Режим записи данных инициируется, когда этот вход переводится в лог. 0 и работа интерфейса разрешена ( <b>CS#</b> также в лог. 0).

		В режиме последовательного интерфейса <b>I2C</b> или <b>SPI</b> этот вывод не используется, и должен быть подключен к <b>V<sub>SS</sub></b> .
<b>D7..D0</b>	<b>IO</b>	В режиме 8-разрядных шин это двунаправленная шина данных для связи с микроконтроллером. Когда выбран последовательный интерфейс <b>SPI</b> , <b>D0</b> используется как сигнал тактов, <b>D1</b> как сигнал данных, <b>D2</b> должен оставаться не подключенным. Когда выбран режим <b>I2C</b> , сигналы <b>D2</b> , <b>D1</b> должны быть соединены друг с другом для обслуживания функции <b>SDA</b> , и <b>D0</b> используется как вход сигнала тактов <b>SCL</b> .
<b>TR0-TR6</b>	-	Зарезервированные выводы, используемые для тестирования. Не должны быть никуда подключены.
<b>SEG127..SEG0</b>	<b>O</b>	Эти выводы предназначены для управления сегментами панели <b>OLED</b> . Когда панель выключена, то на этих выводах уровень <b>V<sub>SS</sub></b> .
<b>COM63..COM0</b>	<b>O</b>	Эти выводы предназначены для управления общими сигналами панели <b>OLED</b> . Когда панель выключена, то эти выводы находятся в состоянии высокого сопротивления.
<b>NC</b>	-	Пустой вывод, который не должен никуда быть подключенным. Не группируйте друг с другом и не соединяйте другие выводы <b>NC</b> .

### [Функциональное описание]

Интерфейс с внешним микроконтроллером (**MCU**) контроллера **SSD1306** состоит из **8** выводов для сигналов данных и **5** выводов для сигналов управления. Назначение выводов в различных режимах интерфейса показана в таблице 8-1. Различные режимы интерфейса **MCU** может быть установлены уровнями на выводах **BS[2:0]** (см. таблицу 7-1).

Таблица 7-1. Выводы для выбора типа интерфейса с **MCU**.

<b>BS[2:0]</b>	<b>Тип выбранного интерфейса</b>
<b>000</b>	<b>4-проводный SPI</b>
<b>001</b>	<b>3-проводный SPI</b>
<b>010</b>	<b>I2C</b>
<b>011</b>	Зарезервировано
<b>100</b>	<b>8-бит параллельная шина 6800</b>
<b>101</b>	Зарезервировано
<b>110</b>	<b>8-бит параллельная шина 8080</b>
<b>111</b>	Зарезервировано

Примечание: **0** означает подключение к  $V_{SS}$ , **1** подключение к  $V_{DD}$ .

Таблица 8-1. Назначение выводов интерфейса MCU в различных режимах.

Имена выводов Тип интерфейса	Интерфейс данных / команд (D/C#)								Сигналы управления				
	D7	D6	D5	D4	D3	D2	D1	D0	E	R/W#	CS#	D/C#	RES#
8-bit 8080	D[7:0]								RD#	WR#	CS#	D/C#	RES#
8-bit 6800	D[7:0]								E	R/W#	CS#	D/C#	RES#
3-wire SPI	Лог. 0					NC	SDIN	SCLK	Лог. 0		CS#	Лог. 0	RES#
4-wire SPI	Лог. 0					NC	SDIN	SCLK	Лог. 0		CS#	D/C#	RES#
I <sup>2</sup> C	Лог. 0					SDA <sub>OUT</sub>	SDA <sub>IN</sub>	SCL	Лог. 0			SA0	RES#

Примечание: "Лог. 0" означает подключение к земле ( $V_{SS}$ ), NC означает "никуда не подключено". Символ # обозначает сигнал с активным нулем, например D/C# означает, что данные передаются при D/C# = лог. 1, а команда передается при D/C# = лог. 0.

**Параллельный интерфейс шины MCU 6800.** Этот параллельный интерфейс состоит из



**8** двунаправленных сигналов данных (выводы **D[7:0]**) и сигналов управления **R/W#**, **D/C#**, **E** и **CS#**.

Лог. **0** на выводе **R/W#** указывает операцию записи (**WRITE**), и лог. **1** операцию чтения (**READ**). Лог. **0** на выводе **D/C#** показывает чтение или запись команды (**COMMAND read/write**), и лог. **1** показывает чтение или запись данных (**DATA read/write**). Вход **E** обслуживает функцию сигнала защелки, когда сигнал выборки **CS#** находится в лог. **0**. Данные защелкиваются по спаду уровня **E**.

Таблица 8-2. Выводы управления интерфейса **6800**.

Функция	E	R/W#	CS#	D/C#
Запись команды	↓	L	L	L
Чтение статуса	↓	H	L	L
Запись данных	↓	L	L	H
Чтение данных	↓	H	L	H

Примечание: ↓ означает спад уровня. **H** обозначает лог. **1** сигнала, **L** означает лог. **0**.

Чтобы рабочая частота **RAM** экрана подходила к микроконтроллеру, внутри **SSD1306** делается некоторая конвейерная обработка, которая требует вставки пустого чтения (**dummy read**) перед первым реальным чтением данных дисплея. Это показано на рис. 8-1.

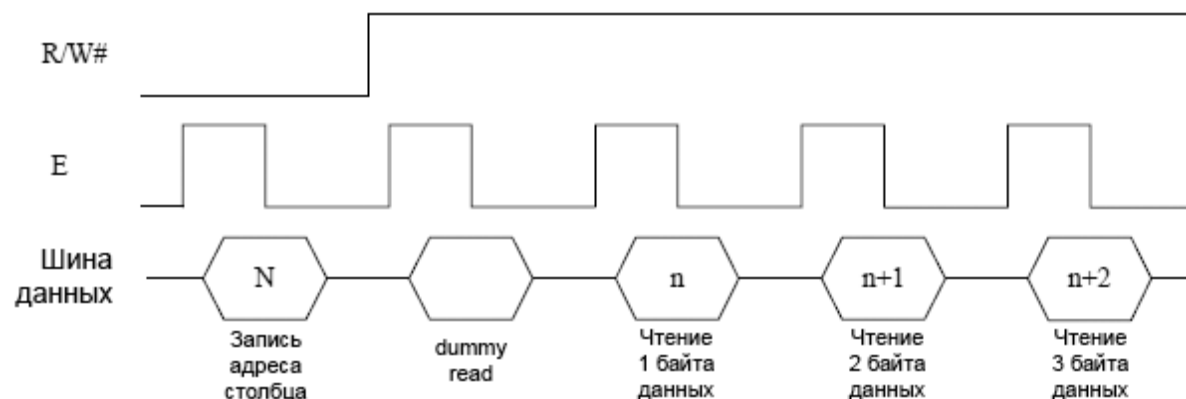


Рис. 8-1. Процедура обратного вычитывания данных из памяти дисплея - вставка **dummy read**.

**Параллельный интерфейс MCU стиля 8080.** Этот параллельный интерфейс состоит из 8 выводов двунаправленных данных (**D[7:0]**) и сигналов управления **RD#**, **WR#**, **D/C#** и **CS#**.

Лог. 0 сигнала **D/C#** показывает чтение/запись команды (**COMMAND read/write**), и лог. 1 показывает чтение/запись данных (**DATA read/write**). Фронт нарастания на входе **RD#** служит сигналом захвата данных операции чтения, когда сигнал выборки **CS#** удерживается в лог. 0. Фронт нарастания на входе **WR#** служит сигналом захвата записи, когда выборка **CS#** удерживается в лог. 0.

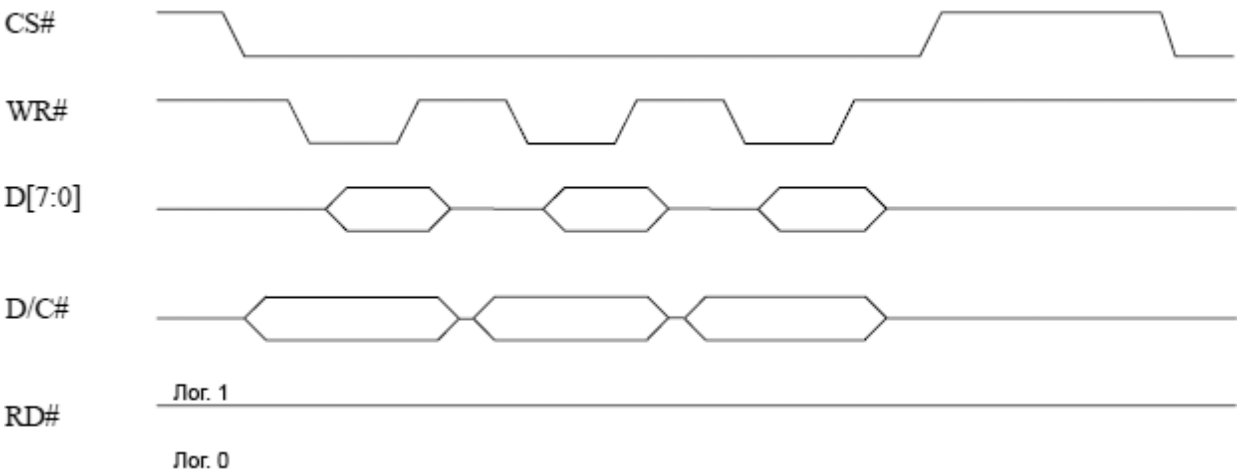


Рис. 8-2. Пример процедуры записи в режиме параллельного интерфейса 8080.

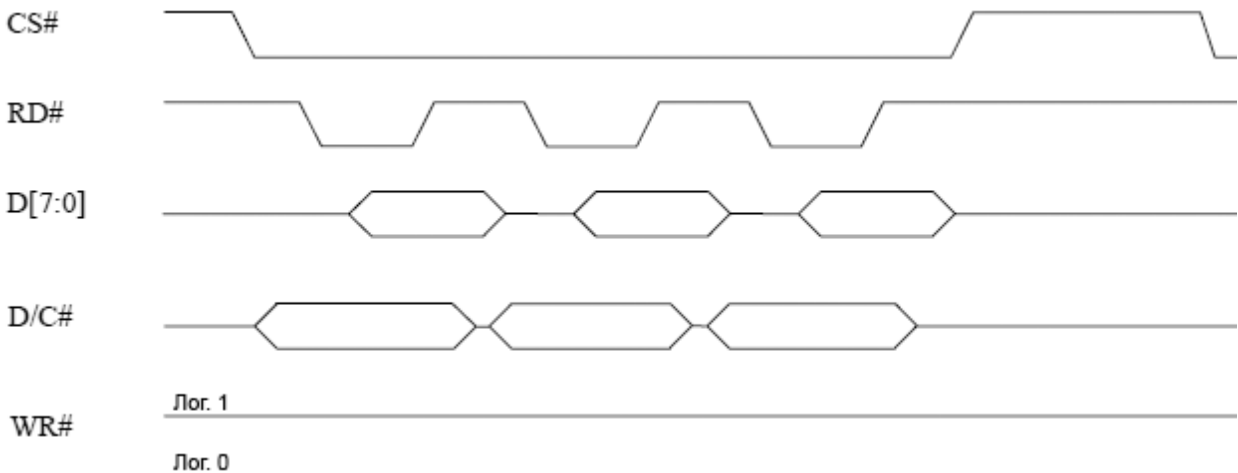


Рис. 8-3. Пример процедуры чтения в режиме параллельного интерфейса 8080.

Таблица 8-3. Выводы управления интерфейса 8080.

Функция	RD#	WR#	CS#	D/C#
---------	-----	-----	-----	------

Запись команды	H	↑	L	L
Чтение статуса	↑	H	L	L
Запись данных	H	↑	L	H
Чтение данных	↑	H	L	H

Примечание: ↑ означает фронт нарастания сигнала. H обозначает лог. 1, L обозначает лог. 0.

Чтобы рабочая частота RAM экрана подходила к микроконтроллеру, внутри SSD1306 делается некоторая конвейерная обработка, которая требует вставки пустого чтения (**dummy read**) перед первым реальным чтением данных дисплея. Это показано на рис. 8-4.

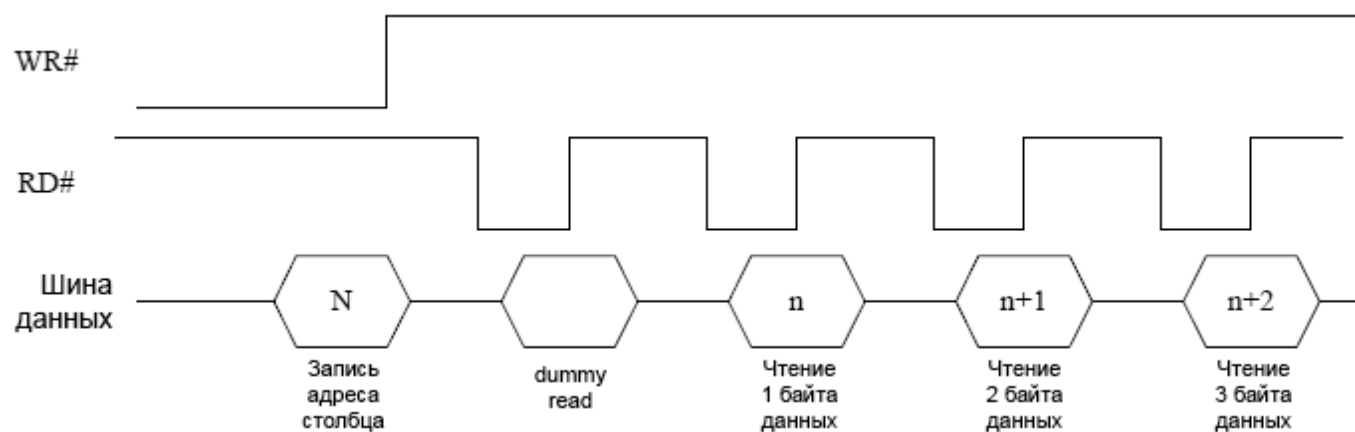


Рис. 8-4. Процедура обратного вычитывания данных из памяти дисплея - вставка **dummy read**.

