



Общие положения и введение в логику работы шины I²C™ Практические рекомендации

Шина I²C

Отличительные особенности

- Двухнаправленный обмен по двум линиям
- Высокая скорость — обмена до 100 кбит и выше
- Возможность адресации до 128 устройств
- Простота программной реализации «Master»-абонента
- Временная независимость процесса передачи

Данное введение не является формальным описанием шины I²C и предназначено только для достаточно подробного ознакомления с целью самостоятельной реализации частных алгоритмов связи. Для получения более полной информации обращайтесь к соответствующей литературе фирмы Philips, например, [Л].

Разработанная фирмой Philips шина I²C («Inter-Integrated Circuit»), — это двухнаправленная асинхронная шина с последовательной передачей данных и возможностью адресации до 128 устройств. Физически шина I²C содержит две сигнальные линии, одна из которых (SCL) предназначена для передачи тактового сигнала, вторая (SDA) для обмена данными. Для управления линиями применяются выходные каскады с открытым коллектором, поэтому линии шины должны быть подтянуты к источнику питания +5 В через резисторы сопротивлением 1...10 кОм, в зависимости от физической длины линий и скорости передачи данных. Длина соединительных линий в стандартном режиме может достигать 2-х метров, скорость передачи — до 100 кбит/с. Суммарная ёмкость линий должна быть не больше 400 пФ, входная ёмкость на каждую ИС должна быть в пределах 5...10 пФ.

Все абоненты шины делятся на два класса — «Master» и «Slave». Устройство «Master» генерирует тактовый сигнал (SCL) и, как следствие, является ведущим. Оно может самостоятельно выходить на шину и адресовать любое «Slave»-устройство с целью передачи или приёма информации. Все «Slave»-устройства «слушают» шину на предмет обнаружения собственного адреса и, распознав его, выполняют предписываемую операцию. Кроме того, возможен так называемый «Multi Master»-режим когда, на шине установлено несколько «Master»-абонентов, которые либо совместно разделяют общие «Slave»-устройства, либо попеременно являются то «Master»-устройствами, когда сами инициируют обмен информацией, то «Slave», когда находятся в режиме ожидания обращения от другого «Master»-устройства. Режим «Multi Master» требует арбитража и распознавания конфликтов. Естественно, он сложнее в реализации (имеется ввиду программная реализация) и, как следствие, реже используется в реальных изделиях.

В начальный момент времени — в режиме ожидания — обе линии SCL и SDA находятся в состоянии логической единицы (транзистор выходного каскада с ОК закрыт). В режиме передачи (рис. 1) бит данных SDA строится положительным импульсом SCL. Смена информации на линии SDA производится при нулевом состоянии линии SCL. «Slave»-устройство может «придерживать» линию SCL в нулевом состоянии, например, на время обработки очередного принятого байта, при этом «Master»-устройство обязано дождаться освобождения линии SCL, прежде чем продолжить передачу информации.

