



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., 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

Главная > I2C > AVR Урок 16. Интерфейс TWI (I2C). Часть 2

# AVR Урок 16. Интерфейс TWI (I2C). Часть 2

Posted on Декабрь 17, 2016 by Narod Stream  
Опубликовано в I2C, Программирование AVR — Нет комментариев ↓

Мета

- Регистрация
- Войти
- RSS записей
- RSS комментариев
- WordPress.org

Искать здесь

Нужны кнопки управления?

Кнопки управления для различного оборудования. Дополнительные контакты.  
otenergo.by Адрес и телефон

Очень горячая аниме игра

Эта аниме игра поглощает с первых минут, начнешь играть и забудешь про сон 18+  
promo.101xp.com

Уроки по программированию МК

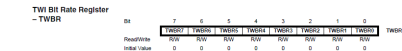
Программирование МК PIC

Тесты устройств и аксессуаров

## Урок 16 Часть 2

# Интерфейс TWI (I2C)

В предыдущей части занятия мы кратко ознакомились с шиной I2C или TWI (Two-wire Serial Interface), а также узнали, как именно она работает, как к ней можно что-то подключить, также познакомились с её регистрами, ну и кое-какие узнали нюансы, как, например настроить частоту или скорость данной шины. Продолжим наши познания в данной области познакомимся с регистрами подробнее и все-таки начнём писать какой-то исходный код, а то уже руки чешутся. Давайте поконкретнее разберёмся с расчетом скорости шины и её конфигурированием. Как мы уже знаем из предыдущей части нашего урока, за установку скорости отвечает регистр TWBR



С формулой расчета скорости мы познакомились. Но там у нас

ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР ПВТ

КУРС iOS РАЗРАБОТЧИКА

Высокая востребованность на рынке IT!

Заходите на канал  
Narod Stream

- [Июнь 2017](#)
- [Май 2017](#)
- [Март 2017](#)
- [Февраль 2017](#)
- [Январь 2017](#)
- [Декабрь 2016](#)
- [Ноябрь 2016](#)

неизвестной величиной является скорость шины I2C.

Но, как правило, в жизни больше случается моментов, когда скорость нам наоборот известна та, которой мы хотим добиться, и тогда нам в формулу придется подбирать величину TWBR, а это крайне неудобно. Поэтому давайте выразим из формулы значение регистра и формула у нас перевернется вот в такой вот вид

$$TWBR = ((F\_CPU/F\_SCL)-16)/(2*4*TWPS))$$

Вот так будет считать гораздо удобнее.

Например. мы хотим к контроллеру, работающему на частоте 8 МГц подключить какой-то девайс и работать мы с данным девайсом хотим на частоте 400 кГц. Биты TWPS пусть у нас остаются в нуле, тогда мы получим значение TWBR, равное 2. Вот так вот и считаем.

В качестве подопытного устройства мы возьмём микросхему **EEPROM** — **AT24C32D**, которая установлена в модуле с часовой микросхемой DS1307. Я думаю, такие модули у многих есть, а у кого нет, то их можно заказать и приобрести, стоят они вообще копейки. А раз уж на данном модуле бонусом идёт такая микросхема, так почему бы ей не воспользоваться в качестве тестирования шины I2C.

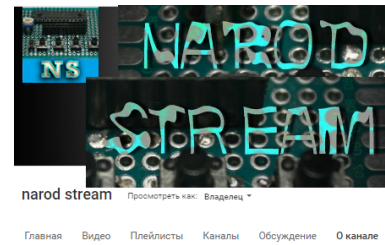
Создадим стандартный проект в Atmel Studio, подключим туда файлы для работы usart из соответствующего проекта, также создадим файл main.h, так же стандартно со всеми обычными стандартными библиотечными файлами, подключив его затем к DS1307Eeprom.c. Затем создадим и подключим также обычным образом две пары файлов. **eeepromext.h** и **eeepromext.c** для специфических функций по работе с внешним EEPROM и, соответственно, **twi.h** и **twi.c** для работы с шиной TWI.

