

Шина I²C

© Александр Торрес, Харьков.
(altor@geocities.com АКА 2:461/28)
По материалам фирмы Филипс.

Шина I²C широко используется в бытовой электронике, передаче данных и промышленной электронике. Разработанная фирмой Philips простая двунаправленная 2-проводная шина для эффективного управления и взаимодействия различных блоков телевизоров, она стала применяться для связи между собой однокристалльных микроконтроллеров, ЖКИ-индикаторов, портов ввода-вывода, микросхем памяти (особенно энергонезависимой), аналого-цифровых и цифро-аналоговых преобразователей, цепях цифровой настройки, DTMF кодеров и декодеров, часов реального времени и т.д.

В настоящее время, ассортимент микросхем с шиной I²C только одной фирмы Philips составляет больше чем 150 КМОП и биполярных приборов, а в области энергонезависимой памяти средней емкости (от сотен байт до десятков килобайт) она стала стандартом де-факто (серии 24XXX фирм Microchip, Atmel, и др.)

Шина I²C представляет собой концепцию, которая решает многие проблемы интерфейса, с которыми сталкиваются при проектировании различной аппаратуры, и имеет следующие достоинства:

- Только две линии - последовательная линия данных (SDA) и последовательная линия синхронизации (SCL)
- Каждый элемент, соединенный с шиной является программно-адресуемым своим уникальным адресом. При этом отношения между ними могут быть построены по простому принципу master/slave или же может быть построена multi-master система.
- Это настоящая шина, с возможностью работы в multi-master среде, включая проверку на пересечение и арбитраж.
- Последовательная, 8-разрядная двунаправленная передача данных может проводиться со скоростью от 0 до 100 kbit/s в стандартном режиме или до 400 kbit/s в быстром режиме (не для всех типов микросхем). При реализации собственных устройств возможно применение и более высоких скоростей при сохранении идеологии шины I²C.
- Фильтрация сигналов внутри микросхем обеспечивает нечувствительность к выбросам на линии шины данных.
- Число устройств, которые могут быть соединены одной шиной, ограничено только максимальной емкостью шины 400 pF.

