



Яндекс.Директ




**Ноутбук  
Lenovo C930  
по низкой  
цене!**

evindo.ru >



**Мощный  
планшет  
Lenovo Yoga  
за 3890р!**

yoga.bigshopsone.top >

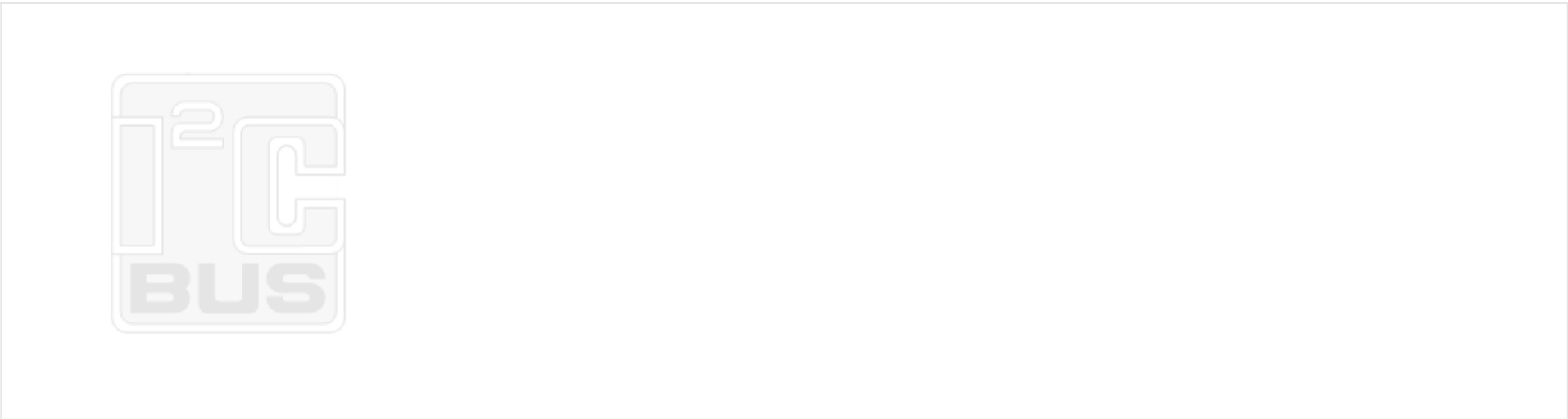


**Планшет  
Lenovo Yoga  
Book  
от 34200р**

market.yandex.ru >

# Интерфейсная шина IIC (I2C)

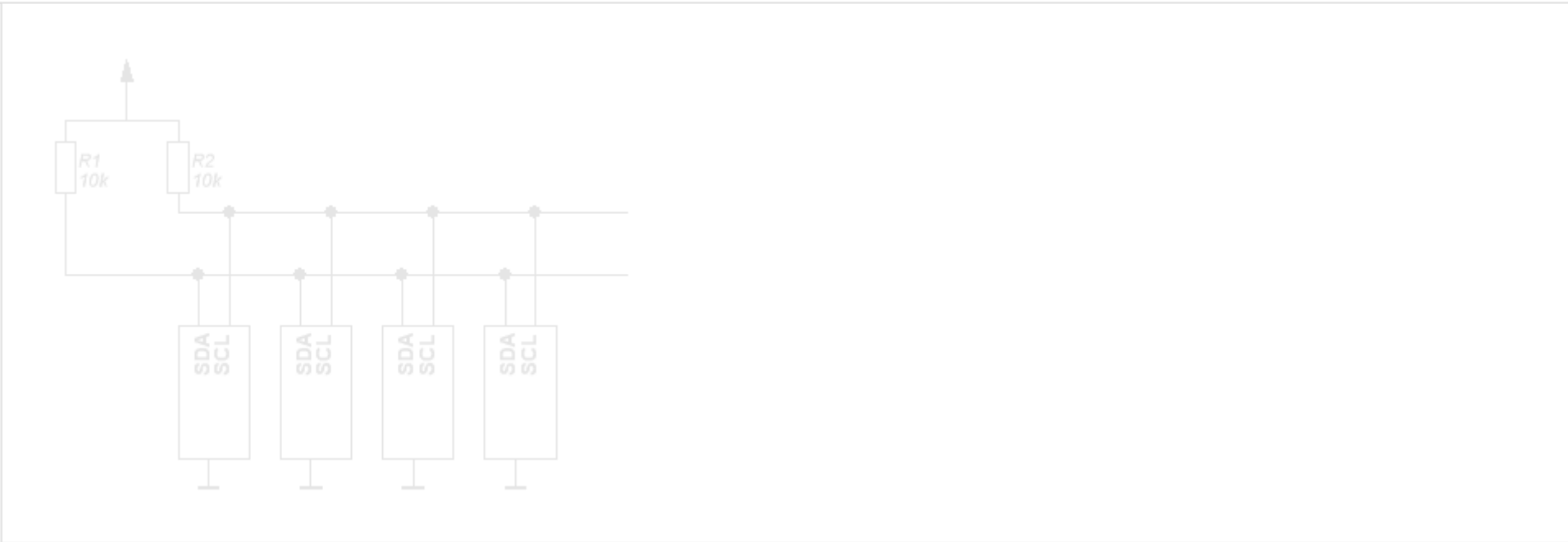
Начинающим | 16 апреля 2009 | DI HALT | 182 комментария



Один из моих самых любимых интерфейсов. Разработан в компании **Philips** и право на его использование стоит денег, но все на это дружно положили и пользуют в свое удовольствие, называя только по другому. В **Atmel** его зовут **TWI**, но от этого ничего не меняется :) Обычно при разборе **IIC** во всех книгах ограничиваются примером с **EEPROM** на этом и ограничиваются. Да еще юзают софтверный Master. Не дождетесь, у меня будет подробный разбор работы этой шины как в режиме Master так и Slave, да еще на аппаратных блоках с полным выполнением всей структуры конечного автомата протокола. Но об этом после, а сейчас основы.

## Физический уровень.

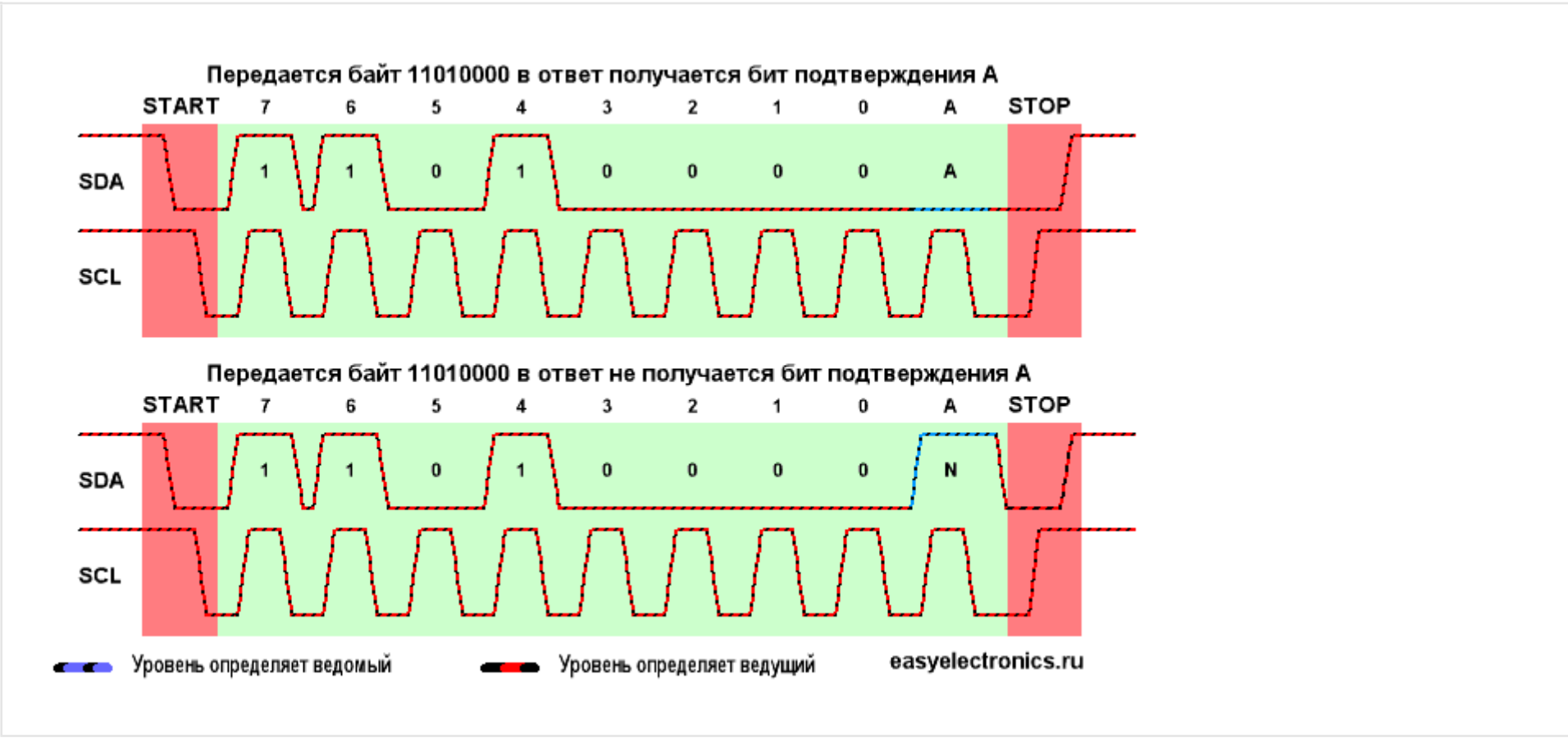
Данные передаются по двум проводам — провод данных и провод тактов. Есть **ведущий**(master) и **ведомый** (slave), такты генерирует master, ведомый лишь поддакивает при приеме байта. Всего на одной двупроводной шине может быть **до 127 устройств**. Схема подключения — **монтажное И**

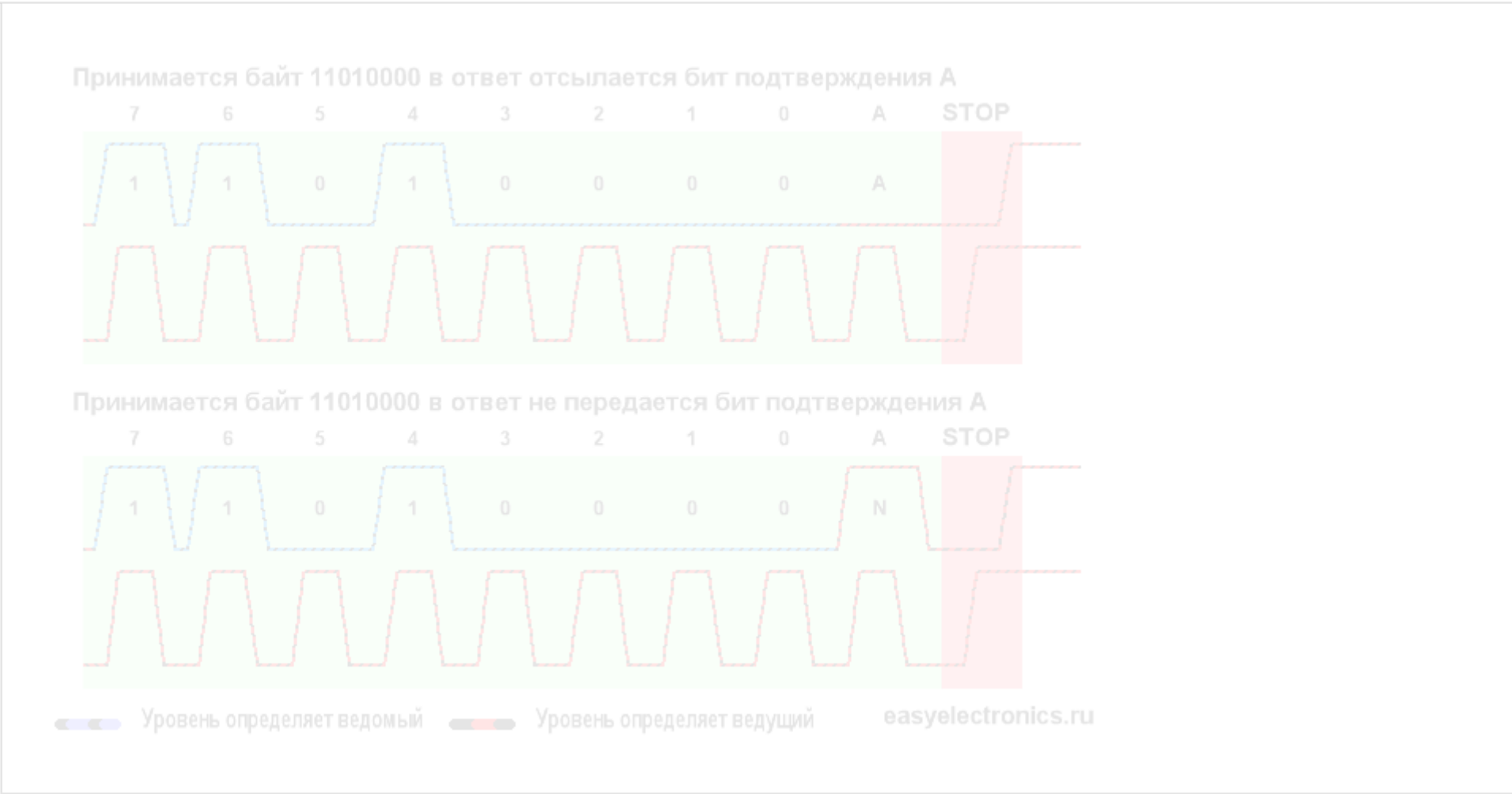


Передача/Прием сигналов осуществляется **прижиманием линии в 0**, в единичку **устанавливается сама, за счет подтягивающих резисторов**. Их ставить обязательно **всегда!** Стандарт! Резисторы на 10к оптимальны. Чем больше резистор, тем дольше линия восстанавливается в единицу (идет перезаряд паразитной емкости между проводами) и тем сильнее заваливаются фронты импульсов, а значит скорость передачи падает. Именно поэтому у **I<sup>2</sup>C** скорость передачи намного ниже чем у **SPI**. Обычно IIC работает либо на скорости 10кбит/с — в медленном режиме, либо на 100кбит/с в быстром. Но в реальности можно плавно менять скорость вплоть до нуля.

