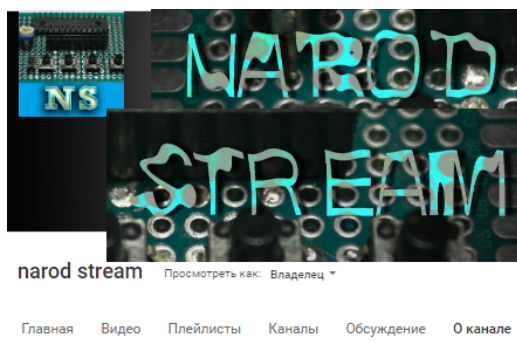


Сайт narodstream.ru

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](#)

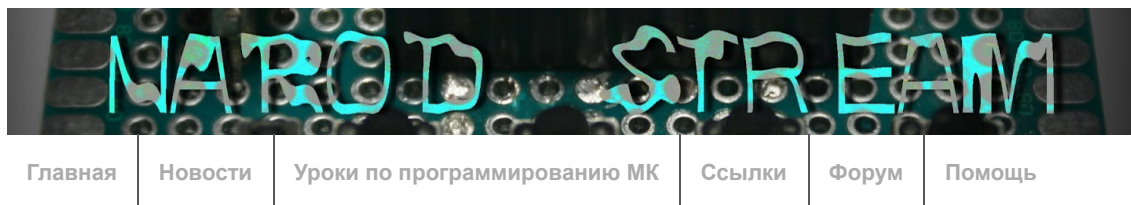
Последние ответы на форуме

-  [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: 43

[Главная](#) > [AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 4](#)




Урок 34 Часть 4

Дисплей TFT 240×320 8bit

Мета

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

Уроки по

	<p>Бесплатный урок по Simatic Step 7 ×</p> <p>Научитесь настройке ПЛК Siemens и другого оборудования в Simatic Step 7.</p>	 <p>Что случилось в Минской области? ×</p> <p>Происшествия и чрезвычайные ситуации в Минской области.</p>
	<p>Atmel Studio программирование на C ×</p> <p>Программирование микроконтроллеров на языке C в среде Atmel Studio.</p> <p>О курсе Содержание Частые вопросы Другие видеокурсы</p> <p>msu-c.ru/-Новый-Видеокурс-</p>	

В [предыдущей части](#) нашего занятия по дисплею мы написали функцию ориентации дисплея, а также завершили писать функцию инициализации дисплея.

ие МК PIC

Тесты устройств и аксессуаров

Напишем теперь функцию заливки определённой области памяти данными пикселей одного цвета для последующего использования в функции заливки одним цветом прямоугольной области

```
//-----
void TFT9341_Flood(unsigned short color, unsigned
long len)
{
}
//-----
```

Первый входной параметр — это цвет пикселей, а второй длина области памяти для заливки. Адресация будет в отдельной функции.

Добавим в функцию несколько переменных, причём две из них сразу проинициализируем определённым образом

```
void TFT9341_Flood(unsigned short color, unsigned
long len)
{
    unsigned short blocks;
    unsigned char i, hi = color>>8, lo=color;
```

То есть 16 бит цвета мы здесь распределим по двум переменным.

Опустим ножку выбора и ножку команд/данных

```
unsigned char i, hi = color>>8, lo=color;
CS_ACTIVE;
CD_COMMAND;
```

Отправим цвет в соответствующий для этого регистр

```
CD_COMMAND;
TFT9341_Write8(0x2C);
CD_DATA;
TFT9341_Write8(hi);
TFT9341_Write8(lo);
```

То есть здесь мы уже не используем специальные функции для отправки команд и данных для экономии времени, здесь уже время очень важно, иначе пострадает скорость работы с дисплеем. Поэтому мы уже пользуемся функциями более низкоуровневыми без лишнего дрыгания ножками.

