

[Портал о науке и технике](#)

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

[Вход](#)

Вход

Логин:

Пароль

Войти

[Забыли пароль?](#)

[Регистрация](#)

Войти через социальные сети:



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

Воспользуйтесь строкой поиска,
чтобы найти нужный материал

Введите слово

Найти

[Поддержать проект](#)

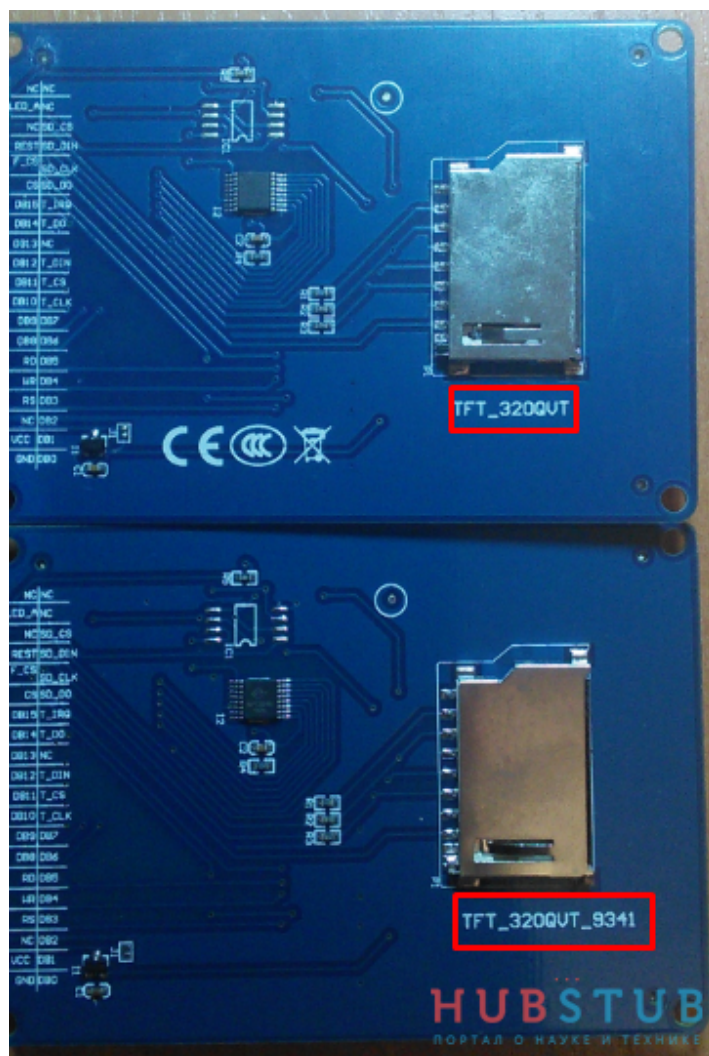
[Главная](#) [Дисплей](#) Инициализация TFT дисплея на примере ILI9341 для AVR.

Инициализация TFT дисплея на примере ILI9341 для AVR.

Около года тому назад на сайте появилась статья о том [как инициализировать TFT дисплей, под управлением SSD1289](#), а где-то около месяца назад мне написал один из посетителей сайта. Суть письма была в том, что он заказал дисплей по указанной в статье ссылке, но запустить его не получалось и он предложил этот дисплей и ещё несколько других выслать мне, а я, в свою очередь, должен буду выложить код если получится их запустить.

Как оказалось, дисплей, который ему прислали с али управляется драйвером **ILI9341**, об этом помогла догадаться надпись на нём.

Верхний дисплей с драйвером SSD1289, нижний с **ILI9341**.



Поэтому как только ко мне пришёл этот дисплей, сразу начал изучать даташит на **ILI9341**.

В отличие от SSD1289 у **ILI9341** нет регистров, для общения с ним используются команды. Сначала посылаешь команду, а затем набор параметров, то есть после того, как мы послали команду дисплей уже ждёт, что ему передадут параметры. Также надо сказать, что общение с дисплеем осуществляется по одному из двух протоколов: интеловский i8080 и мотероловский M6800, чем они отличаются описывал [тут](#), не стал изменять традициям и выбрал i8080. У этого протокола существует две реализации, для получения более подробной информации можно почитать даташит.

Первым делом необходимо реализовать низкоуровневые функции и тут важно понять, что мы правильно понимаем даташит. Самый простой способ проверить, правильность понимания даташита, это считать какую-то информацию с дисплея, например, его ID.