**4-проводный SPI.** Этот последовательный интерфейс состоит из сигналов **SCLK** (такты), **SDIN** (входные данные), **D/C#**, **CS#**. В 4-проводном режиме **SPI** ножка **D0** действует как **SCLK**, **D1** как **SDIN**. Для не используемых выводов данных **D2** должна оставаться не подключенной, ножки **D3..D7**, **E** и **R/W# (WR#)** можно подключить к земле.

Таблица 8-4. Выводы управления 4-проводного SPI.

Функция	E(RD#)	R/W#(WR#)	CS#	D/C#	DO
Запись команды	Замкнуто на V <sub>SS</sub>		L	L	↑
Запись данных	Замкнуто на V <sub>SS</sub>		L	H	↑

Примечание: H означает лог. 1, L означает лог. 0. "Замкнуто на V<sub>SS</sub>" означает постоянную подачу лог. 0.

Данные на SDIN вдвигаются в 8-битный регистр сдвига по каждому фронту нарастания SCLK, в следующем порядке: D7, D6, ... D0. Сигнал D/C# анализируется на каждом восьмом такте, и байт данных, накопленный в регистре сдвига, записывается либо в буфер **Graphic Display Data RAM (GDDRAM)** контроллера, либо в регистр команд на том же самом такте.

В последовательном режиме разрешены только операции записи.

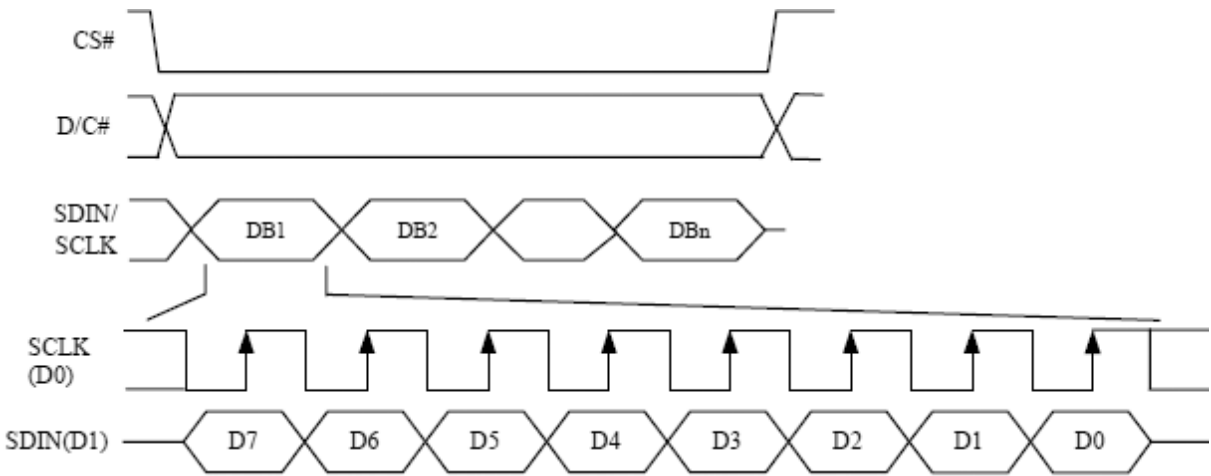


Рис. 8-5. Процедура записи в режиме 4-проводного последовательного интерфейса (4-wire Serial interface mode).

**3-проводный SPI.** Этот последовательный интерфейс состоит из сигнала **SCLK** (такты последовательных данных), **SDIN** (последовательные данные) и **CS#**. В 3-проводном режиме **SPI**, ножка **D0** функционирует как **SCLK**, **D1** как **SDIN**. Для не используемых ножек данных **D2** должна оставаться не подключенной, выводы **D3 .. D7**, **R/W# (WR#)#**, **E** и **D/C#** можно подключить к земле.

Функционирование 3-проводного режима **SPI** подобно 4-проводному **SPI** с тем лишь отличием, что ножка **D/C#** не используется. В регистр сдвига вдвигаются **9**-бит в следующей последовательности: бит **D/C#**, биты **D7 .. D0**. Бит **D/C#** первый бит в последовательных данных будет определять назначение следующих **8** бит (байт данных) - он должен быть записан в буфер экрана **Display Data RAM (D/C# = 1)** или в регистр команд (**D/C# = 0**).

В последовательном режиме разрешены только операции записи.

Таблица 8-5. Выводы управления 3-проводного последовательного интерфейса.

Функция	E(RD#)	R/W#(WR#)	CS#	D/C#	D0
Запись команды	Замкнуто на V <sub>SS</sub>		L	Замкнуто на V <sub>SS</sub>	↑
Запись данных	Замкнуто на V <sub>SS</sub>		L	Замкнуто на V <sub>SS</sub>	↑

Примечание: L означает лог. 0. "Замкнуто на V<sub>SS</sub>" означает постоянную подачу лог. 0.

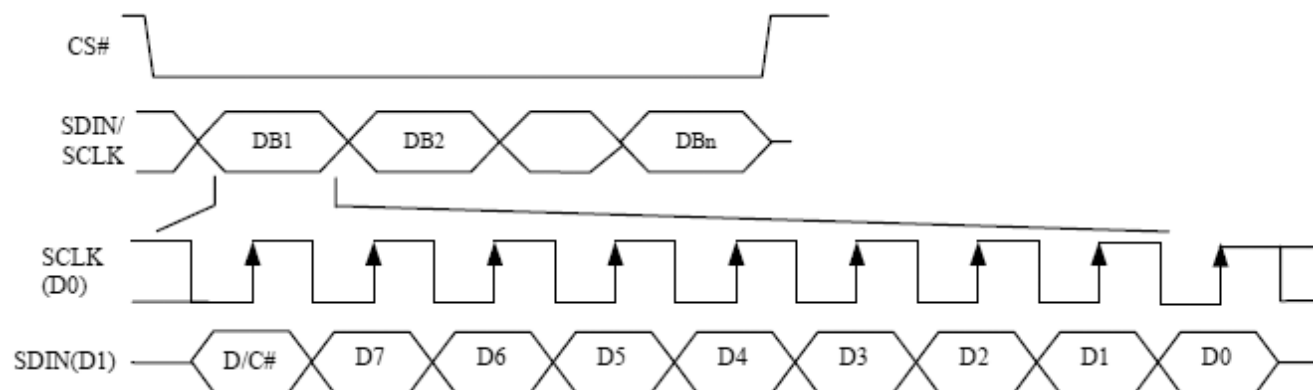


Рис. 8-6. Процедура записи в 3-проводном режиме SPI.

### [I2C]

Коммуникационный интерфейс **I2C** состоит из ножки установки адреса подчиненного устройства **SA0**, сигнала данных **SDA** (**SDAOUT/D2** для выхода и **SDAIN/D1** для входа) и сигнала тактов **SCL (D0)**. Сигналы **SDAOUT** и **SDAIN** обычно соединяются параллельно для получения одного сигнала **SDA**. Оба сигнала **SDA** и **SCL** должны быть подтянуты к лог. 1 с помощью внешних резисторов (**pull-up**) типичным номиналом от 4.7 кОм до 10 кОм. Сигнал **RES#** используется для инициализации устройства (в реальности его использование не обязательно, достаточно **RES#** подтянуть резистором к лог. 1).

**Ножка бита адреса подчиненного устройства SA0.** SSD1306 распознает адрес подчиненного устройства перед передачей или приемом какой-либо информации по шине **I2C**. Устройство будет отвечать на адрес подчиненного устройства (**slave address**), где присутствует бит **SA0** и бит выбора операции чтение/запись (**read/write**, бит **R/W#**). Этот адрес имеет следующий формат:

b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	1	1	0	SA0	R/W#

**SA0** предоставляет возможность расширения **slave**-адреса. Таким образом, для коммуникации с **SSD1306** можно выбрать адрес **0111100 (0x3C)** или **0111101 (0x3D)**, позволяя

тем самым к одной шине **I2C** параллельно подключить два контроллера **SSD1306**. Ножка **D/C#** работает как бит **SA0**. Бит **R/W#** в **slave**-адресе используется для определения рабочего режима шины **I2C**. Если **R/W#=1**, то это режим чтения (**read mode**). Если **R/W#=0**, то это режим записи (**write mode**).

**Сигнал данных SDA.** **SDA** функционирует как канал обмена данными между передатчиком и приемником. Данные и сигнал подтверждения (**acknowledgement, ACK**) посылаются через **SDA**.

Следует отметить, что сопротивление дорожки **ITO (Indium tin oxide, материал анода)** и сопротивление подтяжки **SDA** становятся делителем потенциала напряжения. В результате подтверждение было бы невозможным для достижения допустимого уровня лог. **0** на **SDA**.

Сигналы **SDAIN** и **SDAOUT** соединяются вместе, создавая один двунаправленный сигнал **SDA**. Ножка **SDAIN** должна быть подключена к **SDA** обязательно, а ножку **SDAOUT** можно не подключать. Когда ножка **SDAOUT** отключена, то сигнал подтверждения будет игнорироваться шиной **I2C**.

**Сигнал тактов SCL.** Передача информации по шине **I2C** осуществляется по сигналу тактов **SCL**. Каждая передача бита данных происходит за один период тактов **SCL**.

**Запись данных по шине I2C.** Интерфейс **I2C** дает доступ к записи данных и команд в устройство. На рис. 8-7 показан режим записи (**write mode**) в хронометрическом порядке.



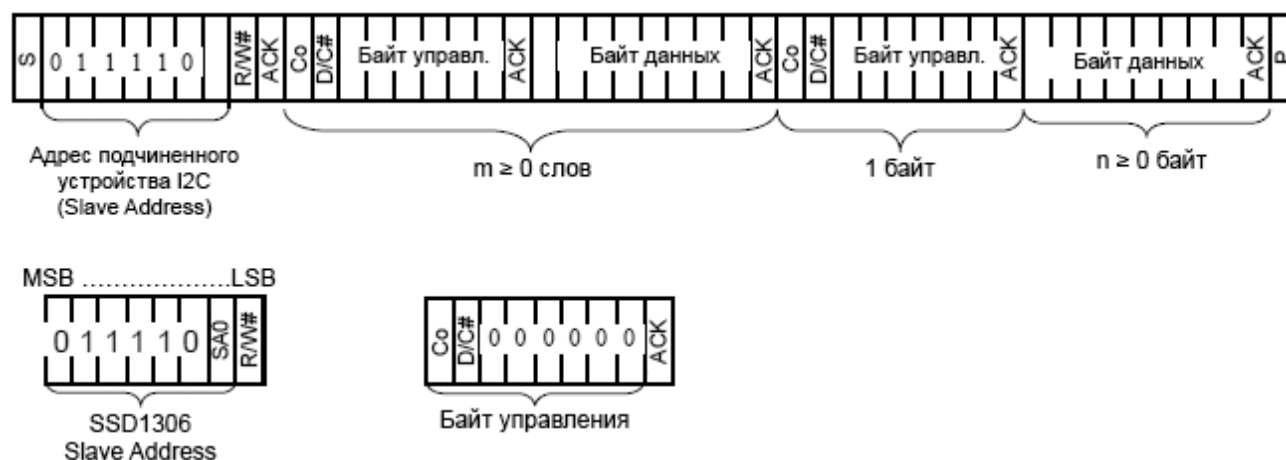


Рис. 8-7. Формат данных шины I2C.

Легенда мнемоник рисунка 8-7:

**Co Continuation**, бит продолжения.

**D/C# Data / Command Selection**, бит выбора данные или команда.

**ACK Acknowledgement**, бит сигнала подтверждения протокола I2C.

**SA0 Slave address**, бит 0 адреса подчиненного устройства протокола I2C.

**R/W# Read / Write Selection**, бит выбора операции чтение / запись протокола I2C.

**S START condition**, сигнал старта протокола I2C.

**P STOP condition**, сигнал останова протокола I2C.

Запись работает следующим образом:

1) **Master**-устройство (управляющий микроконтроллер, **MCU**) инициирует коммуникацию посылкой сигнала **START (start condition)**. Определение **START** показано диаграммой рисунка 8-8. **START** устанавливается переводом **SDA** от лог. 1 к лог. 0 при постоянном лог. 1 на **SCL**.

2) За **START** передается **slave**-адрес, чтобы на него отреагировало подчиненное устройство. В нашем случае это **SSD1306**, для которого допустимым адресом может быть **b0111100** или

**b0111101**, что определяется уровнем на **SA0** (ножка **D/C#** работает как **SA0**).

3) Режим записи устанавливается передачей лог. **0** в бите **R/W#** этого адреса.

4) Сигнал подтверждения **ACK** будет генерироваться каждый раз после приема одного байта данных, включая **slave**-адрес и бит **R/W#**. На рис. **8-9** показаны диаграммы сигнала подтверждения **ACK**. **ACK** определяется как подтяжка **SDA** к лог. **0** во время уровня лог. **1** импульса **SCL**, связанного с выдачей подтверждения.

5) После передачи **slave**-адреса передается либо байт управления, либо байт данных **SDA**. Байт управления состоит главным образом из бит **Co** и **D/C#**, за которыми идут шесть нулевых бит.

a. Если бит **Co** установлен в лог. **0**, то следующие передачи будут содержать только байты данных.

b. Бит **D/C#** определяет, чем будет следующий байт данных - командой или данными. Если бит **D/C#** установлен в лог. **0**, то следующий байт это команда. Если бит **D/C#** установлен в лог. **1**, то следующий байт это байт данных, который должен быть сохранен в графический буфер **GDDRAM**. Указатель адреса столбца **GDDRAM** будет увеличиваться автоматически на **1** с каждым записанным байтом.

6) Бит подтверждения **ACK** будет генерироваться после каждого приема байта управления или байта данных.

7) Режим записи завершается, когда выдается сигнал **STOP (stop condition)**. Сигнал **STOP** также показан на рис. **8-8**. **STOP** устанавливается путем перевода **SDA** из лог. **0** в лог. **1**, когда сигнал **SCL** находится на уровне лог. **1**.

