

Графический OLED дисплей UG-2864ASGGG14: первое включение

В.А. Жмудь, А.С. Каменская, К.В. Курбетьев, В.Г. Трубин
ФГБОУ ВПО НГТУ, Новосибирск, Россия

Аннотация: Рассматриваются основные вопросы, возникающие при освоении OLED дисплея UG-2864ASGGG14 с управлением от отладочной платы STM32VLDISCOVERY.

Ключевые слова: Микроконтроллер, STM32, STM32VLDISCOVERY, UG-2864ASGGG14, SH1106, OLED.

ВВЕДЕНИЕ

В цифровой технике зачастую необходимо обеспечить обмен информацией между оборудованием и его пользователем. Это осуществляется с помощью устройств ввода-вывода информации. На данный момент доминирующими устройствами отображения информации являются ЖКИ дисплеи. Однако у данной технологии существует достойный и активно развивающийся конкурент.

Дисплеи на основе органических светодиодов (англ. *organic light-emitting diode*, сокр. *OLED*) обладают следующими свойствами, которые могут позволить им стать лидерами в области плоскопанельных технологий отображения информации:

- отсутствие необходимости в подсветке, т.к. дисплеи *OLED* являются светоизлучающими по своей природе;
- большие углы обзора – изображение видно без потери качества с любого угла;
- высокая яркость и контрастность;
- высокое быстродействие – такие дисплеи обладают практически мгновенным откликом;
- Рабочий температурный диапазон от -40 °С.

ДИСПЛЕЙ UG-2864ASGGG14

Прежде чем приступить к описанию конкретного выше обозначенного дисплея, важно отметить факт многообразия моделей дисплеев. Различия проявляются в следующем: разрешение экрана, цвет пикселей, тип управляющего контроллера, структура памяти, управляющие команды. Следовательно, вся информация, приведенная в данной статье, относится в первую очередь именно к конкретному дисплею *UG-2864ASGGG14* и, если попробовать, например, запустить тестовый пример, приведенный в статье, подключив дисплей с другим управляющим драйвером, то с высокой вероятностью устройство не заработает. Тем не менее, некоторые общие принципы применимы ко всем моделям.

Вследствие того, что дисплеи *OLED* производятся, в основном, в Китае, может оказаться довольно трудным определить модель конкретного дисплея и найти документацию. Рекомендуется в таком случае в первую очередь внимательно осмотреть плату на предмет идентифицирующих обозначений, также, можно попробовать связаться с продавцом.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

Данное устройство представляет собой монохромный *OLED* дисплей с разрешением экрана 128x64 точек, управляемый встроенным драйвером *SH1106*. Обмен информацией производится по интерфейсу *I²C*. Дисплей может работать в нескольких режимах: нормальный, с инверсией изображения, пустой экран, полностью заполненный экран. Основные параметры дисплея приведены ниже, более подробную информацию можно найти в документации к дисплею [1]:

- напряжение питания модуля: 5 В;
- напряжение логической «1»: 2.4 – 3.5 В;
- ток в линии питания: 23 – 33 мА;
- ток в линии передачи данных: 180 – 300 мкА;
- диапазон рабочих температур: от -40 до +70 °С;
- время работы: до 40 тысяч часов.

СТРУКТУРА ДИСПЛЕЯ

Дисплей состоит из матрицы органических микросветодиодов и драйвера управления ими. Имеются четыре вывода, о назначении которых будет рассказано в дальнейшем, и перемычка для выбора адреса устройства, о чем так же подробнее говорится в разделе об управлении. Внешний вид дисплея изображен на *Рис. 1*. Блок-схема драйвера дисплея изображена на *Рис. 2*.

