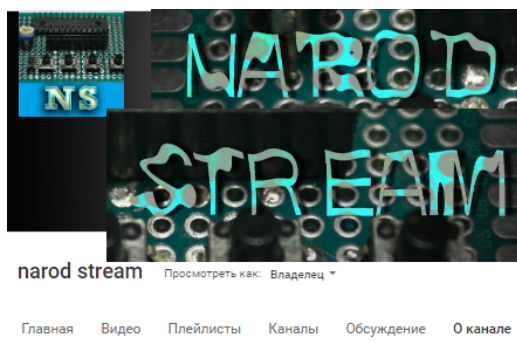


# Сайт narodstream.ru

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](#)

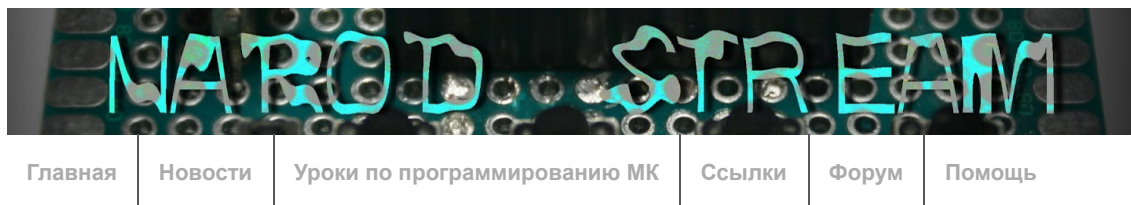
## Последние ответы на форуме

-  [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: 110

[Главная](#) > [AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 1](#)

### Урок 34 Часть 1


# Дисплей TFT 240×320 8bit

## Мета

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

## Уроки по

Яндекс.Директ




**[Микроконтроллеры на языке C](#)**

Научись профессионально программировать микроконтроллеры на языке C

[mcu-c.ru](#)

Яндекс.Директ



**[Бесплатный урок по Simatic Step 7](#)**

Научитесь настройке ПЛК Siemens и другого оборудования в Simatic Step 7.

[Основной курс](#) [Отзывы](#) [Об авторе](#) [Содержание курса](#)

[step7-kurs.ru](#)

Сегодня мы вернёмся к **Atmega8**, так как что-то у меня случилось с моей Atmega328. Но для наших задач нам вполне подойдёт и восьмая серия.

ие МК PIC

Тесты устройств и аксессуаров

Сегодня мы попробуем подключить к нашему контроллеру дисплей уже графический цветной разрешением 320 на 240 точек, управляемый по восьмибитному параллельному интерфейсу, Дисплей выполнен с помощью модуля и управляется контроллером ILI9341.

Вообще, у данного контроллера дисплея существует несколько режимов подключения к управляющему микроконтроллеру, в том числе есть и SPI, но мне попался дисплей именно с прараллельным способом подключения, о чём я нисколько не жалею. С таким интерфейсом также немало дисплеев, но что обидно, очень мало информации по реализации кода для дисплеев именно с таким интерфейсом. Поэтому наша задача – данный информационный пробел устранить на корню.

Вот перечень режимов подключения

- Select the MCU interface mode

IM3	IM2	IM1	IM0	MCU-Interface Mode	DB Pin in use	
					Register/Content	GRAM
0	0	0	0	80 MCU 8-bit bus interface I	D[7:0]	D[7:0]
0	0	0	1	80 MCU 16-bit bus interface I	D[7:0]	D[15:0]
0	0	1	0	80 MCU 9-bit bus interface I	D[7:0]	D[8:0]
0	0	1	1	80 MCU 18-bit bus interface I	D[7:0]	D[17:0]
0	1	0	1	3-wire 9-bit data serial interface I	SDA: In/OUT	
0	1	1	0	4-wire 8-bit data serial interface I	SDA: In/OUT	
1	0	0	0	80 MCU 16-bit bus interface II	D[8:1]	D[17:10], D[8:1]
1	0	0	1	80 MCU 8-bit bus interface II	D[17:10]	D[17:10]
1	0	1	0	80 MCU 18-bit bus interface II	D[8:1]	D[17:0]
1	0	1	1	80 MCU 9-bit bus interface II	D[17:10]	D[17:9]
1	1	0	1	3-wire 9-bit data serial interface II	SDI: In SDO: Out	
1	1	1	0	4-wire 8-bit data serial interface II	SDI: In SDO: Out	

Включить тот или иной режим удаётся благодаря управлению уровнем на ножках IM3:IM0.

