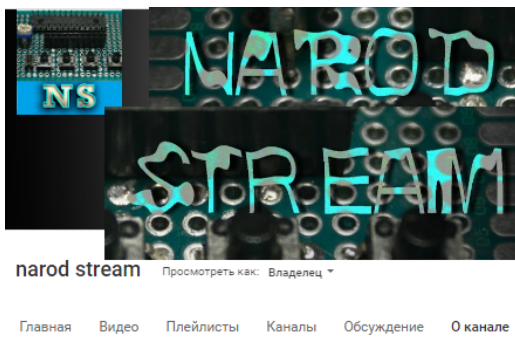


# Сайт narodstream.ru

создан в поддержку  
канала YouTube

## NAROD STREAM







## Рубрики


- [Uncategorized](#)
- [Программирование AVR](#)
- [Программирование STM32](#)

## Свежие записи

- [STM32. Урок 94. DS18B20. Несколько датчиков на одной шине. Часть 1](#)
- [Ограничение доступа к сайту за чрезмерную активность](#)
- [STM32. Урок 93. LAN. W5500. HTTP Server. Сокеты. Часть 2](#)
- [STM32. Урок 93. LAN. W5500. HTTP Server. Сокеты. Часть 1](#)
- [STM32. Урок 92. Датчик температуры DS18B20. Часть 3](#)

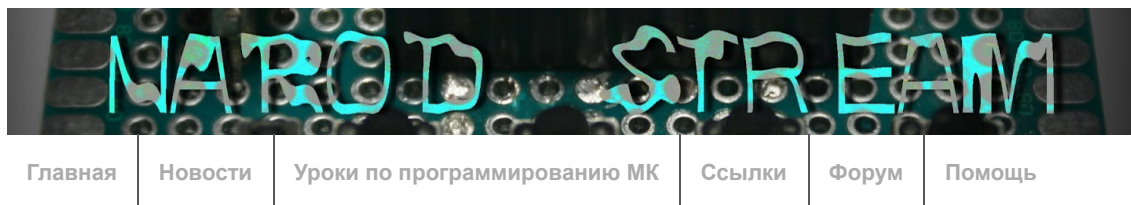
## Последние ответы на форуме

-  [Narod Stream](#) в [Программирование МК STM32](#)  
2 дн., 3 час. назад
-  [Mihail](#) в [Программирование МК STM32](#)  
2 дн., 4 час. назад
-  [Dmitriy](#) в [Программирование МК AVR](#)  
2 нед., 5 дн. назад
-  [nsk](#) в [Программирование МК STM32](#)  
2 нед., 6 дн. назад

-  [Narod Stream](#) в [Программирование МК STM32](#)  
3 нед. назад

## Свежие комментарии

- [Narod Stream](#) к записи [AVR Урок 4](#).  
[Смотрим результаты работы](#)
- [Narod Stream](#) к записи [STM Урок 44](#).  
[SDIO. FATFS](#)
- Виктор к записи [AVR Урок 4](#). [Смотрим результаты работы](#)
- [Narod Stream](#) к записи [STM Урок 56](#).  
[System Workbench. Подключаем библиотеку BSP. Часть 1](#)
- [3k](#) к записи [STM Урок 56](#). [System Workbench. Подключаем библиотеку BSP. Часть 1](#)



Просмотров:

[Главная](#) > [AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 2](#)

### Урок 34 Часть 2

## Дисплей TFT 240×320 8bit

В [предыдущей части](#) нашего урока мы познакомились с модулем графического дисплея на контроллере **ILI9341**, Также мы создали проект, добавили в него ряд макроподстановок и создали функцию инициализации.

Правда, у данной функции пока ещё нет тела, но этот пробел мы сегодня устраним.

Но чтобы продолжить писать функцию инициализации, нам необходима будет ещё функция, так как чтобы управлять контроллером дисплея, нам нужно сначала инициализировать как-то уровни на его ножках. Напишем её выше.

```
//  
void port_ini(void)
```

### Мета

- [Регистрация](#)
- [Войти](#)
- [RSS записей](#)
- [RSS комментариев](#)
- [WordPress.org](#)

#### Уроки по программированию МК

[Программирование МК AVR](#)

[Программирование МК STM32](#)

[Программирование МК PIC](#)

[Тесты устройств и аксессуаров](#)

```
{
DATA_PORT=0x00;
DATA_DDR=0xFF;//Шина данных на выход
COMMAND_DDR=0x1F;//Командные лапки также все на выход
}
//_____
```

Вызовем данную функцию в функции инициализации

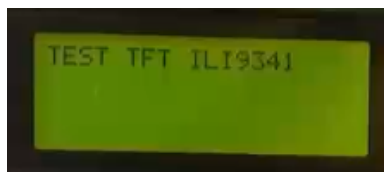
```
void TFT9341_ini(void)
{
    port_ini();
}
```

Перейдём в главный модуль и немного напишем в главную функцию

```
int main(void)
{
    I2C_Init();//Инициализируем TWI
    LCD_ini();//инициализируем символьный дисплей
    clearlcd();//очистим символьный дисплей
    setpos(0,0);
    str_lcd("TEST TFT ILI9341");
    TFT9341_ini();
}
```

Мы включили шину I2C, инициализировали и очистили символьный дисплей, спозиционировали курсор и вывели первое сообщение на экран данного дисплея, а затем вызвали и функцию инициализации нашего графического дисплея.