Продекрементируем длину области, так как отсчёт у нас происходит не с 1, а с 0

```
TFT9341_Write8(lo);
len--;
```

Разобьём нашу длину на блоки по 64 пикселя

```
len--;
blocks=(unsigned short)(len/64); //64 pixels/block
```



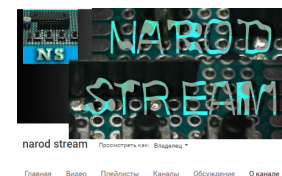

Картотека

Быстрая проверка компаний

Снижайте бизнес-риски на kartoteka.by

Получить доступ

Заходите на канал Narod Stream



Дальше условие, если младшая часть 16-битного цвета равна старшей

```
blocks=(unsigned short)(len/64);//64 pixels/block
if (hi==lo)
{
}
else
{
}
```

В теле условия цикл, равный количеству блоков

```
if (hi==lo)
{
    while(blocks-)
    {
    }
}
```

В теле цикла ещё один цикл, равный 16

```
while(blocks-)
{
    i=16;
    do
    {

WR_STROBE;WR_STROBE;WR_STROBE;WR_STROBE;//2bytes/pixe
l
        WR_STROBE;WR_STROBE;WR_STROBE;WR_STROBE;//x4
pixel
    } while (--i);
}
```

В данном цикле мы просто стробируем, отправляя один и тот же байт в контроллер дисплея, уровни байта ведь у нас на ножках порта установлен. Получается, что мы один и тот же байт отправим 128 раз, то есть если разбить на пары, то 64 пары и, следовательно, 64 пикселя. То есть при условии равенства старшего и младшего байтов цвета мы отправили столько блоков, сколько у нас есть, а за счёт того, что мы каждый раз не устанавливаем уровень на ножках порта данных, мы значительно выигрываем во времени.

Выйдя из цикла отправки всех блоков, мы отправим оставшуюся часть байтов, если у нас длина не будет делиться на 64 без остатка

```
    } while (--i);
}
//Fill any remaining pixels(1 to 64)
for (i=(unsigned char)len%64;i--;)
{
    WR_STROBE;
    WR_STROBE;
}
}
```

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](#)



31 ДЕНЬ	88 083
	9 064
07 ДНЕЙ	22 948
	2 994
24 ЧАСА	3 036
	713
СЕГОДНЯ	1 413
	385
НА ПУТИ	89
	25

Вот так. Теперь заходим в "противную часть" нашего условия, когда у нас старший и младший байт цвета будут разными. Я думаю, это будет чаще. Там также будет цикл, равный количеству полных блоков

```
else
{
    while(blocks-)
    {
    }
}
```

В цикле также цикл, равный 16, в котором мы передаём весь блок

```
while(blocks-)
{
    i=16;
    do
    {

TFT9341_Write8(hi);TFT9341_Write8(lo);TFT9341_Write8(
hi);TFT9341_Write8(lo);

TFT9341_Write8(hi);TFT9341_Write8(lo);TFT9341_Write8(
hi);TFT9341_Write8(lo);
    } while (--i);
}
```

Ну, тут понятно, раз байты не равны, передаём их по очереди.

Затем также передаём оставшуюся часть пикселей

```
    } while (--i);
}
//Fill any remaining pixels(1 to 64)
for (i=(unsigned char)len&63;i--;)
{
    TFT9341_Write8(hi);
    TFT9341_Write8(lo);
}
}
```

Ну и в конце поднимем ножку выбора

```
TFT9341_Write8(lo);
}
}
CS_IDLE;
}
```

Теперь напишем функцию записи в регистр 32-битного числа

```
//-----
void TFT9341_WriteRegister32(unsigned char r,
unsigned long d)
{
    CS_ACTIVE;
    CD_COMMAND;
    TFT9341_Write8(r);
    CD_DATA;
    _delay_us(1);
    TFT9341_Write8(d>>24);
    _delay_us(1);
    TFT9341_Write8(d>>16);
    _delay_us(1);
    TFT9341_Write8(d>>8);
    _delay_us(1);
    TFT9341_Write8(d);
    CS_IDLE;
}
//-----
```

Я думаю, тут даже объяснять ничего не надо. Мы сначала передаём адрес регистра, а затем с некоторыми задержками передаём наши данные.