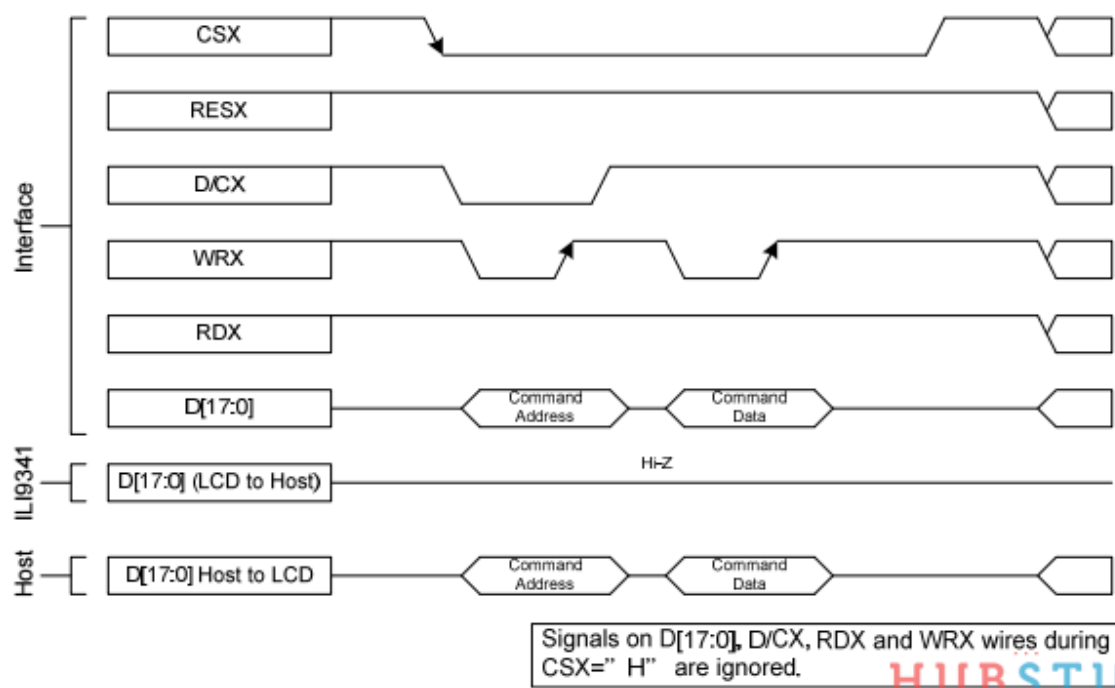
Read ID4	0	↑	1	XX	1	1	0	1	0	0	1	1	D3h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	0	0	0	0	0	0	0	0	00
	1	↑	1	XX	1	0	0	1	0	0	1	1	93
	1	↑	1	XX	0	1	0	0	0	0	0	1	41

На картинке видно, что если контроллеру отправить команду 0xD3h и 4 раза считать данные, то в последние два считывания он должен вернуть ID(**9341**) драйвера. Но для начала надо разобраться как отправлять команды и читать данные, для этого нам понадобятся три картинки из даташита.

Первая, определяет управляющие сигналы для выбранного интерфейса.