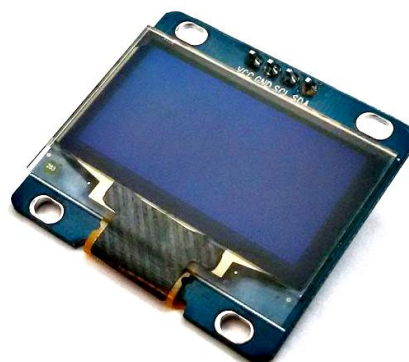


Рис. 1. Внешний вид дисплея UG-2864ASGGG14

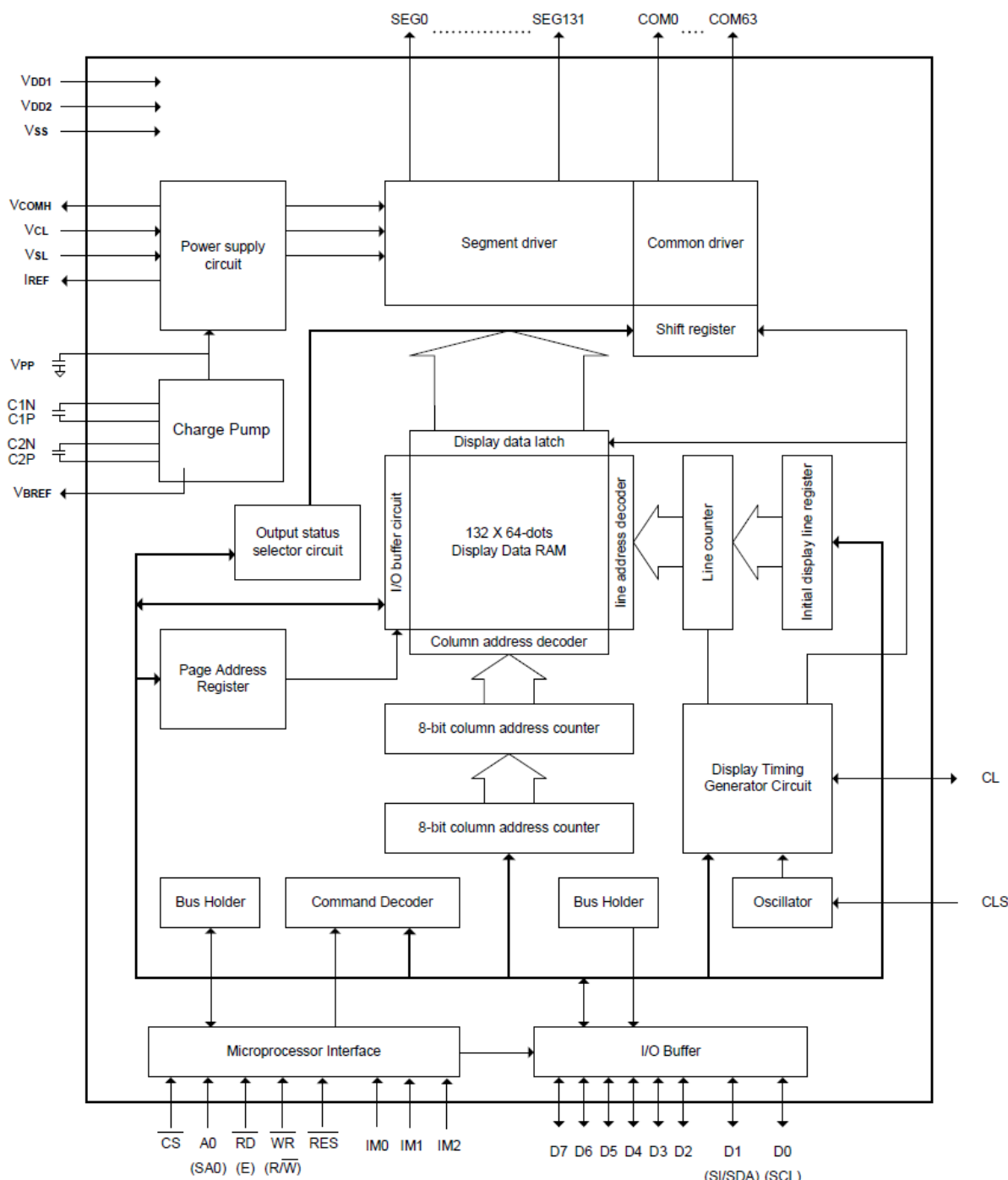


Рис. 2. Блок-схема драйвера SH1106

На данной блок-схеме видно, что информация о состоянии дисплея хранится в оперативной памяти управляющего контроллера (*Display Data RAM*). Кроме того, можно сделать вывод о наличии внутреннего генератора подкачки напряжения (*Charge Pump*), который избавляет от необходимости во внешнем источнике питания. В нашем случае можно будет воспользоваться питанием +5 В от платы *STM32VLDISCOVERY*.

