

Свежие комментарии

- Сергей к записи [PIC Урок 3. Бегущие огни](#)
- Narod Stream к записи [PIC Урок 5. Таймеры](#)
- Артем к записи [PIC Урок 5. Таймеры](#)
- Narod Stream к записи [AVR Урок 13. ШИМ. Мигаем светодиодом плавно. Часть 1](#)
- Narod Stream к записи [STM Урок 10. HAL. Изучаем PWM \(ШИМ\). Мигаем светодиодами плавно](#)

Форум. Последние ответы

- [Narod Stream](#) в [Программирование МК STM32](#)
2 дн., 3 час. назад
- [Zandy](#) в [Программирование МК STM32](#)
2 дн., 10 час. назад
- [Narod Stream](#) в [Программирование МК STM32](#)
1 неделя, 6 дн. назад
- [Narod Stream](#) в [Программирование МК STM32](#)
1 неделя, 6 дн. назад
- [fireweb](#) в [Программирование МК STM32](#)
2 нед., 2 дн. назад

Январь 2018

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
« Дек						

Архивы

- [Январь 2018](#)
- [Декабрь 2017](#)
- [Ноябрь 2017](#)
- [Октябрь 2017](#)
- [Сентябрь 2017](#)
- [Август 2017](#)
- [Июль 2017](#)
- [Июнь 2017](#)
- [Май 2017](#)

[Главная](#) > [Программирование AVR](#) > AVR Урок 33. SPI. Карта SD. FAT. Часть 5

AVR Урок 33. SPI. Карта SD. FAT. Часть 5

Posted on [Январь 11, 2017](#) by [Narod Stream](#) Опубликовано в [Программирование AVR](#)
— [Нет комментариев](#) ↓

Урок 33 Часть 5

SPI. Карта SD. FAT

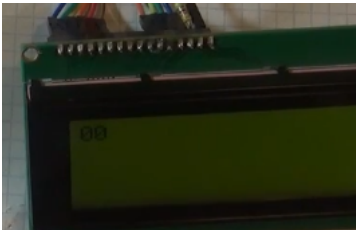
В [прошлой части](#) занятия мы подключили библиотеку для работы с файловой системой и уже смогли не только примонтировать флеш-карту, а уже и открыть файл. Сегодня мы будем продолжать начатое дело и попытаемся уже считать с карты какую-либо информацию.

Чтобы считать какие-то данные из открытого файла, существует функция `pf_lseek`. Поэтому также вызовем её в `main()` и отобразим результат

```
str_lcd(str);
res=pf_lseek(0); //Установим курсор
чтения на 0 в 123.txt
sprintf(str, "%d",res);
setpos(1,0);
str_lcd(str);
```

То есть читать мы будем с самого начала — с нулевого байта.

Соберём проект, прошьём контроллер и посмотрим результат



Мета

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

искать здесь ...

Фильтровать

Уроки по программированию МК

- [Программирование МК AVR](#)
- [Программирование МК STM32](#)
- [Программирование МК PIC](#)
- [Тесты устройств и аксессуаров](#)

Заходите на канал Narod Stream



[Главная](#) [Видео](#) [Плейлисты](#) [Каналы](#) [Обсуждение](#) [0 канал](#)

- Март 2017
- Февраль 2017
- Январь 2017
- Декабрь 2016
- Ноябрь 2016

Мы видим два нуля, что свидетельствует об удачном выполнении функций

Далее уже вызовем непосредственно функцию чтения и также вернём результат

```
str_lcd(str);
res=pf_read(buffer2,128,&s1);
sprintf(str,"%d",res);
setpos(2,0);
str_lcd(str);
```

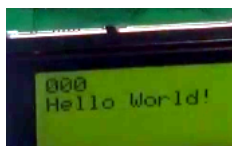
Здесь у функции появились уже какие-то параметры. первый параметр — указатель на наш строковый массив, который у нас инициализирован нулями. Второй — количество байтов для считывания. Функция данная "умная" и она, не смотря на то, что байтов у нас в файле меньше, чем мы пытаемся считать, считает до последнего байта и лишнее считывать пытаться не будет. Третий параметр — это формальный указатель, который ничего нам не даёт, но без него нельзя.

Соберём опять код, прошьём наш контроллер и посмотрим результат на дисплее



Отлично! Значит всё у нас прочиталось. Теперь осталось нам попробовать вывести на дисплей текст из файла

```
str_lcd(str);
setpos(0,1);
str_lcd(buffer2);
```



Всё содержимое нашего файла считалось. Также мы видим, что функция ничего лишнего не считала.