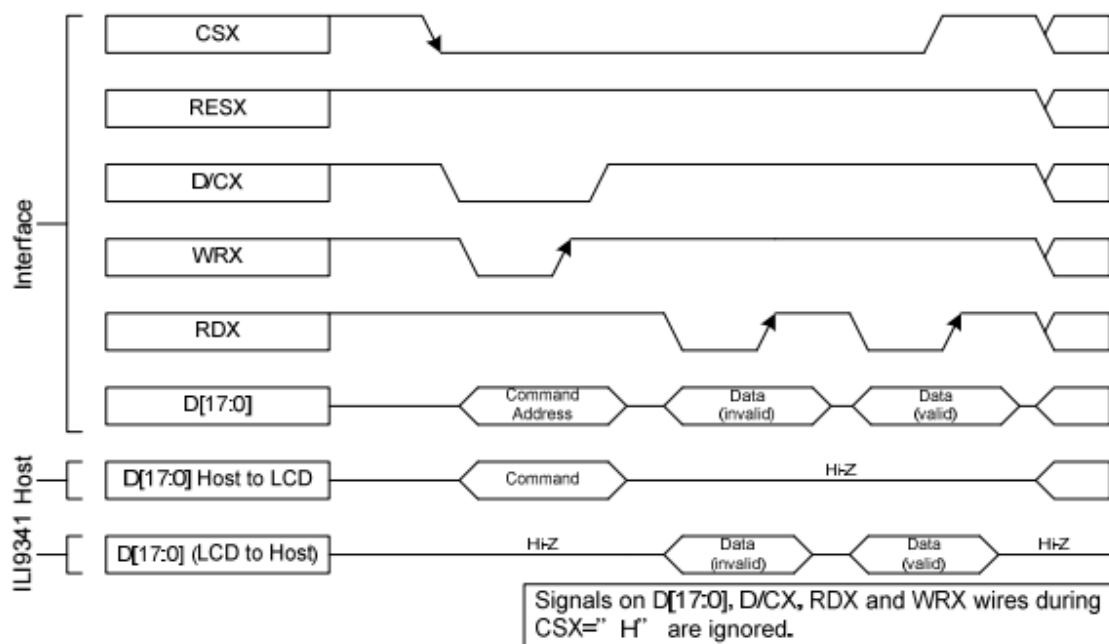
IM3	IM2	IM1	IM0	MCU-Interface Mode	CSX	WRX	RDX	D/CX	Function
0	0	0	0	8080 MCU 8-bit bus interface 1	"L"		"H"	"L"	Write command code.
					"L"	"H"		"H"	Read internal status.
					"L"		"H"	"H"	Write parameter or display data.
					"L"	"H"		"H"	Reads parameter or display data.
0	0	0	1	8080 MCU 16-bit bus interface 1	"L"		"H"	"L"	Write command code.
					"L"	"H"		"H"	Read internal status.
					"L"		"H"	"H"	Write parameter or display data.
					"L"	"H"		"H"	Reads parameter or display data.

Вторая и третья определяют их последовательность, для записи



для чтения

Note: RDX is an unsynchronized signal (It can be stopped).



Note: Read data is only valid when the D/CX input is pulled high. If D/CX is driven low during read then the display information outputs will be High-Z.

HUBSTUB
ПОРТАЛ О НАУКЕ И ТЕХНИКЕ

Тогда функция, для отправки команды, будет выглядеть так.

```
void TFT_Send_Cmd(uint8_t cmd)
{
    COMMAND_PORT &= ~(1<<lcd_dc); //будем слать команду
    COMMAND_PORT |= (1<<lcd_rd); //выставляем на ножке, отвечающей за чтение 1

    COMMAND_PORT &= ~(1<<lcd_cs); //активируем чип

    COMMAND_PORT1 &= ~(1<<lcd_wr); //стробируем битом записи
    DATA_PORT_0 = cmd;
    _delay_us(5);
    COMMAND_PORT1 |= (1<<lcd_wr);

    COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
}
```

А функция которая читает данные так.

```
uint8_t TFT_Read_Data(void)
{
    uint8_t data = 0;
    DATA_DDR_0 = 0X00; //порт на вход с подтяжкой к земле
    DATA_PORT_0 = 0x00;
    COMMAND_PORT |= (1<<lcd_dc); //будем читать ДАННЫЕ
    COMMAND_PORT1 |= (1<<lcd_wr); //выставляем на ножке, отвечающей за запись 1

    COMMAND_PORT &= ~(1<<lcd_cs); //активируем чип

    COMMAND_PORT &= ~(1<<lcd_rd); //стробируем битом чтения
    _delay_us(5);
    data = PINA;
    COMMAND_PORT |= (1<<lcd_rd);

    COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
    DATA_DDR_0 = 0XFF; //порт на выход
}
```

```
return data;
```

```
}
```

```

24 uint8_t TFT_Read_Data(void)
25 {
26     uint8_t data = 0;
27     DATA_DDR_0 = 0X00; //порт на вход с подтяжкой к земле
28     DATA_PORT_0 = 0x00;
29     COMMAND_PORT |= (1<<lcd_dc); //будем читать ДАННЫЕ
30     COMMAND_PORT1 |= (1<<lcd_wr); //выставляем на ножке, отвечающей за запись
31
32     COMMAND_PORT &= ~(1<<lcd_cs); //активируем чип
33
34     COMMAND_PORT &= ~(1<<lcd_rd); //стробируем битом чтения
35     _delay_us(5);
36     data = PINA;
37     COMMAND_PORT |= (1<<lcd_rd);
38
39     COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
40     DATA_DDR_0 = 0XFF; //порт на выход
41     return data;
42 }
43
44 void TFT_Send_Cmd(uint8_t cmd)
45 {
46     COMMAND_PORT &= ~(1<<lcd_dc); //будем слать команду
47     COMMAND_PORT |= (1<<lcd_rd); //выставляем на ножке, отвечающей за чтение :
48 }