Пока мы, конечно, сделали только заготовку нашего проекта, но, тем не менее, давайте соберём проект и прошьём контроллер для того, чтобы пока проверить работоспособность символьного дисплея

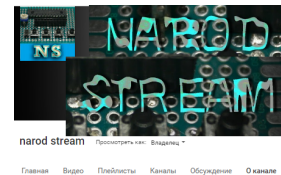


Вернёмся в модуль ili9341.c и напишем ещё две функции для отправки команд и данных в контроллер дисплея

```
//_____
void TFT9341_SendCommand(unsigned char cmd)
{
    CD_COMMAND;//лапка в состоянии посылки команды
    RD_IDLE;//отключим чтение
    CS_ACTIVE;//выбор дисплея
    DATA_PORT=cmd;
    WR_STROBE;
    CS_IDLE;
}
//_____
void TFT9341_SendData(unsigned char dt)
```



Заходите на  
канал Narod  
Stream



narod stream "Просмотреть канал" "Видео"

Главная Видео Плейлисты Каналы Обсуждение О канале

## Архивы

- Октябрь 2017
- Сентябрь 2017

```

{
    CD_DATA; //лапка в состоянии посылки данных
    RD_IDLE; //отключим чтение
    CS_ACTIVE; //выбор дисплея
    DATA_PORT=dt;
    WR_STROBE;
    CS_IDLE;
}
//_____

```

В данных функциях также всё ясно по комментариям. Но немного все равно попытаюсь что-то объяснить.

Сначала мы ножку RS выставляем в уровень, необходимый для нашей задачи (команда или данные), затем отключим чтение, а то вдруг оно включено, иницируя на соответствующей ножке высокий уровень, далее опустим Chip Select, затем на параллельной шине данных выставим уровни, соответствующие значению посылаемого байта, сделаем соответствующий импульс на ножке WR, чтобы данные записались, и поднимем ножку Chip Select.

Далее добавим функцию для полной перезагрузки дисплея

```

//_____
void TFT9341_reset(void)
{
    CS_IDLE;
    WR_IDLE;
    RD_IDLE;
    RESET_ACTIVE;
    _delay_ms(2);
    RESET_IDLE;
    CS_ACTIVE;
    TFT9341_SendCommand(0x01); //Software Reset
    for (uint8_t i=0; i<3; i++) WR_STROBE;
    CS_IDLE;
}
//_____

```

Начало функции похоже на начало предыдущих функций, но не совсем.

Мы поднимаем ножку выбора, поднимаем ножки записи и чтения, затем делаем аппаратную перезагрузку, формируя импульс на соответствующей ножке, сначала мы её опускаем, затем ждём 2 миллисекунда, затем поднимаем, далее опускаем ножку выбора и отправляем команду для программной перезагрузки дисплея. У данной команды параметров нет, поэтому мы их и не отправляем. Затем мы производим три импульса ножкой записи и поднимаем ножку выбора.


Вызовем нашу функцию в функции **TFT9341\_ini**, применив после неё задержку, чтобы контроллер дисплея успел до конца проинициализироваться

```

port_ini();
TFT9341_reset();
_delay_ms(1000);

```

- Август 2017
- Июль 2017
- Июнь 2017
- Май 2017
- Апрель 2017
- Март 2017
- Февраль 2017
- Январь 2017
- Декабрь 2016
- Ноябрь 2016



31 ДЕНЬ	88 083
	9 064
07 ДНЕЙ	22 948
	2 994
24 ЧАСА	3 036
	713
СЕГОДНЯ	1 398
	381
НАПЛИШИ	84
	24

Напишем теперь функцию элементарной передачи байта в дисплей

```
//-----  
void TFT9341_Write8(unsigned char dt)  
{  
    DATA_PORT=dt;  
    WR_STROBE;  
}  
//-----
```

Ну здесь всё понятно, я думаю.