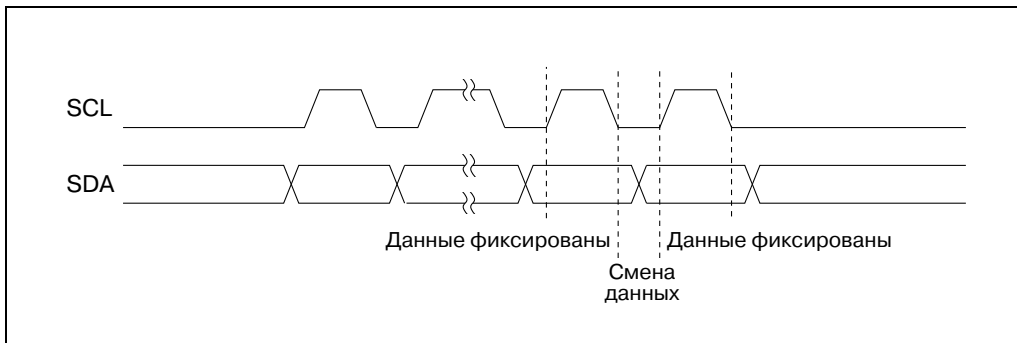


Рис. 1. Диаграмма процесса передачи данных по шине I²C

Для синхронизации пакетов шины I²C различают два условия — «Start» и «Stop», ограничивающие начало и конец информационного пакета (рис. 2). Для кодирования этих условий используется изменение состояния линии SDA при единичном состоянии линии SCL, что недопустимо при передаче данных. «Start»-условие образуется при отрицательном перепаде линии SDA, когда линия SCL находится в единичном состоянии, и наоборот, «Stop»-условие образуется при положительном перепаде линии SDA при единичном состоянии линии SCL.

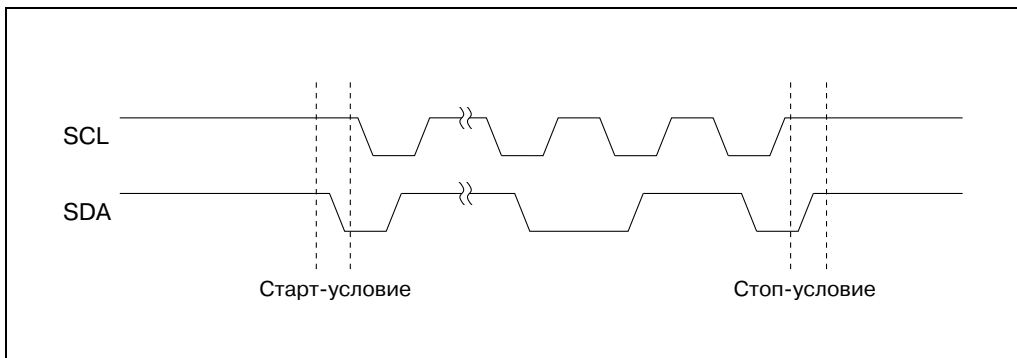


Рис. 2. Диаграмма «Старт»/«Стоп» условия шины I²C

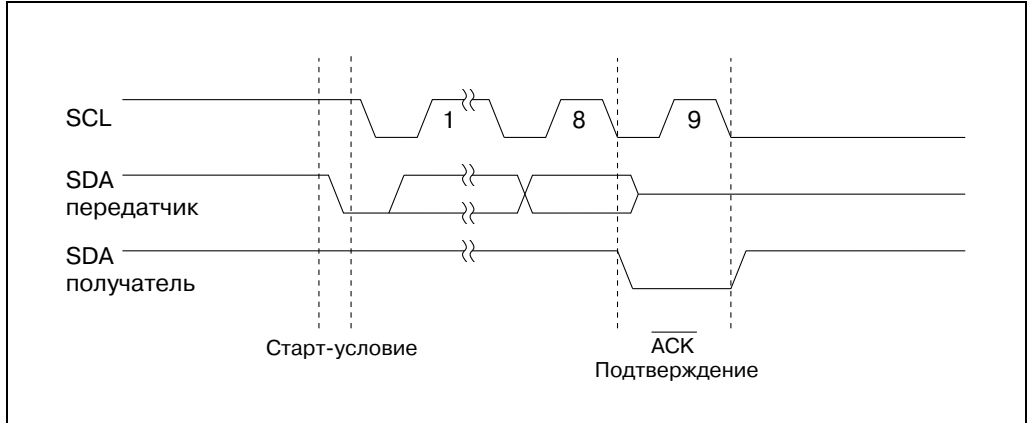


Рис. 3. Диаграмма подтверждения приёма байта по шине I²C

Передача данных начинается по первому положительному импульсу на линии SCL (рис. 3), которым стробируется старший бит первого информационного байта. Каждый информационный байт (8 битов) содежит 9 тактовых периодов линии SCL. В девятом такте устройство-получатель выдаёт подтверждение ($\overline{\text{ACK}}$) — отрицательный импульс, свидетельствующий о «взаимопонимании» передатчика и получателя. Сразу отметим, что любой абонент шины, как «Master», так и «Slave» может в разные моменты времени быть как передатчиком, так и получателем и в соответствии с режимом обязан либо принимать, либо выдавать сигнал $\overline{\text{ACK}}$, отсутствие которого интерпретируется как ошибка.