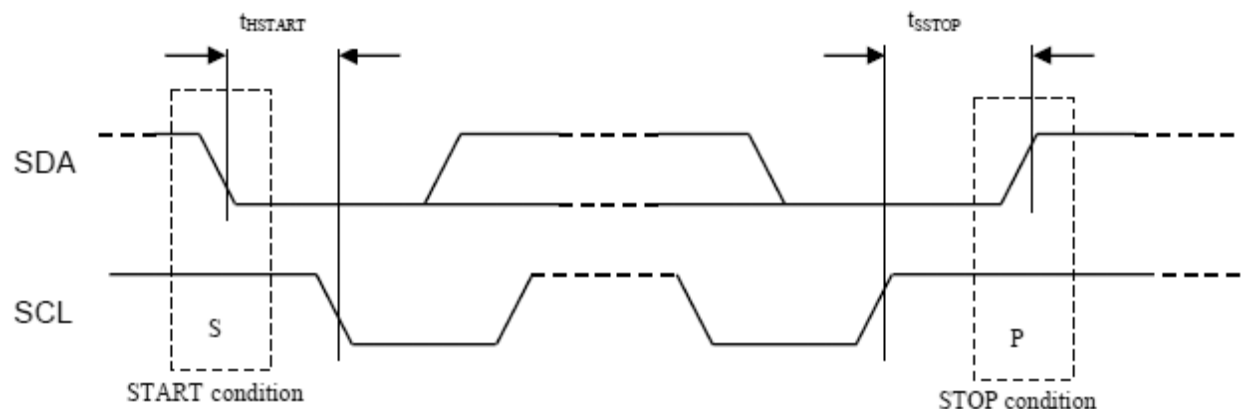


Рис. 8-8. Определение событий сигналов **START** и **STOP** (START condition, STOP condition).

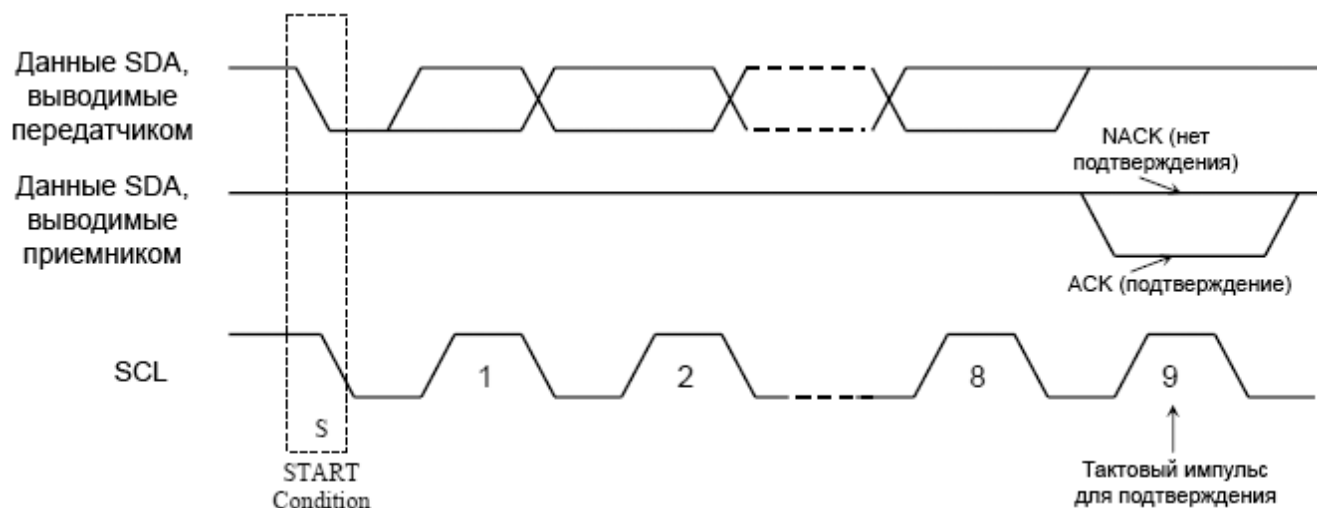


Рис. 8-9. Определение **ACK** (подтверждение, **acknowledgement condition**).

Обратите внимание на некоторые важные особенности протокола шины **I2C**:

1. Бит данных, передаваемый с каждым импульсом **SCL**, должен оставаться в стабильном уровне, пока **SCL** находится в лог. 1. Это показано диаграммами на рис. 8-10. Кроме передачи сигналов **START** или **STOP**, сигнал **SDA** должен переключаться только тогда, когда **SCL**

находится в лог. 0.

2. Оба сигнала, **SDA** и **SCL**, должны получать верхнюю подтяжку к лог. 1 с помощью внешних резисторов.

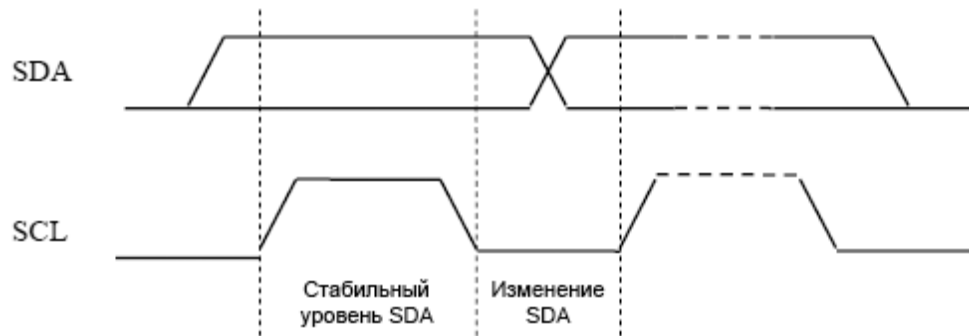


Рис. 8-10. Как передаются биты данных.

**Декодер команды.** Этот блок определяет, как интерпретировать входные данные - либо как данные графики, либо как команду. Данные интерпретируются на основе бита **D/C#** (либо передаваемого через интерфейс, либо с помощью отдельного вывода **D/C#**).

Если **D/C#** в лог. 1, то **D[7:0]** интерпретируются как данные графики, записываемые в буфер **Graphic Display Data RAM (GDDRAM)**. Если **D/C#** в лог. 0, то данные **D[7:0]** интерпретируются как команда. После поступления команды данные будут декодироваться, и записываться в соответствующий регистр команд.

**Схема генератора и генератор интервалов времени (Oscillator Circuit и Display Time Generator).** Этот блок содержит внутренний **RC**-генератор с малым энергопотреблением. Рабочая частота тактов (**CLK**) может генерироваться либо от внутреннего генератора, либо подаваться на вход **CL** от внешнего источника. Выбор источника тактирования делается ножкой **CLS**. Если на **CLS** лог. 1, то выбирается внутренний генератор, тогда ножка **CL** должна оставаться не подключенной. Если на **CLS** лог. 0, то внутренний генератор запрещается, и внешний тактовый сигнал должен быть подан на вывод **CL**. Когда выбран внутренний генератор, его выходная частота может быть изменена битами **A[7:4]** команды

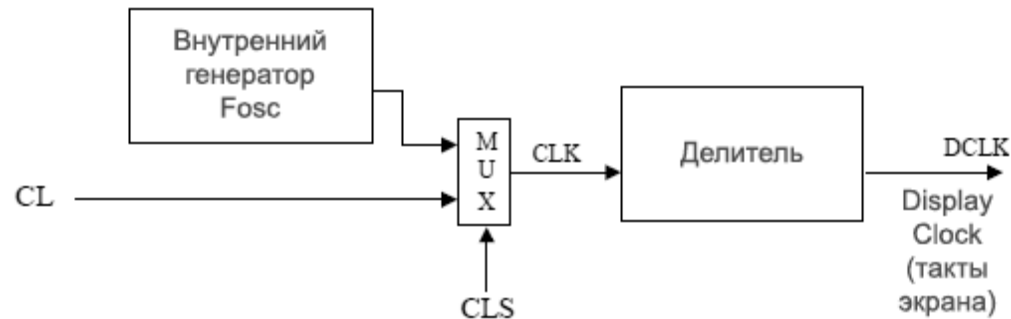
**0xD5.**

Рис. 8-11. Oscillator Circuit и Display Time Generator.

Такты экрана (display clock, DCLK) для Display Timing Generator получаются из сигнала CLK. Команда 0xD5 программирует коэффициент деления D значением от 1 до 16:

$$DCLK = F_{OSC} / D$$

Частота кадров экрана определяется по следующей формуле:

$$F_{FRM} = F_{OSC} / (D \cdot K \cdot NoOfMux)$$

Здесь:

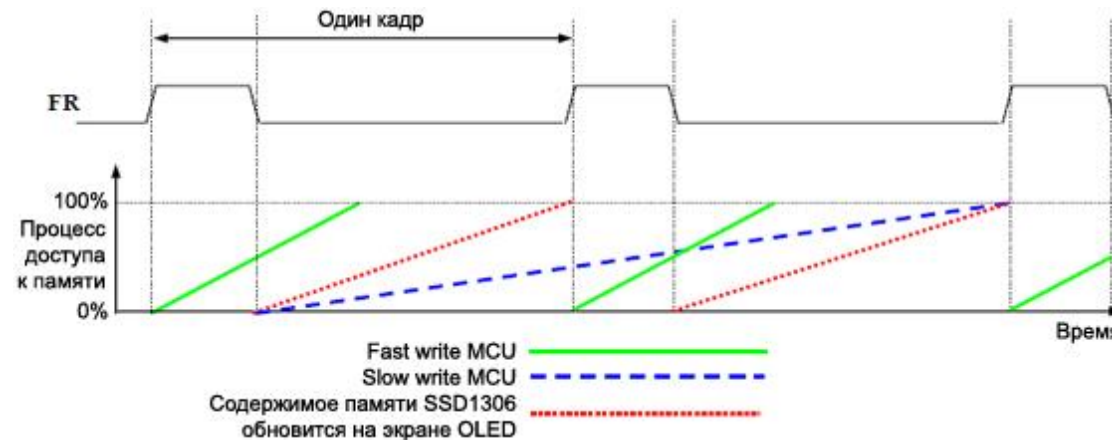
- D обозначает коэффициент деления тактов. Он устанавливается битами A[3:0] команды 0xD5. Коэффициент деления должен быть в диапазоне 1 .. 16.
- K это количество тактов на строку экрана. Значение K вычисляется по формуле:

**K = Phase 1 period + Phase 2 period + BANK0 pulse width**, и после включения питания или сброса **K = 2 + 2 + 50 = 54**.

Подробнее см. секцию "Segment Drivers / Common Drivers", где дано определение для Phase (фаза сигнала управления сегментами).

- **NoOfMux (Number of multiplex ratio, коэффициент умножения)** устанавливается командой **0xA8**. Значение по умолчанию **63** (т. е. **64MUX**).
- **FOSC** это частота генератора. Её можно поменять битами **A[7:4]** команды **0xD5**. Чем больше значение, тем выше частота.

**Синхронизация FR.** Сигнал синхронизации фрейма (кадра) **FR** может использоваться для предотвращения эффекта разрыва изображения (**tearing effect**).



Время момента начала записи нового изображения в драйвер **OLED** зависит от скорости записи управляющего **MCU**. Если **MCU** может завершить запись образа экрана за один период фрейма, то это классифицируется как быстро записывающий **MCU** (**fast write MCU**). Для **MCU**, которому нужно больше времени для завершения записи (больше, чем один фрейм, но в пределах двух фреймов), это классифицируется как медленно записывающий **MCU** (**slow write MCU**).

Для **fast write MCU**: **MCU** должен запустить запись данных нового фрейма сразу после нарастания уровня импульса **FR**, и он должен завершить запись перед следующим нарастанием уровня импульса **FR**.

Для **slow write MCU**: **MCU** должен запустить запись данных нового фрейма после спада

уровня первого импульса **FR** и должен завершить запись данных перед фронтом нарастания уровня третьего импульса **FR**.

**Схема сброса (Reset Circuit).** Когда на вход **RES#** подан лог. 0, то **SSD1306** иницирует следующее состояние:

1. Экран выключается.
2. Активируется режим отображения **128 x 64**.
3. Активируется обычное отображение данных дисплея на сегменты и строки (**SEG0** отображается на адрес **0x00** и **COM0** отображается на адрес **0x00**).
4. Очищается регистр сдвига в последовательном интерфейсе.
5. Начальная строка экрана устанавливается на адрес **0** буфера экрана **RAM**.
6. Счетчик адреса столбца устанавливается в **0**.
7. Устанавливается нормальное направление сканирование выходов **COM**.
8. Регистр управление контрастностью устанавливается в **0x7F**.
9. Устанавливается нормальный режим отображения (эквивалентный команде **0xA4**).

**Segment Drivers / Common Drivers (SEGx/COMx).** Драйверы сегмента **SEG** поставляют **128** источников тока для управления матрицей панели **OLED**. Управляющий ток может быть настроен в диапазоне от **0** до **100** мкА, всего **256** уровней настройки тока. Драйверы **COM** генерируют импульсы сканирования напряжения.

Процесс генерации сигнала управления сегментом делится на три фазы:

1. На **phase 1**, точка **OLED**, получившая заряд от предыдущего изображения, разряжается, чтобы подготовиться к отображению следующего содержимого экрана.

2. На **phase 2**, точка **OLED** устанавливается на целевое напряжение. На точку подается импульс тока, чтобы она достигла соответствующего уровня от напряжения **V<sub>SS</sub>**.

Длительность периода **phase 2** можно запрограммировать в значениями длины в диапазоне от **1** до **15** интервалов тактов **DCLK**. Если значение емкости точки панели **OLED** больше, то требуется больший период времени для заряда конденсатора точки, чтобы она достигла желаемого напряжения.

3. На **phase 3**, драйвер **OLED** переключается на использование источника тока для питания точек **OLED**, поэтому такая фаза называется фазой управляющего тока.