Ни в коем случае **нельзя переключать вывод микроконтроллера в OUT и дергать ногу на +5**. Можно запросто словить КЗ и пожечь либо контроллер либо какой-нибудь девайс на шине. Мало ли кто там линию придавит.

Вся передача данных состоит из **Стартовой посылки, битов и стоповой посылки**. Порядок изменения уровня на шинах задает тип посылки.



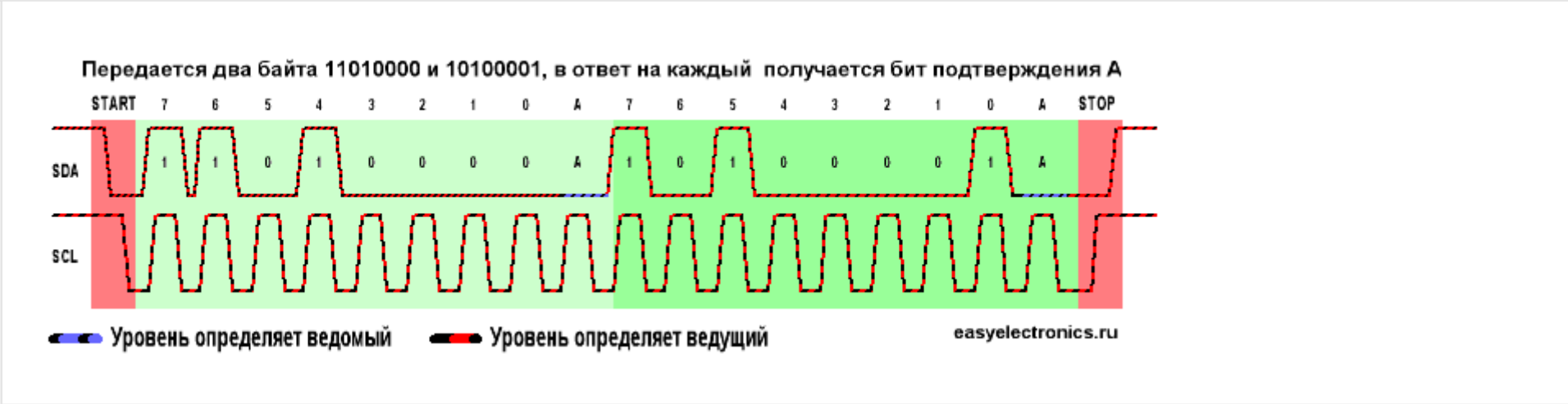


После **старта** передача одного бита данных идет по тактовому импульсу. То есть когда линия **SCL** в нуле **master** или **slave** выставляют бит на **SDA** (прижимают — если 0 или не прижимают — если 1 линию **SDA**) после чего **SCL** отпускается и master/slave считывают бит. Таким образом, у нас протокол **совершенно не зависит от временных интервалов**, только от тактовых битов. Поэтому шину **I<sup>2</sup>C** очень легко отлаживать — если что то не так, то достаточно снизить скорость до байта в минуту и спокойно, обычными вольтметрами, смотреть что у нас происходит. Правда это не прокатит с железным **I<sup>2</sup>C**, там нет таких низких скоростей. Но что нам мешает затактовать микроконтроллер от **ОЧЕНЬ** медленного тактового генератора и отладить все по шагам? ;)

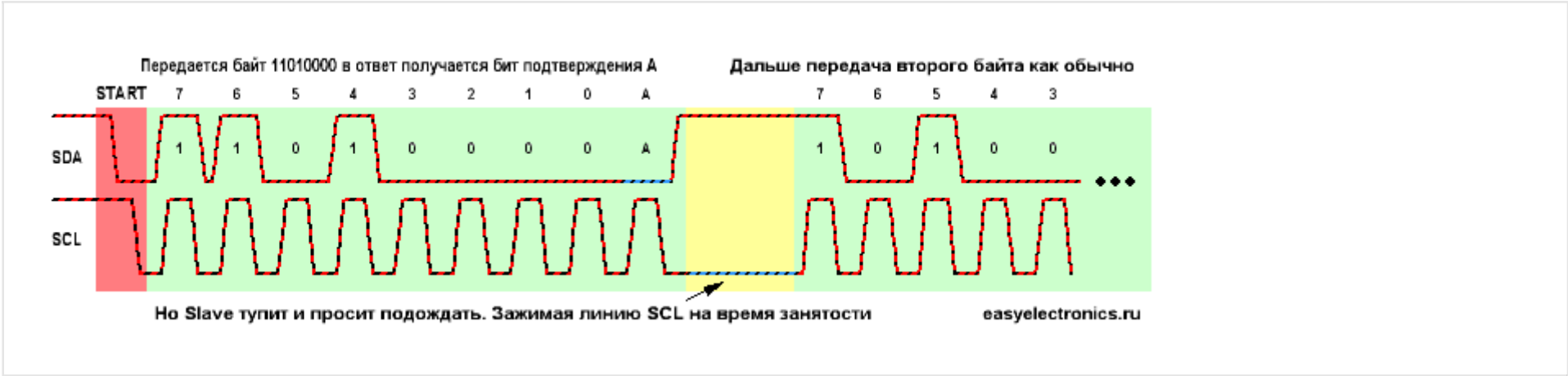
Повторим для ясности:

- Начало передачи определяется **Start** последовательностью — провал **SDA** при высоком уровне **SCL**
- При передаче информации от Master к Slave, **ведущий генерирует такты на SCL** и выдает биты на **SDA**. Которые **ведомый считывает когда SCL становится 1**.
- При передаче информации от Slave к Master, **ведущий генерирует такты на SCL** и смотрит что там ведомый творит с линией **SDA** — считывает данные. **А ведомый, когда SCL уходит в 0, выставляет на SDA бит, который мастер считывает когда поднимет SCL обратно**.
- Заканчивается все **STOP** последовательностью. Когда при высоком уровне на **SCL** линия **SDA** переходит с низкого на высокий уровень.

То есть, **изменение на шине данных в момент приема данных может быть только при низком уровне на SCL**. Когда **SCL** вверху то идет чтение. Если же у нас **SDA** меняется при высоком **SCL**, то это уже служебные команды **START** или **STOP**.



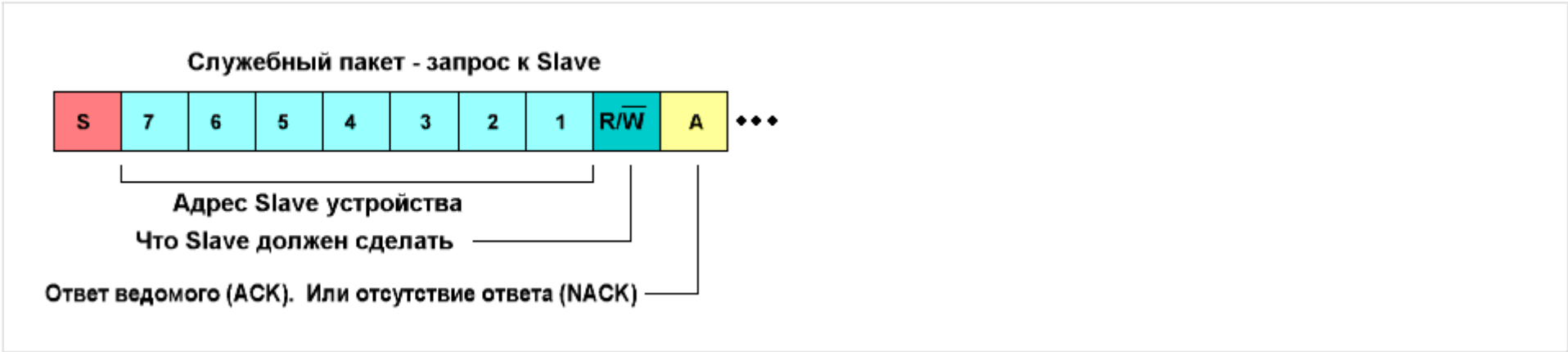
Если **Slave** тормозит и не успевает (у EEPROM, например, низкая скорость записи), то он может **наси́льно положить линию SCL в землю и не давать ведущему генерировать новые такты**. Мастер должен это понять и дать слейву прожевать байт. Так что нельзя тупо генерить такты, при отпускании **SCL** надо следить за тем, что линия поднялась. Если не поднялась, то надо остановиться и ждать до тех пор, пока Slave ее не отпустит. Потом продолжить с того же места.



Логический уровень

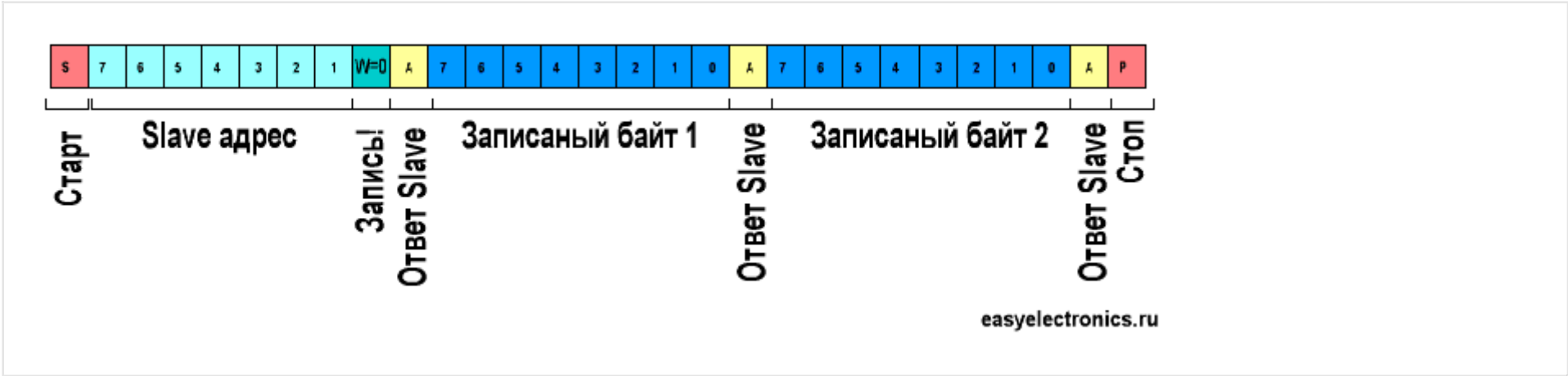
Как передаются отдельные биты понятно, теперь о том что эти биты значат. В отличии от **SPI** тут умная адресная структура. Данные шлются пакетами, каждый пакет состоит из девяти бит. 8 данных и 1 бит подтверждения/не подтверждения приема.

**Первый пакет** шлется от ведущего к ведомому это **физический адрес устройства и бит направления**.

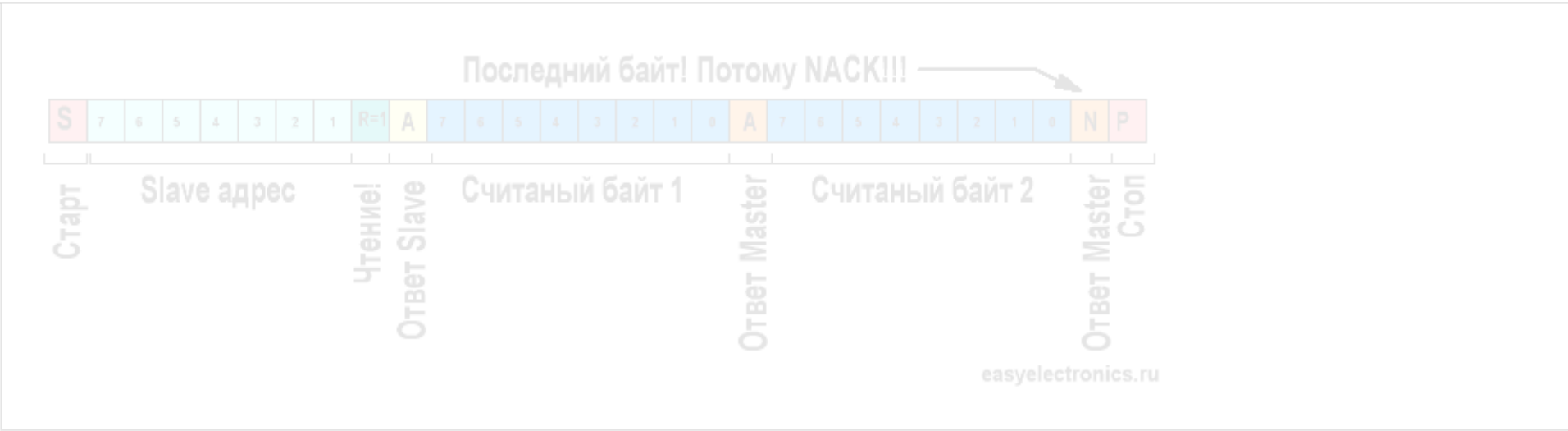


Сам адрес состоит из **семи бит** (вот почему до 127 устройств на шине), а **восьмой бит** означает что будет делать **Slave** на следующем байте — **принимать или передавать данные**. Девятым битом идет бит подтверждения **ACK**. Если Slave услышал свой адрес и считал полностью, то на девятом такте он придавит линию **SDA** в 0, сгенерировав **ACK** — то есть Понял! Мастер, заметя это, понимает, что все идет по плану и можно продолжать. Если **Slave** не обнаружился, прозевал адрес, неправильно принял байт, сгорел или еще что с ним случилось, то, соответственно, **SDA** на девятом такте будет прижать некому и **ACK** не получится. Будет **NACK**. Мастер с горя хлопнет водки и прекратит свои попытки до лучших времен.

После адресного пакета идут **пакеты с данными** в ту или другую сторону, в зависимости от **бита RW** в заголовочном пакете. Вот, например, Запись. В квадратиках идут номера битов. **W=0**



Чтение практически также, но тут есть одна тонкость из-за которой я когда то убил кучу времени. При приеме последнего байта надо дать ведомому понять, что в его услугах больше не нуждаемся и отослать NACK на последнем байте. Если отослать ACK то после стопа Master не отпустит линию — такой уж там конечный автомат. Так что прием двух байтов будет выглядеть так (**R=1**):



Есть еще одно состояние, как **повторный старт**. Это когда мы не объявляя **STOP** вкатываем на шину еще один **START**. После него мы можем обратиться к другому устройству не освобождая шину. Но чаще идет обращение к тому же самому устройству и это связано с особенностями организации памяти.

