

[Портал о науке и технике](#)

- [Статьи](#)
- [Новости](#)
- [Видео](#)
- [Обзоры](#)

[Вход](#)

Вход

Логин:

Пароль

[Забыли пароль?](#)

[Регистрация](#)

Войти через социальные сети:



- [Схемотехника](#)
- [stm32](#)
- [TFT / LCD](#)
- [Программирование](#)
- [Железо](#)
- [Авто](#)

Воспользуйтесь строкой поиска, чтобы найти нужный материал

Введите слово

[Главная](#) [Дисплей](#) Вывод символов и строк на TFT дисплей, на примере ILI9341.

[Поддержать проект](#)

Вывод символов и строк на TFT дисплей, на примере ILI9341.

[В прошлой статье](#) мы рассмотрели как инициализировать TFT дисплей, под управлением драйвера **ILI9341**, в этой будем учиться выводить символы и строки.

Но для начала надо разобраться с одним моментом, если у **SSD1289** для вывода точки на дисплей необходимо указать две её координаты и цвет точки, то у **ILI9341** указывается 4 координаты, которые ограничивают рабочую область, затем указывается цвет вновь отрисовываемой точки, а порядок отрисовки точек задаётся при инициализации.

Что касается порядка отрисовки точек, то он задаётся после отправки команды **36h**.

MADCTL (Memory Access Control)												
D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
0	1	↑	XX	0	0	1	1	0	1	1	0	36h
1	1	↑	XX	MY	MX	MV	ML	BGR	MH	0	0	00

This command defines read/write scanning direction of frame memory.

This command makes no change on the other driver status.

Bit	Name	Description
MY	Row Address Order	These 3 bits control MCU to memory write/read direction.
MX	Column Address Order	
MV	Row / Column Exchange	
ML	Vertical Refresh Order	LCD vertical refresh direction control.
BGR	RGB-BGR Order	Color selector switch control (0=RGB color filter panel, 1=BGR color filter panel)
MH	Horizontal Refresh ORDER	LCD horizontal refreshing direction control.

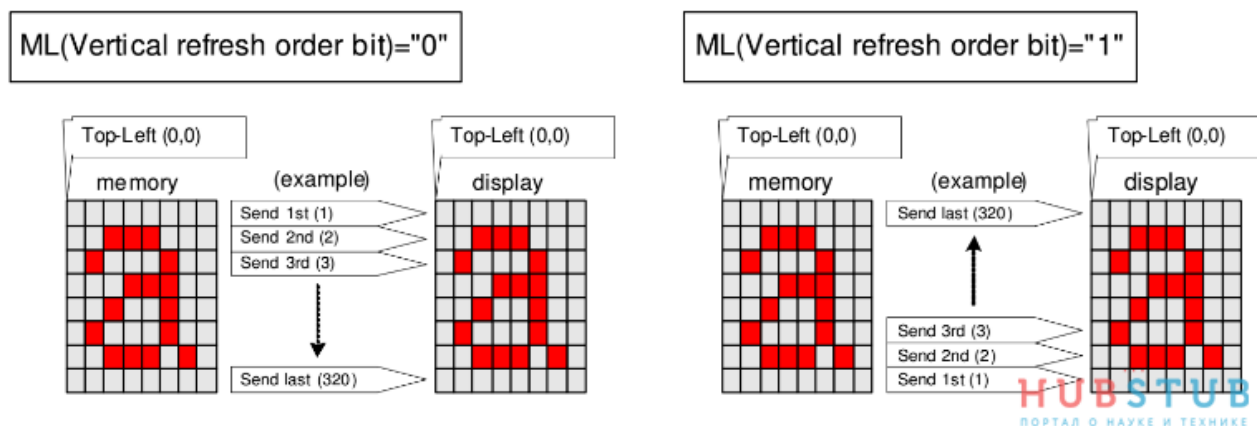
Note: When BGR bit is changed, the new setting is active immediately without update the content in Frame Memory again.

X = Don't care.