СТРУКТУРА ОЗУ И МАТРИЦЫ ДИСПЛЕЯ

Матрица пикселей дисплея имеет размер

128x64. Таким образом, имеется 64 строки, каждая из которых состоит из 128 пикселей. Или, другими словами, имеется 128 колонок, в каждой из которых 64 пикселя.

ОЗУ контроллера дисплея повторяет организацию матрицы пикселей. Если в какой-то бит ОЗУ записана "1", то соответствующий этому биту пиксель на дисплее горит.

Обращение к отдельной строке в данном дисплее не предусмотрено. Обращаться можно только к т.н. странице (*page*). Слово "страница" в данном случае означает просто группу из 8 последовательных строк, не стоит ассоциировать его с привычным нам значением.

Таким образом, всего имеется 8 (64/8=8) страниц (Page0-Page7). Так, строка 10 будет принадлежать странице Page1, строка 60 - странице Page7.

Координата (0,0) соответствует крайнему

верхнему левому пикселю дисплея, координата (127, 63) - крайнему правому нижнему (строки, колонки, страницы нумеруются с "0").

Приведем небольшой пример для пояснения.

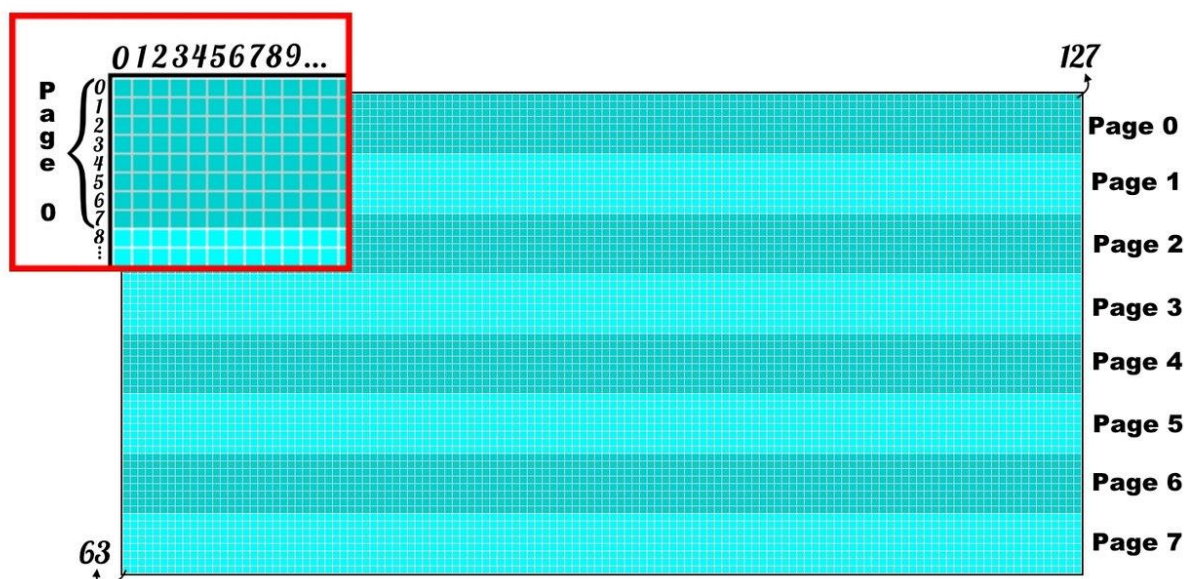


Рис. 3. Схема пиксельной матрицы дисплея

Допустим, необходимо зажечь один пиксель с адресом (78; 22) (22 строка, 78 пиксель при отсчете слева направо). Для этого нужно установить в "1" один бит ОЗУ на соответствующей позиции. Для позиционирования задаем адрес колонки (78) и адрес страницы (страницы, а не строки!). Из вышесказанного очевидно, что 22-ой строке соответствует страница Page2 (которая включает в себя строки 16-23). После задания колонки и страницы позиционирование произведено, осталось записать данные. Необходимо зажечь только один пиксель, то есть установить в "1" один бит в ОЗУ. Однако записать один бит нельзя, можно только байт. Поэтому в остальные 7 бит на данной странице необходимо записать "0", чтобы соответствующие пиксели не светились. Таким образом, записываем байт (0b01000000) который приводит к зажжению пикселя с адресом (78; 22). Если при таком же позиционировании был бы записан, например, байт (0b00000001), зажгется бы пиксель с адресом (78, 16).