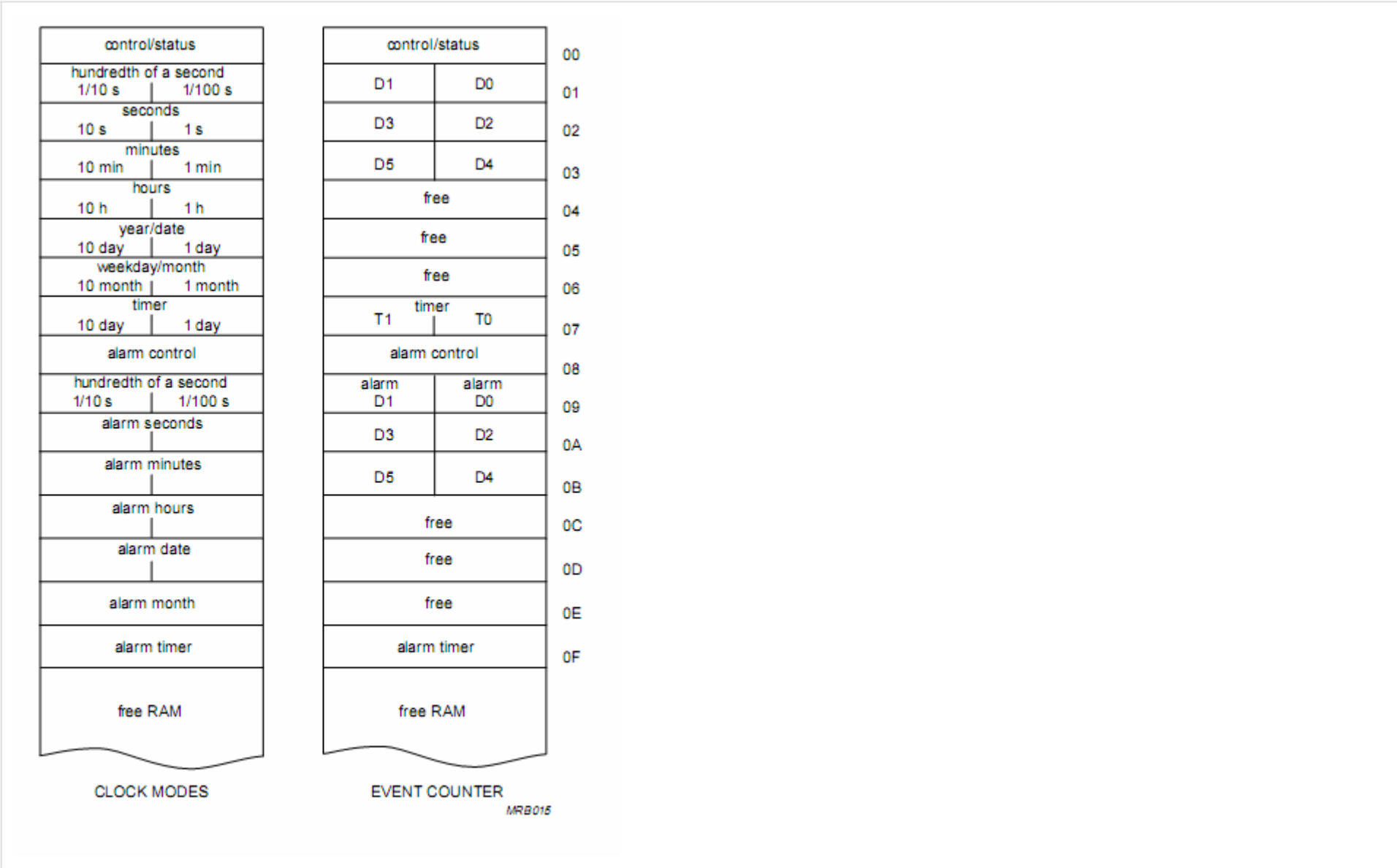
Организация памяти.

Это относится уже не столько к самому протоколу **I<sup>2</sup>C**, сколько к заморочкам создателей разных **EEPROM** и прочих **I<sup>2</sup>C** устройств. Но встречается это повсеместно, поэтому я расскажу про этот момент. Но, повторяюсь, это не аксиома, не стандарт и вообще зависит от конкретного **Slave** устройства. Так что датит в зубы и вкуривать, но обычно так принято.

Итак, о чем речь. Как видно из протокола, в первом байте мы **адресовываем само устройство**, а их может быть до 127 штук. Но в самом устройстве вполне может быть очень сложная структура, с кучей ячеек. Например **EEPROM** с килобайтами данных внутри. Как обращаться с этими данными? Не считывать же все по очереди от нуля до конца — это долго. Поэтому приняли хитрый формат. Это не стандарт, но юзается повсеместно.

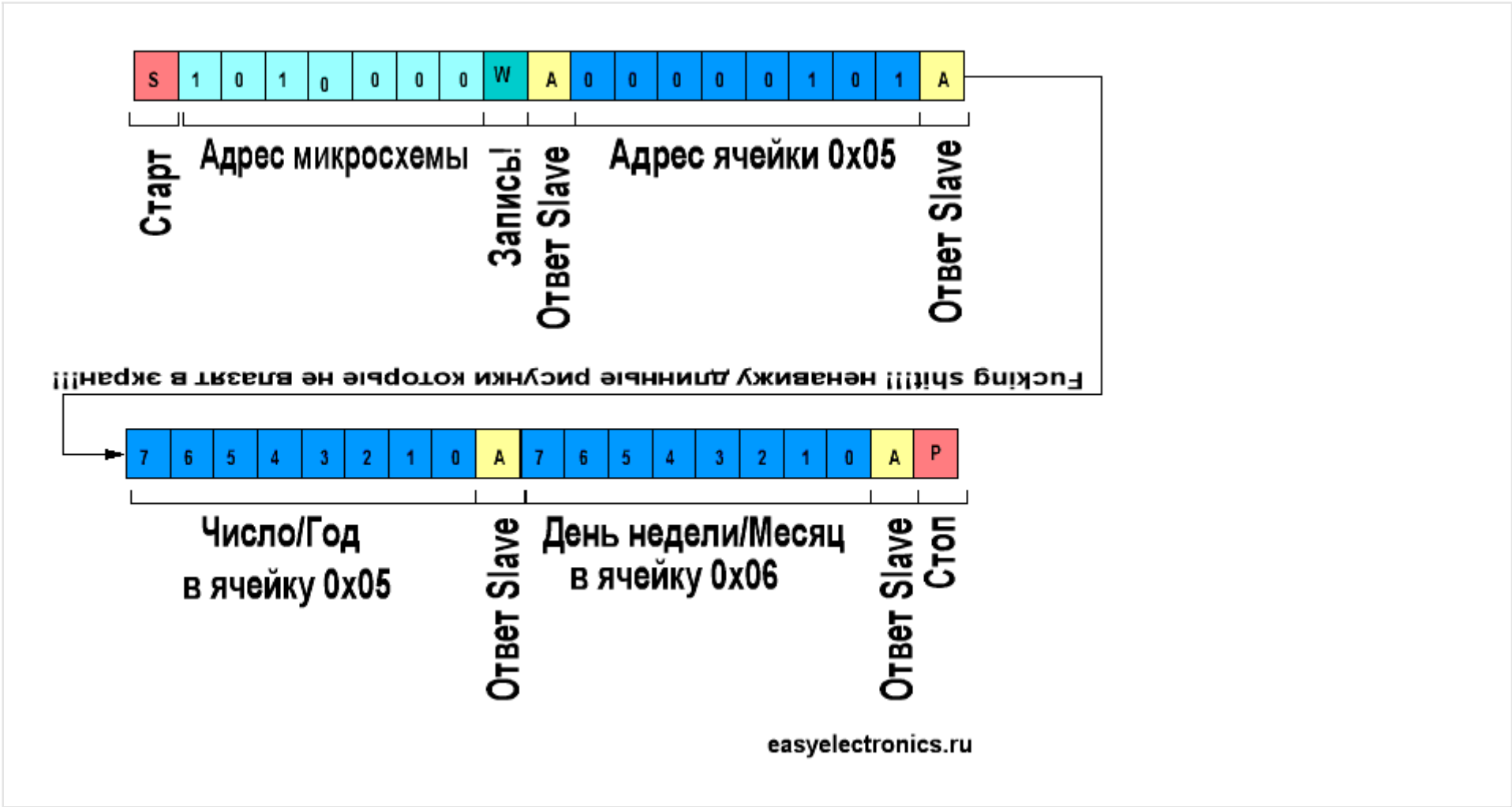
Поясню на примере:

Есть микросхема часов реального времени **PCF8583** (я про нее еще напишу, следите за обновлениями), общающаяся по протоколу **I<sup>2</sup>C**. Внутри нее ячейки памяти, в которых хранятся часы, минуты, секунды, дата, состояние флагов и куча еще всего. До кучи там еще 240 байт просто так, для свободного пользования. Карта адресов этой микросхемы выглядит так:



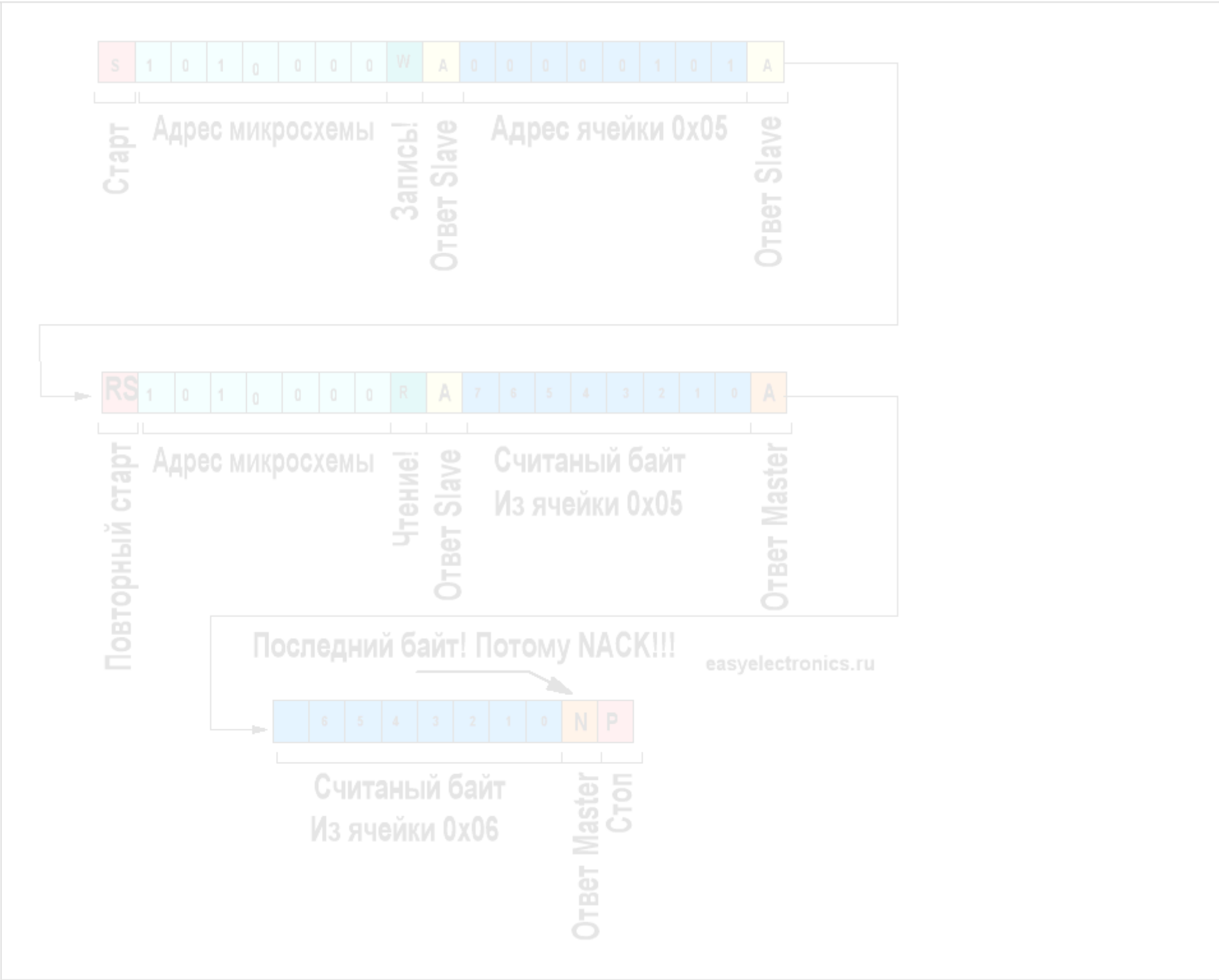
И вот надо мне установить дату. Для этого надо мне записать две ячейки памяти с адресами **0x05** и **0x06**. Как это сделать? Из даташита я узнаю, что **первый байт данных это адрес куда** мы будем обращаться, а потом уже идут данные и со следующим байтом счетчик адреса увеличивается на 1. Там же, в даташите, написано что эти часы откликаются на **Slave-адрес 1010000х** где **х** — **состояние ноги A0** микросхемы. Я эту ногу сразу посадил на 0 так что Slave-адрес у меня 10100000. Очевидно, что на одной шине может быть не более двух экземпляров этой микросхемы с адресами 10100000 и 10100001.

