



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

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 1

AVR Урок 16. Интерфейс TWI (I2C). Часть 1

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

Искать здесь

Нужны кнопки управления?

Кнопки управления для различного оборудования. Дополнительные контакты.

Нужно программирование контроллеров?

Комплексное обучение по продукции Siemens (SIMATIC S7). 5 уровней курсов!

Уроки по программированию МК

Печатные Платы на Заказ. Звоните!

Изготовление печатных плат на заказ. От прототипов до крупных партий. Звони [pcb.electropribor-penza.ru](#) Адрес и телефон

Урок 16 Часть 1

Интерфейс TWI (I2C)

Опыт нашей предыдущей работы с микроконтроллерами AVR показал то, что мы уже многому научились, в том числе общаться с внешними устройствами типа ПК. Теперь, я думаю, все понимают, что этого недостаточно. Есть масса устройств и датчиков, которые нам хочется подключить к микроконтроллерам и, мало того, ещё ими и поуправлять, забрать у них какие-то данные. Только данные девайсы подключаются к МК с помощью проводов и существует несколько видов подключений. Один из таких видов — шина I2C. Данный интерфейс является авторской разработкой компании Phillips и поэтому Atmel, чтобы не нарушать авторские права, немного изменила наименование данной шины и назвала её TWI (Two-wire Serial Interface).

Вот мы сегодня и займёмся изучением данного интерфейса, а также в данном уроке мы попробуем его как-то и использовать, что-то к нему подключить и с этим чем-то пообщаться под интерфейсу TWI.

Мета

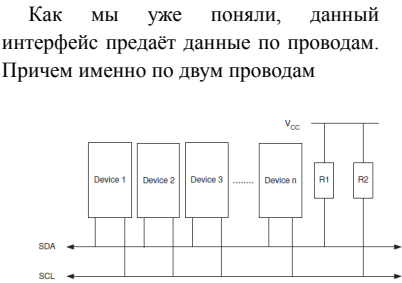
- Регистрация
- Войти
- RSS записей
- RSS комментариев
- WordPress.org

Заходите на канал Narod Stream



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

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



Как мы видим из данной блок-схемы, у нас есть два провода или контакта — SDA и SCL, подсоединённые к контроллеру к его соответствующим ножкам, а затем к этим проводам мы ещё подключаем какое-нибудь устройство или несколько устройств. Теоретически мы можем подключить до 127 устройств, впоследствии мы поймём, почему именно столько. Ножка SDA отвечает за передачу и приём данных по данной шине, а ножка SCL — это ножка тактирования, таким образом шина I2C является синхронной, что увеличивает стабильность и гарантированность передачи данных по ней. Также в данной блок-схеме мы наблюдаем то, что ещё к данным ножкам необходимо подтянуть резисторы на шину питания, причём как правило в каких-то модулях или устройствах данные резисторы уже подтянуты, и. если мы подключим их несколько, то у нас получится ряд параллельно подключённых подтягивающих резисторов, что будет очень нехорошо и придется оставить их в каком-то одном устройстве, а в других выпаять. Всё это происходит потому, что есть ограничение на номинал данных резисторов в зависимости от определенных условия

Rp	Value of Pull-up resistor	$t_{RST} \leq 100\text{ns}$	$V_{CC} = 0.4\text{V}$ 3mA	10000 C_1	Ω
		$t_{RST} > 100\text{ns}$	$V_{CC} = 0.4\text{V}$ 3mA	20000 C_1	

Как показала практика, данные резисторы не должны быть сопротивлением более 10 килоом и менее 4,7 килоом.

Также, само собой, сигналы должны быть толерантными к логическим уровням нашего МК, поэтому подключенные устройства должны быть соответствующим образом запитаны, а также общий провод устройств и контроллера также должен быть объединён.