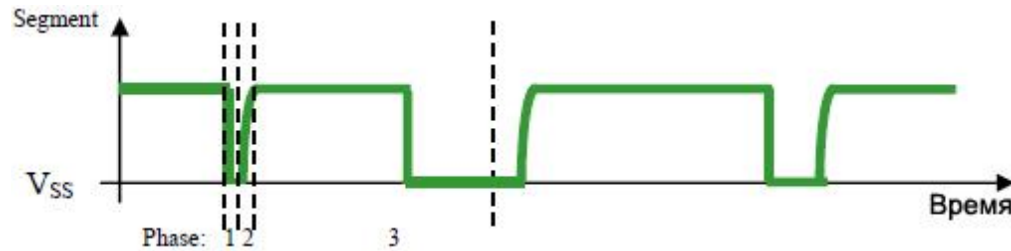


Рис. 8-12. Форма сигнала выхода управления сегмента (SEG), состоящего из 3 фаз.

После завершения **phase 3** драйвер микросхемы **SSD1306** переходит обратно к **phase 1**, чтобы отобразить следующую строку данных изображения. Этот трехфазный цикл непрерывно работает для обновления изображения на панели **OLED**.

На **phase 3** если ширина длительности импульса тока установлена на **50**, то после завершения **50** периодов **DCLK** в фазе управляющего тока микросхема драйвера перейдет обратно к **phase**, чтобы отобразить следующую строку экрана.

### [Graphic Display Data RAM (GDDRAM)]

**GDDRAM** это статическое ОЗУ (**RAM**), каждый бит которого соответствует точке на экране (буфер данных экрана). Размер этого **RAM 128 x 64** бит, и **RAM** поделена на **8** страниц, от **PAGE0** до **PAGE7**, что используется для монохромного матричного экрана **128x64** точек, как показано на рис. 8-13.



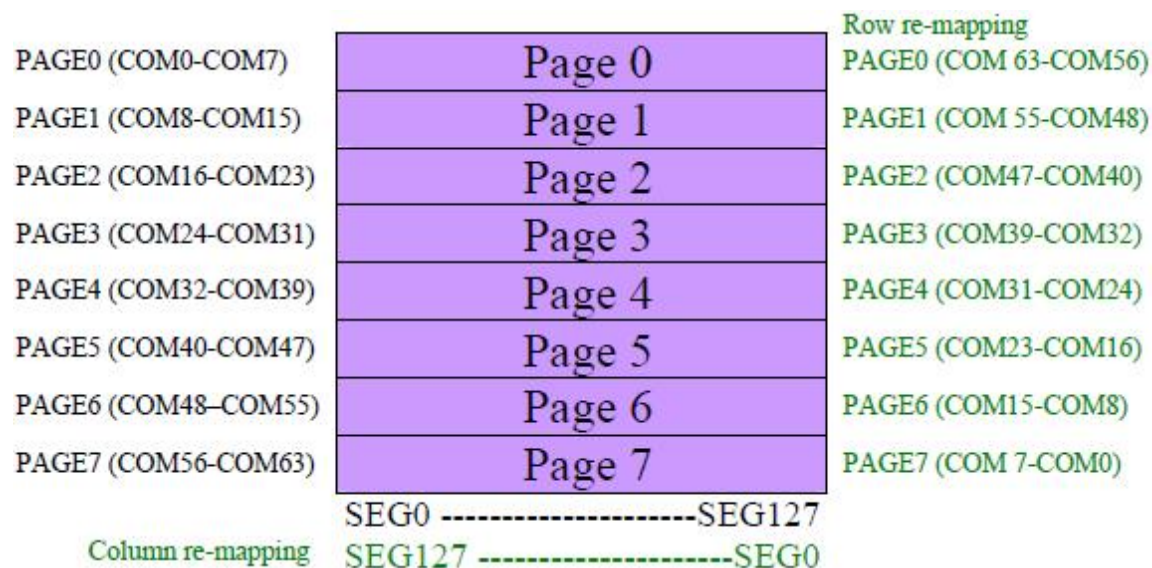


Рис. 8-13. Структура страниц GDDRAM контроллера SSD1306.

Примечание: "**Column re-mapping**" означает переназначение столбцов, "**Row re-mapping**" переназначение строк. **PageX** обозначает страницу памяти GDDRAM.

Когда один байт данных записывается в GDDRAM, то заполняются все данные на строках изображения этой страницы текущего столбца, т. е. весь столбец (8 точек), на который указывает текущий адрес столбца. Бит данных **D0** записывается в верхнюю строку страницы, и так далее (бит **D7** записывается в нижнюю строку страницы), как показано на рис. 8-14.



Рис. 8-14. Организация GDDRAM по умолчанию, нет переназначения строк (без row re-mapping) и столбцов (без column-remapping).

Чтобы достичь гибкости в формировании изображения и разводки сигналов SEG и COM, можно программно изменить их привязку (**Segment re-mapping, Common outputs re-mapping**), как показано на рис. 8-13.

Для вертикального сдвига картинки на экране можно записать внутренний регистр, который хранит начальную строку экрана, что управляет порцией данных RAM, привязанных к экрану (см. команду **0xD3**).

Этот блок используется для распределения поступающего питания на разные внутренние уровни напряжения и тока.

- $V_{CC}$  это положительное напряжение питания.
- $V_{COMN}$  это уровень отмены выборки COM (**Common deselected level**). Генерируется внутри микросхемы контроллера.
- $V_{LSS}$  это путь для аналоговых токов и тока панели.
- $I_{REF}$  это опорный источник тока для драйверов тока сегментов  $I_{SEG}$ . Взаимосвязь между

опорным током и током сегмента следующая:

$$I_{SEG} = \text{Contrast} / 256 \cdot I_{REF} \cdot \text{scalefactor}$$

Значение **Contrast** (0 .. 255) устанавливается командой установки контраста (**Set Contrast 0x81**), и значение **scalefactor** по умолчанию равно 8.

Величина  $I_{REF}$  управляется значением резистора, который подключен между ножкой  $I_{REF}$  и  $V_{SS}$ , как показано на рис. 8-15. Рекомендуется установить  $I_{REF}$  в значение  $12.5 \pm 2$  мкА, чтобы  $I_{SEG} = 100$  мкА был при максимальной контрастности 255.

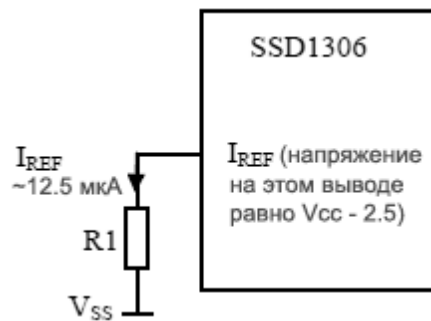


Рис. 8-15. Настройка резистором тока  $I_{REF}$ .

Поскольку напряжение на выводе  $I_{REF}$  равно  $V_{CC} - 2.5V$ , то значение резистора при условиях  $I_{REF} = 12.5$  мкА,  $V_{CC} = 12V$  можно вычислить по формуле:

$$R1 = (\text{Напряжение на } I_{REF} - V_{SS}) / I_{REF} = (12 - 2.5) / 12.5 \text{ мкА} = 760 \text{ кОм}$$

На следующих картинках показана рекомендуемая последовательность включения (**power ON**) и выключения (**power OFF**) для SSD1306.

Последовательность **Power ON**:

1. Подача питания на  $V_{DD}$ .
2. После того, как напряжение  $V_{DD}$  станет стабильным, перевод сигнала  $RES\#$  в лог. 0 на время как минимум 3 мкс ( $t_1$ )<sup>(4)</sup>, после чего перевод  $RES\#$  в лог. 1.
3. После перевода вывода  $RES\#$  в лог. 0 подождать как минимум 3 мкс ( $t_2$ ). Затем подать питание  $V_{CC}$ .<sup>(1)</sup>
4. После того, как  $V_{CC}$  станет стабильным, послать команду  $0xAF$  для включения дисплея.  $SEG/COM$  включатся через 100 мс ( $t_{AF}$ ).

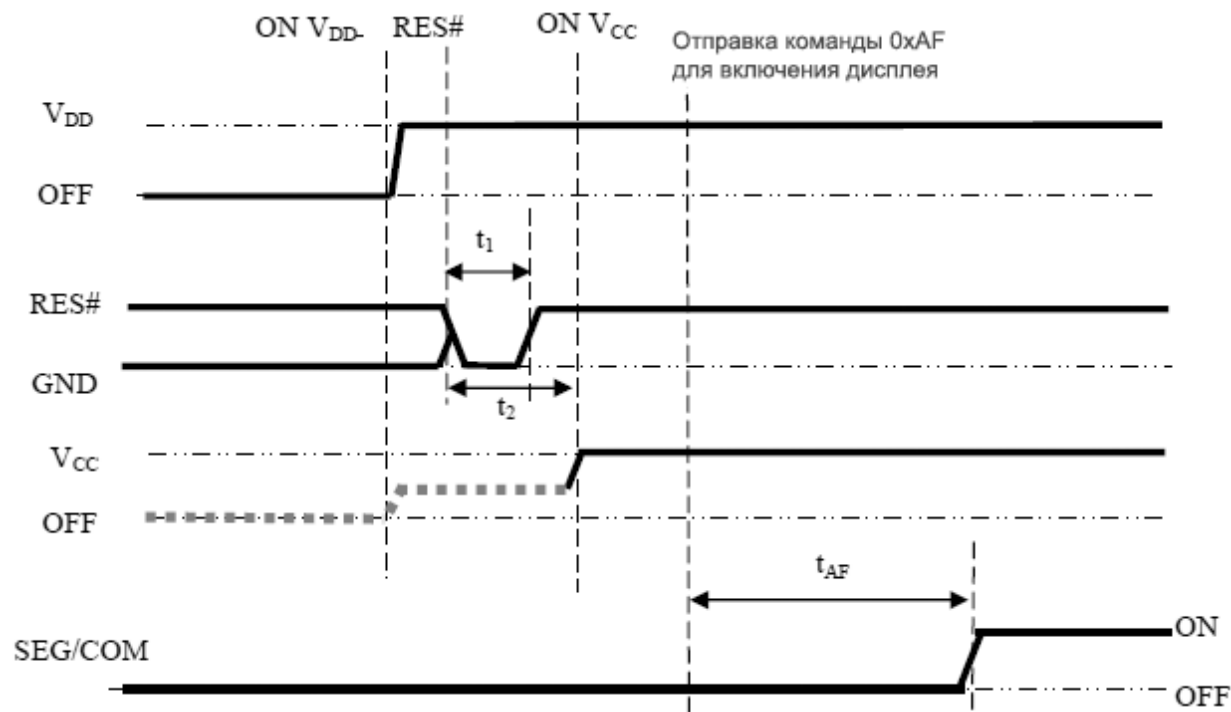


Рис. 8-16. Последовательность включения (Power ON).

Последовательность Power OFF:

1. Послать команду  $0xAE$  для выключения дисплея.

2. Снять питание  $V_{CC}$ . (1)(2)(3)

3. Снять питание  $V_{DD}$  после  $t_{OFF}$  <sup>(5)</sup> (типичное время  $t_{OFF} = 100$  мс).

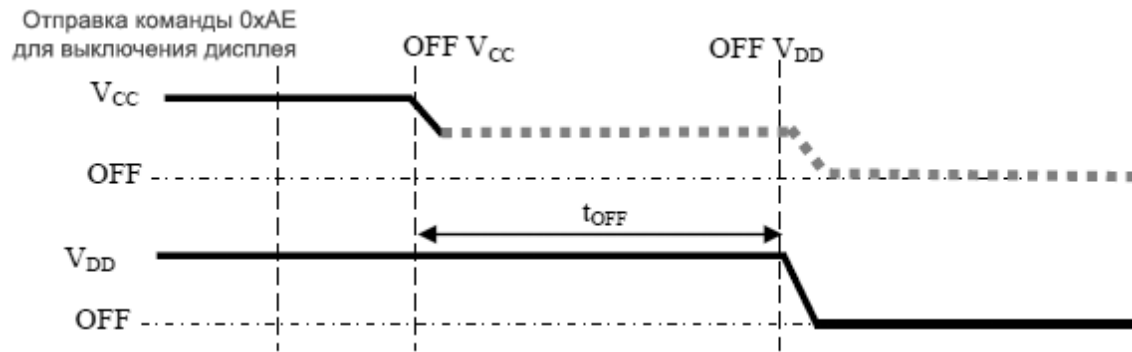


Рис. 8-17. Последовательность выключения (Power OFF).

Примечания:

- (1) Поскольку между  $V_{DD}$  и  $V_{CC}$  подключена схема защиты от ESD (статическое электричество),  $V_{CC}$  становится ниже  $V_{DD}$  всякий раз, когда на  $V_{DD}$  подается питание и на  $V_{CC}$  питания нет, что показано пунктирной линией  $V_{CC}$  на рис. 8-16 и рис. 8-17.
- (2) Уровень  $V_{CC}$  должен удерживаться плавающим (т. е. питание запрещено), когда  $V_{CC}$  выключено.
- (3) Выводы питания ( $V_{DD}$ ,  $V_{CC}$ ) никогда не должны подтягиваться к земле ни при каких условиях.
- (4) Значения регистров сбрасываются после  $t_1$ .
- (5) Питание  $V_{DD}$  не должно быть снято перед снятием питания  $V_{CC}$ .

### [Таблица команд SSD1306]

Таблица 9-1. Таблица команд при  $D/C\#=0$ ,  $R/W\#(WR\#) = 0$ ,  $E(RD\#=1)$ , если не указано нечто другое.