Задача решается так:

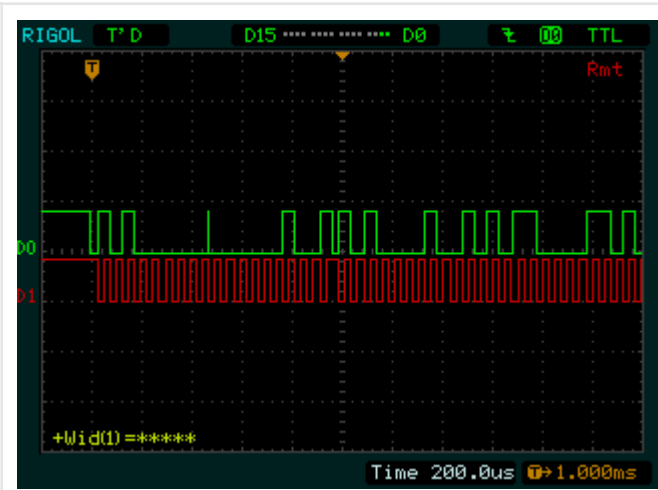


Вот и славно. Часы установлены и начали тикать. Но вот надо нам **считать** те же данные, а вдруг изменились? С записью все понятно — записали вначале адрес, а потом следом записали данные. А умная микросхема все прекрасно поняла и рассовала по ячейкам. А с чтением? А с чтением все через задницу, в смысле через запись.

То есть, мы, вначале, записываем один байт — адрес. Потом делаем повторный старт, затем снова обращаемся к часам по ее **Slave-адресу, но уже с битом R**, на чтение. И умная микруха выдает нам байты с адреса который мы в нее вот только что записали. Выглядит это так:



В этих часах так, у других микрух может быть все по другому, но в 99% очень похоже. Адрес, например, может быть двухбайт-ным или страницу надо будет указать, но сути это не меняет. Цепочка **запись-повстарт-чтение** это повсеместно. Вот так, кстати, выглядит чтение данных из часов PCF8583 на экране моего логического анализатора. Тут не полная посылка (все 5 байт просто не влезли в экран), но тут четко видно запись начального адреса, потом повторный старт, и чтение из девай-са.

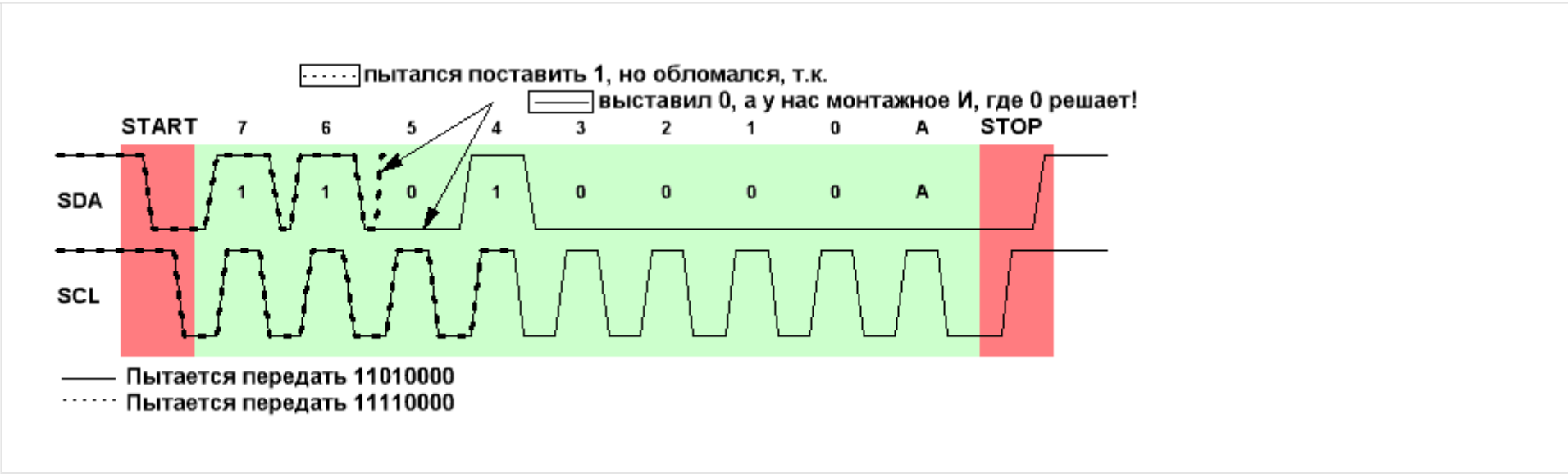


Скриншот с осциллографа RIGOL 1042CD

Арбитраж шины I<sup>2</sup>C.

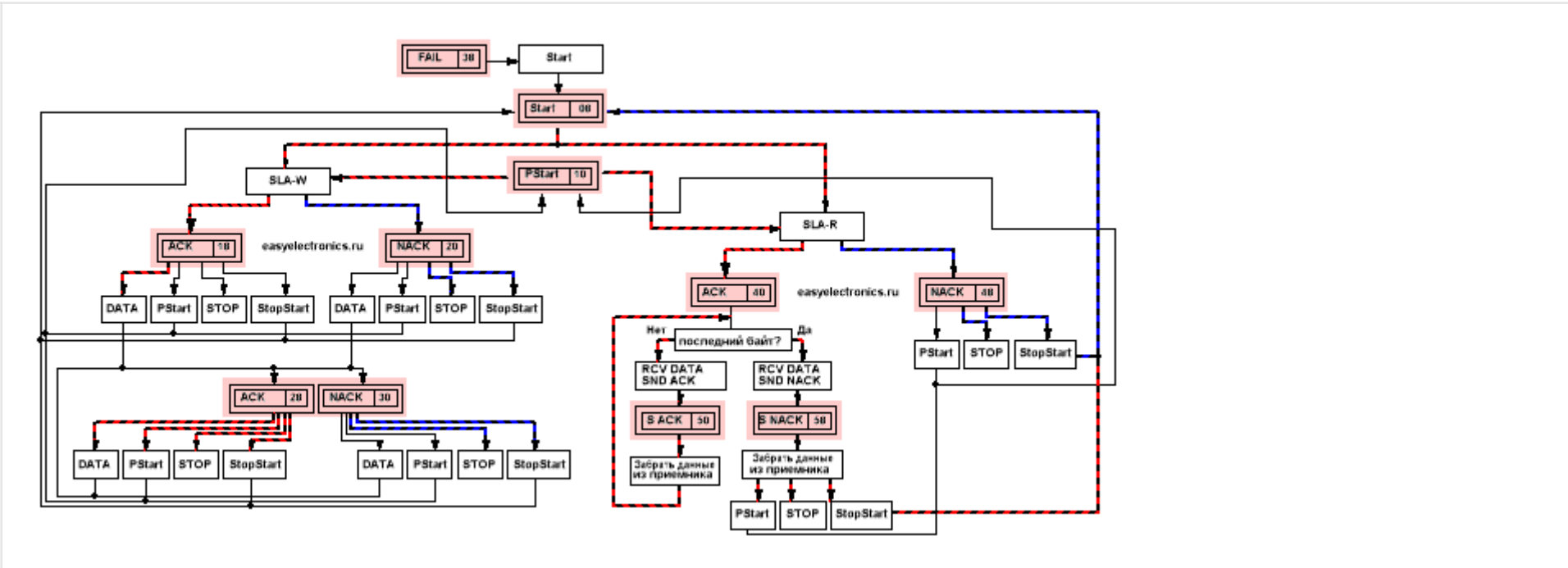
Почему то все мануалы для начинающих в которых рассматривалась тема протокола IIC как то ссыкливо замалчивают возмоз-ность работы двух ведущих на линии. **Master-Slave** и все тут. А если у нас демократия? И каждый сам себе **Master** и сам себе **Slave**? Согласен, редкий случай, но тем не менее, описывать так описывать. Короче, в случае подобного садо-мазо варианта действует железное правило — кто раньше встал того и тапки. В смысле кто первый начал вещать тот и текущий **Master**.





Но вот случилось вообще невероятное — **два Ведущих начали вещать одновременно**. Прямо совсем одновременно. Как быть? А тут нам поможет свойство монтажного И — где против **дома** нуля нет приема. Короче, оба мастера бит за битом играют в простую игру ножик-камень(1 и 0 соответственно). Кто первый выкинет камень против ножика тот и побеждает арбитраж, продолжая вещать дальше. Так что очевидно, что самый важный адрес должен начинаться с нулей, чтобы тот кто к нему пытался обращаться всегда выигрывал арбитраж. Проигравшая же сторона вынуждена ждать пока шина не освободится.

Вроде бы все, практический пример с **AVR** будет потом, а пока помедитируйте над диаграммой работы конечного автомата **TWI** передатчика ATmega8. Скоро я вас буду этим грузить!



Страшна? ;) На самом деле там все не так брутально. Можно обойтись вообще парой десятков строк кода на ассемблере.

UPD:

[Полный перевод оригинальной спецификации IIC от суровых Челябинских электронщиков с радиозавода Полет](#)

182 thoughts on “Интерфейсная шина IIC (I2C)”

- blacklion

16 апреля 2009 в 20:40

(1) OpenID работает криво. Я зашёл как blacklion.livejournal.com и всё равно справа-сверху «войти» (а не «выйти») и не даю комментировать. Пришлось регистрироваться и теперь тут два меня.

(2) Софтового мастера сделать не сложно. А вот слейва как-то просто не получается.



★ DI HALT

16 апреля 2009 в 21:08

Щас попробую с опен ид поиграться.

Более того, многие предпочитают делать софтовый I2C мастер чтобы не заморачиваться с встроенным в TWI конечным автоматом.

blacklion

16 апреля 2009 в 22:11

*Более того, многие предпочитают делать софтовый I2C мастер чтобы не заморачиваться с встроенным в TWI конечным автоматом.*

Я когда первый раз подходил к электронике смотрел на PIC (зачем я это делал!? Зачем на PIC?!) так реализовал мастера в качестве упражнения за вечер. А вот слейва так и не осилил...

★ DI HALT

16 апреля 2009 в 22:17

Я сделал софтового слейва на AT89C2051, но работало жутко медленно.

cahbtexhuk

16 апреля 2009 в 21:47

плюсадин. я уже раз 5 жаловался что логины глючат :D нашел выход, что залогинился, пару минуток подождал и реф-реш. работает, но бесит :\

KVorb

16 апреля 2009 в 21:22

DI HALT спасибо за статью. Давно ждал рассказ про I2C. :)

cahbtexhuk

16 апреля 2009 в 21:49

Могу поделицца опытом работы конкретно с DS1307. Он, сука, еще тот глючный девайс. Вроде просто, но не все просто. ПРОпиши бит СН, запусти часы, не забудь про бит снова, ой а чо горим — а мы забыли Vbat на землю положить...

★ DI HALT

16 апреля 2009 в 21:58

У меня их почти полный аналог ICL12008 ваще работать не хотят — не отзываются на свой адрес, не дают ACK

cahbtexhuk

17 апреля 2009 в 0:13

и у тебя, Брут? я изъебался с их инициализацией — не пашут и все. взял ds, который считал сгоревшим и не рабочим, по по недоразумению не выкинул — и он заработал. в топку эти ISL, от лукавого они..

★ DI HALT

17 апреля 2009 в 0:21

А у меня 5 штук их лежит — сэмплы. Надо бы в ST гневный мессадж накатать. МОл чо вы за говно нам подсунули?

Spirit0

17 апреля 2009 в 18:39

Да, есть такое дело :) Вообще при начале работы с IIC девайсом, желательно вдоль и поперек изучить Datasheet к нему.

В свое время тоже поломал голову с m41 от ST. Вроде все пишется, все читается, но часы стоят и все тут. Оказалось там есть хитрый битик, который при пропадании всего питания — Vcc и Vbat, останавливает часы и пока его не сбросишь они не пойдут.

Ну и еще обычный прикол с часами, что не все могут работать при отключенной батарееке — даже если Vcc есть, внутри у них стоит контроль, который проверяет разницу между напрягой и Vbat, и если Vbat нет, то часы просто ни на что не отвечают :)

p.s. а про NACK в конце чтения это да, самые популярные грабли наверно. Кстати для отлаживания шины если нет крутого осцила, можно прикрутить I2C Sniffer на Atmega8 — оч удобная штука, она в консоли показывает полностью весь обмен в удобоваримом виде.

ArgusB

18 апреля 2009 в 12:32

А чё с ним было голову ломать? Кстати, m41t56 рекомендую, простые и удобные I2C часы. Про битик я давно знал — он у всех часовых изделий от ST присутствует. Наверное, как совместимость с m48t08 — который с батареей на борту. Чтобы батарею не сажать, пока таймкипер лежит на складе, они глушат часы.

Sniper

16 апреля 2009 в 22:08

O)) На самом деле классная тема! Мне бы сначала с простым UARTом разобраться, а потом только к творению Philips)) Буду ждать исходников...

Thi3f

17 апреля 2009 в 3:07

Отличная статья! А есть последняя диаграмма только в чуть лучшем качестве? А то буквы трудно различить.

★ DI HALT

17 апреля 2009 в 4:02

<http://easyelectronics.ru/img/starters/IIC/IIC-state-diagramm.GIF>

Красные пути — нормальная работа  
Синие — возможные косяки.

Thi3f

17 апреля 2009 в 4:07

Спасибо, так намного лучше!

Medved

17 апреля 2009 в 11:48

Про 1-Wire напиши еще :-)

★ DI HALT

17 апреля 2009 в 19:24

Непременно :)

VPSec

27 декабря 2009 в 19:19

1-Wire хорошо описан тут: <http://www.radiokot.ru/articles/13/>  
А здесь хорошая статья про поиск устройств на шине 1-Wire: <http://hardisoft.ru/hard/poisk-ustroystv-na-shine-1-wire-na-assemblere-dlya-mikrokontrollerov-avr-firmy-atmel/>

ploop

17 апреля 2009 в 16:26

>>Ни в коем случае нельзя переключать вывод микроконтроллера в OUT и дергать ногу на +5. Можно запросто словить КЗ и пожечь либо контроллер либо какой-нибудь девайс на шине. Мало ли кто там линию придавит.

Разговаривали на эту тему с одним человеком. Он мысль сказал — если на каждую микросхему между ней и линией поставить резисторы по 100 Ом, то в принципе, проблема будет решена. Так же удобно при отладке.

★ DI HALT

17 апреля 2009 в 19:23

В принципе так и делают на линии с монтажным И. Про это в прошлой теме упоминалось.

Saniok

17 апреля 2009 в 17:41

Похожее разрешение коллизий используется в CAN... но там всёже гораздо более сложный протокол.

http://blrchger.livejournal.com/

18 апреля 2009 в 3:08

Ди, можно вопрос? Вот здесь вот ( [http://www.gaw.ru/html.cgi/txt/interface/iic/iic\\_4.htm](http://www.gaw.ru/html.cgi/txt/interface/iic/iic_4.htm) ) написано, что скорость передачи данных — 100 кбит/с в обычном режиме, у тебя — 10 кбит/с. Кому верить?

