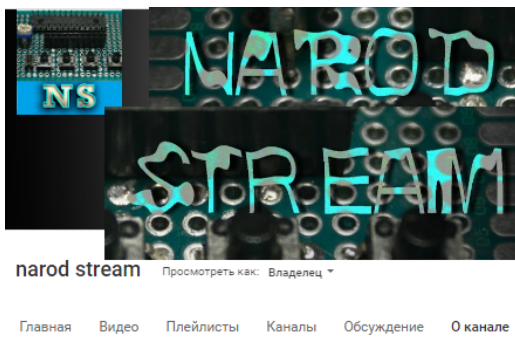


Сайт narodstream.ru

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](#)

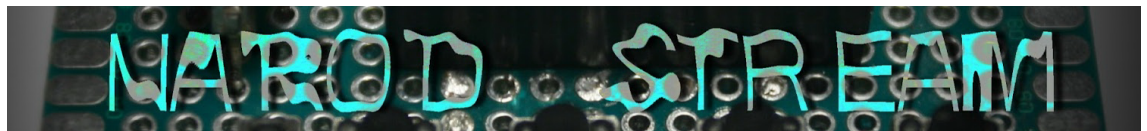
Последние ответы на форуме

-  [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](#)

[Главная](#)[Новости](#)[Уроки по программированию МК](#)[Ссылки](#)[Форум](#)[Помощь](#)

Просмотров:

[Главная](#) > [AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 5](#)

Урок 34 Часть 5

Дисплей TFT 240×320 8bit

В [прошлой части](#) нашего урока мы написали ещё несколько функций для работы с дисплеем и залили экран определенным цветом.

Давайте теперь окрасим дисплей во все те цвета, которые мы прописали в макросах. Зайдём в функцию `main()` и напишем там код после вызова функции инициализации дисплея

```
TFT9341_ini();  
TFT9341_FillScreen(RED);  
_delay_ms(500);  
TFT9341_FillScreen(BLUE);  
_delay_ms(500);  
TFT9341_FillScreen(GREEN);
```

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](#)

```

_delay_ms(500);
TFT9341_FillScreen(CYAN);
_delay_ms(500);
TFT9341_FillScreen(MAGENTA);
_delay_ms(500);
TFT9341_FillScreen(YELLOW);
_delay_ms(500);
TFT9341_FillScreen(WHITE);
_delay_ms(500);
TFT9341_FillScreen(BLACK);
_delay_ms(500);
while (1)

```

Соберём код, прошьём контроллер, посмотрим результат



Продолжаем дальше.

Напишем функцию заливки прямоугольной области экрана, то есть не всю область, а отдельный прямоугольник с заранее заданными координатами. Вернёмся в файл ili9341.c и напомним данную функцию сразу после функции заливки полного экрана

```

//_____
void TFT9341_FillRectangle(unsigned int
color,unsigned int x1, unsigned int y1,
unsigned int x2, unsigned int y2)
{
TFT9341_SetAddrWindow(x1, y1, x2, y2);
TFT9341_Flood(color, (long)(x2-x1+1) * (long)(y2-
y1+1));
}
//_____

```

Здесь, я думаю, всё предельно понятно. Во входных параметрах мы передаём цвет, координаты левого верхнего угла прямоугольной области и координаты правого нижнего угла. Также мы рассчитываем размер заливаемой области умножив вычисленную длину на высоту.

Напишем также в заголовочный файл прототип данной функции.

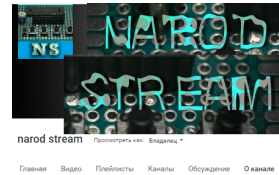
Теперь напомним функцию генерирования случайного цвета

```

//_____
unsigned int TFT9341_RandColor(void)

```


**Заходите на
канал Narod
Stream**



Архивы


- [Октябрь 2017](#)
- [Сентябрь 2017](#)

```

{
    unsigned char c = rand()%8;
    switch(c)
    {
        case 0:
            return BLACK;
            break;
        case 1:
            return BLUE;
            break;
        case 2:
            return RED;
            break;
        case 3:
            return GREEN;
            break;
        case 4:
            return CYAN;
            break;
        case 5:
            return MAGENTA;
            break;
        case 6:
            return YELLOW;
            break;
        case 7:
            return WHITE;
            break;
    }
    return BLACK;
}
//_____

```

- [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	1 414 385
НА ПУНТИ	82 23

Добавим прототип для данной функции в заголовочный файл и добавим в main() небольшой тест сразу после кода по окрашиванию экрана, который мы только что написали

```

_delay_ms(500);
for(i=0;i<8;i++)
{

    TFT9341_FillRectangle(TFT9341_RandColor(),0,0,119,159
    );
    _delay_ms(100);

    TFT9341_FillRectangle(TFT9341_RandColor(),120,0,239,1
    59);
    _delay_ms(100);

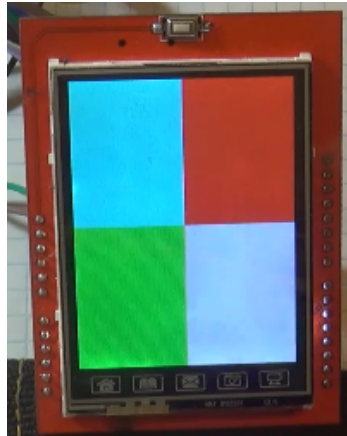
    TFT9341_FillRectangle(TFT9341_RandColor(),0,160,119,3
    19);
    _delay_ms(100);

    TFT9341_FillRectangle(TFT9341_RandColor(),120,160,239

```

```
,319);
    _delay_ms(100);
}
TFT9341_FillScreen(BLACK);
```

Соберём код, прошьём контроллер и посмотрим результат на дисплее



Опять вернёмся в наш файл с функциями и напомним там функцию вывода точки определённого цвета на экран

```
//-----
void TFT9341_DrawPixel(int x, int y, unsigned int
color)
{
    if((x<0)|| (y<0)|| (x>=X_SIZE)|| (y>=Y_SIZE)) return;
    CS_ACTIVE;
    TFT9341_SetAddrWindow(x,y,X_SIZE-1,Y_SIZE-1);
    CS_ACTIVE;
    CD_COMMAND;
    TFT9341_Write8(0x2C);
    CD_DATA;
    TFT9341_Write8(color>>8);TFT9341_Write8(color);
    CS_IDLE;
}
//-----
```

В данной функции мы сначала проверяем входные параметры на валидность, затем опускаем шину выбора, затем адресуем память области экрана, верхний левый угол будет у нас равен координатам точки, а нижний правый — нижнему правому углу дисплея, затем мы ещё раз опускаем шину выбора, так как в функции адресации она поднимается, затем переходим в режим передачи команд, посылаем адрес регистра для передачи данных в область памяти и передаём только один пиксель, ну и по окончании поднимаем шину выбора.

Напишем для данной функции прототип.

Ну и, соответственно, тест в main()

```
TFT9341_FillScreen(BLACK);
for(i=0;i<15000;i++)
{
```

```
TFT9341_DrawPixel(rand()%240,rand()%320,TFT9341_RandC  
olor());  
}  
TFT9341_FillScreen(BLACK);
```

Мы будем выводить здесь 15000 точек случайного цвета в случайное место дисплея.

Соберём код, прошьём контроллер и посмотрим результат нашей работы



В **следующей части** нашего занятия мы напишем ещё несколько функций для работы с дисплеем и попробуем вывести другие примитивы (линии, прямоугольники, окружности) 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)**



Добавить комментарий

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

Имя *

E-mail *

Сайт

× 1 = девять ↺

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

