



Главная | Новости | Уроки по программированию МК | Устройства и интерфейсы | Ссылки | Форум | Помощь

Главная > I2C > AVR Урок 21. Управление DS1307 кнопками. Часть 3

## Свежие комментарии

- SmNikolay к записи [STM Урок 89. LAN. ENC28J60. TCP WEB Server. Подключаем карту SD](#)
- Narod Stream к записи [AVR Урок 3. Пишем код на СИ. Зажигаем светодиод](#)
- strannik2039 к записи [AVR Урок 3. Пишем код на СИ. Зажигаем светодиод](#)
- Dmitriy к записи [AVR Урок 1. Знакомство с семейством AVR](#)
- Narod Stream к записи [STM Урок 9. HAL. Шина I2C. Продолжаем работу с DS3231](#)

## Форум. Последние ответы

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

# AVR Урок 21. Управление DS1307 кнопками. Часть 3

Posted on Декабрь 29, 2016 by Narod Stream  
Опубликовано в I2C, Программирование AVR — Нет комментариев ↓

## Мета

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

искать здесь ...

Фильтровать

Яндекс Директ



### Нужны кнопки управления?

Кнопки управления для различного оборудования. Дополнительные контакты.  
[otnenergo.by](#) Адрес и телефон

Яндекс Директ



### Производство печатных плат

Срочное изготовление за 72 часа! Теплопроводность 1-2 Вт/(м\*К).  
Закажите сегодня!  
[rusalox.ru](#) Адрес и телефон

## Урок 21 Часть 3

# Управление DS1307 кнопками

Продолжаем нашу работу над кнопками.

В [предыдущей части](#) нашего занятия мы обработали нажатие кнопки выбора режима и "научили" данную кнопку управлять режимами изменения показаний. Дальнейшая наша задача — научиться управлять уже не режимами, а самими показаниями, находящимися в регистрах микросхемы DS1307.

Поэтому нам нужно опросить на нажатие другие кнопки.

Ищем в коде в бесконечном цикле, участок, в который мы попадаем в случае выбора режима перевода даты и в конце данного участка добавим следующие строки

```
if (clockmode==CLOCKMODEDATE)
button1state=0;//сбросим статус
if(!(BUTTONPIN&(1<<BUTTONPIN2))//Кнопка
2 нажата
```



Заходите на канал  
Narod Stream

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

```
if(!(BUTTONPIN&(1<<BUTTONPIN3))//Кнопка
3 нажата
```

Мы напишем пока условия. А тела данных условий мы пока писать не будем, так как у нас уже в бесконечном цикле и так предостаточно кода, что уже можно и запутаться. Поэтому тела подобных условий мы оформим в общую для них функцию и напишем её сразу после функции инициализации портов

```
//-----
void ModifyRTC(char index, char
direction)
{
}
//-----
```

Данная функция у нас будет отвечать именно за изменение значений регистров, то есть с помощью неё мы будем переводить показания времени, даты и дня недели в наших часах. Первым входным аргументом функции будет тип показания, которое мы будем модифицировать, а вторым — направление изменение, то есть увеличивать мы будем данное показание или уменьшать.

В тело данной функции мы пока напишем то, что мы будем в ней делать независимо от условий или от входных аргументов

```
void ModifyRTC(char index, char
direction)
{
    I2C_StartCondition();
    I2C_SendByte(0b11010000);
    I2C_StopCondition();
}
```

Прежде чем писать дальше код данной функции, нам нужно завершить условия, которые мы начали писать в бесконечном цикле, иначе мы не сможем проверить код на собираемость из-за возникающей ошибки незаконченной операции условия

```
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
    ModifyRTC(1,0);//1-дата,0-
назад
if(!(BUTTONPIN&(1<<BUTTONPIN3))//
Кнопка 3 нажата
    ModifyRTC(1,1);//1-дата,1-
вперед
}
```

Как видно из кода, мы задаём, какой именно показатель мы редактируем и в каком направлении.