```

Name	Value	Type
data	0x93	uint8_t(r

HUBSTUB
ПОРТАЛ О НАУКЕ И ТЕХНИКЕ

```

37     COMMAND_PORT |= (1<<lcd_rd);
38
39     COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
40     DATA_DDR_0 = 0XFF; //порт на выход
41     return data;
42 }
43
44 void TFT_Send_Cmd(uint8_t cmd)
45 {
46     COMMAND_PORT &= ~(1<<lcd_dc); //будем слать команду
47     COMMAND_PORT |= (1<<lcd_rd); //выставляем на ножке, отвечающей за чтение
48
49     COMMAND_PORT &= ~(1<<lcd_cs); //активируем чип
50
51     COMMAND_PORT1 &= ~(1<<lcd_wr); //стробируем битом записи
52     DATA_PORT_0 = cmd;
53     _delay_us(5);
54     COMMAND_PORT1 |= (1<<lcd_wr);
55
56     COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
57 }

```

Name	Value	Type
data	0x41	uint8_t(r

HUBSTUB
ПОРТАЛ О НАУКЕ И ТЕХНИКЕ

Из скриншотов понятно, что описанные выше функции заработали, осталось реализовать отправку данных по 8 и 16 бит. Дело в том, что в основном при отправке данных используются только младшие 8 бит шины, но, например, при записи данных в память дисплея используются все 16 бит.

```

void TFT_Write_Data(uint8_t data)
{
    COMMAND_PORT |= (1<<lcd_dc); //будем слать ДАННЫЕ
    COMMAND_PORT |= (1<<lcd_rd); //выставляем на ножке, отвечающей за чтение 1

    COMMAND_PORT &= ~(1<<lcd_cs); //активируем чип

```

```

COMMAND_PORT1 &= ~(1<<lcd_wr); //стробируем битом записи
DATA_PORT_0 = data;
_delay_us(5);
COMMAND_PORT1 |= (1<<lcd_wr);

COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
}

void TFT_Write_Data16(uint16_t data)
{
    COMMAND_PORT |= (1<<lcd_dc); //будем слать ДАННЫЕ
    COMMAND_PORT |= (1<<lcd_rd); //выставляем на ножке, отвечающей за чтение 1

    COMMAND_PORT &= ~(1<<lcd_cs); //активируем чип

    COMMAND_PORT1 &= ~(1<<lcd_wr); //стробируем битом записи
    DATA_PORT_0 = (data & 0x00ff);
    DATA_PORT_8 = (data >> 8);
    _delay_us(5);
    COMMAND_PORT1 |= (1<<lcd_wr);

    COMMAND_PORT |= (1<<lcd_cs); //деактивируем чип
}

```

Всё, мы реализовали все необходимые низкоуровневые функции, теперь можно переходить непосредственно к инициализации. Перед тем как перейти к инициализации, надо сказать, что *напряжение питания ЖК-ячейки постоянно изменяет свою полярность, сделано это для того, чтобы избежать явлений гидролиза и диссоциации сложных органических соединений, из которых состоит жидкокристаллический материал.*

```

void Init_ILI9341(void)
{
    //конфигурируем все используемые выводы как выходы
    DATA_DDR_0 = 0xFF;
    DATA_DDR_8 = 0xFF;
    COMMAND_DDR = 0xFF;
    COMMAND_DDR1 = 0x01;

    COMMAND_PORT |= (1<<lcd_res);

    TFT_Send_Cmd(0x01); //Software Reset
    _delay_ms(1000);

    //Power Control 1
    TFT_Send_Cmd(0xC0); //задаём градацию серого цвета
    TFT_Write_Data(0x25);

    //Power Control 2
    TFT_Send_Cmd(0xC1); //настройка step up преобразователя
    TFT_Write_Data(0x11);

    //VCOM Control 1
    TFT_Send_Cmd(0xC5); //контрастность определяется разностью VCOMH - VCOML = 5.2V
    TFT_Write_Data(0x2B); //VCOMH = 3.825
    TFT_Write_Data(0x2B); //VCOML = -1.375

    //VCOM Control 2
    TFT_Send_Cmd(0xC7); //на Vcom по сути ШИМ, а тут мы задаем offset для него
    TFT_Write_Data(0x86); //VML=58 VMH=58

    //Memory Access Control
    TFT_SetOrientation(orient); //выбираем ориентацию дисплея

    //COLMOD: Pixel Format Set