Наш режим, которым мы будем пользоваться — самый первый. Чтобы его включить, нужно на всех данных ножках инициировать низкоуровневой сигнал.

Судя по технической документации, контроллер ILI9341 может управлять 262000 разновидностями цветов.

Вот схема подключения дисплея к контроллеру

искать здесь ...

Фильтровать


Нужно программировать контроллеров? ✕

[festo.com/обучение](https://festo.com/obucheniye)

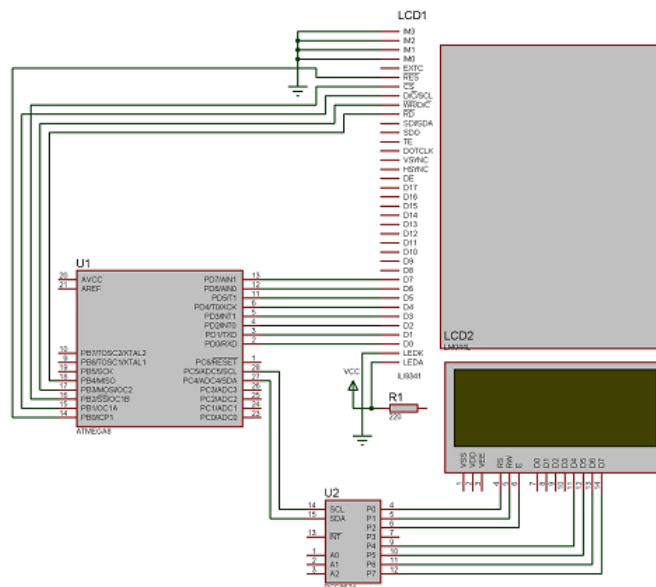


Яндекс.Директ

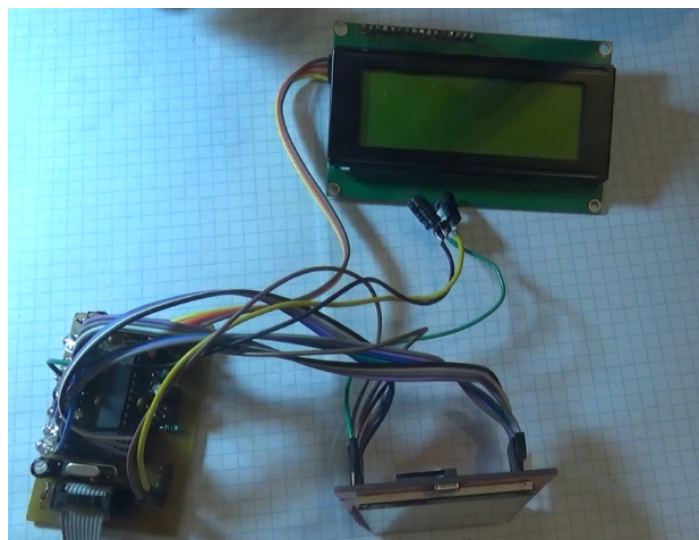
Заходите на канал Narod Stream



[Главная](#) [Видео](#) [Плейлисты](#) [Каналы](#) [Обсуждение](#) [О канале](#)