Хочу подчеркнуть, что данный интерфейс в микроконтроллерах AVR реализован на аппаратном уровне. Можно также организовывать его и программным способом, путём включения в определённое время двух любых ножек портов в определенные логические состояния, но само собой, аппаратная организация всегда лучше по всем показателям. Программная реализация как правило используется, если не хватит одной шины, но как правило хватает, так как мы выше уже видели, что устройств можно к одной

Microsoft.NET: архитектура корпоративных 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)

7

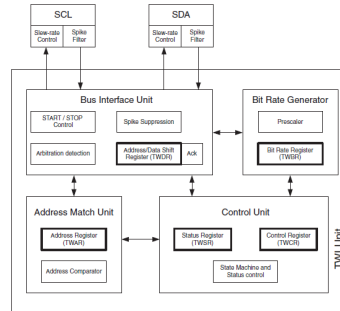
31 ДЕНЬ	124 507
	13 098
07 ДНЕЙ	30 048
	4 366
24 ЧАСА	5 253
	1 071
СЕГОДНЯ	2 568
	580
НАПЯНУ	52
	26

шине подключить несколько. Таким образом, мы будем рассматривать именно аппаратную реализацию — на уровне регистров контроллера.



В следующей блок-схеме мы рассмотрим следующие регистры интерфейса TWI контроллера Atmega8

Figure 76. Overview of the TWI Module



Давайте с ними кратко ознакомимся.

Регистр **TWBR** — это регистр, в котором мы устанавливаем значения для обеспечения определённой скорости шины. Устройства, которые мы подключаем по I2C требуют определённой скорости ну или определённого диапазона скоростей. Рассчитывается эта скорость или битрейт по следующей формуле

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{\text{TWPS}}}$$

Как мы видим, в данной формуле для расчета частоты шины участвует частота тактирования контроллера, значение регистра **TWBR**, а также значение **TWPS**, судя по формуле представляющий собой предделитель, то есть, чем он больше, тем больше будет коэффициент деления, так как в степерь значения данного регистра мы возводим цифру 4.

Причем **TWPS** — это не отдельный регистр, а совокупность всего лишь двух битов статусного регистра **TWSR**, в котором помимо этих битов есть ещё и другие, посредством которых мы будем отслеживать определенные состояния или ошибки обмена данными. Есть перечень кодов статуса, по которым мы и будем узнавать, удачно ли прошла команда или что-то не так.

Также само собой существует ещё и управляющий регистр **TWCR**, с помощью установки в соответствующее состояния битов которого мы будем определённым образом настраивать нашу шину.

Есть ещё адресный регистр **TWAR**, с помощью которого, если мы настраиваем контроллер ведущим устройством, мы передаем ведомому устройству 7-битный



адрес (отсюда и 127 максимально устройств на шине), по которому-то как раз и будет обращение именно к определённому устройству из всех навешанных на наши два провода. А восьмой бит к адресу будет добавляться для того, чтобы передать, читаем мы данные или пишем. А если у нас контроллер будет настроен как ведомое устройство, то в данном регистре мы будем получать от ведущего адрес и сравнивать его с определённым, чтобы понять, обратился ведущий именно к нам или к кому-то другому.

Ну, само собой, есть и регистр данных, в который мы в случае ведущего устройства записываем данные для передачи по шине, а в случае ведомого считываем пришедшие по шине данные.

Вот, вообще, вкратце о регистрах так.

В **следующей части** занятия мы продолжим подробнее заниматься с данной шиной.

*Предыдущий
урок*

*Программирование
МК AVR*

*Следующая
часть*

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

Программатор (продавец надёжный)

USBASP USBISP 2.0

Модуль RTC DS1307 с микросхемой памяти

Смотреть **ВИДЕОУРОК** (нажмите на картинку)



👁 Post Views: 633

◀ AVR Урок 15.

Внутренняя

энергонезависимая

память EEPROM.

Часть 4

AVR Урок 16.

**Добавить
комментарий**

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

Комментарий

Имя *

E-mail *

Сайт

× 3 =

восемнадцать

Отправить комментарий



2 444 ↗
695 ↗
542 ↗