```

```

TFT_Send_Cmd(0x3A);          //один пиксель будет кодироваться 16 битами
TFT_Write_Data(0x05);

//Frame Rate Control
TFT_Send_Cmd(0xB1);
TFT_Write_Data(0x00);
TFT_Write_Data(0x18);      //Frame Rate 79Hz

//Display Function Control
TFT_Send_Cmd(0xB6);
TFT_Write_Data(0x0A);
TFT_Write_Data(0x82);      //восьмой бит определяет нормальный цвет кристалла белый - 1, черный - 0,
TFT_Write_Data(0x27);

// Sleep Out
TFT_Send_Cmd(0x11);

_delay_ms(120);

//Display On
TFT_Send_Cmd(0x29);
}

```

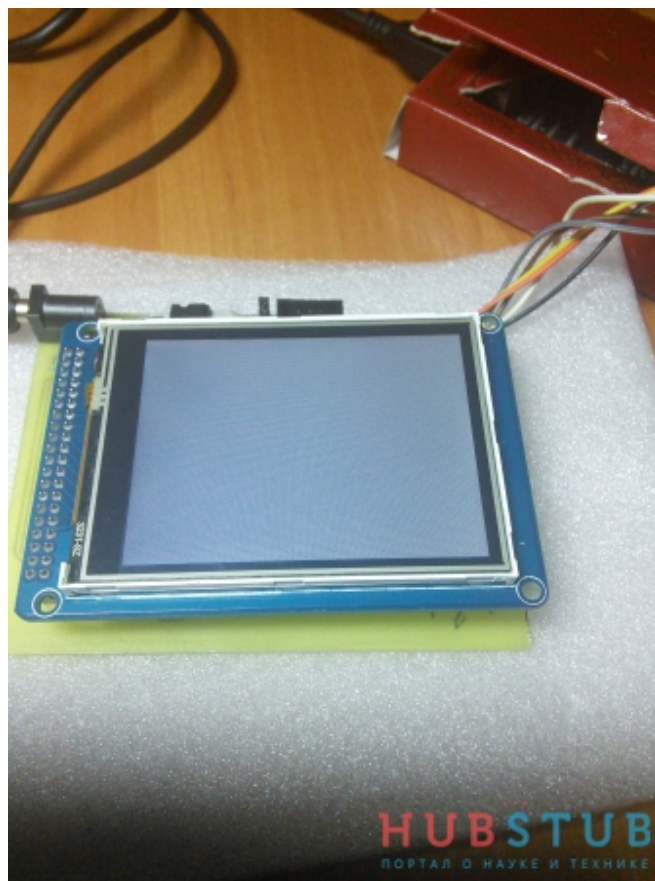
В функции инициализации есть функция TFT_SetOrientation(2), которая определяет порядок отрисовки, MAX_X и MAX_Y объявлены глобально и инициализированы нулями.

```

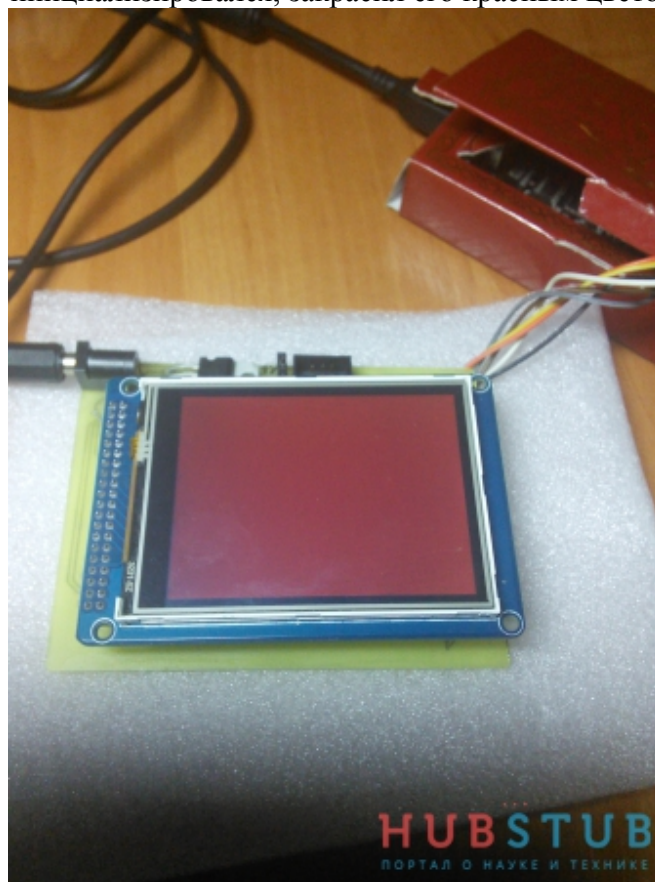
void TFT_SetOrientation(uint8_t orient)
{
    TFT_Send_Cmd(0x36);
    switch (orient)
    {
        case 0: TFT_Write_Data(0x48);
                break;
        case 1: TFT_Write_Data(0x28);
                break;
        case 2: TFT_Write_Data(0x88);
                break;
        case 3: TFT_Write_Data(0xE8);
                break;
    }
    if (orient == 0 || orient == 2)
    {
        MAX_X = 239;
        MAX_Y = 319;
    }
    else
    {
        MAX_X = 319;
        MAX_Y = 239;
    }
}