Для того чтобы понять, что там написано надо мысленно провести две оси, ось **X** по горизонтали и ось **Y** по вертикали, тогда:

- MX – определяет с какой стороны дисплея X будет равен нулю, с правой или с левой, соответственно с другой стороны дисплея X будет максимальным
- MY – то же самое только для оси Y
- MV – меняет оси X и Y местами

- ML – определяет порядок обновления выделенной области по вертикали
- MH – определяет порядок обновления выделенной области по горизонтали



Изменение значений ML и MH, не давали никакого результата, с этим ещё предстоит разобраться.

Для отрисовки чего-либо на экране необходимо выполнить следующую последовательность действий:

- отправляем команду 2A, затем координаты начала и конца области по горизонтали
- отправляем команду 2B, затем координаты начала и конца области по вертикали
- отправляем команду 2C, то есть говорим: “Сейчас будем писать в видеоОЗУ”
- посылаем кодировку цвета, который хотим вывести в текущей ячейке
- снова посылаем кодировку цвета, при этом координаты сами изменятся по выбранному при инициализации алгоритму

Тогда код для отрисовки одного пикселя будет выглядеть так

```
void TFT_Send_Data(uint16_t data)
{
    uint8_t data1 = data>>8;
    uint8_t data2 = data&0xff;
    TFT_Write_Data(data1);
    TFT_Write_Data(data2);
}

/*ф-ция ограничивает координаты рабочей области по оси X*/
void TFT_Set_Column(uint16_t start_column,uint16_t end_column)
{
    TFT_Send_Cmd(0x2A);
    TFT_Send_Data(start_column);
    TFT_Send_Data(end_column);
}

/*ф-ция ограничивает координаты рабочей области по оси Y*/
void TFT_Set_Page(uint16_t start_page,uint16_t end_page)
{
    TFT_Send_Cmd(0x2B);
    TFT_Send_Data(start_page);
    TFT_Send_Data(end_page);
}

/*ф-ция ограничивает координаты рабочей области*/
void TFT_Set_XY(uint16_t x, uint16_t y)
{
    TFT_Set_Column(x, x);
    TFT_Set_Page(y, y);
}

/*ф-ция отрисовывает пиксель по заданным координатам*/
void TFT_Draw_Pixel(uint16_t x, uint16_t y,uint16_t color)
{
    TFT_Set_XY(x, y);
    TFT_Send_Cmd(0x2C);
    TFT_Write_Data16(color);
}
```

А теперь, как и обещал в прошлой статье, давайте рассмотрим как залить экран выбранным цветом.

```
uint16_t constrain(uint16_t a, uint16_t b, uint16_t c)
{
    if (a < b)
    {
        return b;
    }
    if (c < a)
    {
        return c;
    }
    else return a;
}

void TFT_Fill_Screen(uint16_t x_left, uint16_t x_right, uint16_t y_up, uint16_t y_down, uint16_t color)
{
    unsigned long xy=0;
    unsigned long i=0;
    if(x_left > x_right)
    {
        x_left = x_left^x_right;    //если координата левого края больше
        x_right = x_left^x_right;    //координаты правого края они поменяются
        x_left = x_left^x_right;    //местами, было x_left = 5 x_right = 3
        //стало x_left = 3 x_right = 5
    }
    if(y_up > y_down)
    {
        y_up = y_up^y_down;        //то же самое для оси y
        y_down = y_up^y_down;      //название этой операции
        y_up = y_up^y_down;        //"swap без временной переменной"
    }
    //контролируем, что бы передаваемые в функцию координаты
    //входили в область допустимых значений
    x_left = constrain(x_left, MIN_X,MAX_X);
    x_right = constrain(x_right, MIN_X,MAX_X);
    y_up = constrain(y_up, MIN_Y,MAX_Y);
    y_down = constrain(y_down, MIN_Y,MAX_Y);

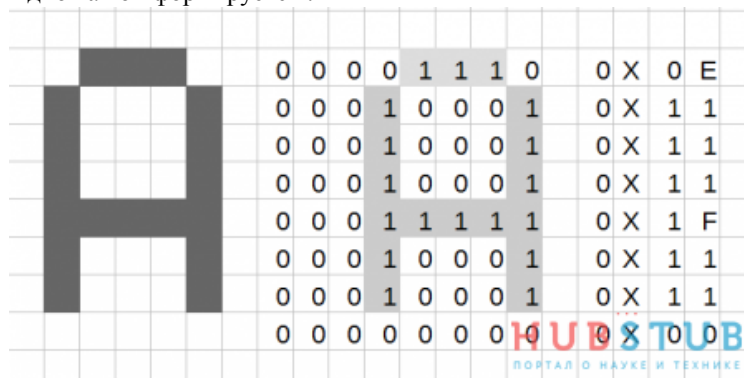
    xy = (x_right - x_left+1);      //рассчитываем количество точек
    xy = xy*(y_down - y_up+1);     //которое надо закрасить

    TFT_Set_Column(x_left,x_right); //задаём рабочую область по x
    TFT_Set_Page(y_up, y_down);    //задаём рабочую область по y
    TFT_Send_Cmd(0x2c);            //будем писать в видео ОЗУ

    for(i=0; i < xy; i++)
    {
        TFT_Write_Data16(color);    //передаём кодировку цвета
    }
}
```