Теперь создадим функцию чтения из регистра контроллера дисплея. Данная функция, ясное дело, будет возвращать результат в виде 32-битной величины. Для начала добавим в эту функцию две переменные

```
//-----  
unsigned long TFT9341_ReadReg(unsigned char r)  
{  
    unsigned long id;  
    unsigned char x;  
}  
//-----
```

---

Далее мы в данной функции опустим кнопку выбора, также иницилируем уровень для команды и передадим байт адреса регистра в контроллер дисплея

```
unsigned char x;  
CS_ACTIVE; //выбор дисплея  
CD_COMMAND; //лапка в состоянии посылки команды  
TFT9341_Write8(r);
```

Затем воспользуемся макросом setReadDir, который установит все ножки данных контроллера дисплея на выход

```
TFT9341_Write8(r);  
setReadDir();
```

Потом установим высокий уровень на ножке RS, соответствующий работе с данными и подождём немного

```
setReadDir();  
CD_DATA;  
_delay_us(50);
```

Затем переведём в активное состояние ножку чтения и подождём ещё немного

```
_delay_us(50);  
RD_ACTIVE;
```

```
_delay_us(5);
```

После всего этого на порте данных контроллера дисплея у нас должен образоваться байт данных. Заберём его в переменную

```
_delay_us(5);
x=DATA_PIN;
```

Это будет самый старший байт. Поднимем ножку чтения и перепишем данный в нашу 32-разрядную переменную и сдвинем её на 8 пунктов, тем самым наш считанный байт станет немного старше, сместившись вправо. И в последствии мы также и поступим со всеми остальными байтами

```
x=DATA_PIN;
RD_IDLE;
id=x;
id<<=8;
```

Напишем то же самое для остальных трёх байтов

```
id<<=8;
RD_ACTIVE;
_delay_us(5);
x=DATA_PIN;
RD_IDLE;
id|=x;
id<<=8;
RD_ACTIVE;
_delay_us(5);
x=DATA_PIN;
RD_IDLE;
id|=x;
id<<=8;
RD_ACTIVE;
_delay_us(5);
x=DATA_PIN;
RD_IDLE;
id|=x;
```

Особый случай будет для регистра EFh. Здесь мы сдвигаем всё ещё раз и принимаем ещё один байт

```
id|=x;
if(r==0xEF)
{
    id<<=8;
    RD_ACTIVE;
    _delay_us(5);
    x=DATA_PIN;
    RD_IDLE;
    id|=x;
}
```

Далее мы поднимаем ножку выбора и воспользуемся макросом **setWriteDir**, который установит включит все

ножки данных на вход

```
id|=x;
}
CS_IDLE;
setWriteDir();
```

Немногождём для стабилизации и вернём считанные данные

```
setWriteDir();
_delay_us(150); //stabilization time
return id;
}
```

Для того, чтобы узнать, с тем ли дисплеем мы и правильно ли работаем, мы попробуем считать его идентификатор. Для этого в начале файла добавим переменную

```
#include "ili9341.h"
//_____
unsigned long dtt=0;
```

Для того, чтобы считать идентификатор, мы воспользуемся данными регистра **D3h**

8.3.23. Read ID4 (D3h)

D3h	RDID4 (Read ID4)												
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	1	XX	1	1	0	1	0	0	1	1	D3h
1 <sup>st</sup> Parameter	1	1	1	XX	X	X	X	X	X	X	X	X	X
2 <sup>nd</sup> Parameter	1	1	1	XX	0	0	0	0	0	0	0	0	00h
3 <sup>rd</sup> Parameter	1	1	1	XX	1	0	0	1	0	0	1	1	93h
4 <sup>th</sup> Parameter	1	1	1	XX	0	1	0	0	0	0	0	1	41h

Считаем данный регистр в функции инициализации дисплея

```
_delay_ms(1000);
dtt=TFT9341_ReadReg(0xD3);
```

Опустим ножку выбора и, применив функцию `sprintf`, отобразим считанную из регистра величину на символьном дисплее на второй строке

```
dtt=TFT9341_ReadReg(0xD3);
CS_ACTIVE;
setpos(0,1);
sprintf(str,"0x%08lx",dtt);
str_lcd(str);
```

Соберём код, прошьём контроллер.  
А вот и результат



Идентификатор считан, значит мы работаем именно с тем контроллером и работаем с параллельной шиной правильно, так как регистры читаются.

В **следующей части** нашего занятия по программированию графического дисплея мы напишем функцию поворота дисплея в горизонтальное и вертикальное состояние.

Также попытаемся заполнить регистры контроллера дисплея определенными данными для инициализации дисплея.

Предыдущая  
часть

Программирование  
МК AVR

Следующая  
часть

### **Техническая документация на контроллер дисплея ILI9341**

Программатор и символьный дисплей LCD 20×4 можно приобрести здесь:

Программатор (продавец надёжный) **USBASP USBISP 2.0**

**Дисплей LCD 20×4**

**Смотреть ВИДЕОУРОК (нажмите на картинку)**





Назначение перевода Я хочу поддержать проект Narod Stream

Сумма  руб.

Отправить

## Добавить комментарий

Ваш e-mail не будет опубликован. Обязательные поля помечены \*

Имя \*

E-mail \*

Сайт

+ семь = 15 

Отправить комментарий

[Главная](#) | [Новости](#) | [Уроки по программированию МК](#)  
| [Программирование микроконтроллеров AVR](#)  
| [Программирование микроконтроллеров STM32](#)  
| [Программирование микроконтроллеров PIC](#) | [Тесты устройств](#)  
и аксессуаров  
| [Ссылки](#) | [Форум](#) | [Помощь](#)



Google



© 2017 Narod Stream