★ DI HALT

18 апреля 2009 в 3:15

У меня написано 10кбит в медленном и 100 в быстром. Обычный режим это быстрый =). А щас еще новая спецификация вышла — там интерфейс разогнали до 400

cahbtexhuk

19 апреля 2009 в 17:52

эээ... О\_о? А я всю жизнь и во всех даташитах видел 100 и 400 как простой и быстырь. Ну не важно

Ridik911

19 апреля 2009 в 15:00

спасибс  
ждемс также как работать с SRAM (и dram)  
(кланюсь.. и пока ушел)

rc-dimon

21 апреля 2009 в 11:47

Спасибо за очень подробную статью. Хотелось бы очень увидеть такую же по SPI

Ivan

27 апреля 2009 в 12:32

«Передача/Прием сигналов осуществляется прижиманием линии в 0, в единичку устанавливается сама, за счет подтягивающих резисторов.»  
  
PORTx.x = 0; //Прижал в 0  
  
тут код

PORTx.x = 1; //Пока так не сделаю, в единицу не устанавливается, подтяжка есть 4k7

Настройка порта:  
PORTC=0x00;  
DDRC=0xFF;

И сразу еще один вопрос, как передать в процедуру номер ноги контроллера?  
пробовал так:

```
void my_proc(unsigned char _port)
{
}
```

вызываю так:

my\_proc(PORTB.3) и не работает...

★ DI HALT

27 апреля 2009 в 14:47

Ногу надо передавать по отдельности. Т.к. PORTB это адрес порта. А 3 — просто номер пина. Так что либо в две переменные либо просто номер пина, а порт уже прописан в функции. Вот только нафига тебе тут функция? Напрямую то нельзя чтоль?

Ivan

27 апреля 2009 в 15:38

Не, напрямую нельзя, нога заранее неизвестна (и порт тоже).  
Интересно, если передать номер порта и ноги отдельно, как их потом лучше всего объединить?

Блин, каша в голове (только учусь) на работу с датчиком ds18b20 как должен быть настроен порт? //Заранее спасибо!

Разобрался... Proteus 7.4 SP3 с краком 1.6.3 — 6.5 от Nemo глючит сильно ...  
скачал 7.5 — ВСЕ СРАЗУ ЗАРАБОТАЛО!! Начинаящим, как и я, обновлять обязательно!  
Вопрос по передаче номера ноги в процедуру в силе, буду благодарен за пример.

vados

22 мая 2009 в 1:30

DI HALT Привет! Очень полезна статья, вот недавно наткнулся на убитую магнитолу с микросхемой LC75373(аудиопроцес-сор), так вот там управление осуществляется, как я понял, что-то наподобие I2C (как написано в datasheet — serial data input). Единственное отличие это не две линии а три — CE, DI и CL. Как я понял управление осуществляется в одну сторо-ну, т.е. только на прием. Вот единственное что я хотел у тебя спросить, возможно ты когда-нибудь сталкивался с подобием такой микрухи, хотел узнать управление происходит только единичной посылкой управления, или возможно как-то по-другому. В datasheet не нашел вразумительного ответа. Конечно вопрос не совсем в тему, но все-таки надеюсь на отклик.

★ DI HALT

22 мая 2009 в 1:51

Это не I2C это почти чистый SPI только в одну сторону.  
Формат данных там написано что 52 бита — 8 адреса и 44 данные.  
Точнее как, вначале CE ложишь вниз, подаешь 8 бит адреса. Потом поднимаешь CE вверх и гонишь 44 бита данных. В этих 44 битах закодированы параметры для резисторных матриц. Как переслал CE бросаешь вниз и при этом происхо-дит запоминание введенных данных.

Как и что слать там нормально написано.Я бы сказал даже очень наглядно нарисовано. На 15й странице даташита

**vados**

22 мая 2009 в 2:05

Спасибо! В принципе так и думал, единственное что до сих пор не внушает доверие в datasheet так это то что получается CE нужно ложить вниз не на 8 бит адреса, а на 9 (или возможно это опечатка), т.е. захватывает старший бит данных D0 (если верить рисунку осциллограммы). Но конечно судя по логике все-таки скорее правильное как ты написал — CE ложишь вниз, подаешь 8 бит адреса.

**Cluster**

9 июля 2009 в 3:29

Жду примера для AVR, а то у меня как-то руки не дошли разобраться в этом :)

**★ DI HALT**

9 июля 2009 в 14:41

Есть уже пример. Тему про часы реального времени посмотри. Там правда на ассемблере, но думаю тебя это не остановит ;)

**voltAVR**

15 сентября 2009 в 22:57

Здравствуйте, подскажите как можно организовать контроль I2C шины, если например есть N-ое количество устройств и одно из них придавило линию и например зависло в таком состоянии, есть ли возможность узнать какое программно, а лучше аппаратно отключить его или показать что оно не работает.

**★ DI HALT**

15 сентября 2009 в 23:01

Со стороны мастера никак. Т.к. придавленная линия где бы то ни было означает облом в коммуникациях. А нет связи — нет мультиков.

Ставить на каждое устройство вачдог. Если девайс не сообщил собаке, что он отпустил линию, то собака укусит его за задницу.

**voltAVR**

15 сентября 2009 в 23:27

Спасибо!

**SAShA\_II**

1 ноября 2009 в 4:50

Вот интересно. Вопрос, конечно, может и дурацкий. А можно ли сделать программное подобие I2C используя USART? Соединив все TX слейвов с RX мастера и все RX слейвов с TX мастера. Будут ли слейвы все слейвы одновременно принимать данные от мастера или есть какие-то подводные камни в этом? I2C мне не подходит из-за ограничения по количеству устройств.

**sseett**

17 декабря 2009 в 12:08

Прекрасное описание шины. Живой язык, яркие рисунки.

По моему, в статье есть неточность.

Автор пишет:

«Если Slave тормоз и не успевает (у EEPROM, например, низкая скорость записи), то он может насильно положить линию SCL в землю и не давать ведущему генерировать новые такты. Мастер должен это понять и дать слейву прожевать байт. Так что нельзя тупо генерить такты, при отпускании SCL надо следить за тем, что линия поднялась. Если не поднялась, то надо остановиться и ждать до тех пор, пока Slave ее не отпустит. Потом продолжить с того же места.»

Пример с EEPROM мне показался неудачным, т.к. обычно у них к SCL только вход.  
Вход же не может притянуть SCL к земле.  
Но для многих других I2C микрух этот вариант прокатывает.

Спасибо автору за труд.

★ DI HALT

17 декабря 2009 в 12:31

Разве??? Впервые про то что слейв девайс может не успеть прожевать данные и прижмет линию я услышал именно при описании работы с еепромкой и2с шной.

sseett

17 декабря 2009 в 14:02

Вы согласны, что вход не может притянуть линию к земле.

[http://www.gaw.ru/pdf/Atmel/at24/AT24C128\\_256.pdf](http://www.gaw.ru/pdf/Atmel/at24/AT24C128_256.pdf)

Table 1. Pin Configuration  
Pin Name Function  
A0 — A1 Address Inputs  
SDA Serial Data  
SCL Serial Clock Input -вход  
WP Write Protect  
NC No Connect  
GND Ground

Смотрите еще Figure 1. Block Diagram  
Там на SDA висит транзистор, а на SCL – нет.  
Если слейв занят внутренними проблемами, то он просто отключается от шины и не реагирует на мастера. Мастер должен отправить условие старта после адресного слова и ждать ответа.  
Вот тут на русском: [http://www.gaw.ru/html.cgi/txt/doc/memory/seeeprom/at24c128\\_256.htm](http://www.gaw.ru/html.cgi/txt/doc/memory/seeeprom/at24c128_256.htm)

★ DI HALT

17 декабря 2009 в 22:05

Да. В самом деле, вы правы. Видимо тут такой подход связан с тем, что запись буфера будет очень долгой чтобы на это время нельзя парализовывать работу шины.

AlexRu1

29 декабря 2009 в 9:13

День добрый. Помогите пожалуйста с ds1307. Уже третий день мучаю ее, запустить не могу — в протеусе работает «как часы».), а в железе — на экране 2.1.2.1 Осца временно нету, посему пытался проверить путем конфигурации ее на вывод SQW с частотой 1 Гц — генерация отсутствует. Здесь я краем уха слышал по поводу батарейного питания — может у меня в этом проблема??? Вывод Vbat у меня висит в воздухе. Заранее спасибо.

★ DI HALT

30 декабря 2009 в 17:59

Без батарейного питания они не стартанут. Либо будут в какомнибудь энергосберегающем режиме висеть. Батарею подцепи, тогда пойдут. Может еще какой регистр внутри надо будет дергнуть. Но это даташит кури.

AlexRu1

12 января 2010 в 4:12

Сами часики вроде в норме. Проблема с контроллером (точнее с программой). Я отключил контроллер от часиков, далее запускаю и проверяю осцом что у меня на SDA и SCL. В итоге вообще запутался — идет передача данных, а в перерывах



шина лежит в нуле. Почему проц может не отпускать шину???? (чтение данных спецом отрубил — контроллер в режиме мастер-передатчик). И как тут можно выложить кусок кода? может кто-то что-то поможет...А то беда — контроллер весь учу всего пару месяцев, а тут на одной шине i2c зависнул уже на две недели.

★ DI HALT

12 января 2010 в 4:25

Последний байт NACK?

AlexRu1

12 января 2010 в 4:44

По поводу NACK:

Вот тут меня мучает такой вопрос:

TWCR — он же не буферизированный? Как же тогда в сях одной командой установить TWEN, TWINT и сбросить TWEA (режим NOASK)???

Я так понимаю — в таком порядке не пройдет:

TWCR &= ~(1<<TWEA)  
TWCR |= ((1<<TWINT) | (1<<TWEN))

по идее в обратном:

TWCR |= ((1<<TWINT) | (1<<TWEN))  
TWCR &= ~(1<<TWEA)

тоже не должно работать. Ведь и в том и в другом случае мы обнуляем TWINT первой командой и сразу начинается работа модуля.

Прав я или где-то запутался???

может так? :

TWCR &= ~((1<<TWEA) | (1<<TWINT))  
TWCR |= ((1<<TWINT) | (1<<TWEN))

Но я даже убрал вообще прием — оставил только передачу (инициализация). Когда идет прием — тогда поинтересней. Вот мой маин:

```
while(1)
{
  read_rtc ();
  _delay_ms (500);
  shou_clock ();
}
```

так на осце идет меандр с периодом, который задан в задержке (мс 500) а передача идет коротко на спаде и фронте импульсов. То есть:

шина в «1!  
чтение  
шина не поднялась  
500 мс  
шина не поднялась  
чтение  
шина в «1» и тд

★ DI HALT

12 января 2010 в 5:16

А что тебе мешает сформировать сразу весь байт?

```
1 TWCR=( 0<<TWEA | 1<<TWINT | 1<<TWEN ) ;
```

AlexRu1

12 января 2010 в 5:29

А разве в WinAVR такой вариант потянет? Как я понимаю — gcc сильно заточен под стандарты языков, а если данную строку разложить, то получим несурязицу...хотя он компилирует. Нет что-то мне кажется, что так можно еще больше запутаться.

★ DI HALT

12 января 2010 в 5:31

Гхм. Срочно! Учить матчасть!!!

Вот такая запись для него как раз родней некуда. Обычное присваивание байта байту. BCE.

AlexRu1

12 января 2010 в 5:37

Я просто не заметил что там = а не |= вот и удивился.

AlexRu1

12 января 2010 в 5:38

Так а если через бит-маску — какой из трех вариантов канает?

★ DI HALT

12 января 2010 в 8:16

Через XOR можно сделать обычную инверсионную маску. Только зачем? Присваивай и все.

AlexRu1

12 января 2010 в 10:53

Суть я уловил. Я в принципе догадывался что надо использовать XOR. Зачем — другой вопрос. Хочу отточить навыки и разобраться со всем (а лог арифметика сюда также входит) и, самое главное, выработать один стиль. Ведь, насколько я знаю, простое присваивание не всегда катит. Нужно следить за всеми регистрами, а для буферизированных — битмаска — что доктор прописал (не пинайте асемблерщики). Вот и пытаюсь на одном примере (раньше взял очередной модуль, написал прогу и пошел дальше, а тут застрял) отточить многие нюансы).

Еще пара вопросов:

- 1) Если я отключил все устройства (оставил один контроллер) — он ведь после всех попыток общение по iic должен отпустить шину??? А резисторы ее притянуть к «1»? Верно?
- 2) С какой частотой надо опрашивать RTC, что-бы, как говориться, и волки сыты и овцы целы?

★ DI HALT

12 января 2010 в 11:35

Чтобы контроллер отпустил шину он должен выйти из конечного автомата. (погляди на диаграмму работы) для этого ему надо получить одно из состояний. Единственно что если ты ТВИ вырубешь вообще, то он должен шину отпустить. Ну а дальше резисторы.

2) С какой тебе надо с такой и опрашивай. Но, сам понимаешь, нет смысла спрашивать чаще чем меняется самый младший разряд часов. А вообще я его спрашиваю когда надо получить значение. Разово, не циклически. Зачем зря шину занимать?

AlexRu1

12 января 2010 в 13:19

По поводу первого вопроса:  
Я отключил все на шине TWI. Перевожу контроллер в режим мастер-передатчик, передаю байт и даю команду СТОП. Контроллер при этом должен в любом случае отпустить шину??? В режиме передатчика это условие является абсолютным???

★ DI HALT

12 января 2010 в 16:26

Не помню. Надо посмотреть на диаграмму переходов.

AlexRu1

12 января 2010 в 18:22

Все РАБОТАЕТ!!!! Спасибо. Проблема в NACK:

Заменял  
TWCR &= ~(1<<TWEA)  
TWCR |= ((1<<TWINT) | (1<<TWEN))

на

TWCR=(0<<TWEA|1<<TWINT|1<<TWEN);  
и все пошло.

AlexRu1

12 января 2010 в 5:33

Да, здесь обновляется весь регистр. Так прокатит. А вот бит-маску вроде так не наложить.