Неприятной особенностью данного дисплея является то, что ОЗУ имеет 132 бита в строке, что больше, чем 128 пикселей в строке дисплея. При записи в первые две колонки (т.е. 1-ая и 2-ая) и последние две колонки (131-ая и 132-ая) ОЗУ на дисплее ничего не происходит. Данную особенность необходимо учитывать при написании управляющей программы.

ИНТЕРФЕЙС I²C

Управление и обмен данными с дисплеем

осуществляются по интерфейсу I²C. Для управления устройствами по этому интерфейсу используются две двунаправленные линии связи: SDA (последовательная линия данных) и SCL (последовательная линия тактирования). Такты генерирует ведущее устройство, ведомое лишь «отчитывается» при приёме байта. Рассматриваемый дисплей может выступать только в качестве ведомого (slave) устройства.

Процедура обмена начинается с того, что ведущий формирует состояние «СТАРТ»: генерирует переход сигнала линии SDA из «ВЫСОКОГО» состояния в «НИЗКОЕ» при «ВЫСОКОМ» уровне на линии SCL. Этот переход воспринимается подключенным к шине устройством как признак начала процедуры обмена. Генерация синхросигнала — это всегда обманность ведущего. Процедура обмена завершается тем, что ведущий формирует состояние «СТОП» — переход состояния линии SDA из низкого состояния в «ВЫСОКОЕ» при «ВЫСОКОМ» состоянии линии SCL. Состояния «СТАРТ» и «СТОП» всегда вырабатываются ведущим. Считается, что шина занята после фиксации состояния «СТАРТ». Шина считается освободившейся через некоторое время после фиксации состояния «СТОП». При передаче посылки по шине I²C ведущий генерирует свой синхросигнал на линии SCL. После формирования состояния «СТАРТ» ведущий опускает состояние линии SCL в «НИЗКОЕ» состояние и выставляет на линию SDA старший бит первого байта сообщения. Количество байт в сообщении не ограничено. Спецификация шины I²C разрешает изменения на линии SDA

только при «НИЗКОМ» уровне сигнала на линии *SCL*. Данные действительны и должны оставаться стабильными только во время «ВЫСОКОГО» состояния синхроимпульса. Для подтверждения приёма байта от ведущего-

передатчика ведомым-приёмником в спецификации протокола обмена по шине *I²C* вводится специальный бит подтверждения, выставляемый на шину *SDA* после приёма 8 бита данных.

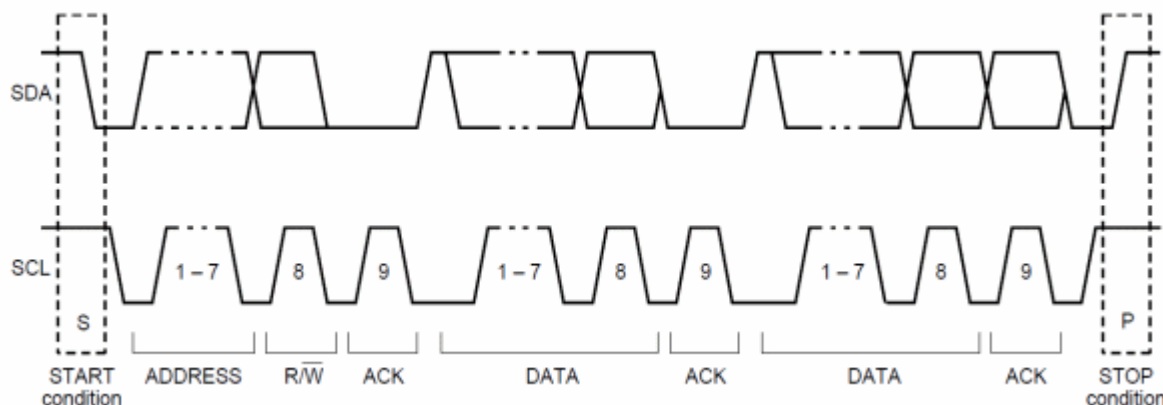


Рис. 4. Порядок сообщения по интерфейсу *I²C*

Передача 8 бит данных от передатчика к приёмнику завершается дополнительным циклом (формированием 9-го тактового импульса линии *SCL*), при котором приёмник выставляет низкий уровень сигнала на линии *SDA*, как признак успешного приёма байта. Подтверждение при передаче данных обязательно, кроме случаев окончания передачи ведомой стороной. Соответствующий импульс синхронизации генерируется ведущим. Передатчик отпускает («ВЫСОКОЕ» состояние) линию *SDA* на время синхроимпульса подтверждения. Приёмник должен удерживать линию *SDA* в течение «ВЫСОКОГО» состояния синхроимпульса подтверждения в стабильном «НИЗКОМ» состоянии.