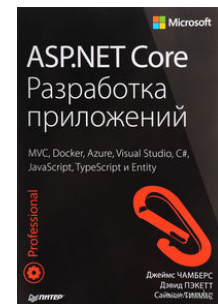
Ну и давайте добавим подобный код для редактирования всех остальных показаний регистров микросхемы, найдя для этого нужные места в бесконечном цикле

```
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
```



narod stream Просмотреть «н»: Владелец

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



ASP.NET Core.  
Разработка  
приложений  
Купить



## Рубрики

- 1-WIRE (3)
- ADC (6)
- DAC (4)
- GPIO (26)
- I2C (19)
- SPI (13)
- USART (8)
- Программирование AVR (131)
- Программирование PIC (7)
- Программирование STM32 (213)
- Тесты устройств и аксессуаров (1)

```

ModifyRTC(2,0);//2-месяц,0-назад
if(!(BUTTONPIN&(1<<BUTTONPIN3))//
Кнопка 3 нажата
ModifyRTC(2,1);//2-месяц,1-вперед
. . . . .
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
ModifyRTC(3,0);//3-год,0-назад
if(!(BUTTONPIN&(1<<BUTTONPIN3))//
Кнопка 3 нажата
ModifyRTC(3,1);//3-год,1-вперед
. . . . .
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
ModifyRTC(4,0);//4-день недели,0-
назад
if(!(BUTTONPIN&(1<<BUTTONPIN3))//
Кнопка 3 нажата
ModifyRTC(4,1);//4-день недели,1-
вперед
. . . . .
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
ModifyRTC(5,0);//5-часы,0-назад
if(!(BUTTONPIN&(1<<BUTTONPIN3))//Кнопка
3 нажата
ModifyRTC(5,1);//5-часы,1-вперед
. . . . .
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
ModifyRTC(6,0);//6-минуты,0-назад
if(!(BUTTONPIN&(1<<BUTTONPIN3))//
Кнопка 3 нажата
ModifyRTC(6,1);//6-минуты,1-вперед
. . . . .
if(!(BUTTONPIN&(1<<BUTTONPIN2))//
Кнопка 2 нажата
ModifyRTC(7,0);//7-секунды,0-
установка 30
if(!(BUTTONPIN&(1<<BUTTONPIN3))//
Кнопка 3 нажата
ModifyRTC(7,1);//7-секунды,1-
установка 0

```

С секундами мы поступили следующим образом. Раз уж у нас две кнопки, то будем их использовать обе. Одна будет сбрасывать их на 0, а другая на 30 секунд.

Теперь можно продолжить начатую функцию модификации показаний. Добавим в нее отслеживание режимов с помощью оператора switch

```

I2C_SendByte(0b11010000);
switch(index){
case 1: //число даты
break;
case 2: //месяц
break;
case 3: //год
break;
case 4: //день недели
break;
case 5: //часы
break;
case 6: //минуты
break;
case 7: //секунды
break;
}
I2C_StopCondition();

```

	7
31 ДЕНЬ	124 507
07 ДНЕЙ	13 098
30 ДНЕЙ	30 048
4 ЧАСА	4 366
5 253	
1 071	
24 ЧАСА	5 253
1 071	
СЕГОДНЯ	2 567
580	
НАПЛИВУ	52
26	



Ну и попробуем написать код для перевода даты. Первым делом надо перейти на определённый адрес регистра, отвечающий именно за перевод даты. Все эти адреса мы давно знаем

```
case 1: //число даты
    I2C_SendByte(4); //Переходим на
    0x04 – байт числа даты
```

Добавим далее отслеживание направление перевода даты

```
I2C_SendByte(4); //Переходим на 0x04
– байт числа даты
//вперед
if (direction==1)
{
}
//назад
if (direction==0)
{
}
break;
```



The Most Popular eCommerce Platform  
Get Started Today For Free

Laun

Пока поработаем с направлением "вперёд". Будет ещё условия, так как у нас существуют варианты различного количества дней в различных месяцах

```
if (direction==1)
{
    if (month==2) //февраль
    {
    }
    else if ((month==4)|(month==6)|
(month==9)|(month==11))
    {
    }
    else
    {
    }
}
```

Вот такие варианты. У нас есть февраль, в котором дней будет или 28 или 29, есть ещё апрель, июнь, сентябрь и ноябрь, в которых 30 дней, и есть остальные месяцы, в которых дней у нас 31. С этим учётом мы и добавили условия.

Работаем с февралём. Тут придётся ещё отследить, високосный у нас год или обычный

```
if (month==2) //февраль
{
    if(year%4==0) //високосный год
    {
    }
    else
    {
    }
}
```

Здесь мы поступили оригинальным способом, используя остаток от деления

величины года на 4, если делится, то год високосный, если нет — обычный.