А вот так схема выглядит на практике



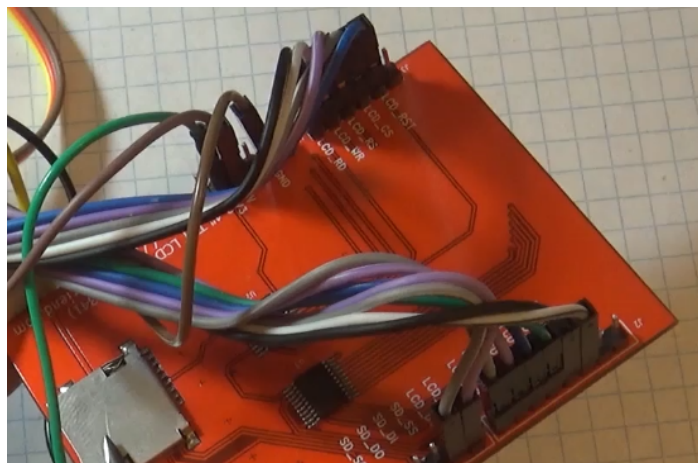
Мы видим также простенький стабилизатор, висящий на проводках, состоящий из микросхемы и конденсатора. Данный стабилизатор служит для подачи питания 3,3 вольта на модуль дисплея, так как такое напряжение для него также требуется. Некоторую информацию, возвращённую из контроллера дисплея мы будем отслеживать с помощью символического дисплея 20x4, подключенного по интерфейсу **i2c**, который также мы видим на картинке.

А вот так выглядит модуль дисплея снизу

## Архивы

- [Октябрь 2017](#)
- [Сентябрь 2017](#)
- [Август 2017](#)
- [Июль 2017](#)
- [Июнь 2017](#)
- [Май 2017](#)
- [Апрель 2017](#)
- [Март 2017](#)
- [Февраль 2017](#)
- [Январь 2017](#)
- [Декабрь 2016](#)
- [Ноябрь 2016](#)

	88 083
31 ДЕНЬ	9 064
07 ДНЕЙ	22 948
24 ЧАСА	3 036
СЕГОДНЯ	1 363
НАПЛИШИ	27



Мы видим ещё на борту данного модуля картоприёмник для карты Micro SD, которая подключается по интерфейсу SPI. Также мы видим, что все контакты у дисплея подписаны, поэтому с подключением, я думаю сложностей не возникнет.

Первые снизу справа четыре ножки служат для того, чтобы общаться с картой памяти, поэтому они у нас свободные. Используем мы следующие ножки дисплея:

**LCD\_RST** — ножка для перезагрузки дисплея;

**LCD\_CS** — Chip Select (выбор), активный уровень низкий;

**LCD\_RS** — данные/команда, для передачи данных высокий уровень, для передачи команды — низкий;

**LCD\_WR** — включение режима записи, активный уровень низкий;

**LCD\_RD** — включение режима чтения, активный уровень низкий;

**GND** — общий провод;

**5V** — питание 5 вольт;

**3V3** — питание 3,3 вольт;

**LCD\_D0:LCD\_D7** — параллельная 8-разрядная шина данных.

Управление контроллером дисплея происходит посредством команд или другими словами отправкой в определённый регистр определённых величин. В технической документации очень подробно расписаны все регистры, причём есть и краткий перечень команд регистров, а есть и подробный для каждой команды. Техническая документация на контроллер ILI9341 прикреплена внизу данной страницы. Сначала мы отправляем адрес регистра, а затем шлём параметры команды. Причём количество параметров в различных командах также различается, что накладывает некоторые трудности для создания универсальной функции отправки команды. Но ничего, что-нибудь придумаем.

У нас создан проект **TFT9341**. Проект вполне стандартный, подключены к нему наши библиотеки, написанные на прошлых занятиях — это **twi** и **lcdtwi**.

Также мы создадим и подключим к проекту ещё 2 файла для библиотеки уже непосредственно нашего нового дисплея — это файлы **ili9341.h** и **ili9341.c**.