<b>1. Основные команды</b>											
<b>D/C#</b>	<b>Hex</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	<b>Команда</b>	<b>Описание</b>
<b>0 0</b>	<b>81 A[7:0]</b>	<b>1 A7</b>	<b>0 A6</b>	<b>0 A5</b>	<b>0 A4</b>	<b>0 A3</b>	<b>0 A2</b>	<b>0 A1</b>	<b>1 A0</b>	<b>Set Contrast Control</b>	Команда из двух байт для выбора одной из <b>256</b> ступеней контрастности. Чем больше значение во втором байте, тем выше контрастность. Значение по умолчанию после сброса <b>0x7F</b> .
<b>0</b>	<b>A4/A5</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>X0</b>	<b>Entire Display ON</b>	<b>0xA4, X0=0:</b> возобновить отображение содержимого <b>RAM</b> (состояние после сброса). <b>0xA5, X0=1:</b> полное включение дисплея, вывод игнорирует содержимое <b>RAM</b> .
<b>0</b>	<b>A6/A7</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>X0</b>	<b>Set Normal/Inverse Display</b>	<b>0xA6, X[0]=0:</b> нормальное отображение (состояние по умолчанию после сброса). Лог. <b>0</b> бита соответствует

											погашенной точке, 1 светящейся точке. A7, X[0]=1: инверсное отображение.
0	AE/AF	1	0	1	0	1	1	1	X0	Set Display ON/OFF	0xAE, X[0]=0: экран выключен (sleep mode, режим сна. Состояние после сброса). 0xAF X[0]=1: экран включен в нормальном режиме.

## 2. Команды скроллинга

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Команда	Описание
0	26/27	0	0	1	0	0	1	1	X0	Continuous Horizontal Scroll Setup	0x26, X0=0, горизонтальный сдвиг вправо, 0x27, X0=1, горизонтальный сдвиг влево (на один столбец). A пустой байт. B[2:0] определяет адрес начальной страницы: 000 PAGE0, .., 111 PAGE7. C[2:0] установка интервала времени между каждым шагом в количестве кадров: 000 5 кадров, 001 - 64
0	A[7:0]	0	0	0	0	0	0	0	0		
0	B[2:0]	*	*	*	*	*	B2	B1	B0		
0	C[2:0]	*	*	*	*	*	C2	C1	C0		
0	D[2:0]	*	*	*	*	*	D2	D1	D0		
0	E[7:0]	0	0	0	0	0	0	0	0		
0	F[7:0]	1	1	1	1	1	1	1	1		

											кадра, <b>010</b> - <b>128</b> кадров, <b>011</b> - <b>256</b> кадров, <b>100</b> - <b>3</b> кадра, <b>101</b> - <b>4</b> кадра, <b>110</b> - <b>25</b> кадров, <b>111</b> - <b>2</b> кадра. <b>D[2:0]</b> определяет адрес конечной страницы так же, как и <b>B[2:0]</b> . Однако значение <b>D[2:0]</b> должно быть больше или равно <b>B[2:0]</b> . Е и F пустые байты.
<b>0</b> <b>0</b> <b>0</b> <b>0</b> <b>0</b> <b>0</b>	<b>29/2A</b> <b>A[7:0]</b> <b>B[2:0]</b> <b>C[2:0]</b> <b>D[2:0]</b> <b>E[5:0]</b>	<b>0</b> <b>0</b> <b>*</b> <b>*</b> <b>*</b> <b>*</b>	<b>0</b> <b>0</b> <b>*</b> <b>*</b> <b>*</b> <b>*</b>	<b>1</b> <b>0</b> <b>*</b> <b>*</b> <b>*</b> <b>E5</b>	<b>0</b> <b>0</b> <b>*</b> <b>*</b> <b>*</b> <b>E4</b>	<b>1</b> <b>0</b> <b>*</b> <b>*</b> <b>*</b> <b>E3</b>	<b>0</b> <b>0</b> <b>B2</b> <b>C2</b> <b>D2</b> <b>E2</b>	<b>X1</b> <b>0</b> <b>B1</b> <b>C1</b> <b>D1</b> <b>E1</b>	<b>X0</b> <b>0</b> <b>B0</b> <b>C0</b> <b>D0</b> <b>E0</b>	<b>Continuous Vertical and Horizontal Scroll Setup</b>	<b>0x29, X1X0=01</b> , вертикальный и горизонтальный сдвиг вправо, <b>0x2A, X1X0=10</b> , вертикальный и горизонтальный сдвиг влево (на один столбец). <b>A</b> пустой байт. <b>B[2:0]</b> определяет адрес начальной страницы: <b>000 PAGE0, ..., 111 PAGE7</b> . <b>C[2:0]</b> установка интервала времени между каждым шагом в количестве кадров: <b>000</b> 5 кадров, <b>001</b> - <b>64</b> кадра, <b>010</b> - <b>128</b> кадров, <b>011</b> - <b>256</b> кадров, <b>100</b> - <b>3</b> кадра, <b>101</b> - <b>4</b> кадра, <b>110</b> -



											<p>25 кадров, 111 - 2 кадра.  <b>D[2:0]</b> определяет адрес конечной страницы так же, как и <b>B[2:0]</b>. Однако значение <b>D[2:0]</b> должно быть больше или равно <b>B[2:0]</b>.  <b>E[5:0]</b> смещение прокрутки по вертикали. Например <b>0x01</b> означает смещение на 1 строку, <b>0x3F</b> смещение на 63 строки.  Примечание: непрерывный скроллинг по вертикали недоступен.</p>
0	2E	0	0	1	0	1	1	1	0	Deactivate scroll	<p>Останавливает прокрутку, которая была сконфигурирована командами <b>0x26</b>, <b>0x27</b>, <b>0x29</b>, <b>0x2A</b>.  Примечание: после отправки команды <b>0x2E</b> данные <b>RAM</b> должны быть перезаписаны.</p>
0	2F	0	0	1	0	1	1	1	1	Activate scroll	<p>Запуск прокрутки, которая была сконфигурирована командами <b>0x26</b>, <b>0x27</b>, <b>0x29</b>, <b>0x2A</b>, следующими</p>

											<p>допустимыми последовательностями команд:</p> <p><b>0x26 0x2F</b></p> <p><b>0x27 0x2F</b></p> <p><b>0x29 0x2F</b></p> <p><b>0x2A 0x2F</b></p> <p>Например, если была выдана последовательность команд <b>0x26 0x2A 0x2F</b>, то будет запущена прокрутка последняя настройка прокрутки для команды <b>0x2A</b>. Другими словами, настройка последней команды установки прокрутки перезапишет настройки предыдущих команд установки прокрутки.</p>
<p><b>0</b></p> <p><b>0</b></p> <p><b>0</b></p>	<p><b>A3</b></p> <p><b>A[5:0]</b></p> <p><b>B[6:0]</b></p>	<p><b>1</b></p> <p><b>*</b></p> <p><b>*</b></p>	<p><b>0</b></p> <p><b>*</b></p> <p><b>B6</b></p>	<p><b>1</b></p> <p><b>A5</b></p> <p><b>B5</b></p>	<p><b>0</b></p> <p><b>A4</b></p> <p><b>B4</b></p>	<p><b>0</b></p> <p><b>A3</b></p> <p><b>B3</b></p>	<p><b>0</b></p> <p><b>A2</b></p> <p><b>B2</b></p>	<p><b>1</b></p> <p><b>A1</b></p> <p><b>B1</b></p>	<p><b>1</b></p> <p><b>A0</b></p> <p><b>B0</b></p>	<p><b>Set Vertical Scroll Area</b></p>	<p><b>A[5:0]</b> установит количество строк в верхней фиксированной области. Количество строк в верхней фиксированной области относится к верхней области <b>GDDRAM</b> (например к строке <b>0</b>). Значение по умолчанию</p>

после сброса 0. В[6:0] установит количество строк в области прокрутки. Это количество строк, которые будут использоваться в вертикальной прокрутке. Область прокрутки начинается в первой строке ниже фиксированной области. Значение по умолчанию после сброса 64.

Примечания:

1.  $A[5:0] + B[6:0] \leq \text{MUX ratio}$ .
2.  $B[6:0] \leq \text{MUX ratio}$ .
3. Смещение прокрутки по вертикали E[5:0] в командах 0x29h/0x2A.
4. Установка начальной строки экрана (X5..X0 от 0x40~0x7F < B[6:0]).
5. Последняя строка области прокрутки смещается в первую строку области прокрутки.
6. Для экрана с MUX=64: A[5:0]=0, B[6:0]=64 задает прокрутку всего

											<p>экрана,  <math>A[5:0]=0, B[6:0] &lt; 64</math>  задает прокрутку верхней части,  <math>A[5:0] + B[6:0] &lt; 64</math>  задает прокрутку средней части,  <math>A[5:0] + B[6:0] = 64</math>  задает прокрутку нижней части.</p>
--	--	--	--	--	--	--	--	--	--	--	--

### 3. Команды установки адреса

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Команда	Описание
0	00~0F	0	0	0	0	X3	X2	X1	X0	Set Lower Column Start Address for Page Addressing Mode	Устанавливает младший ниббл регистра начального адреса столбца для режима адресации страниц (Page Addressing Mode) с помощью бит X[3:0]. После сброса начальное значение 0000. Примечание: эта команда используется только для Page Addressing Mode.
0	10~1F	0	0	0	1	X3	X2	X1	X0	Set Higher Column Start Address for Page	Устанавливает старший ниббл регистра начального адреса столбца для режима

										<b>Addressing Mode</b>	адресации страниц ( <b>Page Addressing Mode</b> ) с помощью бит X[3:0]. После сброса начальное значение <b>0000</b> . Примечание: эта команда используется только для <b>Page Addressing Mode</b> .
<b>00</b>	<b>20</b> A[1:0]	<b>0*</b>	<b>0*</b>	<b>1*</b>	<b>0*</b>	<b>0*</b>	<b>0*</b>	<b>0</b> A1	<b>0</b> A0	<b>Set Memory Addressing Mode</b>	A[1:0] = 00, Horizontal Addressing Mode A[1:0] = 01, Vertical Addressing Mode A[1:0] = 10, Page Addressing Mode (значение по умолчанию после сброса) A[1:0] = 11, недопустимый режим
<b>000</b>	<b>21</b> A[6:0] B[6:0]	<b>0*</b> *	<b>0</b> A6 B6	<b>1</b> A5 B5	<b>0</b> A4 B4	<b>0</b> A3 B3	<b>0</b> A2 B2	<b>0</b> A1 B1	<b>1</b> A0 B0	<b>Set Column Address</b>	Настройка адреса начального и конечного столбца. A[6:0] адрес начального столбца в диапазоне 0..127 (значение по умолчанию после сброса 0). B[6:0] адрес конечного столбца в диапазоне

											<b>0..127</b> (значение по умолчанию после сброса <b>127</b> ). Примечание: эта команда применяется только для режимов горизонтальной или вертикальной адресации.
<b>0</b> <b>0</b> <b>0</b>	<b>22</b> <b>A[2:0]</b> <b>B[2:0]</b>	<b>0</b> * *	<b>0</b> * *	<b>1</b> * *	<b>0</b> * *	<b>0</b> * *	<b>0</b> <b>A2</b> <b>B2</b>	<b>1</b> <b>A1</b> <b>B1</b>	<b>0</b> <b>A0</b> <b>B0</b>	<b>Set Page Address</b>	Настройка адреса начального и конечного столбца. <b>A[2:0]</b> адрес начального столбца в диапазоне <b>0..7</b> (значение по умолчанию после сброса <b>0</b> ). <b>B[2:0]</b> адрес конечного столбца в диапазоне <b>0..7</b> (значение по умолчанию после сброса <b>7</b> ). Примечание: эта команда применяется только для режимов горизонтальной или вертикальной адресации.
<b>0</b>	<b>B0~B7</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>X2</b>	<b>X1</b>	<b>X0</b>	<b>Set Page Start Address for</b>	Установка битами <b>X[2:0]</b> адреса начальной страницы <b>GDDRAM</b>

											<b>Page Addressing Mode</b>	<b>(PAGE0~PAGE7) для Page Addressing Mode.</b> Примечание: эта команда применяется только для режима страничной адресации (Page Addressing Mode).
--	--	--	--	--	--	--	--	--	--	--	-----------------------------	--

#### 4. Команды аппаратной конфигурации (относятся к разрешающей способности панели и ориентации графики)

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Команда	Описание
0	40~7F	0	1	X5	X4	X3	X2	X1	X0	Set Display Start Line	Устанавливает битами X[5:0] регистр начальной строки в диапазоне 0..63. После сброса начальное значение 0.
0	A0/A1	1	0	1	0	0	0	0	X0	Set Segment Re-map	0xA0, X0=0: адрес столбца 0 отображается на SEG0 (состояние по умолчанию после сброса). 0xA1, X0=1: адрес столбца 127 отображается на SEG0.
0 0	A8 A[5:0]	1 *	0 *	1 A5	0 A4	1 A3	0 A2	0 A1	0 A0	Set Multiplex Ratio	Установка коэффициента мультиплексирования (MUX ratio) для N+1

											<b>MUX.</b> <b>N=A[5:0]:</b> от <b>16MUX</b> до <b>64MUX</b> , значение по умолчанию после сброса равно <b>63</b> , т. е. <b>64MUX</b> . Значения <b>0..14</b> недопустимы.
<b>0</b>	<b>C0/C8</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>X3</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>Set COM Output Scan Direction</b>	<b>0xC0, X3=0:</b> нормальный режим (состояние по умолчанию после сброса), сканирование от <b>COM0</b> до <b>COM[N-1]</b> . <b>0xC8, X3=1: remapped mode</b> , сканирование от <b>COM[N-1]</b> до <b>COM0</b> . Здесь <b>N</b> это <b>MUX ratio</b> (коэффициент мультиплексирования).
<b>0</b> <b>0</b>	<b>D3</b> <b>A[5:0]</b>	<b>1</b> <b>*</b>	<b>1</b> <b>*</b>	<b>0</b> <b>A5</b>	<b>1</b> <b>A4</b>	<b>0</b> <b>A3</b>	<b>0</b> <b>A2</b>	<b>1</b> <b>A1</b>	<b>1</b> <b>A0</b>	<b>Set Display Offset</b>	Установка вертикального смещения <b>COM</b> от <b>0~63</b> . Значение после сброса по умолчанию <b>0</b> .
<b>0</b> <b>0</b>	<b>DA</b> <b>A[5:4]</b>	<b>1</b> <b>0</b>	<b>1</b> <b>0</b>	<b>0</b> <b>A5</b>	<b>1</b> <b>A4</b>	<b>1</b> <b>0</b>	<b>0</b> <b>0</b>	<b>1</b> <b>1</b>	<b>0</b> <b>0</b>	<b>Set COM Pins Hardware Configuration</b>	<b>A4=0</b> , конфигурация последовательных выводов <b>COM</b> . <b>A4=1</b> (значение по умолчанию после сброса)