Ну вот теперь уже конкретное изменение показаний. У нас февраль и год високосный

```
if(year%4==0) //високосный год
{
    if(date<29)
    I2C_SendByte(RTC_ConvertFromBinDec(date+1));
    else
    I2C_SendByte(RTC_ConvertFromBinDec(1));
}
```

Мы здесь инкрементируем показания до тех пор, пока не дойдём до 29, Если у нас уже 29, то мы дальше прибавлять не можем так как 30 февраля не бывает даже в високосном году, поэтому мы в этом случае установим 1 число месяца.

Теперь обработаем февраль невисокосного года подобным образом

```
else
{
    if(date<28)
    I2C_SendByte(RTC_ConvertFromBinDec(date+1));
    else
    I2C_SendByte(RTC_ConvertFromBinDec(1));
}
```

Здесь просто вместо 29 будет 28, так как в обычный год в феврале, как водится, 28 дней.

Теперь подобным образом обработаем месяцы, в которых 30 дней

```
else if ((month==4)|(month==6)|
(month==9)|(month==11))
{
    if(date<30)
    I2C_SendByte(RTC_ConvertFromBinDec(date+1));
    else
    I2C_SendByte(RTC_ConvertFromBinDec(1));
}
```

Ну и, соответственно, остальные месяцы

```
else
{
    if(date<31)
    I2C_SendByte(RTC_ConvertFromBinDec(date+1));
    else
    I2C_SendByte(RTC_ConvertFromBinDec(1));
}
```

Здесь нам может показаться, что мы уже закончили с показаниями даты. Но не тут то было, у нас ещё перевод назад. Добавим туда также отслеживание вариантов месяцев

```
//назад
if (direction==0)
{
```

```

    if (month==2) //февраль
    {
        if(year%4==0) //високосный год
        {
        }
        else
        {
        }
    }
    else if ((month==4)|(month==6)|
(month==9)|(month==11))
    {
    }
    else
    {
    }
}
break;

```

Обработаем високосный февраль

```

if(year%4==0) //високосный год
{
    if(date>1)
    I2C_SendByte(RTC_ConvertFromBinDec(d
ate-1));
    else
    I2C_SendByte(RTC_ConvertFromBinDec(2
9));
}

```

Алгоритм здесь похожий. Мы декрементируем показания до тех, пор пока не дойдём до 1 числа, как только дошли, дальше нам декрементировать некуда и мы устанавливаем максимальный день. В данном случае у нас 29.

Таким же образом обработаем невисокосный февраль и все остальные месяцы

```

    else
    {
        if(date>1)
        I2C_SendByte(RTC_ConvertFromBinDec(d
ate-1));
        else
        I2C_SendByte(RTC_ConvertFromBinDec(2
8));
    }
    else if ((month==4)|(month==6)|
(month==9)|(month==11))
    {
        if(date>1)
        I2C_SendByte(RTC_ConvertFromBinDec(d
ate-1));
        else
        I2C_SendByte(RTC_ConvertFromBinDec(3
0));
    }
    else
    {
        if(date>1)
        I2C_SendByte(RTC_ConvertFromBinDec(d
ate-1));
        else
        I2C_SendByte(RTC_ConvertFromBinDec(3
1));
    }
}

```

Ну вот мы и закончили с алгоритмом изменения даты.

Соберём код, прошьём контроллер и на настоящей схеме, протеусу больше доверять пока не будем.

Также нет смысла здесь показывать картинку, так как мы здесь не увидим процесс. Для того, чтобы его увидеть, нужно будет посмотреть видеоверсию данной части урока, ссылка на который содержится внизу и пройти по ней возможно, нажав на картинку видеоурока. Поэтому поверьте мне на слово, что всё у нас отлично переводится.

С переводом остальных показателей мы займёмся в [следующей части](#) нашего урока.

[Предыдущая часть](#)

[Программирование МК AVR](#)

[Следующая часть](#)

Программатор, модуль RTC DS1307 с микросхемой памяти и дисплей можно приобрести здесь:

Программатор (продавец надёжный)

[USBASP USBISP 2.0](#)

[Модуль RTC DS1307 с микросхемой памяти](#)

[Дисплей LCD 16×2](#)

**Смотреть ВИДЕОУРОК**  
(нажмите на картинку)



👁 Post Views: 536

◀ AVR Урок 21.

Управление

DS1307 кнопками.

Часть 2

AVR Урок 21.

Управление

DS1307 кнопками.

Часть 4

**Добавить комментарий**


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

**Комментарий**

**Имя \***

**E-mail \***

**Сайт**

+ восемь = 15 

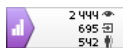
Отправить комментарий

[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