Надеюсь, что как работает функция понятно из комментариев.

Теперь давайте рассмотрим как рисовать символы на дисплее. Каждая буква размером 8x8 представляет собой массив из 8 элементов, на рисунке ниже видно как он формируется .



Для того чтобы вывести букву А, надо построчно выводить элементы массива. Для удобства все буквы вместе сложены в один большой **двумерный массив** и хранятся во флеше контроллера, на что указывает атрибут PROGMEM.

[Показать / Скрыть текст](#)

Функция для отрисовки символов выглядит так.

```

void TFT_Draw_Char(uint16_t x, uint16_t y, uint16_t color, uint16_t phone, uint8_t ascii, uint8_t size)
{
    for (int i = 0; i < FONT_Y; i++ )
    {
        for(uint8_t f = 0; f < FONT_X; f++)
        {
            if((pgm_read_byte(&simbols[ascii-0x20][i])>>(7-f))&0x01)
            {
                TFT_Fill_Rectangle(x+f*size, y+i*size, size, size, color);
            }
            else
            {
                TFT_Fill_Rectangle(x+f*size, y+i*size, size, size, phone);
            }
        }
    }
}

```

Работает она следующим образом, с помощью функции

pgm_read_byte(&simbols[ascii-0x20][i])

из флэша извлекается i-тый байт нужного нам символа. А как мы помним, символ кодируется 8 байтами, а **каждый байт, в свою очередь, хранит в себе информацию о том как закрашивать строку**. У этого байта смещаем в правую крайнюю позицию интересующий нас бит

(pgm_read_byte(&simbols[ascii-0x20][i])>>(7-f))

и с помощью операции **побитового и** определяем чему он равен

(pgm_read_byte(&simbols[ascii-0x20][i])>>(7-f))&0x01

Если он равен единице рисуем точку соответствующую цвету символа, иначе точку цвета фона. Таким образом, построчно отрисовываются символы.

А вот и функция для вывода строк.

```

void TFT_Draw_String(uint16_t x, uint16_t y, uint16_t color, uint16_t phone, char *string, uint8_t size)
{
    //определить конец строки очень просто если знать, что она ВСЕГДА заканчивается нулём
    while(*string)
    {
        //проверяем не вылезем ли мы за пределы экрана при отрисовке следующего символа,
        // если да, то переходим на следующую строчку
        if((x + FONT_X) > MAX_X)
        {
            x = 1;
            y = y + FONT_X*size;
        }
        TFT_Draw_Char(x, y, color, phone, *string, size); //отрисовываем символ
        x += FONT_X*size; //изменяем координату для отрисовки следующего символа
        *string++; //увеличиваем значение указателя, чтобы он ссылался на следующий символ
    }
}

```

Думаю, как работает эта функция тоже понятно из комментариев.