В том случае, когда ведомый-приёмник не может подтвердить свой адрес (например, когда он выполняет в данный момент какие-либо функции реального времени), линия данных должна быть оставлена в «ВЫСОКОМ» состоянии. После этого ведущий может выдать состояние «СТОП» для прерывания пересылки данных. Если в пересылке участвует ведущий-приёмник, то он должен сообщить об окончании передачи ведомому-передатчику путем не подтверждения последнего байта. Ведомый-передатчик должен освободить линию данных для того, чтобы позволить ведущему выдать состояние «СТОП» или повторить состояние «СТАРТ». Более подробную информацию можно найти на Википедии [2] или других электронных ресурсах.

ОСОБЕННОСТИ УПРАВЛЕНИЯ

Дисплей принимает два вида байт: байты управления, записываемые в регистры управления контроллера, и байты данных, записываемые в оперативную память. Перед отправкой команды или данных сначала посылается управляющий байт, в котором значение определенного бита определяет, будет

ли следующий передаваемый байт командой или данными для записи в оперативную память. Выбор режима записи или считывания данных определяется другим битом, значение которого устанавливается в процессе работы по *I²C*.

В управлении данным дисплеем имеется ряд нюансов, которые необходимо осветить. Хотя при отправлении данных автоматический инкрементируется счетчик адреса колонки, что позволяет записывать несколько байт информации в автоматическом режиме, этого не происходит со счетчиком адреса страницы, поэтому страницы необходимо переключать вручную с помощью предусмотренной для этого команды управления (см. Список команд управления). Адреса страниц и колонок не зависят друг от друга. Кроме того, как отмечалось выше, в памяти дисплея на 4x8 ячеек больше, нежели имеется точек на дисплее, поэтому при записи данных во избежание некорректного (смещенного) отображения информации следует устанавливать адрес колонки с дополнительным сдвигом на две колонки.

В случае, если необходимо подключить к управляющему устройству два дисплея и выводить на каждый разную информацию, следует на одном из дисплеев переставить резистор-перемычку, отвечающий за адрес устройства. Он расположен на обратной стороне дисплея и позволяет установить адрес устройства как 0x68 или 0x69.

Возможности дисплея позволяют без особых трудностей организовать с помощью встроенных команд управления вертикальную прокрутку и различные способы отображения информации на дисплей. (см. Список команд управления и документацию).

СПИСОК КОМАНД УПРАВЛЕНИЯ

Ниже приведены команды управления дисплеем в шестнадцатеричной системе

счисления. Некоторые команды, например, установка адреса страницы, представляют собой некоторый диапазон, соответствующий в обозначенном примере количеству страниц.

Установка старших битов адреса колонки (00H - 0FH).

Установка младших битов адреса колонки (10H - 1FH).

Устанавливает восьмибитный адрес колонки, разделенный на старшую и младшую составляющую, в оперативной памяти дисплея.

Управление встроенным источником напряжения.

При вызове команд управления состоянием источника напряжения дисплей должен быть выключен.

Вход в режим управления источником напряжения (ADH);

Выкл/вкл внутренний источник напряжения (8AH/8BH).

Установка значения подкачки напряжения (30H - 3FH).

Устанавливает выходное напряжение встроенного источника напряжения (7.4 – 9 В).

Установка начальной строки (40H-7FH).

Устанавливает адрес строки, с которой начинается вывод информации. Выбранная в памяти контроллера строка окажется верхней на дисплее.

Настройка контрастности.

Устанавливает значение контраста дисплея из 256 возможных в диапазоне 00H – FFH. Включает две команды:

Вход в режим управления контрастом (81H);

Установка значения контраста (00H – FFH). После отправки данной команды, происходит автоматический выход из режима управления контрастом.

Установка режима отображения (A0H/A1H).

Устанавливает инверсный или нормальный режим отображения информации. Иными словами, позволяет отображать информацию зеркально.

Выкл/вкл режим подсветки всего дисплея (A4H /A5H).