Временная диаграмма сигналов SCL и SDA шины I²C приведена на рис. 4. Здесь S обозначает «Start»-условие, P — «Stop»-условие. Значения временных характеристик приведены в табл. 1.

Чтобы начать операцию обмена устройство «Master» выдаёт на шину «Start»-условие, за которым следует байт с адресом «Slave»-устройства (рис. 5), состоящий из семибитового адреса устройства (занимает биты 1...7) и однобитового флага операции — «R/ $\overline{\text{W}}$ » (бит 0) определяющего направление обмена, причём 0 означает передачу от «Master» к «Slave» (рис. 5, а), а 1 — чтение из «Slave» (рис. 5, б). Все биты передаются по шине I²C в порядке старший-младший, то есть первым передаётся 7-ой бит, последним 0-ой. За адресом могут следовать один или более информационных байтов (в направлении, определённом флагом R/ $\overline{\text{W}}$), биты которых стробируются сигналом SCL из «Master»-устройства.

При совершении операции чтения «Master»-абонент должен сопровождать прочитанный байт сигналом $\overline{\text{ACK}}$, если необходимо прочитать следующий байт, и не выдавать сигнала $\overline{\text{ACK}}$, если собирается закончить чтение пакета (рис. 5, б).

Допускается многократное возобновление «Slave»-адреса в одном цикле передачи, то есть передача повторного «Start»-условия без предварительного «Stop»-условия. Такой принцип широко применяется в управлении I²C абонентами, когда

выдача нового «Start»-условия служит для синхронизации начала нового пакета данных, сопровождаемого, например, новым управляющим словом, уточняющим адресацию пакета. Логическая реализация протоколов на шине I²C не нормируется документами фирмы Philips, содержащими формальные описания шины, и может быть произвольной для каждой конкретной ИС.

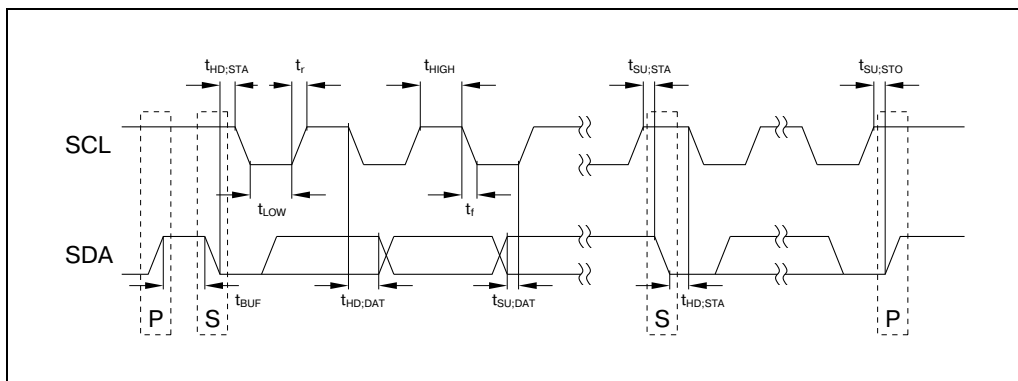


Рис. 4. Временная диаграмма работы шины I²C

Табл. 1. Значения временных характеристик шины I²C

Параметр	Обозн.	Мин.	Макс.	Един.
Частота сигнала SCL	f_{SCL}	0	100	кГц
Свободная шина	t_{BUF}	4.7	—	мкс
Фиксация «Start»-условия	$t_{HD;STA}$	4.0	—	мкс
Длительность «LOW» полупериода SCL	t_{LOW}	4.7	—	мкс
Длительность «HIGH» полупериода SCL	t_{HIGH}	4.0	—	мкс
Готовность повторного «Start»-условия	$t_{SU;STA}$	4.7	—	мкс
Удержание данных	$t_{HD;DAT}$	0	—	мкс
Готовность данных	$t_{SU;DAT}$	250	—	нс
Фронт сигналов SCL и SDA	t_r	—	1000	нс
Спад сигналов SCL и SDA	t_f	—	300	нс
Готовность «Stop»-условия	$t_{SU;STO}$	4.0	—	мкс