Теперь, прежде чем нам что-то записать на карту SD, размонтируем файловую систему

```
str_lcd(buffer2);
res=pf_mount(0x00); //Демонтируем
//файловую систему карты
sprintf(str,"%d",res);
setpos(3,0);
str_lcd(str);
```

Я думаю, нет смысла проверять, что система отмонтировалась. Вставим задержку, очистим дисплей, вызовем заново функцию монтирования и опять отобразим результат, можно с первой позиции, мы же очистили дисплей. И

Рубрики

- 1-WIRE (3)
- ADC (6)
- DAC (4)
- GPIO (25)
- I2C (19)
- SPI (13)
- USART (8)
- Программирование AVR (131)
- Программирование PIC (6)
- Программирование STM32 (211)
- Тесты устройств и аксессуаров (1)

		7
31 ДЕНЬ	113 936	
	12 522	
07 ДНЕЙ	30 487	
	4 156	
24 ЧАСА	5 599	
	1 152	
СЕГОДНЯ	2 198	
	529	
НАПЛИШУ	198	
	35	

заново откроем тот же файл, вставив перед этим задержку.

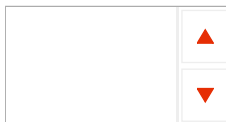
```
str_lcd(str);
_delay_ms(2000);
clearlcd();//очистим дисплей
res=pf_mount(&fs);
sprintf(str, "%d", res);
setpos(4,0);
str_lcd(str);
_delay_ms(2000);
clearlcd();//очистим дисплей
res=pf_open("/123.txt"); //Попытка
открыть файл 123.txt
sprintf(str, "%d", res);
setpos(5,0);
str_lcd(str);
```

Пока проверять ничего не будем, потом всё проверится.

Указатель мы никуда устанавливать не будем, он автоматически встанет на нулевую позицию. Вызовем теперь функцию записи в файл

```
str_lcd(str);
sprintf(buffer2, "World Hello!");
res=pf_write(buffer2, strlen(buffer2)
,&s1);//Запишем текст из buffer2 в
файл 123.txt
sprintf(str, "%d", res);
setpos(6,0);
str_lcd(str);
```

Что мы тут заметили нового? Мы перевернули слова наоборот в тексте, а также использовали новую функцию для измерения длины строки. Измеряются байты до первого вхождения нулевого байта.



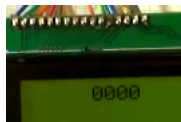
Только есть одна загвоздка. Приме данную функцию, мы ничего не запиш в файл, чтобы всё записалось, нужно финализировать запись. То есть мы можем записывать частями и каждый раз не финализировать, так будет процесс идти быстрее. А затем перед размонтированием файловой системы мы финализируем запись. Поэтому вызовем ещё функцию

```
str_lcd(str);
res=pf_write(0,0,&s1);//Финализируем
файл 123.txt
sprintf(str, "%d", res);
setpos(7,0);
str_lcd(str);
```

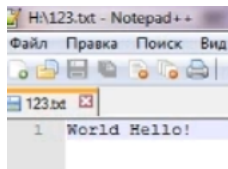
Вот так. Осталось нам только отмонтировать файловую систему

```
str_lcd(str);
res=pf_mount(0x00);//Демонтируем
файловую систему карты
sprintf(str, "%d", res);
setpos(8,0);
str_lcd(str);
```

Соберём код, прошьём контроллер и проверим результат



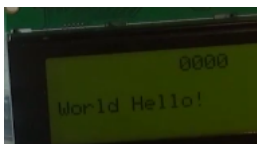
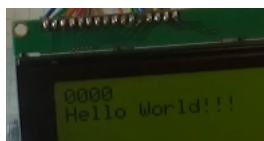
Все результаты у нас хорошие. Вынем карту из картоприёмника и поместим в картовод, чтобы считать файл на ПК. Считаем файл:



Теперь давайте отобразим новое содержание файла на дисплее, дописав код для чтения

```
read file
```

Проверим результат, воткнув перед этим SD-карту, но не ту же, так как на ней уже файл с новым содержимым, а другую



Всё отлично считывается и пишется.

На этом мы данную часть завершим.

В **следующей части** урока мы устраним мелкие ошибки, а также вместо цифрового отображения результата действий с картой SD мы напишем код для отображения в более удобочитаемом текстовом виде. Также мы удалим лишние неиспользуемые функции и фрагменты исходного кода.

Предыдущая
часть

Программирование
МК AVR

Следующая
часть

**Техническая документация на
Secure Digital**

Программатор, модуль SD и дисплей можно приобрести здесь:

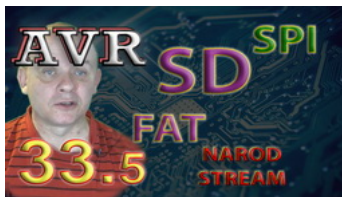
Программатор (продавец надёжный)

USBASP USBISP 2.0

Модуль карты SD SPI

[Дисплей LCD 20×4](#)

Смотреть ВИДЕОУРОК
(нажмите на картинку)



👁 Post Views: 274

← AVR Урок 33.

SPI. Карта SD.

FAT. Часть 4

AVR Урок 33.

SPI. Карта SD.

FAT. Часть 6 →

Добавить комментарий

Ваш e-mail не будет опубликован.

Обязательные поля помечены *

Комментарий

Имя *

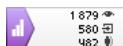
E-mail *

Сайт

+ ● = 10 ↺

Отправить комментарий

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



© 2018 Narod Stream