[В следующей статье](#) мы будем учиться рисовать геометрические фигуры.

Проект для **Atmega16** в Atmel Studio 6.2 в архиве [ili9341_i8080_example.rar](#) [9,84 Kb] (скачиваний: 286).

Источник: hubstub.ru

[Статья](#)

1

Похожие статьи

24-04-2016, 15:57

[Инициализация TFT дисплея на примере ILI9341 для AVR.](#)

[Дисплей](#)

3-06-2015, 14:19

[Вывод символов и строк на LCD дисплей 1602A, с управляющим контроллером KS0066U.](#)

[Дисплей](#)

4-03-2015, 14:36

[Как получить координаты точки касания и произвести калибровку резистивной сенсорной панели на примере ХРТ2046.](#)

[Дисплей](#)

24-02-2015, 19:41

[Рисование геометрических фигур на TFT дисплее, на примере SSD1289.](#)

[Дисплей](#)

17-02-2015, 18:49

[Вывод символов и строк на TFT дисплей, на примере SSD1289.](#)

[Дисплей](#)

комментарии

2



Добро пожаловать в Disqus! Находите больше отличных обсуждений как это. Мы гораздо больше чем просто комментарии.

Начать

Dismiss ✕

2 Комментариев

HubStub

 Alexandr Pirs ▾

 Рекомендовать

 Поделиться

Новое в начале ▾



Присоединиться к обсуждению...

**Юрий Ерофеев** • год назад

На изображении, где приводится кодировка символа 8x8, в предпоследней строке правого столбца разве не должно быть 0x11, так же, как и в предыдущих строках?

^ | ▾ • Ответить • Поделиться ›

**HubStub** Модератор ➔ Юрий Ерофеев • год назад

Спасибо, исправил!!!

^ | ▾ • Ответить • Поделиться ›

ТАКЖЕ НА HUBSTUB

Как выбрать mosfet.

4 комментария • 2 года назад

Аватар HubStub — Уже как-то писал, что он образуется в процессе изготовления.

Как работает компаратор на операционном усилителе(ОУ).

2 комментария • 2 года назад

Аватар HubStub — На счёт побольше сайтов мне сказать нечего, а вот побольше статей - это в наших силах.

Инициализация TFT дисплея на примере ILI9341 для AVR.

20 комментариев • год назад

Аватар schneider taylor — ..., не стал изменять традициям и выбрал i8080....Как выбрали? Ног IM0-IM3 на модуле не выведены, а на ...

Вывод символов и строк на TFT дисплей, на примере SSD1289.

4 комментария • 2 года назад

Аватар HubStub — 1. Что Вы изменяли мне не известно, да и программой этой не пользовался. Попробуйте проверить нарисовав один из ...

 Подписаться

 Добавь Disqus на свой сайтДобавить DisqusДобавить

 Конфиденциальность

Статьи раздела

[Вывод символов и строк на T...](#)[Вывод символов и строк на TE...](#)[Вывод символов и строк на LC...](#)[Инициализация LCD дисплея 1...](#)

[Инициализация TFT дисплея на...](#)

[Инициализация TFT дисплея на...](#)

[Как получить координаты точк...](#)

[Как работает резистивный сенс...](#)

[Подключение TFT дисплея по F...](#)

[Подключение TFT дисплея по F...](#)

[Подключение резистивной сенс...](#)

[Рисование геометрических фиг...](#)

[Рисование геометрических фиг...](#)

[Посетите наш канал](#)

Разделы

- [Статьи](#)
- [Новости](#)
- [Видео](#)
- [Обзоры](#)

ИНформация

- [О сайте](#)
- [Правила](#)
- [Помощь](#)
- [Контакты](#)

Услуги

- [Реклама](#)

support@hubstub.ru

HubStub.ru © 2014 - 2016 Все права защищены