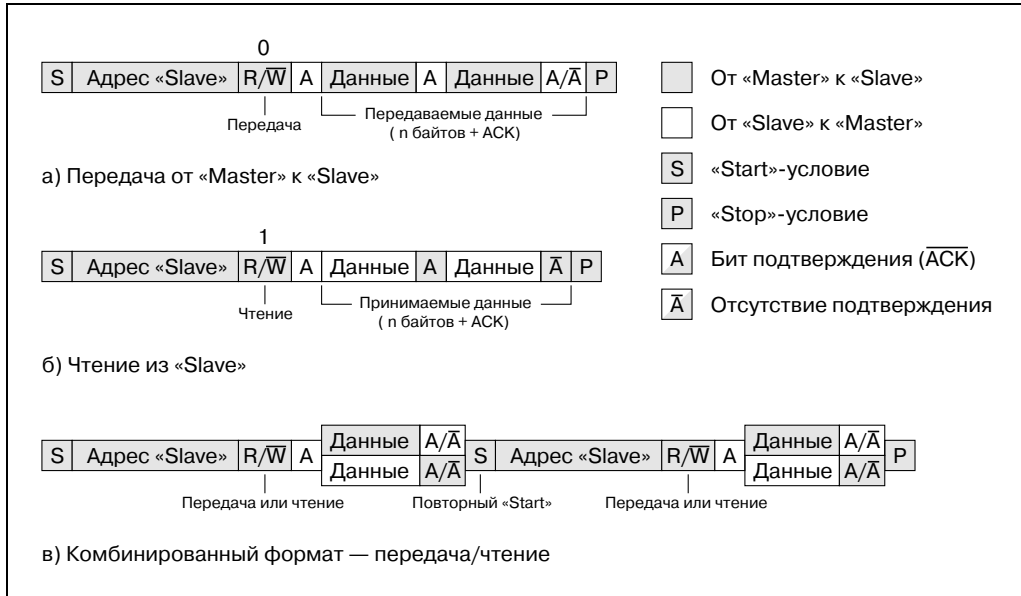


Рис. 5. Формат операций чтения/записи

Практические рекомендации

Ниже мы рассмотрим основные трудности, связанные с использованием шины I²C в целом, а также типичные проблемы и ошибки, возникающие на начальных этапах реализации алгоритмов обмена.

Удобства применения шины I²C очевидны — малое количество соединительных линий и высокая скорость обмена, простота аппаратной реализации линии связи. Наиболее широко поддерживает шину I²C, конечно же, фирма Philips, производящая множество ИС различной сложности с управлением по I²C. В первую очередь, можно выделить микросхемы энергонезависимой памяти (EEPROM) серии 24Схх в 8-ми выводных корпусах, фактически ставшие промышленным стандартом. Из широко распространённых ИС можно выделить: микросхемы часов PCF8583, параллельный порт PCF8574, 4-х канальный 8-ми разрядный АЦП PCF8591. Существует множество модификаций этих ИС и более специализированные контроллеры, в частности, фирма КТЦ-МК выпускает специализированные контроллеры с управлением по I²C: контроллер цифрового светодиодного дисплея и клавиатуры (CE210, CE220), контроллер ЖКИ-модуля с параллельной шиной, совмещённый с контроллером клавиатуры (CE110), контроллер сети на основе шины RS-485 (CE302).