После всего этого у нас получится вот такой вот main.h

```
#ifndef MAIN_H_
#define MAIN_H_
#define F_CPU 8000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>
#include <stdlib.h>
#include "usart.h"
#include "twi.h"
#include "eeepromext.h"
#endif /* MAIN_H_ */
```

А файл DS1307Eeprom.c вместе с инициализацией USART — вот такой

```
#include "main.h"
int main(void)
{
    USART_Init (8);
    while(1)
```



## Рубрики

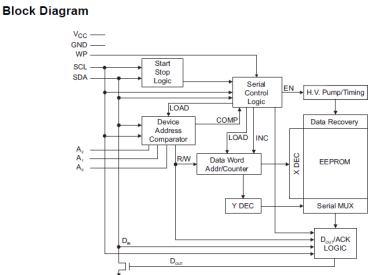
- [1-WIRE](#) (3)
- [ADC](#) (6)
- [DAC](#) (4)
- [GPIO](#) (26)
- [I2C](#) (19)
- [SPI](#) (13)
- [USART](#) (8)
- [Программирование AVR](#) (131)
- [Программирование PIC](#) (7)
- [Программирование STM32](#) (213)
- [Тесты устройств и аксессуаров](#) (1)

```
{
}
}
```

↗

31.01.2018	124 507
07.01.2018	13 098
07.01.2018	30 048
07.01.2018	4 366
24.01.2018	5 253
24.01.2018	1 071

Посмотрим блок-схему микросхемы EEPROM



Как мы видим, здесь имеется само собой шина I2C, контакт WP — Write Protect, при соединении которого с общим проводом мы работаем с данной микросхемой в обычном режиме, а если мы его подтянем к шине питания, то, соответственно, память микросхемы будет только читаться.

Также мы видим здесь адресную шину, состоящую из трёх проводов, соединение каждого из которых с землёй сбрасывает соответствующий бит в адресе. Это сделано с учётом того, чтобы на одной шине можно было использовать несколько подобных микросхем. Вот так вот выглядит адрес устройства при условии, если все эти ножки будут соединены с общим проводом. Соединение с питанием каждой из данных ножек соответствующий бит переведёт в логическую 1

Figure 7-1. Device Addressing

Package	Device Type Identifier				Hardware Address Select Bits				R/W Select
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SOIC, TSSOP, LDIP, PDIP, and VFBGA	1	0	1	0	A2	A1	A0	R/W	
SOT-23 and WLCSP	1	0	1	0	0	0	0	R/W	

MSB LSB

Данная микросхема также умеет работать очень быстро. При питании 1,7 вольт она может работать до 400 кГц, а при 2,5-5 в — до 1 мегагерца.

Конечно, мы таких частот на нашем контроллере не добьёмся, так как частота шины I2C у микроконтроллера AVR не может быть больше, чем частота тактирования, делённая на 16. Поэтому при 8МГц мы можем разогнаться по I2C только до 500 кГц.

Но нам спешить некуда, нам главное надёжность. Поэтому с шиной мы будем работать на частоте 100 кГц и путём подсчета по вышенаписанной формуле мы получим значение TWBR, равное 32 или шестнадцатеричных 0x20.

Также ещё скажу интересную вещь, что инициализация нашей шины I2C будет состоять только лишь из настройки скорости. Напишем эту функцию в

файле twi.c и создадим на неё прототип в соответствующем хедер-файле

```
#include "twi.h"
void I2C_Init(void)
{
    TWBR=0x20; //скорость передачи
                (при 8 МГц получается 100 кГц)
}
```

Ну, прототипы, я думаю все уже писать умеют, поэтому я не буду здесь его постить, чтобы не порождать лишний текст.

Также вызовем данную функцию в main()

```
USART_Init (8);
I2C_Init();
```

Попробуем собрать проект.

Если всё нормально собралось, то на этом закончим данную часть занятия.

Предыдущая  
часть

Программирование  
МК AVR

Следующая  
часть

### Техническая документация на микросхему AT24C32

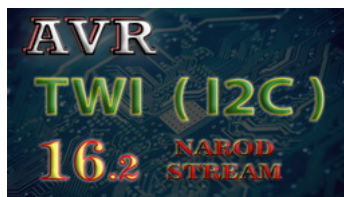
Программатор и модуль RTC DS1307 с микросхемой памяти можно приобрести здесь:

Программатор (продавец надёжный)

[UUSBASP USBISP 2.0](#)

[Модуль RTC DS1307 с микросхемой памяти](#)

Смотреть ВИДЕОУРОК (нажмите на картинку)



👁 Post Views: 584

◀ AVR Урок 16.

Интерфейс TWI

(I2C). Часть 1

AVR Урок 16.

Интерфейс TWI

(I2C). Часть 3 ▶

**Добавить комментарий**

Ваш e-mail не будет опубликован.

Обязательные поля помечены \*

Комментарий

**Имя \***

**E-mail \***

**Сайт**

•••

- 4 =

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](#)