```

А вот как выглядит дисплей после инициализации.

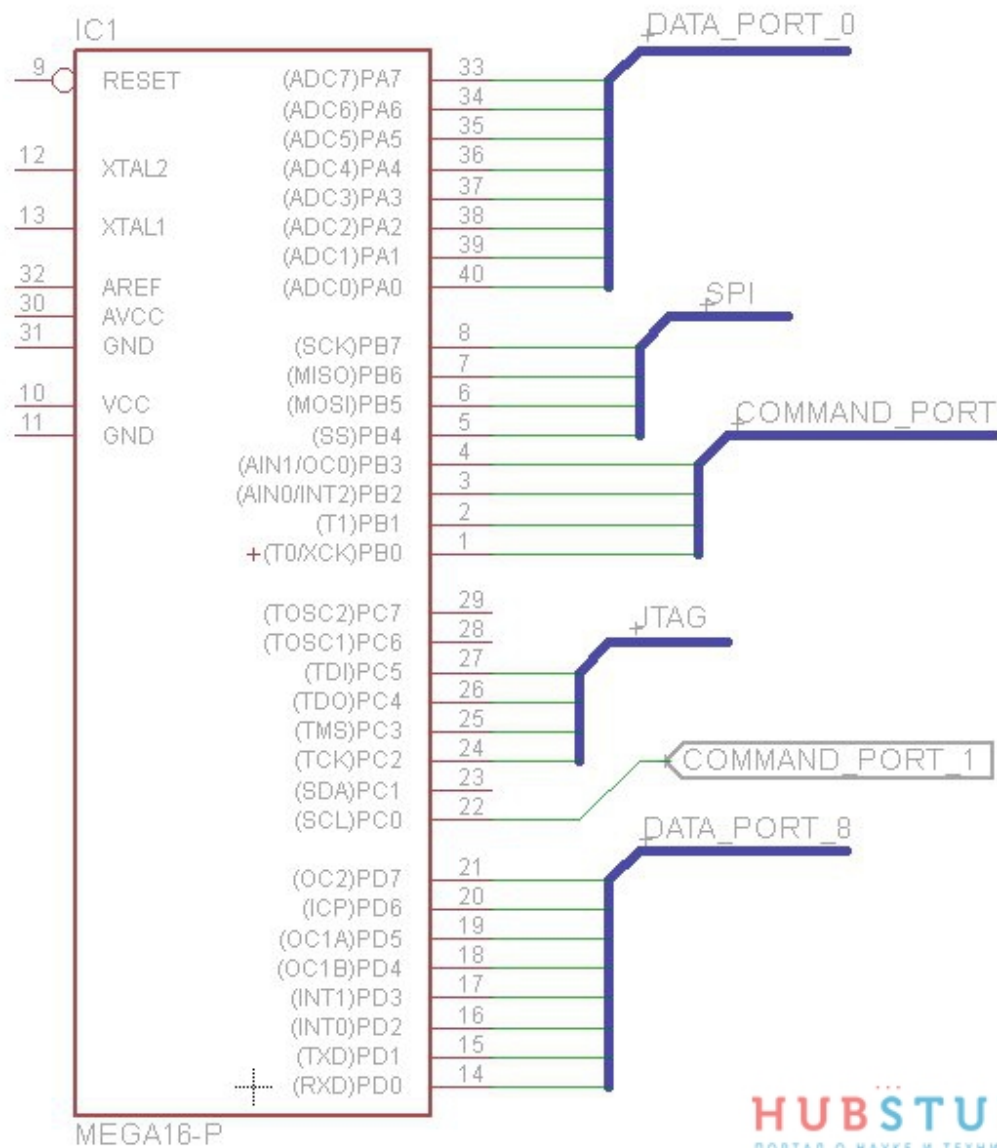


Чтобы понять, что он точно инициализировался, закрасил его красным цветом.