I²C-абоненты жёстко разделяются по классам: «Master»- и «Slave»-устройство. Тот факт, что сигнал SCL всегда генерируется «Master»-устройством означает,

что «Master»-абонент может быть достаточно легко реализован чисто программными средствами, так как все изменения на шине будут происходить только по сигналу SCL. И наоборот, реализация «Slave»-устройства требует аппаратной поддержки, кроме случая очень низких скоростей обмена. Существуют однокристалльные микро-ЭВМ (ОМЭВМ) поддерживающие «Slave»-операции шины I²C. Это прежде всего Philips PCF80C552 (652), Microchip PIC16C62 (64, 73 и др.), Motorola MC68HC705CJ4 (BD3, E5). Кроме того, ОМЭВМ фирмы Philips аппаратно поддерживают «Master»-протокол, хотя практически аппаратные «Master»-контроллеры применяются редко.

В типичной микропроцессорной системе с применением I²C-устройств, обычно существует центральная ОМЭВМ, две линии портов которой отведены для управления линиями шины I²C SCL и SDA. Линии шины I²C должны быть выполнены с открытым коллектором (ОК), и подтянуты резисторами к линии питания +5 В, что позволяет абонентам шины производить операции типа «Монтажное ИЛИ» для торможения процесса обмена в случае, если они не успевают обрабатывать операции со скоростью «Master»-абонента.

Типичная ошибка при реализации программ «Master»-абонента — управление значением порта ОМЭВМ для установки нулевого и единичного состояний линий SCL и SDA. Если для ОМЭВМ семейства MCS-51 это нормальный режим работы, так как единичное состояние порта у них реализуется встроенным подтягивающим резистором ≈ 50 кОм, то для ОМЭВМ с симметричными портами (Motorola 68HCxxx, Microchip PIC, Atmel AVR) это будет порождать электрические конфликты. Например, в руководстве «Microchip. Embedded Control Handbook 1994/1995» приведены практические программы для связи PIC с ИС EEPROM, содержащие подобные грубые ошибки. Положение усугубляется тем, что в случае микросхем EEPROM такой вариант может сработать, так как они являются 100% аппаратными схемами и не вносят задержек в связной протокол, а паузу ожидания окончания цикла программирования производят переходом в пассивное состояние. Использование таких подпрограмм с ИС, производящими захват линии SCL (практически любой «Slave»-абонент, реализованный с применением ОМЭВМ), приведёт к невозможности связи, а возможно, и к выходу ИС из строя.

Реализовать настоящую имитацию ОК (мы назвали этот режим имитацией ОК, так он не позволяет устанавливать на линии напряжение выше напряжения питания, что было бы нормально для настоящего ОК, но так как по спецификации I²C напряжение на линиях SCL и SDA не должно превышать напряжение питания, его вполне законно можно считать выходом с ОК) на порте с симметричным выходом можно, если установить значение порта постоянно в ноль, а управлять состоянием линии через манипуляции с регистром направления данных. Для ОМЭВМ PIC это будет регистр «TRISx», переводящий порт либо в третье состояние, либо подключающий линии в соответствии с состоянием регистра «PORTx». Практически так же это реализуется в AVR и MC68HC05 (08, 11), где «DDRx» коммутирует порт «PORTx», с той лишь разницей, что у них другая полярность управляющего сигнала — у PIC ноль в «TRISx» соответствует нулю на выходе, а у AVR и MC68HC05

единица в «DDRx» соответствует нулю на выходе.