★ DI HALT

12 января 2010 в 8:16

Можно сделать хитрую маску которая инверснет нужные биты. ПО ксору.

AlexRu1

12 января 2010 в 18:50

Всем спасибо за помощь.

redradist

21 февраля 2010 в 4:11

Вопрос: а можно ли осуществлять по данному протоколу следующие действия:  
Старт, передача адреса устройства + W, передача адреса ячейки, ПОВ-Старт, передача адреса устройства + R, считывание ячейки, считывание следующей ячейки,(вот со следующего действия и начинаются проблемы) ПОВ-Старт, передача адреса устройства + W, передача адреса ячейки, запись одного байта, запись второго байта, СТОП ... Моделирую в протесе, нихера не получается, а до этого все нормально ... Помогите ...

★ DI HALT

21 февраля 2010 в 8:06

считывание ячейки, считывание следующей ячейки, [?????ACK или NACK?????](вот со следующего действия и начинаются проблемы)Повстарт...

Должен быть NACK иначе линию не отдаст.

|   |
|---|
| <div><div>redradist</div><div>21 февраля 2010 в 17:12</div><div>DI HALT, уже делал НАК, он передает, но результат на индикаторах не тот ... (((</div></div>   |
| <div><div>Stydent</div><div>16 марта 2010 в 15:14</div><div>DI HALT, посоветуй пожалуйста! ломаю голову не знаю какой интерфейс выбрать IIC или SPI для связи 17 МК(каждый принимает и передает).</div></div>   |
| <div><div>★ DI HALT</div><div>16 марта 2010 в 17:05</div><div>тут только i2c или 1-wire spi тут в принципе не годится.</div></div>  |
| <div><div>Stydent</div><div>16 марта 2010 в 18:13</div><div>Хочу уточнить: один МК связан с 16-тью МК ?</div></div>   |
| <div><div>★ DI HALT</div><div>16 марта 2010 в 18:43</div><div>Там шина, где все МК разноправные. В SPI же может быть связь только между двумя устройствами.</div></div>   |
| <div><div>Stydent</div><div>16 марта 2010 в 22:00</div><div>Спасибо большое за помощь!</div></div>  |
| <div><div>Stydent</div><div>17 марта 2010 в 16:02</div><div>DI HALT, подскажи пожалуйста, а что будет происходить, если в момент передачи информации возникнет не запрещенное прерывание в ведущем передающем МК? и 2-й вопрос если связь будет прост оборвана, сохранятся ли в ведомом приемнике, успевшие прийти до прерывания,байты?</div></div> |
| <div><div>★ DI HALT</div><div>17 марта 2010 в 17:38</div><div>Смотря как реализован мастер. Если аппаратно, то он дошлет текущий байт и выставит нужные уровни сам. Просто передача чутко встрянет на время обработки прерывания. ИИС это же синхронная шина и остановка мастера дает лишь остановку передачи, а не срыв передачи.</div></div>      |
| <div><div>Stydent</div><div>21 марта 2010 в 15:58</div><div>DI HALT, помоги разобраться, после каждой операции модулем TWI устанавливает флаг TWINT, дальше необходимо проверить код статуса (размер 1байт), на сколько я понял этот код надо искать в регистре TWSR в пяти битах TWS. Как может хранится код \$38 в пяти битах ?</div></div>       |
| <div><div>★ DI HALT</div><div>21 марта 2010 в 16:16</div><div>Почему в пяти битах?</div></div>  |

Stydent

21 марта 2010 в 19:53

Регистр TWSR состоит из след. битов: 0-1 — это коэф деления предделителя,2 — зарезервирован, 3-7 — биты состояния модуля TWI (5 битов TWS). В TWS должен же хранится код статуса, или нет?

★ DI HALT

21 марта 2010 в 20:29

2 может и зарезервирован, но он все равно читается как 0. Обрати внимание, все коды имеют в младшем разряде 0. Это и есть этот зарезервированный бит.

Так что просто сдвигаешь вправо на два бита значение TWSR и вот тебе код статуса.

mmm

26 марта 2010 в 22:13

Очень доходчиво! Коротко и понятно. Огромное спасибо за популярное объяснение. Читал 2 дня официалку — недопер). Дым из башки валит а понимания «0».

kingdomof

4 мая 2010 в 11:45

Здравствуйте!

Можно ли связать через интерфейс I2C (TWI) несколько МК Attiny13 (Slave) и Atmega64 (Master)? Если нет то, какие интерфейсы лучше использовать?

★ DI HALT

4 мая 2010 в 22:13

Можно

kingdomof

5 мая 2010 в 9:02

Разве в данном случае не обязательно чтобы Attiny13 поддерживал I2C (TWI), т.к. в даташите нету информации, что они поддерживают TWI.

Проблема в том что, я думаю что если использовать I2C, то те приборы (датчики и т.п.) которые подключаются к этой шине должны поддерживать I2C.

★ DI HALT

5 мая 2010 в 9:15

Если в контроллере чего то нет это всегда можно сделать программно. Хотя I2C Slave программно сделать сложно, но можно. Ну и придется все писать на ассемблере и очень жоско оптимизировать, т.к. памяти там совсем мало. Есть другие 8ми ножки с большим числом памяти (Тини88) с ними будет проще.

kingdomof

5 мая 2010 в 15:31

По твоему совету поискал ATtiny88 в <http://www.chip-dip.ru>, но не нашел. Подскажи, плиз что лучше использовать из дешевых МК для I2C в стиле Plug&Play

★ DI HALT

5 мая 2010 в 23:10

Да я прогнал. Я имел ввиду тини25/45/85 (они различаются только числом памяти) У него не совсем полная поддержка

TWI, а скорей полуфабрикат — USI, но самую сложную задачу: ловлю стартов/стопов на slave и отлов байтов аппаратно он тебе делает.

Самый же дешевый из тинек, пожалуй это Тини2313 но она на 20 ног.

**maze**

17 июня 2010 в 23:49

А если сделать несколько модулей с подтяжкой на каждом, то будет ли работать система с включенными несколькими модулями?

★ **DI HALT**

18 июня 2010 в 0:00

Если подтяжка не сильно мощная, такая что ее вывод контроллера пересилит то без проблем. Стандарт допускает.

**SergeyB**

3 октября 2010 в 19:57

Режим fast 400kbit/s подразумевает 400 Кгц по линии SCL или же 400кбит чистой информации в секунду?

★ **DI HALT**

4 октября 2010 в 13:58

SCL но за один период SCL проходит один бит.

**SergeyB**

4 октября 2010 в 20:00

ага, т.е. 400кбит/с означает что за секунду проходит:  
(кол-во старт битов)+(количество стоп битов)+(кол-во Ack/Nack)+(кол-во бит данных)=400кбит

**vlad49**

5 января 2011 в 3:15

Прочитал на твоём сайте статью про 1 wire протокол. Самая понятная и доходчивая из тех что я знаю. Сегодня хотел поделиться инфой с другом и не смог найти.Третий час рыскаю по сайту. Помогите, подскажи где она?

★ **DI HALT**

5 января 2011 в 4:08

<http://easyelectronics.ru/klyuch-ot-vsex-dverej-emulyator-klyuchej-ot-domofona.html>

Эта?

**Alesandr**

15 февраля 2011 в 16:09

Кто каритнки рисовал? В некоторых стоповый бит выделен цветом а по уровням невнопад.

★ **DI HALT**

15 февраля 2011 в 18:41

Я картинки рисовал. Так и задумано. Чего непонятного?

**Alesandr**

15 февраля 2011 в 19:15



SDA линия не меняет свой уровень с низкого на высокий а остается постоянной в двух верхних цветных картинках. а это уже не стоповый бит. после стопового бита уровень должен уйти вниз что бы при высоком уровне SCL подняться вверх.

Alesandr

15 февраля 2011 в 19:17

не стоповая последовательность.

★ DI HALT

15 февраля 2011 в 19:36

Да, признаю. Там не стоповая последовательность, а просто выход из автомата по NACK и конец передачи. Надо бы по другому обозовать.

AleksanderG

15 февраля 2011 в 19:28

DI HALT большая просьба. Открыть (написать) тему про работу SD карты и МЭГи 32. Подключение, обмен, исходный код для обучения (желательно на ассемблере), в общем все так, как ты это делаешь доходчиво и со вкусом.

d-lun

29 апреля 2011 в 19:38

DI, большое спасибо за диаграмму состояний автомата I2C! Сразу доходит, как все работает.

★ DI HALT

29 апреля 2011 в 19:44

Старался :)

Тут еще погляди, если не видел:  
<http://easyelectronics.ru/avr-uchebnyj-kurs-ispolzovanie-avr-twi-dlya-raboty-s-shinoy-iic-i2c.html>

d-lun

29 апреля 2011 в 20:10

А, забыл спросить, что значит StopStart на диаграмме?

★ DI HALT

29 апреля 2011 в 20:16

То и значит. Будет сначала сделан Stop, а потом сразу Start — например чтобы постучаться к другому slave

d-lun

29 апреля 2011 в 20:18

Ясно. Спасибо за помощь:)

valio

30 мая 2011 в 19:18

> .. Схема подключения — монтажное И.

Почему «И» не пойму ? Косяк, это т.н. монтажное «ИЛИ» называется или «wired OR». То бишь А V В в записи или еще есть такое слово — дизъюнкция, или логическое «ИЛИ», иногда включающее «ИЛИ», еще обзывают как логическим сложение, чаще просто «ИЛИ».

А насчет 100R резюков в сигнальной шине перед слейвом, де-факто прописано в многих апнотах по iic, тот же STM если посмотреть.

★ DI HALT

30 мая 2011 в 19:36

Не тупи. Это именно «И» т.к. если один из слейвов дал на шину 0 то по всей шине будет 0.

В каком апноте по iic есть резюк перед слейвом? Покажи пример. Ни разу такого не видел.

valio

30 мая 2011 в 19:57

Точно, замкнуло самого) Принудительная к земле подтяжка, в записи 0/1 = 0.

buddah

9 ноября 2011 в 1:48

Здравствуйте!

Работаю с PIC16Fxxx,I2C программный, работает !!!, как на чтение, так и на запись, с этим порядок.И все,вроде бы просто —

```
lab
call read_rtc ;чтение DS1307.
call display ;отображение прочитанного (динамическая индикация).
goto lab
```

, но при таком варианте — на дисплее одна динамическая белеберда.  
А вот если читаю RTC один раз, а отображение в цикле, то все нормуль.  
И паузы втыкал после call read\_rtc и регистры чистил после call display (нафиг, правда не понятно...). В общем, RTC тикают,правильно тикают,а читать могу ОДИН, блин, раз. Понятное дело, чтение будет по прерыванию, но блин, чего оно не читается в цикле..? Может кто-то сталкивался с такой какой, помогите плиз. (...а инета скока перерыл, ниче подобного не нашел.)

ArxangelRUS

14 ноября 2011 в 22:57

Про бит подтверждения А. Как то странно работает, он что инверстно пашет? т.е. если посылка удалась, то ведомый на линии не выставляет высокого уровня?

Orienta

29 ноября 2011 в 18:26

Эх, DI HALT, как же мне нравится Ваша манера объяснять и непринужденный язык. Читаю сайт как произведение искусства :-)

Arseniy Muradov

20 мая 2012 в 1:28

ээээ так?)) <http://easyelectronics.ru/repository.php?act=view&id=87>

★ DI HALT

20 мая 2012 в 1:36

Что это за ад ваще? Это обработчик прерываний? Если да, то какого хрена ты в нем в цикле крутишься? Сами прерывания инициализированы? Вектора прописаны? Если хочешь в лоб, то возьми мой пример и просмотри как он себя ведет.

Arseniy Muradov

20 мая 2012 в 1:39

не, это в лоб. там нет прерываний))  
посмотрел вот это  
<http://easyelectronics.ru/chasy-realnogo-vremeni-pcf8583.html>  
вот это такой макрос, который ты назвал «IIC\_BYTE»  
Отличий собственно не очень много:)

Arseniy Muradov

20 мая 2012 в 1:45

Вся проблема как раз в том, что програма нормально работает, но в какой-то момент twi перестает выставлять флаг TWINT и все зависает.  
Не понимаю откуда беерться ошибка 0x00 и зачем он придавливает линию в ноль.  
И это не зависит от того есть кто на линии или нет. Все и так и так — одно и тоже... :(

Раньше такой способ работал :(

★ DI HALT

20 мая 2012 в 2:14

А какого хрена там тогда в конце reti? Если у тебя прерываний нет и ты его не вызывал, то эта строчка сорвет тебе стек.

Arseniy Muradov

20 мая 2012 в 2:21

Ну не ругайся))))  
Это не прерывание, это вызов, но не в этом суть))  
С этим все ок))  
Меня волнует почему twi зависает и выдает ошибку 00 когда я передаю бит...  
Причем не всегда. иногда он проходит. вообще я во время этого даже стараюсь не дышать — раз сработает, два сработает, а потом зависнет. иногда сразу. в общем полный рандом :(

и этот бит TWWC — он чего забыл тут тоже не понятно :(

★ DI HALT

20 мая 2012 в 2:28

Я не ругаюсь, а луплю за дурной стиль :) не стоит юзать reti там где неположено :)

А автомат работает железно. Ты его пнул, он завертелся. По идее он должен отработать и на пустой линии, по крайней мере выдать старт последовательность и получить ответ, что там никого нет. Полковнику никто не пишет и так далее. Попробуй. Бусфейл это серьезная лажа, может конфликт, может емкость или помехи. Прогони на пустой линии без всего вообще. Если нормально будет выходить на старт последовательность, то проблема у тебя не в проге, а в линии. Тут без осцила ты много не наловишь.

Жтаг вещь хорошая, но он (особенно в авр) кривоват и может косячить.

Нафаршируй код отладочным выводом состояний автомата в некий буфер в ОЗУ ,а по переполнению выдавай его в уарт. Увидишь полную картину последовательности кодов, после чего ты получаешь фейл.

Arseniy Muradov

20 мая 2012 в 2:39

Эх, да вот как раз на пустой тоже пробовал) только подтяжку отставлял)

осцил да, все коплю на него ><

Хорошо) попробую отладочный вывод сделать :)  
Уф, надеюсь поможет)  
А то чет реально непонятки — раньше работал, а щас через раз))

Спасибо)) Постараюсь отписаться если будет чего интересного или странного о\_О

Arseniy Muradov

21 мая 2012 в 3:01

Your bunny wrote!! Вот это неожиданность :D  
Я забыл проверять готовность после стоповой и стартовой последовательности...