Как это сделать буду описывать [в следующей статье](#).

Теперь что касается схемы подключения так, как распиновка 40 выводных дисплеев SSD1289 и **ILI9341** одинаковая, схему подключения оставил без изменения и выглядит она так.



Таким образом, можно сделать одну отладочную плату для обоих дисплеев и ещё важно чтобы напряжение питания МК и дисплея было 3.3 вольта, при питании от 5 вольт дисплей не запустился. Подсветку дисплея тоже запитал от 3.3 вольта через резистор 470м.

А на этом всё, [в следующей](#) статье мы рассмотрим как выводить символы на дисплей и там же будет проект для **Atmega16** в AtmelStudio6.2.

Для желающих быстро проверить работает ли их дисплей, оставляю тут прошивку для **Atmega16**, которая заливает дисплей разными цветами [ili9341_i8080.hex](#) [3,54 Kb] (скачиваний: 181).

Недавно приобрёл пару дисплеев [тут](#).

Источник: hubstub.ru

[Статья](#)

Похожие статьи

27-05-2015, 11:40

[Инициализация LCD дисплея 1602A, с управляющим контроллером ks0066U.](#)

[Дисплеи](#)

8-04-2015, 13:51

[Подключение TFT дисплея по FSMC к STM32F103VET6 на примере SSD1289.](#)

[Дисплеи](#) / [STM32](#)

31-03-2015, 00:25

[Вывод картинки на TFT дисплей с SD карты на примере SSD1289.](#)

[Дисплеи](#)

20-03-2015, 18:35

[Вывод картинки на TFT дисплей с SD карты на примере SSD1289\(преобразование картинки в массив\)](#)

[Дисплеи](#)

12-02-2015, 00:19

[Инициализация TFT дисплея на примере SSD1289 для AVR.](#)

[Дисплеи](#)

комментарии

20

Оставить комментарий

20 Комментариев

HubStub

 Alexandr Pirs ▾

 Рекомендовать

 Поделиться

Новое в начале ▾



Присоединиться к обсуждению...



Schneider taylor • месяц назад

..., не стал изменять традициям и выбрал i8080....

Как выбрали? Ноги IM0-IM3 на модуле не выведены, а на контроллере есть!

Следовательно производителями этого экрана жестко установлен! Другой вариант не включишь.

Да и еще в даташите четко по английски написано что есть режим rgb у этого экрана. На контроллере есть входы HSYNC VSYNC... Получается можно видеокамеру подключить при желании! Но на этом модуле входа эти тоже не оформлены.

При дальнейшем чтении мануала вижу что мотололовский M6800 в этом контроллере отсутствует! Поправьте текст.

А два варианта i8080 + сериал есть, но как я сказал выше видимо не переключишь, да и ноги контроллера отвечающего за сериал не выведены на разъем.

^ | ▾ • Ответить • Поделиться ›



Сергей • 3 месяца назад

Вы могли бы привести полностью пример вашей программы, инициализирующей дисплей ILI9341_16, читающий его ID и окрашивающий экран в красный цвет? Я подключаю его через шилд к Ардуино Меге 2560 и 2560 R3, но он не хочет работать, хотя экран светится.

Перепробовал кучу библиотек и примеров - не помогло. Ваш пример заинтересовал - думаю, позволит определить пошагово, где происходит глюк. Но мне не понятно, где и как определяются COMMAND_PORT, COMMAND_PORT1, PORTA, PORTB, PORTD и другие.

Прошу выложить пример полностью. Заранее благодарен!

PS Экран 320x240

^ | ▾ • Ответить • Поделиться ›



Александр Чернобровка → Сергей • 2 месяца назад

**Сергей**

я то же никак не могу понять как его подключать по той схеме
у вас что то получилось?

^ | v • Ответить • Поделиться ›

**Сергей** → Александр Чернобривко • 2 месяца назад

Мне удалось заставить работать два комплекта из трех, но для этого пришлось дополнительно припаять несколько проводников на шильд и изменить библиотеку обработки нажатий. Давайте ваш емейл, напишу подробнее.
Мой sergeclause@mail.ru

^ | v • Ответить • Поделиться ›

**Александр Чернобривко** → Сергей • 2 месяца назад

BEST-83154@yandex.ru

^ | v • Ответить • Поделиться ›

**Василий** • 4 месяца назад

Могу также выслать свой 2.8" на ili9341. Только он 8-битный. Запустить не получается.