Но, прежде чем отправлять в память байты, нам нужно объявить область памяти, в которую будет вся наша цепочка одинаковых пикселей отправляться. Для этого мы напомним специальную функцию

```
//-----
void TFT9341_SetAddrWindow(unsigned int x1,unsigned
int y1,unsigned int x2,unsigned int y2)
{
    unsigned long t;
    CS_ACTIVE;
    t = x1;
    t<<=16;
    t |= x2;
    TFT9341_WriteRegister32(0x2A,t); //Column Address Set
    t = y1;
    t<<=16;
    t |= y2;
    TFT9341_WriteRegister32(0x2B,t); //Page Address Set
    CS_IDLE;
}
//-----
```

Ну здесь мы в специализированные регистры **2Ah** и **2Bh** передаём наши координаты сначала начала и окончания вертикальной области, а затем горизонтальной, распределив соответственно эти координаты в четырехбайтовые величины. Отправка в регистры стандартная.

Также нам потребуется какой-то шрифт, если мы захотим выводить какой-то текст. Я с помощью определённой программы создал этот шрифт, причём написал не весь шрифт, а несколько букв, ибо памяти у нас не так много. Сделал я их высотой в 16 пикселей

font 16

Ну и далее собственно заливка всей области экрана определённым цветом

```
//-----  
void TFT9341_FillScreen(unsigned int color)  
{  
    TFT9341_SetAddrWindow(0,0,X_SIZE-1,Y_SIZE-1);  
    TFT9341_Flood(color,(long)X_SIZE*(long)Y_SIZE);  
}  
//-----
```

Мы использовали наши вышенаписанные функции, чтобы сначала передать координаты, а затем уже отправить цвет и длину всей области, получив её способом перемножения горизонтального и вертикального размеров.

Напишем на данную функцию прототип и вызовем её в main() после инициализации дисплея

```
TFT9341_ini();  
TFT9341_FillScreen(BLACK);
```

Соберём код, пошьём контроллер и посмотрим результат



Как мы видим, дисплей наш окрасился в чёрный цвет. Поменяем цвет на красный и ещё раз проверим

```
TFT9341_ini();  
TFT9341_FillScreen(RED);
```



В **следующей части** нашего урока мы напишем ещё несколько функций для работы с дисплеем и попробуем вывести некоторые примитивы на экран 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

7 комментариев на “AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 4”**Alex:**

Апрель 10, 2017 в 9:21 пп

Спасибо за видео, мне нравится. Только вот что-то у меня не пошло вроде все проверил, подсоединено все верно и программу проверил все также, но что — то не идет, не могу понять куда копать?

[Ответить](#)**admin:**

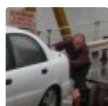
Апрель 11, 2017 в 5:07 дп

Спасибо за оценку ресурса! Возможно, другой дисплей немного, идентификатор хотя бы считывается?

[Ответить](#)**Максим:**

Май 1, 2017 в 6:19 пп

Добрый день,
спасибо за столь подробное описание.
Я пишу на CodeVisionAVR. Решил переделать код под нее.
Но вот незадача, качаю исходник, беру готовых hex, прошиваю.
Дисплей инициализируется, но вместо теста выдает вот такую картинку
Открываю проект в AtmelStudio 7
Компилирую по новой и экран перестает инициализироваться.
Подскажите в чем может быть причина?
И не совсем понял, как можно изменить номера портов Data.
С командными все понятно, они явно заданы.
А у дата портов указывается только буква порта

[Ответить](#)**admin:**

Май 2, 2017 в 5:38 дп

Добрый день!
Номера ножек совпадают с номерами битов в 8-битной переменной cmd.

[Ответить](#)**Максим:**

Май 2, 2017 в 7:55 дп

А по какой причине скопмилированный hex из
исходника может не работать?
Просто в AtmelStudio не работал. Возможно есть
какие-то нюансы при компиляции

[Ответить](#)**Максим:**[Май 2, 2017 в 7:56 дп](#)

И есть ли возможность указания номеров ножек
не по порядку?

[Ответить](#)**Максим:**[Май 2, 2017 в 7:58 дп](#)[Ответить](#)

Добавить комментарий

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

<input type="text" value="Источник"/>	
<input type="text"/>	<input type="text"/>
<input type="text"/>	
<input type="text"/>	
<input type="text"/>	
<input type="text"/>	
<input type="text"/>	<input type="text"/>
<input type="text" value="Стили"/>	<input type="text" value="Формат..."/>
<input type="text" value="Шрифт"/>	
<input type="text" value="Ра..."/>	
<input type="text"/>	<input type="text"/>

Имя *

Е-mail *

Сайт

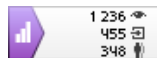
× **6** = двенадцать ↻

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