											сброса) альтернативная конфигурация выводов COM. A5=0 (значение по умолчанию после сброса), Disable COM Left/Right remap. A5]=1, Enable COM Left/Right remap.
--	--	--	--	--	--	--	--	--	--	--	---

### 5. Команды настройки интервалов времени и схемы драйверов

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Команда	Описание
0 0	D5 A[7:0]	1 A7	1 A6	0 A5	1 A4	0 A3	1 A2	0 A1	1 A0	Set Display Clock Divide Ratio/Oscillator Frequency	A[3:0] определение коэффициента деления (D) для тактов дисплея (DCLK). Коэффициент деления = A[3:0]+1. Значение по умолчанию после сброса 0, что соответствует коэффициенту деления 1. A[7:4] установка частоты генератора F <sub>OSC</sub> . Чем больше значение A[7:4], тем выше частота

											генератора. Значение по умолчанию после сброса <b>8</b> , допустимый диапазон <b>0..15</b> .
<b>00</b>	<b>D9</b> <b>A[7:0]</b>	<b>1</b> <b>A7</b>	<b>1</b> <b>A6</b>	<b>0</b> <b>A5</b>	<b>1</b> <b>A4</b>	<b>1</b> <b>A3</b>	<b>0</b> <b>A2</b>	<b>0</b> <b>A1</b>	<b>1</b> <b>A0</b>	<b>Set Pre-charge Period</b>	<b>A[3:0]</b> установка периода <b>Phase 1</b> до <b>15</b> импульсов <b>DCLK</b> . Значение <b>0</b> недопустимо, значение по умолчанию после сброса <b>2</b> . <b>A[7:4]</b> установка периода <b>Phase 2</b> до <b>15</b> импульсов <b>DCLK</b> . Значение <b>0</b> недопустимо, значение по умолчанию после сброса <b>2</b> .
<b>00</b>	<b>DB</b> <b>A[6:4]</b>	<b>1</b> <b>0</b>	<b>1</b> <b>A6</b>	<b>0</b> <b>A5</b>	<b>1</b> <b>A4</b>	<b>1</b> <b>0</b>	<b>0</b> <b>0</b>	<b>1</b> <b>0</b>	<b>1</b> <b>0</b>	<b>Set V<sub>COMH</sub> Deselect Level</b>	<b>0x00</b> ~ <b>0.65 · V<sub>CC</sub></b> <b>0x20</b> ~ <b>0.77 · V<sub>CC</sub></b> <b>0x30</b> ~ <b>0.83 · V<sub>CC</sub></b> Значение по умолчанию после сброса <b>0x20</b> .
<b>0</b>	<b>E3</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>NOP</b>	Пустая команда (нет операции).

Примечание: \* означает "Don't care", т. е. "не имеет значения".

Таблица 9-2. Команда чтения (Read Command).

Биты данных	Команда	Описание
D[7:0]	Status Register Read	D7: зарезервированный бит D6: 1 когда дисплей выключен, 0 когда включен D[5:0] зарезервированные биты

Примечание: данные команд, не присутствующие в таблицах, запрещены для ввода команды в чип **SSD1306**, иначе могут быть получены неожиданные результаты.

**Чтение / запись данных.** Чтобы прочитать данные из **GDDRAM**, установите лог. 1 для сигнала **R/W# (WR#)** и сигнала **D/C#** для режима параллельного интерфейса стиля **6800**, и выберите лог. 0 для вывода **E (RD#)** и лог. 1 для вывода **D/C#** для параллельного интерфейса стиля **8080**. В последовательных режимах работы интерфейса чтение данных не предоставляется.

В нормальном режиме чтения данных адрес столбца **GDDRAM** автоматически увеличивается при каждом чтении данных.

Также перед первым чтением требуется пустое чтение (**dummy read**).

Чтобы записать данные в **GDDRAM**, установите лог.0 на выводе **R/W# (WR#)** и лог. 1 на выводе **D/C#** для режима параллельного интерфейса стиля **6800** и стиля **8080**. Последовательные режимы интерфейса всегда работают в режиме записи. Адрес столбца **GDDRAM** автоматически увеличивается после каждой записи данных.

Таблица 9-3. Автоматический инкремент адреса.

D/C#	R/W#(WR#)	Комментарий	Инкремент адреса
------	-----------	-------------	------------------

0	0	Write Command (запись команды)	Отсутствует
0	1	Read Status (чтение состояния)	Отсутствует
1	0	Write Data (запись данных)	Есть
1	1	Read Data (чтение данных)	Есть

## [Описание команд]

### Основные команды

Эта команда задает младший ниббл 8-битного адреса столбца для буфера RAM экрана в режиме адресации страницы (**Page Addressing Mode**). Адрес столбца будет инкрементироваться при каждом доступе к данным. Обратитесь к таблице 9-1 и описанию команды **Set Memory Addressing Mode (0x20)**.

Эта команда задает старший ниббл 8-битного адреса столбца для буфера RAM экрана в режиме адресации страницы (**Page Addressing Mode**). Адрес столбца будет инкрементироваться с каждым доступом к данным. Пожалуйста обратитесь к таблице 9-1 и описанию команды **Set Memory Addressing Mode (0x20)**.

Имеется 3 разных режима адресации памяти **SSD1306: page addressing mode, horizontal addressing mode** и **vertical addressing mode**. Эта команда устанавливает способ адресации памяти в один из этих трех режимов. Здесь **COL** означает столбец данных **RAM** графического дисплея.

### Page addressing mode (A[1:0] = 10)

В этом режиме адресации после того, как был осуществлен доступ к **RAM** (на чтение или запись), указатель адреса столбца автоматически увеличивается на 1. Если адрес столбца достигает конечного столбца, то указатель адреса столбца сбрасывается к адресу начального столбца, и адрес указателя на страницу не меняется. Пользователи должны установить адрес новой страницы и адрес нового столбца, чтобы получить доступ к содержимому другой

страницы **RAM**. Последовательность перемещения указателя адреса **PAGE** и **COL** для **page addressing mode** показана на рис. 10-1.

	COL0	COL 1	.....	COL 126	COL 127
PAGE0	→	→	→	→	→
PAGE1	→	→	→	→	→
⋮	⋮	⋮	⋮	⋮	⋮
PAGE6	→	→	→	→	→
PAGE7	→	→	→	→	→

Рис. 10-1. Перемещение указателя адреса в **Page Addressing Mode**.

Для нормального чтения или записи данных **RAM** экрана в **page addressing mode** требуются следующие шаги для определения начального адреса доступа к **RAM**:

- Установите начальный адрес страницы целевой области дисплея командами от **0xB0** до **0xB7**.
- Установите нижний начальный адрес указателя столбца командой **0x00 .. 0x0F**.
- Установите верхний начальный адрес указателя столбца командой **0x10 .. 0x1F**.

Например, если адрес страницы установлен на **0xB2**, нижний адрес столбца **0x03** и верхний адрес столбца **0x10**, то это означает начальный столбец **SEG3** страницы **PAGE2**. Указатель доступа к **RAM** находится в месте, показанном на рис. 10-2. Входной байт данных будет записан в позицию **RAM** столбца 3.

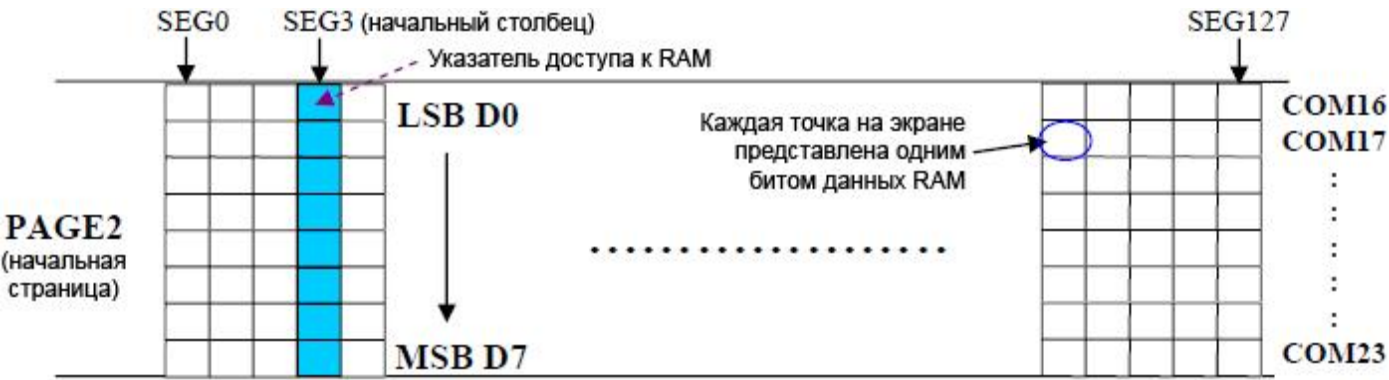


Рис. 10-2. Пример установки указателя доступа GDDRAM в Page Addressing Mode (без переназначений строк и столбцов).

Horizontal addressing mode (A[1:0]=00)

В режиме горизонтальной адресации после каждого доступа к данным **RAM** экрана адрес указателя столбца автоматически увеличивается на 1. Если адрес столбца достиг последнего столбца, то адрес сбрасывается к начальному столбцу, и указатель на адрес страницы увеличивается на 1. Последовательность изменения адресов страницы и столбца в этом режиме показана на рис. 10-3. Когда оба адреса, и адрес столбца, и адрес страницы, достигают конца, то эти указатели сбрасываются в адрес начального столбца и адрес начальной страницы (пунктирная линия на рис. 10-3).

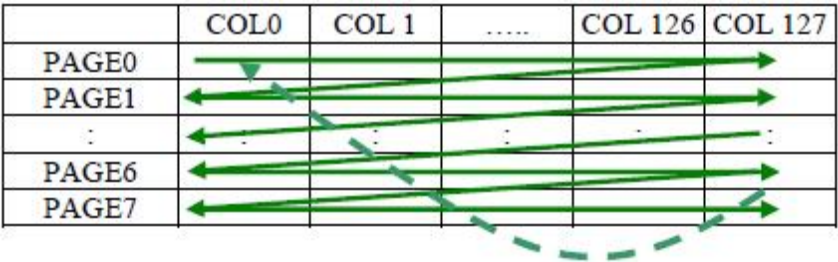


Рис. 10-3. Перемещение указателя адреса в Horizontal Addressing Mode.

Vertical addressing mode: (A[1:0]=01)

В режиме вертикальной адресации после доступа к **RAM** дисплея указатель адреса страницы автоматически увеличивается на 1. Если указатель адреса страницы достиг последней страницы, то адрес страницы сбрасывается в адрес начальной страницы, и адрес начального столбца увеличивается на 1. Последовательность изменения адресов страницы и столбца в этом режиме показана на рис. 10-4. Когда оба адреса, и адрес столбца, и адрес страницы, достигают конца, то эти указатели сбрасываются в адрес начального столбца и адрес начальной страницы (пунктирная линия на рис. 10-4).

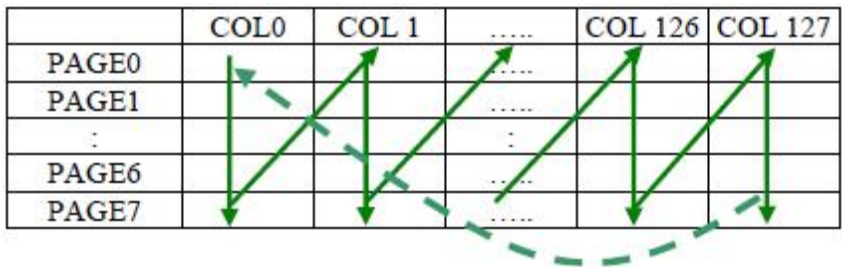


Рис. 10-4. Перемещение указателя адреса в **Vertical Addressing Mode**.

Чтобы осуществить обычный доступ к данным **RAM** экрана для чтения или записи в режимах адресации **Horizontal / Vertical**, требуются следующие шаги для определения места указателя доступа к **RAM**:

- Установите начальный и конечный адреса столбца места на экране командой **0x21**.
- Установите адрес начальной страницы и адрес конечной страницы командой **0x22**.

Пример показан на рис. 10-5 в описании команды **Set Page Address (0x22)**.

Эта команда из трех байт задает начальный и конечный адреса данных **RAM** экрана. Эта команда также устанавливает указатель адреса столбца на начальный адрес столбца. Этот указатель используется для определения текущего адреса столбца для чтения/записи данных **RAM** экрана. Если командой **0x20** разрешен **horizontal address increment mode**, то после завершения чтения/записи одного столбца данных указатель автоматически инкрементируется на следующий адрес столбца. Всякий раз, когда адрес столбца завершает

доступ к адресу последнего столбца, он сбрасывается обратно к начальному адресу столбца, и адрес строки увеличивается для доступа к следующей строке.

Эта команда из трех байт задает начальный адрес страницы и конечный адрес страницы данных **RAM** экрана. Эта команда также устанавливает указатель адреса страницы на начальный адрес страницы. Этот указатель используется для определения текущей позиции чтения/записи данных **RAM** экрана. Если командой **0x20** разрешен **vertical address increment mode**, то после завершения чтения/записи одной страницы данных адрес автоматически инкрементируется на адрес следующей страницы. Всякий раз, когда указатель адреса страницы завершает доступ к последнему адресу страницы, то он сбрасывается обратно к начальному адресу страницы.