^ | v • Ответить • Поделиться ›

**александр** • 4 месяца назад

извиняюсь!

можете мне помочь с подключением?

если не затруднит можно в протеусе показать?

^ | v • Ответить • Поделиться ›

**максим фил** • 10 месяцев назад

Да, перечитал даташит, вы полностью правы, работает от 2.7в. Но вопрос нейтральности к 5в у дисплея остается открытым

^ | v • Ответить • Поделиться ›

**HubStub** Модератор → максим фил • 10 месяцев назад

Думаю этот вопрос Вам стоит задать производителю дисплея.

Вы наверное хотели написать толерантности к 5 вольтам, а не нейтральности)

^ | v • Ответить • Поделиться ›

**максим фил** → HubStub • 10 месяцев назад

Ну можно и так)))

^ | v • Ответить • Поделиться ›

**HubStub** Модератор → максим фил • 10 месяцев назад

SSD1289 без проблем работает от 5 вольт, а ILI9341 только от 3.3 о чем написал в конце статьи.

.....ещё важно чтобы напряжение питания МК и дисплея было 3.3 вольта, при питании от 5 вольт дисплей не запустился.

^ | v • Ответить • Поделиться ›

**максим фил** → HubStub • 10 месяцев назад

Дело в том, что я заказал дисплей и в описании указано, что нейтрален к 5в

поэтому сей момент мне и непонятен...

^ | v • Ответить • Поделиться ›



максим фил • 10 месяцев назад

Дико извиняюсь, у вас обычная atmega16 или atmega16L, просто у меня atmega 16A и она вроде только от 4,5 вольт работает, как вы контроллер от 3,3 запитали? И вроде пишут что они 5в нейтральны ... в смысле дисплеи

^ | v • Ответить • Поделиться ›



HubStub Модератор ➔ максим фил • 10 месяцев назад

У меня обычная ATmega16A.

Вот выдержка из даташита

Operating Voltages

2.7 - 5.5V

^ | v • Ответить • Поделиться ›



Виктор • год назад

Спасибо, разобрался. Он такой же как у вас, под 16 бит.

^ | v • Ответить • Поделиться ›



HubStub Модератор ➔ Виктор • год назад

В описании должна быть такая строчка
управление TFT по протоколу 6800, 8080, поддержка FSMC.
Не за что.

^ | v • Ответить • Поделиться ›



Виктор • год назад

И ещё- как определить какой интерфейс у дисплея? На сайте где заказывал написан SPI, но в посылке нет никакой документации.

^ | v • Ответить • Поделиться ›



Виктор • год назад

Спасибо за ответ, но мне всё равно не понятно. Я хочу подключить дисплей по SPI. Вижу на дисплее ноги для сигналов RS, WR, RD, CS, F_CS, REST. RS-это команда/данные, WR-запись(SCK с контроллера), RD мне не надо (читать не буду), CS-выбор дисплея, REST-сброс дисплея, а куда мне прицепить MOSI?

^ | v • Ответить • Поделиться ›



Виктор • год назад

Здравствуйте. Интересует назначение ног дисплея RS,WR,RD,F_CS. Документацию искал, но увы.....

^ | v • Ответить • Поделиться ›



HubStub Модератор ➔ Виктор • год назад

Вот в этой статье есть описание выводов дисплея.

<https://hubstub.ru/programm...>

^ | v • Ответить • Поделиться ›

Как рассчитать разность фаз по фигуре Лиссажу.

2 комментария • 2 года назад



HubStub — Спасибо, исправил.

STM32 АЦП. » Хабстаб

6 комментариев • 2 года назад



Arthur Pirozhkov — Невнятно написано про группы и каналы. Что такое длинна последовательности? Количество групп? ...

Как работает компаратор на операционном усилителе(ОУ).

2 комментария • 2 года назад



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

Подключение sd карты к микроконтроллеру.

6 комментариев • 2 года назад



Александр Костик — подскажите пожалуйста, в чем может быть проблема при такой форме записи <http://dpaste.com/0BD905T> я считываю

Статьи раздела

[Вывод символов и строк на TF...](#)

[Вывод символов и строк на LC...](#)

[Инициализация LCD дисплея 1...](#)

[Инициализация TFT дисплея ...](#)

[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...](#)

[Рисуем свой символ на LCD ди...](#)

[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