Устанавливает нормальный режим работы дисплея или принудительно зажигает все светодиоды.

Выкл/вкл инверсию состояний точек дисплея (A6H /A7H).

Определяет нормальный режим работы дисплея, либо инверсный, в котором в зависимости от содержимого оперативной памяти все выключенные светодиоды включаются, а включенные – выключаются.

Выкл/вкл дисплей (AEH/AFH).

Установка адреса страницы (B0H-B7H).

Приведенный список команд является неполным. Полный список управляющих команд и их подробное описание можно найти в

документации [1].

ПОДКЛЮЧЕНИЕ ДИСПЛЕЯ К STM32VLDISCOVERY

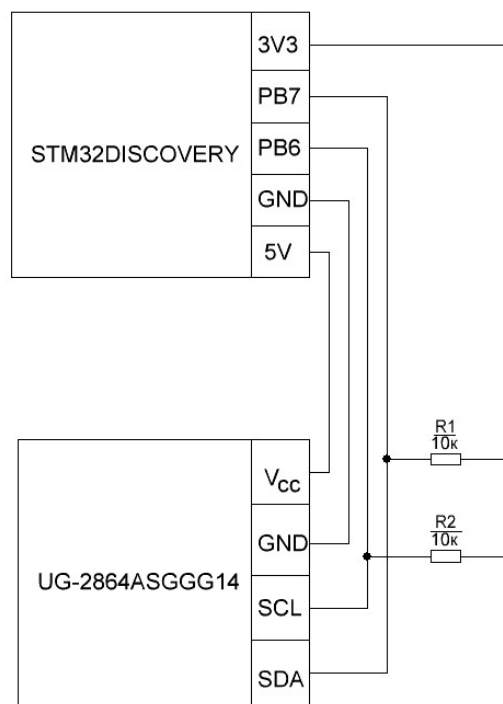


Рис. 5. Схема подключения дисплея к плате

Для подключения дисплея к плате используются четыре вывода, однако для работы интерфейса I²C строго необходимы подтягивающие к напряжению уровня логической «1» резисторы. Это объясняется тем, что передача сигналов осуществляется «притягиванием» линии к «0», а в состоянии «1» она устанавливается автоматически благодаря подтяжке.

На вход V_{cc} питания дисплея подается напряжение 5 В, которое поступает на стабилизатор и питает контроллер, а также на множитель напряжения, чтобы питать матрицу светодиодов. Следует обратить внимание, что контроллер дисплея питается от 3,3 В, соответственно и уровень логической «1» так же близок к этому напряжению. В нашем случае, при инициализации дисплея, необходимо подключить внутренний генератор подкачки напряжения.

ТЕСТОВЫЙ ПРИМЕР

В качестве тестового примера рассмотрим заполнение дисплея константой. Последовательность действий:

1. Настройка подсистемы I²C №1 в микроконтроллере.
2. Настройка дисплея.
3. Вывод тестовой информации на дисплей.