Рис. 10-5 показывает на примере, как меняются адреса столбца и страницы: начальный адрес столбца устанавливается на **2**, конечный адрес столбца устанавливается на **125**, начальный адрес страницы устанавливается на **1** и конечный адрес страницы устанавливается на **6**. Командой **0x20** разрешается **horizontal address increment mode**. В этом случае допустимый диапазон доступа к столбцам устанавливается между **2** и **125** столбцами, и допустимый диапазон доступа к страницам устанавливается между **1** и **6**. Дополнительно указатель адреса столбца устанавливается на **2**, и указатель адреса страницы устанавливается на **1**. После завершения чтения/записи одной точки данных адрес столбца автоматически увеличивается на **1** для доступа к следующей ячейке **RAM** на следующей операции (сплошная линия на рис. 10-5). Всякий раз, когда указатель адреса столбца достигает конечного столбца **125**, он сбрасывается обратно к столбцу **2**, и адрес страницы автоматически увеличивается на **1** (сплошная линия на рис. 10-5). Когда осуществлен доступ на последней странице **6** и последнему столбцу **125**, адрес страницы сбрасывается обратно к **1**, и адрес столбца сбрасывается обратно к **2** (пунктирная линия на рис. 10-5).



	Col 0	Col 1	Col 2	.....	.....	Col 125	Col 126	Col 127
PAGE0								
PAGE1								
:								
PAGE6								
PAGE7								




Рис. 10-5. Пример перемещения указателя адреса столбца и строки.

Эта команда устанавливает регистр начальной строки (**Display Start Line**), чтобы определить начальный адрес **RAM** экрана, путем выбора значения от **0** до **63**. Когда значение равно **0**, строка **0 RAM** отображается на **COM0**. Когда значение равно **1**, строка **1 RAM** отображается на **COM0**, и так далее.

Для дополнительной информации обратитесь к таблице **10-1** в описании команды **Set Display Offset (0xD3)**.

Эта команда устанавливает настройку контрастности экрана. Имеется **256** шагов настройки контрастности от **0x00** до **0xFF**. Выходной ток сегмента увеличивается при увеличении значения контрастности.

Эта команда меняет привязку данных адреса столбца и драйвера сегмента. Это добавляет гибкости в разработке модуля **OLED**. См. таблицу команд **9-1**.

Эта команда влияет только на последующий ввод данных. Данные, уже сохраненные в **GDDRAM**, не меняются.

Команда **0xA4** разрешает отображение на экране в соответствии с содержимым **GDDRAM**. Если выдана команда **0xA5** и затем выдана команда **0xA4**, то продолжится отображение содержимого **GDDRAM**. Другими словами, команда **0xA4** возобновляет отображение экрана всего экрана от стадии включения. Команда **0xA5** принудительно включает отображение, независимо от содержимого данных **RAM** экрана.

Эта команда устанавливает отображение данных в нормальном или инверсном виде. При нормальном отображении лог. 1 данных **RAM** соответствует включенному (засвеченному) пикселю экрана. В инверсном режиме все наоборот, значение бита в лог. 0 соответствует включенному пикселю экрана.

Эта команда переключает режим мультиплексирования по умолчанию **63** на любой коэффициент мультиплексирования в диапазоне от **16** до **63**. Выходные выводы **COM0 .. COM63** будут переключать соответствующий сигнал **COM**.

Эти однобайтные команды используются для включения и выключения панели экрана **OLED**. Когда экран включен, то выбранные командой **Set Master Configuration** схемы будут включены. Когда экран выключен, то эти схемы будут выключены, и выходы **SEG** и **COM** будут соответственно в состоянии  $V_{SS}$  и высокого сопротивления. Эти команды устанавливают экран в одно из двух состояний:

**0xAE: Display OFF** (экран выключен)

**0xAF: Display ON** (экран включен)



Рис. 10-6. Переход между разными режимами.

Эта команда устанавливает начальный адрес страницы **GDDRAM** в значение от 0 до 7 в режиме **Page Addressing Mode**. См. таблицу 9-1 и описание команды **Set Memory Addressing Mode (0x20)**.

Эта команда устанавливает направление сканирования выхода **COM**, добавляя гибкость для дизайна модуля **OLED**. Дополнительно на экране отобразятся изменения при выдаче этой команды. Например, если эта команда была отправлена во время нормального отображения графики на экране, то изображение немедленно будет перевернуто по вертикали. Подробности см. в таблице 10-3 из описания команды **Set COM Pins Hardware Configuration**

(0xDA).

Это двухбайтная команда. Второй байт задает привязку начальной строки дисплея к одному из выходов драйвера **COM0 .. COM63** (предполагается, что если **COM0** соответствует начальной строке экрана, тогда значение регистра начальной строки экрана равно 0).

Например, чтобы переместить строку **COM16** в направлении **COM0** на 16 строк, то 6-битные данные во втором байте должны быть **b010000**. Чтобы переместить картинку в противоположном направлении на 16 строк, эти 6-битные данные должны быть равны **64 – 16**, что соответствует второму байту **b100000**. Следующие две таблицы (таблица 10-1, таблица 10-2) показывают пример установки команд **0xC0/0xC8** и **0xD3**.

Таблица 10-1. Пример Set Display Offset и Display Start Line без переназначения (no Remap).

[]

Таблица 10-2. Пример Set Display Offset и Display Start Line с переназначением (Remap).

[]

Примечание: таблицы 10-1 и 10-2 здесь не приведены из-за своего большого размера, см. оригинальный документ [1].

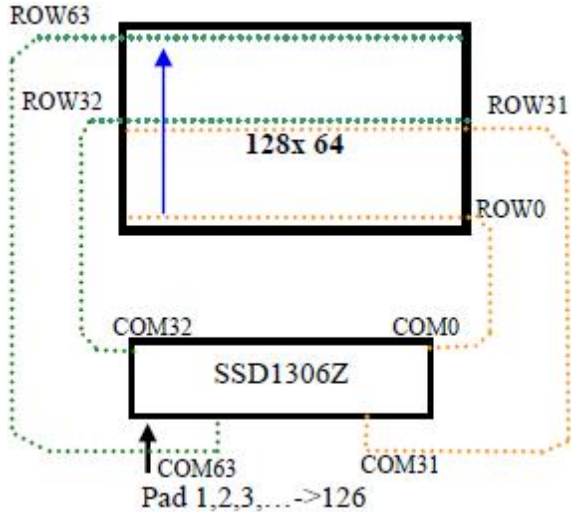
Эта команда состоит из двух функций:

- **Display Clock Divide Ratio** (установка **D**, биты **A[3:0]**). Это установит коэффициент деления для генерации **DCLK (Display Clock)** из **CLK**. Коэффициент деления может быть в диапазоне от 1 до 16, значение сброса = 1. Подробнее про соотношение между **DCLK** и **CLK** см. в секции "Схема генератора и генератор интервалов времени".
- **Oscillator Frequency** (биты **A[7:4]**). Программируют частоту **Fosc**, которая является источником для генерации **CLK**, если на выводе **CLS** лог. 1. Это 4-битное значение дает 16 разных значений частот, что показано ниже. Значение по умолчанию **b1000**.

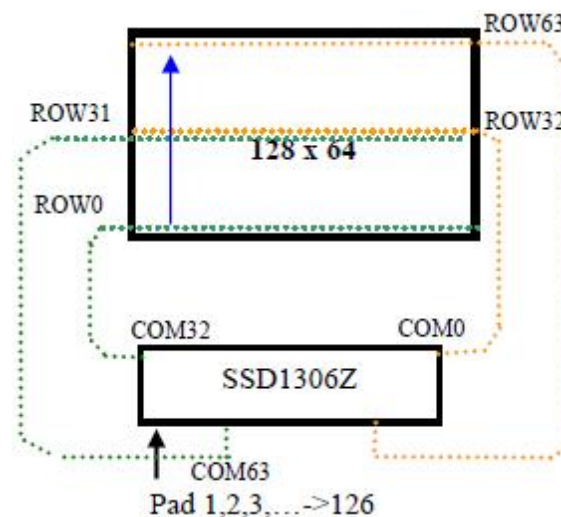
Эта команда используется для установки длительности периода **pre-charge**.  
Устанавливаемый интервал измеряется в количестве **DCLK**, где **RESET** эквивалентен 2 периодам **DCLK**.

Эта команда устанавливает конфигурацию сигналов **COM**, чтобы она соответствовала аппаратной разводке сигналов панели **OLED**. Таблица ниже показывает конфигурацию выводов **COM** при разных условиях (для коэффициента **MUX = 64**):

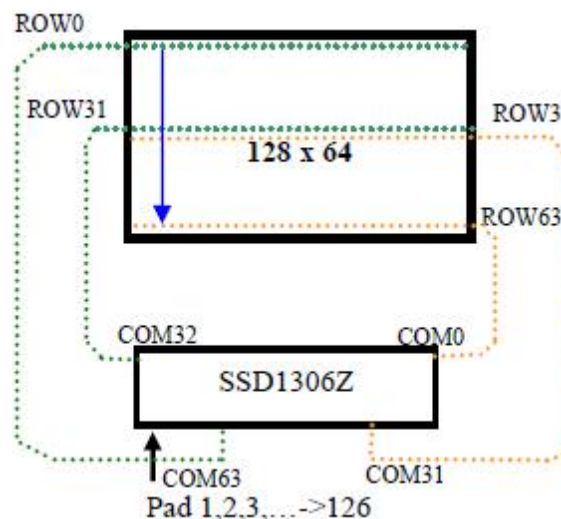
Таблица 10-3. Аппаратная конфигурация выводов **COM**.

Условия	Конфигурации выводов COM
1. Последовательная конфигурация выводов COM ( <b>Sequential COM</b> ), команда <b>0xDA</b> со значением <b>A4=0</b> . Направление сканирования выходов COM от <b>COM0</b> до <b>COM63</b> ( <b>0xC0</b> ). <b>Disable COM Left/Right remap</b> , команда <b>0xDA</b> со значением <b>A5=0</b> .	

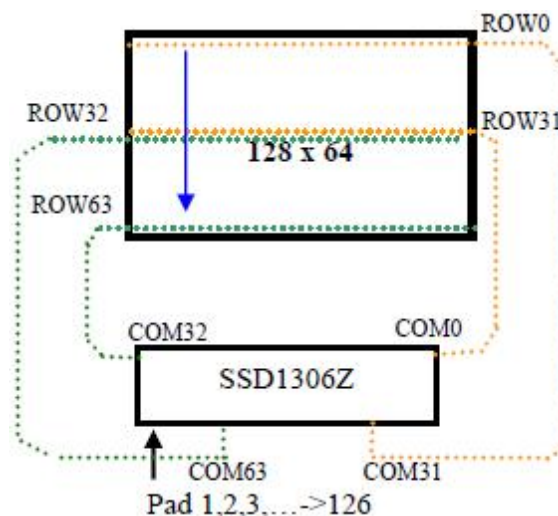
2. То же самое, что и 1, с отличием **Enable COM Left/Right remap**, команда **0xDA** со значением **A5=1**.



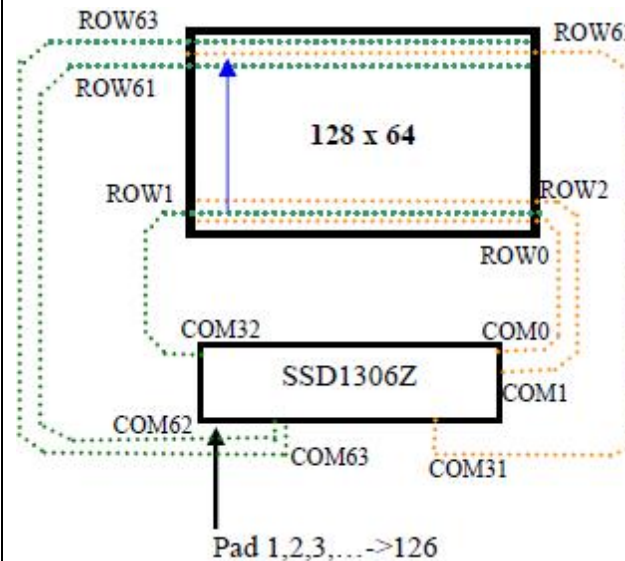
3. То же самое, что и 1, с отличием в направлении сканирования от **COM63** до **COM0** (0xC8).



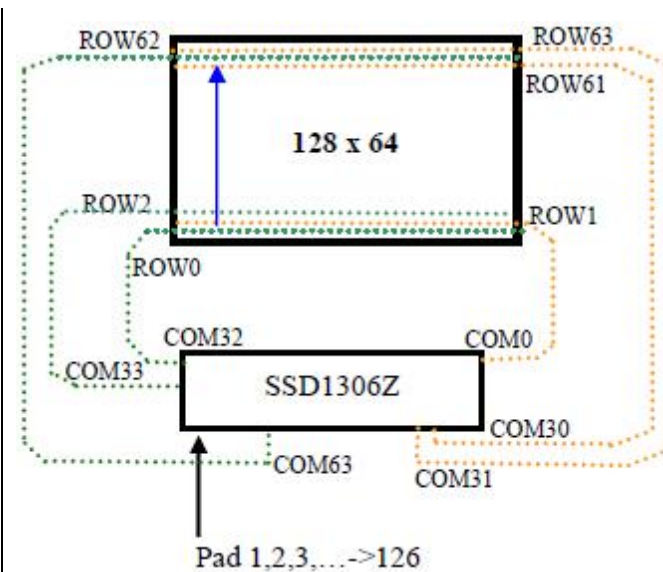
4. Последовательная конфигурация выводов COM (Sequential COM), команда 0xDA со значением A4=0. Направление сканирования выходов COM от COM63 до COM0 (0xC8). Enable COM Left/Right remap, команда 0xDA со значением A5=1.