Вот почему он через раз работал. Когда я пошагово отлаживал — у него хватало времени, когда сразу включал — twi просто не успевал все сделать и захлебывался

ну кошкин еж, как же я так ... :D

★ DI HALT

21 мая 2012 в 3:13

Как вычислил?

Arseniy Muradov

21 мая 2012 в 3:49

ну я в конечном счете заметил, что, мол, как так, вручную крутится, а автоматом — нет? Ну и решил немного смоделировать этот ручной режим программно  
тоесть в лоб — написал цикл тупой задержки и тыкал вызов ее в подозрительные места.  
Потом заметил закономерность и, вычислив стремные места, перечитал по-слогам сначала старт, ну а там и понял что к чему и со стопом разобрался))))

Чисто случайно, отвлекся может, забыл написать проверку к ним)))

Dorian Gray

20 мая 2012 в 15:29

У меня небольшой вопросик, на какой частоте работает первая мега (главная) и на какой скорости передаются посылки по UART... полазив по коду, нашел, что около 25000 скорость, но хотелось бы точно знать. Возможно, это в статье написано, но я что-то не вижу.

★ DI HALT

20 мая 2012 в 15:30

8мгц, а с какой скоростью идет передача хз, не помню. Посмотри инициализацию и2с.

Dorian Gray

20 мая 2012 в 15:42

и еще небольшой вопросик — у меня стоит не 16 мега, а 32я, в проекте меняю с 16ой на 32ю и компилю, потом в отладчике прохожу пошагово код, но вот в чем проблема дальше строки инициализации периферии не хочет идти, каждый раз начинает сначала... в чем может быть проблема ?

★ DI HALT

20 мая 2012 в 15:46

А адреса векторов прерываний, RAM END и конца памяти правильно прописал? Разница между мегой16 и 32 только

в размерах памяти.

**Dorian Gray**

20 мая 2012 в 16:11

ошибка в том, что в дебагере не поменял 16ю на 32ю. спасибо за наводку!!!

**Aronsky**

14 июня 2012 в 22:10

А если мне, к примеру, попалоcь тупое Slave устройство, которое опускает линию и просто держит? У меня постоянно горит флаг «line busy» и я могу смотреть на него, пока мне не надоест и я не сделаю хард мануал ресет устройства. Что может подсказать протокол в таком случае?

**Arseniy Muradov**

16 сентября 2012 в 0:03

Почему все же написано 127 устройств?)  
вообще можно на шину спокойно повесить сколько угодно устройств, но тока 128 будут с адресами.  
так например если TWI модуль посадить на шину, как мастера, то адреса у нее не будет(ну, если так настроить) и тогда на нее можно вешать 128 слейвов, а всего 129 устройств вполне рабочих получается. ну это чисто пример. Просто не понял почему 127, а не 128  
  
такие дела.  
  
Ну, вроде как. Проверить я это не могу)

**★ DI HALT**

16 сентября 2012 в 0:09

0 адрес это вроде бы бродкаст. Остается 127  
  
А больше нельзя. Все должны иметь адрес.

**Arseniy Muradov**

16 сентября 2012 в 0:18

Ну нет) иметь адрес не обязательно)  
В конце концов не с проста модулю TWI можно выставить бит, который запретит отзываться и вообще реагировать, на адрес, который в него записали)  
Да и по докам написано, что можно 128)  
  
А что такое broadcast?)

**Arseniy Muradov**

16 сентября 2012 в 0:23

Или я не правильно понял, за что отвечает бит TWGCE

**Arseniy Muradov**

16 сентября 2012 в 0:39

что такое broadcast понял)  
Ни когда не видел такого в IIC)

**★ DI HALT**

16 сентября 2012 в 1:03

Широковещательный запрос. На который отвечают все кто его слышит. Вроде бы у него адрес все нули. Или наоборот все единицы. А адрес должен быть обязательно. На него слейв отвечает аском. Нет ответа нет работы. Если без него то это уже не ииц, а чтото другое.

Arseniy Muradov

16 сентября 2012 в 1:07

У слейва само собой, я об этом и сказал в самом начале, а вот у мастера — вроде как нет(и реально нафиг он ему?))) )

Но на счет broadcast не понятно... а где ты углядел эту инфу?)  
Я перерыл доки, какие у меня были, по IIC и по модулям, но там ни где об этом не говорится :(

★ DI HALT

16 сентября 2012 в 1:21

Мастер может быть и слейвом. В ииц вообще анархия на шине. А по поводу бродкастов... Не знаю как насчет оригинальных филипсовских спек на стандарт, они менялись много раз, а вот в железе, бродкаст обычно поддерживается. У авр для него даже свои выделенные состояния автомата есть. Почитай шит на авр.

Arseniy Muradov

16 сентября 2012 в 1:51

Может быть) а может и не быть)  
Все зависит от желаний разработчика)  
поэтому можно и без адреса)  
Если точно уверен, что мастер не станет слейвом)

хм, ну я читал, как раз там говорится про 128 подчиненных устройств)  
Но да, еще там есть» Slave addresses can freely be allocated by the designer, but the address 0000 000 is reserved for a general call.» — реально, штудировал эту главу(TWI) несколько раз и ни когда не замечал... так что ты прав на счет общего вызова. Значит это за его функционирование отвечает бит TWGCE.) тогда понятно)  
Надо будет потестить этот 0000000 на других микрах)  
Вполне вероятно, что это весьма полезной финтифлюшкой окажется o\_O

★ DI HALT

16 сентября 2012 в 1:56

Ну с точки зрения слейва не важно сколько мастеров на шине. Не его это дело. А с точки зрения мастера адрес дает всего 127+1 слейв. Бродкаст может быть полезен для синхры всех устройств шины. Как вариант.

Arseniy Muradov

16 сентября 2012 в 2:22

Точно, щас наконец-то нашел этот broadcast в оригинальной спецификации на IIC.  
Реально есть такое)  
ппц)  
Значит реально больше 127 слейвов не повесишь.  
Или 128 вешать, но на всех, кроме одного отключить разрешение на общий вызов)  
Это это муторно как-то)))) так что пусть 127 будет)

Спасибо)) а то я значит в заблуждении был)

magomedoff

17 сентября 2012 в 20:37

Если отослать ACK то после стопа Master не отпустит линию — такой уж там конечный автомат.



тут ошибки нет ? разве не мастер управляет линией ? прошу пояснить..., может просто я чего-то недопонял...

★ DI HALT

17 сентября 2012 в 20:44

Мастер управляет. Но я же имею ввиду железного мастера. Им мы можем управлять только установкой флажков. Мы не можем заставить его отпустить линию, только дать ему приказ закончить передачу. А по стандарту конец передачи от слейва к мастеру это нак данный мастером. Вот там и зашито так.

magomedoff

17 сентября 2012 в 21:00

все, понял, извиняюсь, собственно я так и подумал, когда прочел это предложение, до него думал про программного мастера..., еще раз пардон !

limburan

8 октября 2012 в 18:12

Приветствую!

У статьи явно намечалось продолжение в виде практического примера с AVR. Но я после долгого пролистывания вперед так его и не обнаружил, что довольно таки грустно.

Если я просмотрел, можешь дать ссылочку? Иначе скажи пожалуйста, будет ли когда-либо продолжение?

★ DI HALT

8 октября 2012 в 22:48

Ищи поиском по сайту IIC там статей несколько.

nwanomaly

19 декабря 2012 в 21:08

Внезапно возник вопросик — про подтягивающие резисторы. Где их лучше ставить, если устройства разнесены? Только в одном месте (например, около одного мастера) или надо у каждого девайса?  
Ещё у Евстифеева написано, что при использовании в качестве ведущих-ведомых МК с нужными ножками без резисторов таки можно обойтись.

★ DI HALT

19 декабря 2012 в 21:49

ИИЦ работает на дистанциях в десяток сантиметров. В пределах одного устройства, короче. Так что тут пофигу где, мож-но возле передатчика и приемника поставить по R/2 резистору.

Евстифеев скорей всего имел ввиду внутренний подтяг порта. Но он слабоват.

nwanomaly

19 декабря 2012 в 22:18

1. Странно, видал примеры, когда устройства были разнесены почти на метр, и в каждом предусмотрен подтяг (для примера, регуляторы квадрокоптера и главная плата). Вот потому и засомневался, в даташитах на линии только два резистора нарисовано.  
Кстати, если максимум 127 девайсов, то сложновато их на 10 см разместить будет, если вдруг понадобится )
2. Про внутренний подтяг конечно. Сейчас добрался до даташитов, пока подобных примеров не нахожу.

nwanomaly

19 декабря 2012 в 22:19

И ещё в авто такую же систему видал — там вообще по всему салону линии раскиданы были.

★ DI HALT

19 декабря 2012 в 23:17

1. Ну потому и два, т.к. линия длинная. Но вообще ииц заточена для внутриплатной коммуникации.
2. Можно, но его может не хватить на быстрый подтяг. Если у получившейся линии связи будет большая емкость, то фронты затянет, особенно на больших скоростях.

nwanomaly

19 декабря 2012 в 23:23

Понял, спасибо!

Поставлю два резистора, на каждый девайс подтяг делать не буду, но площадку под них оставляю — вдруг не заработает )

MS24

1 января 2013 в 17:48

Отличная статья, даже школьнику понятно будет. Спасибо тебе огромное DI HALT за твою работу в просвещении!

Был бы сисятой телкой, совокуплялся бы с тобой за каждую статью!

magomedoff

1 января 2013 в 20:24

надо же, какая признательность ! :-)

С новым годом друзья !

★ DI HALT

1 января 2013 в 23:51

Ахахаха

ssqI

17 апреля 2013 в 1:35

Приветствую всех! Отличный сайт. Большое спасибо автору!! Товарищи, а кто-нить работал с AD7746 преобразует емкость в код) передает это все через I2C?? Являюсь начинающим))) и пока жутко туплю) Поделитесь знаниями)

Bagler

4 сентября 2013 в 13:40

Здравствуйте, может немного глупый вопрос, хочу прояснить для себя верно ли я понял: если устройство на шине после стартовой последовательности увидело не свой адрес, то оно не обращает внимания на данные, бегающие по шине до тех пор, пока не увидит стоповой последовательности. Это так?

★ DI HALT

4 сентября 2013 в 13:50

Да

FreshManGood

12 октября 2013 в 0:23

вы пишете что когда Slave не справляется с приемом данных то он просто прижимает SCL  
а как быть когда он прижал и «заснул» ?

тогда все повиснет и работу уже не возобновить ?  
каков есть выход с данной ситуации ?

★ DI HALT

12 октября 2013 в 0:30

Никакого выхода. Это поломка шины. Такого слейва надо будить и бить по почкам пока не проснется. Можно молотком.

FreshManGood

12 октября 2013 в 0:47

понял )))  
2) если мастер посылает 7 битный адрес устройства, но по непонятным причинам слейв не отвечает, но мы точно знаем что он там есть....., это как в фильме ДМБ: видиш суслика....., нет, не вижу....., и я не вижу, но он то там есть )))....., что надо делать мастеру ? кричать пока его не услышат ?

Tsegorah

19 марта 2015 в 20:41

Если слабая 1 устанавливается вне схемы, а внутри неё значение шины только «прижимается» в 0, то не понимаю, если у меня внутри схемы на VHDL всё описано, то, чтобы она работала, мне надо устанавливать слабую 1 внутри неё. Тогда при синтезе на транзисторах получится лишний устанавливающий слабую 1 кусок схемы?

surganova

1 сентября 2015 в 0:29

Здравствуйте,  
помогите советом как найти неисправность. LPC1769 на моей плате должен управлять цифровым потенциометром по этой шине.  
Но что то идет или пошло не так перестала управляться микросхема MCP4451. Писал изготовителям платы. Посоветовали поменять MCP. Сделал. Результата нет.  
Сейчас пытаюсь поределить проходят ли команды в момент загрузки.  
Спасибо за ответ.

★ DI HALT

1 сентября 2015 в 4:17

А вы разберитесь вначале с адресацией их, может они на один адрес сели, а лпц оказалась шустрой и шину перехватывает. Дальше смотрите кто перехватывает управление. Сделать это можно вкорячив небольшое сопротивление в шину (ом на 100 между лпц и мсп) и по разнице напряжений смотреть какая из микросхем шину использует, передаче это мешать не должно, а на осциллографе будет видно.

surganova

2 сентября 2015 в 8:28

Здрастье,  
повесил ардуино прочитать ситуацию. глухо как и предполагалось.  
там вторая шина есть. на ней ардуино быстро и честно показывает что никого нет. что верно.  
а на этой низкий на SCL, и не знаю куды бечь без помощи.

surganova

2 сентября 2015 в 8:31

Спс за ответ.  
Там такой плотный монтаж, что всунуть что то нереально.  
Правда выведены пины SCL и SDA. К ним можно подключиться.  
Осциллографа нет.  
Пробник показывает «0» на шине SCL всё время.

А если я прикручу ардуино прочитав что там записывает контроллер, и попробую с него запрограммировать/регулировать потенциометр?

**surganova**

1 сентября 2015 в 9:24

Спас за ответ.

Там такой плотный монтаж, что вставить что то нереально.

Правда выведены пины SCL и SDA. К ним можно подключиться.

Осциллографа нет.

Пробник показывает «0» на шине SCL всё время.

А если я прикручу ардуино прочитав что там записывает контроллер, и попробую с него запрограммировать/регулировать потенциометр?