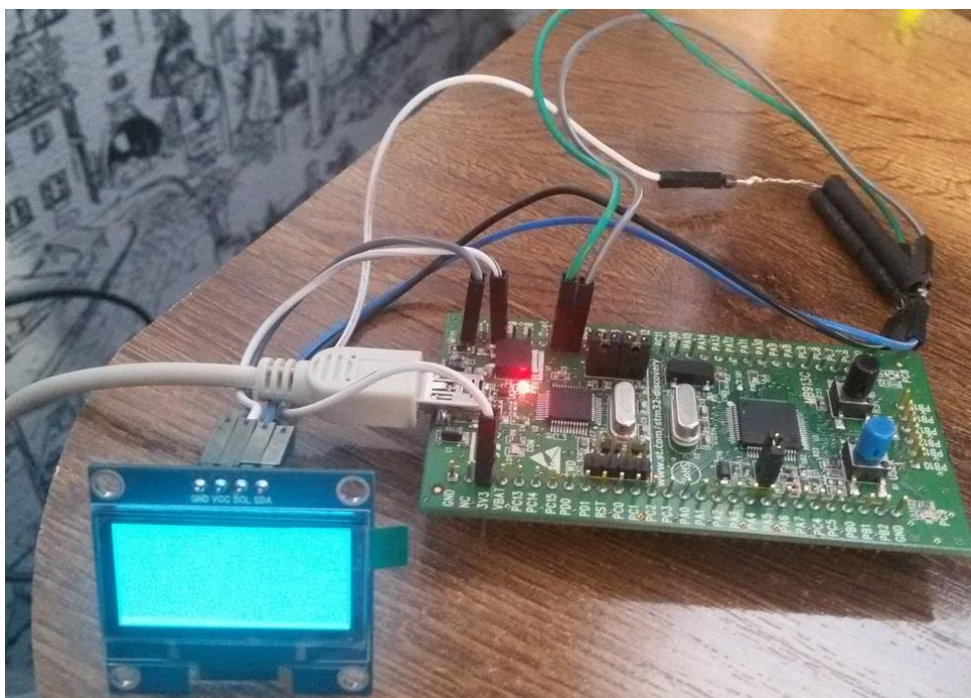


Рис. 6. Схема в собранном состоянии

Содержимое файла SH1106.h:

```
#ifndef __SH1106_H
#define __SH1106_H
#include "stm32f10x.h"
#define SH1106_I2C_ADDRESS          0x78
#define SH1106_I2C_SPEED            100000 //300000
// GPIO OLED
#define SH1106_I2C                   I2C1
#define SH1106_I2C_CLK                RCC_APB1Periph_I2C1
#define SH1106_I2C_SCL_PIN            GPIO_Pin_6
#define SH1106_I2C_SCL_GPIO_PORT     GPIOB
#define SH1106_I2C_SCL_GPIO_CLK      RCC_APB2Periph_GPIOB
#define SH1106_I2C_SDA_PIN            GPIO_Pin_7
#define SH1106_I2C_SDA_GPIO_PORT     GPIOB
#define SH1106_I2C_SDA_GPIO_CLK      RCC_APB2Periph_GPIOB
#define SH1106_I2C_DR                  ((uint32_t)0x40005410)
void OLED_SH1106_I2C_init(void);
void UG2864ASGGG14_Init(void);
void OLED_SH1106_Clear(uint8_t data);
#endif
```

Содержимое файла SH1106.c:

```
#include "stm32f10x.h"
#include "sh1106.h"
//===== Настройка интерфейса I2C =====
void OLED_SH1106_I2C_init(void) {
    I2C_InitTypeDef I2C_InitStructure;
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB1PeriphClockCmd(SH1106_I2C_CLK, ENABLE);
    RCC_APB2PeriphClockCmd(SH1106_I2C_SCL_GPIO_CLK | SH1106_I2C_SDA_GPIO_CLK,
    ENABLE);

    GPIO_InitStructure.GPIO_Pin = SH1106_I2C_SCL_PIN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_10MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_OD;
    GPIO_Init(SH1106_I2C_SCL_GPIO_PORT, &GPIO_InitStructure);

    GPIO_InitStructure.GPIO_Pin = SH1106_I2C_SDA_PIN;
    GPIO_Init(SH1106_I2C_SDA_GPIO_PORT, &GPIO_InitStructure);
}
```

```

I2C_InitStructure.I2C_Mode = I2C_Mode_I2C;
I2C_InitStructure.I2C_DutyCycle = I2C_DutyCycle_2;
I2C_InitStructure.I2C_OwnAddress1 = SH1106_I2C_ADDRESS;
I2C_InitStructure.I2C_Ack = I2C_Ack_Enable;
I2C_InitStructure.I2C_AcknowledgedAddress= I2C_AcknowledgedAddress_7bit;
I2C_InitStructure.I2C_ClockSpeed = SH1106_I2C_SPEED;

I2C_Cmd(SH1106_I2C, ENABLE);
I2C_Init(SH1106_I2C, &I2C_InitStructure);
}

//===== Передача команды по I2C =====
void WriteC(uint8_t COM) {
while (I2C_GetFlagStatus(I2C1, I2C_FLAG_BUSY));

I2C_GenerateSTART(I2C1, ENABLE);
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_MODE_SELECT));

// адрес устройства
I2C_Send7bitAddress(I2C1, 0x78, I2C_Direction_Transmitter);
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED));

I2C_SendData(I2C1, 0x00); // контрольный байт
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_BYTE_TRANSMITTED));

I2C_SendData(I2C1, COM); // команда
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_BYTE_TRANSMITTED));

I2C_GenerateSTOP(I2C1, ENABLE);
}

//===== Передача данных по I2C =====
void WriteD(uint8_t DAT) {
while (I2C_GetFlagStatus(I2C1, I2C_FLAG_BUSY));

I2C_GenerateSTART(I2C1, ENABLE);
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_MODE_SELECT));

// адрес устройства
I2C_Send7bitAddress(I2C1, 0x78, I2C_Direction_Transmitter);
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED));

I2C_SendData(I2C1, 0x40); // контрольный байт
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_BYTE_TRANSMITTED));

I2C_SendData(I2C1, DAT); // данные
while (!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_BYTE_TRANSMITTED));

I2C_GenerateSTOP(I2C1, ENABLE);
}

//===== Минимальная настройка дисплея UG-2864ASGGG14 =====
void UG2864ASGGG14_Init(void) {
WriteC(0xAE); //выключить дисплей*/
WriteC(0x00); /*младшие биты адреса колонки*/
WriteC(0x10); /*старшие биты адреса колонки*/
WriteC(0x40); /*стартовая строка*/
WriteC(0xB0); /*стартовая страница*/
WriteC(0x81); /*установка контраста*/
WriteC(0x80); /*128*/
WriteC(0xA1); /*отображение слева направо*/
WriteC(0xA6); /*нормальный режим*/
WriteC(0xad); /*включить генератор подкачки 7.4В*/
}