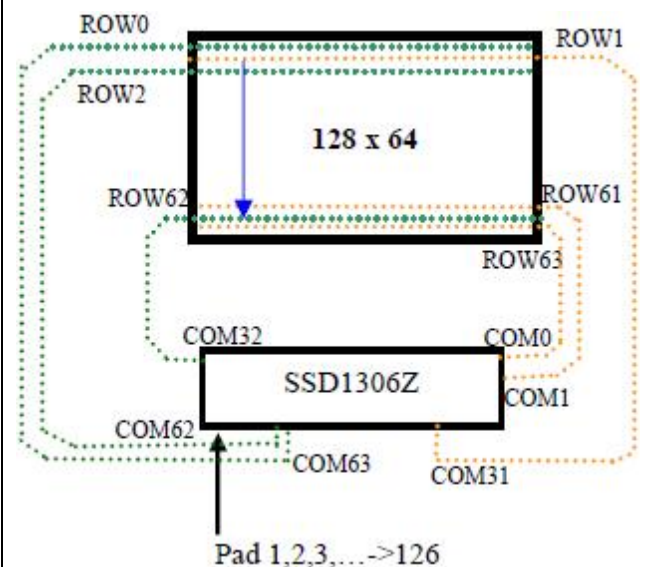
5. Альтернативная конфигурация выводов COM (Alternative COM), команда 0xDA со значением A4=1. Направление сканирования выходов COM от COM0 до COM63 (0xC0). Disable COM Left/Right remap, команда 0xDA со значением A5=0.



6. То же самое, что и 5, с отличием **Enable COM Left/Right remap**, команда **0xDA** со значением **A5=1**.

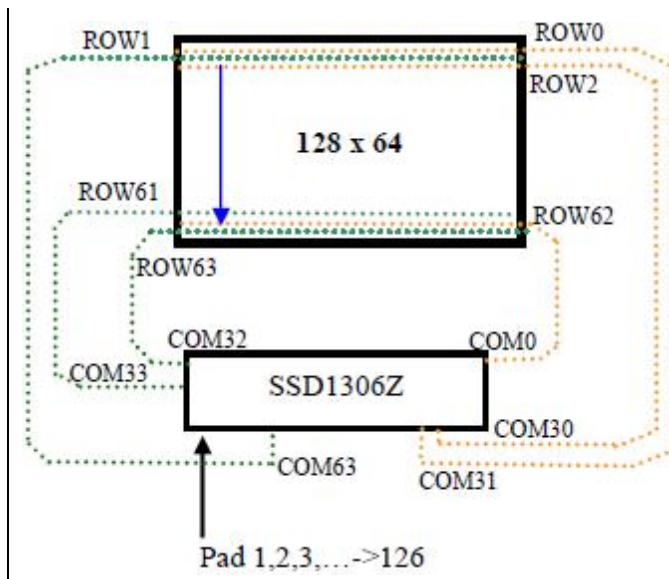


7. То же самое, что и 5, с отличием в направлении сканирования от **COM63** до **COM0** (**0xC8**).





**8. Альтернативная конфигурация выводов COM (Alternative COM), команда 0xDA со значением A4=1.** Направление сканирования выходов COM от COM63 до COM0 (0xC8). Enable COM Left/Right remap, команда 0xDA со значением A5=1.



Примечание: строки на картинках, помеченные желтыми пунктирными линиями, расширяют размеры экрана.

Эта команда настраивает выход регулятора напряжения VCOMH.

**No Operation Command**, команда пустой операции.

**Чтение регистра состояния (Status register Read).** Эта команда выдается путем установки D/C# в лог. 0 на время чтения данных (см. рис. 13-1 и рис. 13-2, где показаны сигналы параллельного интерфейса). Это позволяет MCU мониторить внутреннее состояние чипа SSD1306. В режиме последовательных интерфейсов чтение состояния не предоставляется.

### Команды управления графикой (Graphic Acceleration Command)

Эта команда состоит из последующих байт для установки параметров горизонтального скроллинга (прокрутки изображения), и определяет начальную страницу скроллинга, конечную страницу скроллинга и скорость скроллинга.



Перед выдачей этой команды горизонтальный скроллинг должен быть деактивирован (командой **0x2E**). Иначе содержимое **RAM** может быть повреждено.

Горизонтальный скроллинг **SSD1306** разработан для прокрутки **128** столбцов. Рисунки **10-7**, **10-8**, **10-9** даташита [1] показывают примеры использования горизонтального скроллинга.

Эта команда состоит из **6** последующих байт, предназначенных для установки параметров вертикального скроллинга (прокрутки изображения). Параметры определяют начальную, конечную страницу скроллинга, скорость скроллинга и смещение вертикального скроллинга.

Байты **B[2:0]**, **C[2:0]** и **D[2:0]** команд **0x29/0x2A** предназначены для установки непрерывного горизонтального скроллинга. Байт **E[5:0]** предназначен для установки смещения непрерывного вертикального скроллинга. Все эти байты вместе используются для установки непрерывного диагонального (прокрутка по горизонтали + прокрутка по вертикали) скроллинга. Если байт смещения вертикального скроллинга **E[5:0]** установлен в **0**, то выполняется только горизонтальный скроллинг (наподобие действия команд **0x26/0x27**).

Перед выдачей этой команды скроллинг должен быть деактивирован (**0x2E**). Иначе содержимое **RAM** может быть повреждено. Рис. **10-10** даташита [1] показывает пример использования вертикальной и горизонтальной прокрутки.

Эта команда останавливает прокрутку. После того, как отправка команды **0x2E** деактивирует действие прокрутки, данные **RAM** должны быть перезаписаны.

Эта команда запускает движение скроллинга, и она должна выдаваться только после того, как настроены параметры прокрутки командами **0x26/0x27/0x29/0x2A**. Установка последней команды настройки скроллинга перезаписывает установки предыдущих команд настройки скроллинга.

После активации скроллинга запрещены следующие действия:

1. Доступ к **RAM** (на чтение и запись).
2. Изменение параметров горизонтальной прокрутки.

Эта команда состоит из 3 последующих байт, предназначенных для настройки области вертикальной прокрутки. Для функции непрерывной прокрутки по вертикали (команда **0x29/0x2A**), количество строк прокрутки по вертикали может быть меньше или равно коэффициенту MUX.

### [Предельные параметры]

Таблица 11-1. Предельно допустимые параметры (напряжения измерены относительно  $V_{SS}$ ).

Символ	Параметр	Значение	Ед.
$V_{DD}$	Питание ядра логики контроллера	-0.3 .. +4	V
$V_{CC}$	Питание сегментов экрана	0 .. 16	V
$V_{SEG}$	Выходное напряжение SEG	0 .. $V_{CC}$	V
$V_{COM}$	Выходное напряжение COM	0 .. 0.9 $V_{CC}$	V
$V_{in}$	Входное напряжение	$V_{SS}-0.3$ .. $V_{DD}+0.3$	V
$T_A$	Рабочая температура окружающего воздуха	-40 .. +85	°C
$T_{stg}$	Температура хранения	-65 .. +150	°C

Предельные параметры это значения, эксплуатация изделия возле которых может его повредить. Функционирование должно быть ограничено пределами, указанными в таблицах электрических характеристик или в секции описания выводов.

Это устройство может быть чувствительно к действию света. Следует избегать облучение устройства источниками света во время его нормальной работы. Это устройство не защищено от радиации.

### [Характеристики постоянного тока]

Даны параметры для следующих условий (если не указано что-то другое): напряжения измерены относительно  $V_{SS}$ ,  $V_{DD} = 1.65V \dots 3.3V$ ,  $T_A = 25^\circ C$ .

Таблица 12-1. DC-характеристики (постоянный ток).

[]

Примечание: таблицу 12-1 см. в даташите [1].

### [Характеристики переменного тока]

Даны параметры для следующих условий: напряжения измерены относительно  $V_{SS}$ ,  $V_{DD} = 1.65V \dots 3.3V$ ,  $T_A = 25^\circ C$ .

Таблица 13-1. AC-характеристики (переменный ток).

Символ	Параметр	MIN	typ	MAX	Ед.
$F_{OSC}^{(1)}$	Частота генератора для формирования интервалов времени управления экраном. Приведены значения для $V_{DD} = 2.8V$ .	333	370	407	кГц
$F_{FRM}$	Частота кадров для режима <b>64MUX</b> . Режим отображения <b>128x64</b> точки, экран включен, разрешен внутренний генератор.	-	$F_{OSC} / (D \cdot K \cdot 64)^{(2)}$	-	Гц
<b>RES#</b>	Длительность лог. 0 импульса сброса	3	-	-	мкс

Примечания:

(1)  $F_{OSC}$  означает значение частоты внутреннего генератора, и это значение было измерено, когда командой **0xD5** (биты **A[7:4]**) установлено значение по умолчанию.

(2) **D**: коэффициент деления (значение по умолчанию = 1). **K**: количество тактов дисплея

(значение по умолчанию = 54).

Обратитесь к таблице 9-1 (описание команды **Set Display Clock Divide Ratio / Oscillator Frequency, 0xD5**) для получения подробного описания.

Значения интервалов времени и временные диаграммы для параллельного интерфейса стиля **6800** см. в таблице 13-2 и на рис. 13-1 даташита [1]. Значения интервалов времени и временные диаграммы для параллельного интерфейса стиля **8080** см. в таблице 13-3 и на рис. 13-2 даташита [1].

Значения интервалов времени и временные диаграммы для 4-проводного **SPI** см. в таблице 13-4 и на рис. 13-3. Значения интервалов времени и временные диаграммы для 3-проводного **SPI** см. в таблице 13-5 и на рис. 13-4.

Таблица 13-6. Характеристики интервалов времени интерфейса **I2C**.

Символ	Параметр	MIN	typ	MAX	Ед.
$t_{\text{cycle}}$	Время периода тактов данных.	2.5	-	-	мкс
$t_{\text{HSTART}}$	Время удержания для <b>START condition</b> .	0.6	-	-	мкс
$t_{\text{HD}}$	Время удержания данных для вывода <b>SDA<sub>OUT</sub></b>	0	-	-	нс
	Время удержания данных для вывода <b>SDA<sub>IN</sub></b>	300	-	-	нс
$t_{\text{SD}}$	Время установки данных	100	-	-	нс
$t_{\text{SSTART}}$	Время установки для <b>START condition</b> (параметр актуален только для повторяющихся сигналов <b>START condition</b> )	0.6	-	-	мкс
$t_{\text{SSTOP}}$	Время установки для <b>STOP condition</b>	0.6	-	-	мкс

$t_R$	Время нарастания лог. уровня сигналов <b>SDA</b> и <b>SCK</b>	-	-	<b>300</b>	нс
$t_F$	Время спада лог. уровня сигналов <b>SDA</b> и <b>SCK</b>	-	-	<b>300</b>	нс
$t_{IDLE}$	Время ожидания перед тем, как может начаться новая передача	<b>1.3</b>	-	-	мкс

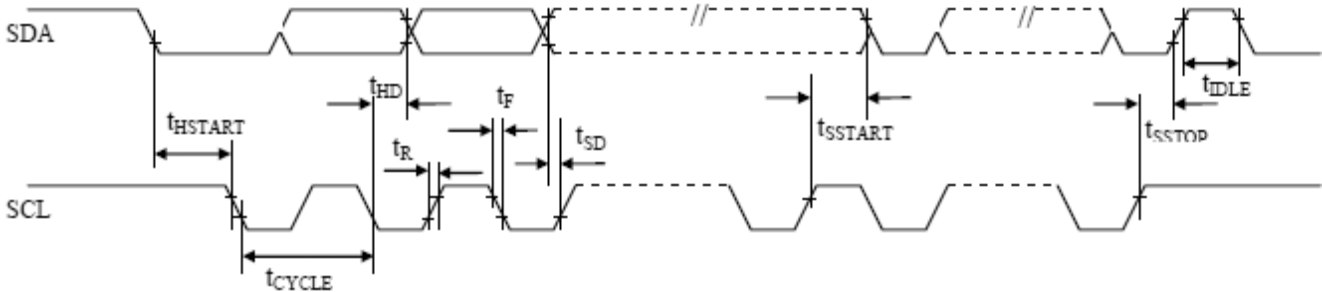


Рис. 13-5. Характеристики интервалов времени интерфейса I2C.

[Программное конфигурирование]

У контроллера **SSD1306** имеется внутренние регистры команд, которые используются для конфигурирования работы узлов драйвера **OLED**. После сброса или включения питания эти регистры должны быть установлены в нужные параметры, чтобы дисплей **OLED** функционировал правильно. К регистрам можно получить доступ через один из вариантов интерфейса **MCU** (типа **6800**, **8080**, **SPI** с выводом **D/C#**, подтянутым к лог. **0**, или через интерфейс **I2C**). Ниже дан пример алгоритма инициализации **SSD1306**. Значения регистров зависят от разных условий и требований приложения.

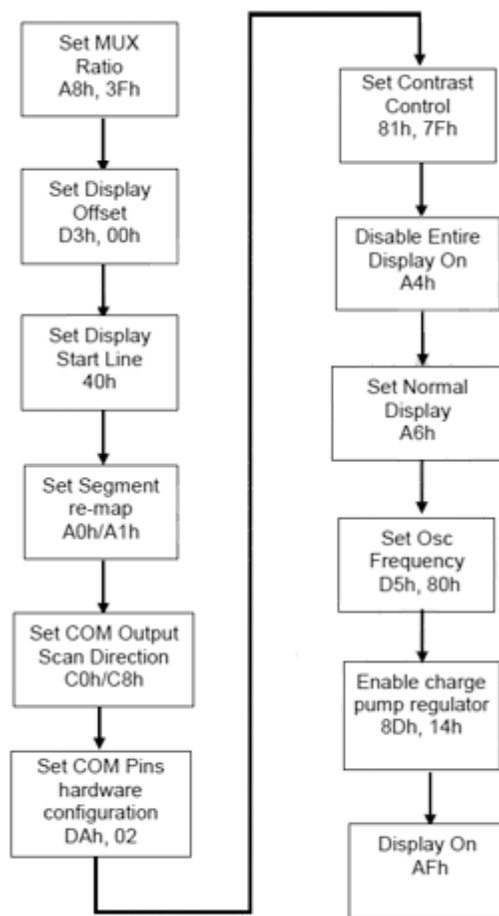


Рис. 2. Последовательность программной инициализации SSD1306.

[Ссылки]

1. 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller.
2. [SSD1322: контроллер/драйвер для дисплеев OLED/PLED](#).