Ещё одна тонкость связанная с PIC, не имеющая собственно к I²C никакого отношения заключается в том, что у него совмещены регистр чтения состояния линий порта и регистр записи значения порта, в результате чего, если часть линий порта запрограммирована на вывод, а другая находится в третьем состоянии, то при выполнении PIC'ом операции модификации в регистр порта переписываются значения непосредственно с выводов PIC'a установленных в третье состояние, что может нарушить (и скорее всего нарушит) работу программы обмена (потеряются нули обеспечивающие имитацию выходов с ОК). Поэтому лучше производить принудительную установку в ноль битов в «PORTx», например, перед каждой операцией обмена, скажем такой последовательностью:

```
movlw b'11100111' ; Маска линий SCL и SDA в порте C (RC3 и RC4)
andwf portc,f ; Сбр. значений линий SCL и SDA в порте C в ноль
```

Другая важная сторона вопроса — необходимость тщательного соблюдения параметров временной диаграммы процесса обмена. Несмотря на то, что шина I²C асинхронная и позволяет затягивать передачу бита (байта) на сколь угодно длительное время (это свойство позволяет реализовывать программы I²C-обмена на самом низком уровне приоритета, прерывая процесс передачи в любое время), требования к минимальным значениям длительностей импульсов очень жёсткие. Ситуация усугубляется тем, что положительные перепады состояния линии имеют склонность затягиваться, так как несимметричные управляющие выходы не могут создать крутые положительные фронты.

При написании программ очень важно контролировать время между операциями на шине, реализуемыми различными подпрограммами, например выдача «Start» и «Stop»-условия, передача бита, передача байта. При состыковке этих подпрограмм не должны быть нарушены минимальные значения времени, что очень легко происходит при использовании высокоскоростных процессоров (AVR, PIC). Кроме того, необходимо следить, чтобы время между изменением на линии SDA и стробированием положительным импульсом на линии SCL было не меньше половины минимальной длительности полупериода SCL ($4.7 \text{ мкс}/2 = 2.4 \text{ мкс}$). Помимо этого, некоторые «Slave»-приборы могут ужесточить требования к максимальной частоте обмена (например, CE110 — 48 кГц), в этом случае необходимо пропорционально снижению частоты обмена увеличивать значения минимумов временных допусков.

Ещё одна распространённая ошибка — игнорирование требования слежения за захватом линии SCL «Slave»-абонентом. Грамотно реализованные программы операций «Master»-абонента должны контролировать возврат линии SCL после того, как переводят её в единичное состояние, и только дождавшись реальной установки линии SCL в единичное состояние продолжать операции приёма-передачи.

Теперь мы опишем минимально необходимый набор операций для реализации «Master»-абонента. Прежде всего, это подпрограммы выдачи «Start»- и «Stop»-условия, подпрограмма передачи байта, подпрограмма приёма байта, выдающая

сигнал \overline{ACK} и подпрограмма приёма байта, не выдающая сигнал \overline{ACK} . Три последние используют подпрограммы передачи бита и приёма бита. Во временных промежутках между подпрограммами линия SCL имеет нулевое значение. Как уже говорилось ранее, важно правильно согласовать все временные характеристики подпрограмм. Если, к примеру, вы вызываете повторный «Start» сразу за передачей (приёмом) байта, то необходимо, чтобы полностью завершился нулевой полупериод (≥ 4.7 мкс), предшествующий «Start»-условию, а подпрограмма, реализующая «Start»-условие, переведя линии SDA и SCL в единичное состояние, дождавшись их реального возврата, должна удерживать её в таком состоянии не менее установленного времени (≥ 4.7 мкс), и так во всех случаях.

Оперируя этими пятью подпрограммами можно легко наладить обмен по шине I²C. Можно выдать на шину «Start»-условие, за ним «Slave Adress», далее данные, в соответствии с протоколом конкретной I²C-ИС. Можно выдать повторное «Start»-условие, новый «Slave Adress» и так далее, формируя процесс обмена как из элементов конструктора. Обычно протоколы обмена «Slave»-устройств довольно логично соответствуют программированию в таком стиле.

Литература

- [1] The I²C-bus and how to use it. © 1997 Philips.

Общие положения и введение в логику работы шины I²C

Практические рекомендации

1-е издание

© 1997 КТЦ-МК

Автор: А.А.Голов

Вёрстка произведена в пакете SugiTUG-emTeX

с использованием кириллических шрифтов семейства LH