```

```
WriteC(0x8b);
WriteC(0x30);
WriteC(0xAF);    /*включить дисплей*/
}

//===== Заполнение всего дисплея константой =====
void OLED_SH1106_Clear() {
static uint8_t page = 0xB0;
uint8_t x, y;
for(y = 0; y < 8; y++) {
    WriteC(page);
    page++;
for(x = 0; x < 132; x++){
    WriteD(0x02); }
}
//-----
page = 0xB0;
}
```

Содержимое файла main.c:

```
#include <stm32f10x.h>
#include "sh1106.h"
int main(void)
{
uint32_t i32;
for (i32=0; i32<100000; i32++) ; // Программная задержка
OLED_SH1106_I2C_init(); // Настройка I2C
for (i32=0; i32<100000; i32++) ;
UG2864ASGGG14_Init(); // Настройка дисплея
for (i32=0; i32<100000; i32++) ;
while (1) {
    for (s=0; s<256; s++) OLED_SH1106_Clear();
}
}
```



Рис. 7. Тестовый пример



Рис. 8. Демонстрация возможностей дисплея по выводу растровой графики

ВЫВОДЫ

- Дисплей *OLED* является относительно простым в подключении устройством.

• Характеристики данного устройства позволяют в короткие сроки создавать качественные и гибкие в настройке устройства вывода информации с богатыми возможностями.

Работа выполнена при финансовой поддержке Минобрнауки России по государственному заданию №2014/138, тема проекта «Новые структуры, модели и алгоритмы для прорывных методов управления техническими системами на основе наукоемких результатов интеллектуальной деятельности», проект № 471.

ЛИТЕРАТУРА

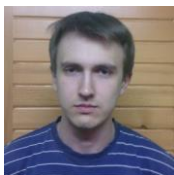
- [1] SH1106 datasheet [Электронный ресурс]. Режим доступа: http://www.rhydolabz.com/documents/sh1106_datashet.pdf
- [2] I²C [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/I%C2%B2C>



Вадим Аркадьевич Жмудь – заведующий кафедрой Автоматики НГТУ, профессор, доктор технических наук.
E-mail: oao_nips@bk.ru



Каменская Анна Сергеевна - студентка группы АА-36 кафедры Автоматики НГТУ.
E-mail: ladymacbeth94@yandex.ru



Курбетьев Константин Вячеславович - студент группы АА-37 кафедры Автоматики НГТУ.
E-mail: kkurbetev@mail.ru



Виталий Геннадьевич Трубин - зав. лаб. кафедры Автоматики НГТУ, директор ООО «КБ Автоматика».
E-mail: trubin@ngs.ru

Graphic OLED Display UG-2864ASYGG14: First Start

V.A. Zhmud, A.S. Kamenskaya, K.V. Kurbetyev, V.G. Trubin

Abstract: The paper discusses main issues arising according the development of OLED display based on UG-2864ASGGG14, which is controlled by the debug board STM32VLDISCOVERY.

Key words: microcontroller, STM32, STM32VLDISCOVERY, UG-2864ASGGG14, SH1106, OLED.