Вот так выглядит главный заголовочный файл **main.h**

```

#ifndef MAIN_H_
#define MAIN_H_
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <avr/pgmspace.h>
#include "twi.h"
#include "lcdtwi.h"
#include "ili9341.h"
#endif /* MAIN_H_ */

```

В файле ili9341.h подключим

```

#ifndef ILI9341_H_
#define ILI9341_H_
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "twi.h"
#include "lcdtwi.h"

```

Ну и, как вы все знаете, чтобы любой дисплей подключить и его заставить что-то показывать, необходимо пройти определённую процедуру инициализации. По большому счёту это относится не только к дисплеям. В файле ili9341.c подключим заголовочный файл и напишем каркас для функции инициализации дисплея

```

#include "ili9341.h"
//-----
void TFT9341_ini(void)
{
}

```

Не забываем на данную функцию добавить прототип в хедер-файле, а также напишем некоторые макроподстановки, которые нам потребуются в процессе работы с дисплеем

```

#include "lcdtwi.h"
#define swap(a,b) {int16_t t=a;a=b;b=t;}
#define DATA_DDR DDRD
#define DATA_PORT PORTD
#define DATA_PIN PIND
#define COMMAND_DDR DDRB
#define COMMAND_PORT PORTB
#define LCD_CS 2//Chip Select
#define LCD_CD 1//Command/Data
#define LCD_WR 3//LCD Write
#define LCD_RD 4//LCD Read
#define LCD_RESET 0//LCD Reset
#define RESET_IDLE COMMAND_PORT|=(1<<LCD_RESET)
#define CS_IDLE COMMAND_PORT|=(1<<LCD_CS)
#define WR_IDLE COMMAND_PORT|=(1<<LCD_WR)

```

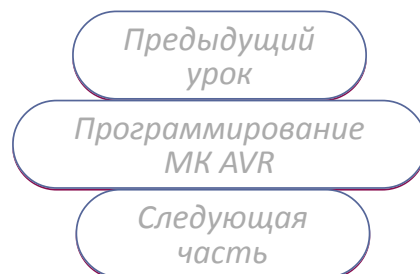
```

#define RD_IDLE COMMAND_PORT|=(1<<LCD_RD)
#define RESET_ACTIVE COMMAND_PORT&=~(1<<LCD_RESET)
#define CS_ACTIVE COMMAND_PORT&=~(1<<LCD_CS)
#define WR_ACTIVE COMMAND_PORT&=~(1<<LCD_WR)
#define RD_ACTIVE COMMAND_PORT&=~(1<<LCD_RD)
#define CD_COMMAND COMMAND_PORT&=~(1<<LCD_CD)
#define CD_DATA COMMAND_PORT|=(1<<LCD_CD)
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0x0F800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF
#define setReadDir() DATA_DDR=0x00
#define setWriteDir() DATA_DDR=0xFF
#define WR_STROBE {WR_ACTIVE;WR_IDLE;}
void TFT9341_ini(void);
#endif /* ILI9341_H_ */

```

Я думаю, назначение каждого макроса объяснять нет смысла, так как всё ясно из наименований. Но если вдруг кому-то что-то непонятно, то посмотрите видеоурок, кликнув по картинке внизу страницы, там объяснение более подробное.

В **следующей части** нашего занятия мы попытаемся написать ещё некоторые важные функции для работы с контроллером дисплея, а также считать его идентификатор, что позволит нам убедиться, что мы работаем именно с таким контроллером.



### **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)





Сумма	руб.
-------	------

Отправить

2 комментария на “AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 1”



Роман:

Март 16, 2017 в 6:09 пп

Добрый день. Есть экран HX8347 у него включение 8bit режима совсем по другому на IM3-IM0 0001 В Вашем коде где это можно изменить?

Ответить



admin:

Март 17, 2017 в 4:14 дп

А зачем же менять в моём коде? Напишите свой. Ну, хотя, даже если и измените, авторство никакое не предъявлю. Так что дерзайте!

Ответить

## Добавить комментарий

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

[illegible]



Pa... ▾

▾ ▾

Имя \*

E-mail \*

Сайт

x 4 = 12 ↺

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