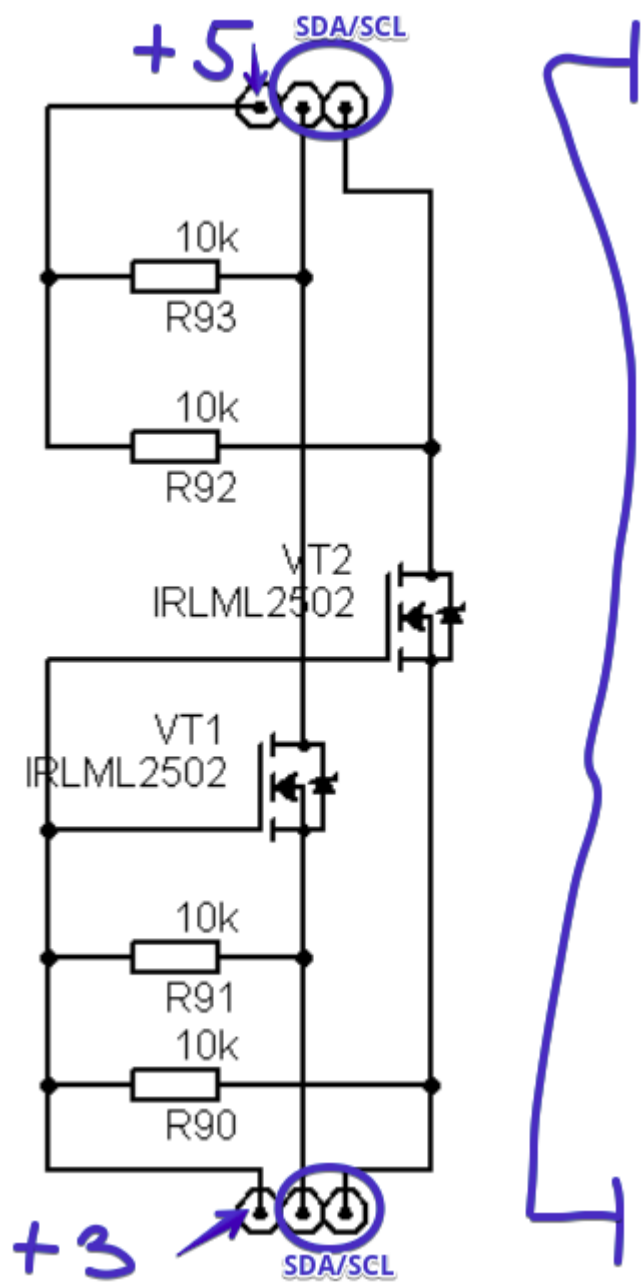
**MrFarik**

25 ноября 2015 в 23:50

А как быть если подключены разные микросхемы на одну шину. Но у них разные логические уровни. Как их согласовывать?

★ **DI HALT**

27 ноября 2015 в 0:18



Как то так, например.

**MrFarik**

27 ноября 2015 в 21:59

А какие еще транзисторы можно применить?

**Massaraksh47**

26 декабря 2016 в 21:10

Добрый день, DI HALT.

Правильно ли я понимаю, что если:

«Прием сигналов осуществляется прижиманием линии в 0, в единичку устанавливается сама, за счет подтягивающих резисторов.»

то уровень 1 для линии — это DDRC(соотв.бит)=0,PORTC(соотв.бит)=0, т.е. input, tristate,

а уровень 0 — это DDRC(соотв.бит)=1,PORTC(соотв.бит)=0, т.е. output,0

И аппаратный TWI работает таким же образом?

И если это так, то нужно ли вообще преобразование логических уровней, если контроллер питается от 5в, а устройство от 3.3в?

Evgen

29 июня 2017 в 9:59

Недавно работал с TWI на Atmega64. Реализовывал мастера, работающего с датчиком давления BMP180. Столкнулся с проблемами в духе повторных стартов и переходов в режим приема. Проблемы сами собой разрешились, когда в TWCR начал записывать не побитно, а байт целиком.

EdMa

17 июля 2017 в 2:26

Опечатка в описании адресации часиков — «Slave-адрес 1010000x где x — состояние ноги A0 микросхемы.»  
Лишний нулевой бит, иначе адрес получается в байт длиной.

Юрий

15 сентября 2017 в 14:12

Использовал данный материал для реализации подключения BMP180 к Mega32. Частота опроса датчика 1 с. Но никак не могу выловить причину регулярных сбоев (где-то раз в пару минут). Судя по логам iic передатчика сбой случается при передаче предпоследнего байта с датчика. То есть получаю последовательность статусов 0x08 0x18 0x28 0x10 0x40 0x50 и все. Статуса 0x58 после не приходит — последний байт не приходит. Не знаю что делать уже несколько дней бьюсь и все бестолку. Помогите, если есть кто живой здесь

★ DI HALT

15 сентября 2017 в 14:42

Т.е. сбой не постоянный? В целом логика работает, но иногда глючит. Возможные причины:

Электрические.

Сделайте синхронизацию по сбою и в момент сбоя поглядите на уровни шины, питания и черт знает что еще. Может увидите.

Глюки в проге выше уровня. Может где то стек подрывает, может еще что. Сделайте подробное логирование по шагам всего автомата в циклический буфер и по сбою выгружайте его в уарт куда-нибудь. Поймете на каком шаге произошло что то не то. Может забыли что или у меня в коде ошибка. Всякое может быть.

extman72

19 января 2018 в 4:29

У меня после прочтения данной статьи возникли следующие 2 вопроса. Тут написано: » Если Slave тормоз и не успевает (у EEPROM, например, низкая скорость записи), то он может насильно положить линию SCL в землю и не давать ведущему генерировать новые такты. Мастер должен это понять и дать слейву прожевать байт». С таким подходом тормоза при тормозить мастера дабы самому успеть проглотить байт я читал и на других сайтах, всё понятно вроде, но вот наткнулся я на описание шины вот здесь: <http://robocraft.ru/blog/communication/780.html> и прочитал там следующее: » Прожевав восьмой бит, ведомая микросхема должна сказать уверенное “Ага”, если всё понятно — послать сигнал подтверждения (ask, acknowledge) — прижав линию SDA (отпущенную ведущим) на время 9-го такта на SCL. Мастер с интересом это дело выслушает (и тактично подождёт если туговатая микросхема не сразу отпустит SDA) » А по сему вопрос: такое поведение слейва тоже имеет место быть? Это я по поводу того, что слэйв не сразу отпустит SDA. Т.е. типа слэйв может её держать

прижатой дольше чем надо, как бы продляя ACK себе в угоду. Или автор там перепутал SDA и SCL? Или можно и SCL прижимать, а можно и SDA после ACK, чтобы получить выигрыш по времени?

Теперь далее. В этой статье написано, что: » Короче, в случае подобного садо-мазо варианта действует железное правило — кто раньше встал того и тапки. В смысле кто первый начал вещать тот и текущий Master.» А вот как написано на том сайте: » Во-первых, тактирование. Естественно, все мастера ломанутся дёргать SCL самостоятельно, они ж мастера=) Вот только пока самые резвые (начавшие первыми) сгенерировав первый такт, отпустят линию — более тормозные и неторопливые будут её держать, отпуская по одному, пока не найдётся самый главный тормоз — начавший вещать последним, он то и продолжит, а остальные останутся нервно ждать.» Вот и не понятно, как правильно. Кто последний того и тапки выходит? Кто прав?

★ DI HALT

20 января 2018 в 2:23

По первому, я такой реакции не встречал. Почитайте описание стандарта i2c от филипс. Может там есть такое.

По второму. Тут все проще. Определяет не столько кто первый, а у кого больше доминантных нулей. Это на всех шинах с подтяжкой так.

extman72

20 января 2018 в 21:44

Ну вообще так. Почитал я «Полный перевод оригинальной спецификации IIC от суровых Челябинских электронщиков с радиозавода Полет». Выводы такие. По первому вопросу, да действительно я не увидел там, что слэйв может удерживать SDA после ACK, видимо автор того сайта что-то напутал. У слэйва есть возможность придерживать только SCL. По второму вопросу так. В этой статье механизм синхронизации никак не упомянут вообще. Описан механизм арбитража, причём описана ситуация только когда два мастера начинают вещать одновременно. На том сайте автор видимо до конца что-то не понял и разделил понятия описав механизм синхронизации и арбитраж как разные вещи, при этом арбитраж рассмотрен там так же как у вас, а именно, когда 2 мастера начинают вещать одновременно, описывая их, мастеров как «однойцевых», а так же тот автор рассматривает синхронизацию отдельно типа получается как ситуацию не одновременного начала передачи мастерами. На самом деле, почитав спецификацию шины от Филлипс, я понял следующее. Стандарт вообще не разделяет таких понятий как одновременно начали вещать мастера, ну типа момент в момент, или не одновременно. Там есть такое понятие, как время минимального удерживания (Thd;sta). Так вот согласно пункту 7.2 Арбитраж, Ведущий может начинать пересылку данных только если шина свободна. Два и более ведущих могут сгенерировать сигнал СТАРТ за время минимального удерживания (Thd;sta), что ведет к определенному сигналу СТАРТ на шине. Т.е. за время (Thd;sta), и не важно одновременно или нет. А вот далее при возникновении такой ситуации и начинается собственно сам арбитраж, который одинаково описан, что в этой статье, что на том сайте, правда, что здесь, что там почему-то арбитраж рассматривается именно как ситуация именно одновременного, прям раз в раз, начала вещания мастерами. Что же касемо синхронизации, которая подробно описана в главе 7.1 Синхронизация стандарта, видим, что она имеет место быть именно во время арбитража на шине, когда несколько мастеров начинают дёргать SCL, каждый по своему. И об этом кстати напрямую в стандарте не сказано, хотя это становится понятным из названия рисунка

Рисунок 9, в котором написано, что это Синхронизация во время арбитража. Ну или не из названия, а из здравых рассуждений, типа ну когда ж ещё такая ситуауия может быть, если не во аремя арбитража, ведь мастера начинают генерить SCL, только после старт условия, которое генерят только при условии что линия свободна. Ну не заострили они на этом внимание, да и ладно, мы сами это поняли, да и название рисунка увидели, так что мы знаем, что синхронизация эта работает именно во время арбитража, который может начаться, когда несколько мастеров сгенерят старт условие во время Thd;sta, и начнут генерить каждый SCL, но вот результирующая SCL, от их действий будет сформирована по прааилу монтажног И, и будет такой, что НИЗКИЙ период синхролинии SCL определяется наидлиннейшим периодом синхронизации из всех задействованных устройств, а ВЫСОКИЙ период определяется самым коротким периодом синхронизации устройств. Так что получается, что тапки не того, кто первый начал вещать, как здесь написано, и не того, кто начал вещать последний как написано на том сайте, а того, кто выиграл арбитраж за счёт того, что сгенерил нолик, а не единичку вперёд других в последовательности битов, причём вся эта арбитражная битва происходит в условиях синхронизации, которая искажает выдаваемый SCL каждым мастером согласно правилам монтажного И, выдавая на линию SCL результат правила. По этому предлагаю расширить статью и описать более подробно арбитраж и синхронизацию, чтобы более полно описать шину.

Виктор

22 января 2018 в 22:02

Мне кажется у вас ошибки в предложении:



Slave-адрес 1010000х где х — состояние ноги A0 микросхемы. Я эту ногу сразу посадил на 0 так что Slave-адрес у меня 10100000. Очевидно, что на одной шине может быть не более двух экземпляров этой микросхемы с адресами 10100000 и 10100001

Адреса ведь 7-битные...

Michael

23 января 2018 в 20:00

а как формируются адреса на шине?

★ DI HALT

19 апреля 2018 в 1:14

Обычно они зашиты железно в устройства + конфигурируются парочкой адресных ног на самом устройстве, которые за-дают два последних бита.

Александр

18 апреля 2018 в 21:53

А я вот что-то не знаю такого слова, как «слейв». Английский я не плохо знаю, но в русском языке такого слова нет! Даже в самом толстом и современном орфографическом словаре такого не найдешь. Зачем вы загрязняете красивый русский язык?

★ DI HALT

19 апреля 2018 в 1:13

ЛОЛ

m72

3 мая 2019 в 4:50

«При приеме последнего байта надо дать ведомому понять, что в его услугах больше не нуждаемся и отослать NACK на последнем байте. Если отослать ACK то после стопа Master не отпустит линию — такой уж там конечный автомат.»  
Тут немного другая глубинная причина. Люди думавшие над I2C ой как не дураки.  
Что такое STOP? Это положительный фронт SDA / при SCL = 1. Т.е. MASTER должен уронить SDA \ при SCL = 0 и поднять его когда SCL = 1.  
Что такое передача? Это изменение уровня SDA по спаду SCL. А теперь чпокус. После последнего бита данных мастер выдал не NACK, а ACK.  
SCL \  
MASTER выставляет ACK  
SCL /  
SLAVE читает ACK  
SCL \  
SLAVE выводит на SDA новый бит, и о чудо, он 0  
MASTER как бы тоже выводит на SDA 0 чтобы потом сделать из этого STOP  
SCL /  
SLAVE считает, что его данные кому-то нужны до следующего SCL \ и держит 0  
MASTER при SCL = 1 отпускает SDA /, но т.к. SLAVE продолжает держать его в 0, то STOP не происходит  
Шина висит.  
Конечно, если SLAVE выведет 1, то шина поймает STOP и все будут счастливы. Но ведь единица выпадает не всегда. Именно потому MASTER обязан остановить SLAVE с помощью NACK.

m72

4 мая 2019 в 0:44

Следствие: проверять наличие микросхемы на линии надо командой записи (WRITE=0).  
При запросе чтения (READ=1) после ACK SLAVE сразу начнёт выдавать данные и если первый бит будет 0, то MASTER

не сможет вставить STOP и прервать поток. Придётся слушать либо до первой единицы либо уж до конца байта.

**Вадим**

5 мая 2019 в 11:35

Крутая статья. Спасибо. У меня такой вопрос. Есть некоторое устройство, которое работает мастером и все время шлет данные только в одну сторону и только по одному жестко зашитому адресу (0x70). Прочитал эти данные через Logic Analyzer, все прекрасно видно. А вот прочитать эти данные через Ардуину не могу. Ставлю адрес слейва как 0x38, но вижу, что мастер стучится, а самих данных не приходит. В чем может быть проблема?

**★ DI HALT**

5 мая 2019 в 12:15

Трудно сказать. Смотрите на ардуиновские библиотеки, как они работают с и2с. Возможно у вас адресация внутренняя не совпала.

**Аноним**

10 июня 2019 в 12:57

Интерфейс устроен, как коровьи титьки: чтобы получить «единицу», надо дёрнуть вниз...

Этот сайт использует Akismet для борьбы со спамом. [Узнайте как обрабатываются ваши данные комментариев.](#)