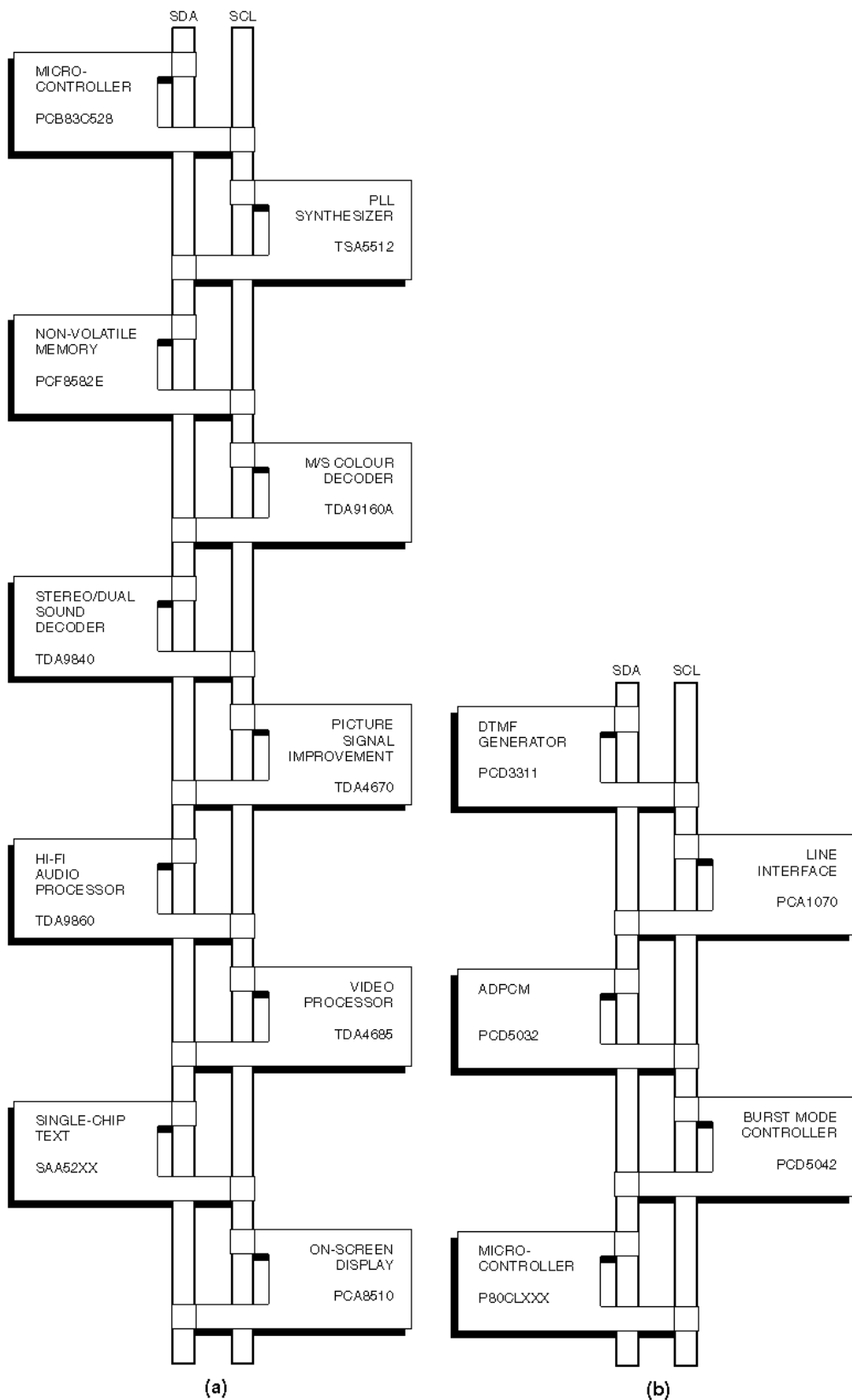


Рис. 1 - два примера применения шины I²C.

В дополнение ко всем преимуществам, микросхемы с шиной I²C в КМОП исполнении являются особенно привлекательными для переносного оборудования и с аварийным батарейным питанием, особенно в сочетании с низко потребляющими системами управления, например на базе микроконтроллеров фирмы Microchip, т.к. многие из них имеют:

- чрезвычайно низкое потребление электроэнергии;
- высокую помехоустойчивость;
- широкий диапазон питающего напряжения;
- широкий рабочий диапазон температуры.

Для производителей микросхем шина I²C привлекательна тем, что:

- простой 2-проводной последовательный интерфейс минимизирует соединения, что уменьшает количество выводов;
- полностью интегрированный I²C протокол шины устраняет потребность в декодерах адреса и т.п.;
- возможность дистанционного контроля и настройки через внешние соединения на компьютере сборочного конвейера;
- возможность исполнения в любом типе корпуса - SO (small outline), VSO (very small outline), DIL (dual in line), что уменьшает массу и габариты.

ACCESS.BUS

Другой привлекательной стороной шины I²C для проектировщиков и изготовителей является то, что простой 2-проводной интерфейс и возможность программной адресации это идеальная платформа для ACCESS.BUS (Рисунок 2). Это более дешевая альтернатива интерфейсу RS-232C для периферийных устройств компьютеров через простой 4-контактный соединитель, включающий 2 линии I²C и две линии питания (+5в и 0в).

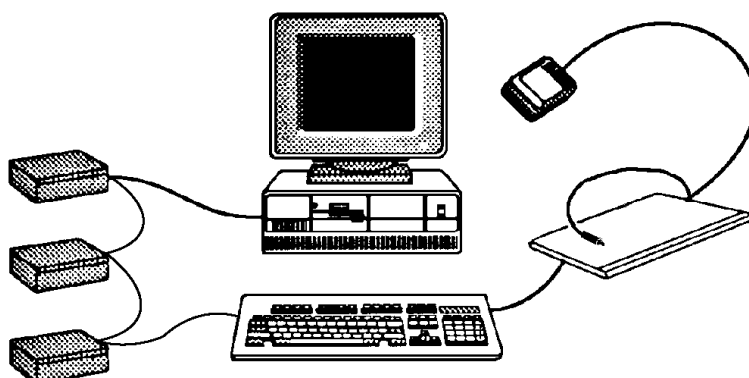


Рис.2 ACCESS.bus - недорогая альтернатива RS232

ВВЕДЕНИЕ В СПЕЦИФИКАЦИЮ I²C

Обычно для многих приборов выдвигаются следующие критерии:

- полная система обычно состоит, по крайней мере, из одного микроконтроллера и периферийных устройств типа блоков памяти и расширителей ввода - вывода.
- стоимость соединения различных элементов внутри системы должна быть минимальной.
- система, которая выполняет только функции управления, как правило, не требует быстросействующей передачи данных.
- общая эффективность зависит от выбранных элементов и характера межсоединений.

Последовательная структура шины удовлетворяет этим условиям. Хотя последовательные шины не имеют такой производительности как параллельные шины, они требуют меньшего количества проводов и меньшего количества выводов микросхемы.

Элементы, общающиеся друг с другом по последовательной шине должны иметь некоторый протокол, который определяет логику функционирования, избегает всех возможностей беспорядка, потерь данных и блокирования информации, при этом быстрые элементы должны быть способны связаться с медленными элементами.

КОНЦЕПЦИЯ ШИНЫ

Все операции по шине I²C осуществляются при помощи двух проводов:

- линия последовательных данных (SDA):
- линия синхронизации (SCL).

Каждый элемент определяется своим уникальным адресом, в который входит группа приборов и номер конкретного прибора. Группа определяет, является ли это микроконтроллером, LCD-индикатором, памятью или чем-то еще. Например, все устройства памяти имеют код 0Ah, таймеры и часы реального времени - 0Dh, устройства телетекста - 02h и т.д.

Любой элемент, инициирующий передачу, является мастером, любой адресуемый элемент является подчиненным. В системах с несколькими мастерами, один и тот же элемент может в разное время выступать или как мастер или как подчиненный.

ТЕРМИНОЛОГИЯ ШИНЫ I²C

Передатчик - элемент, который посылает данные в шину.

Приемник - элемент, который получает данные из шины.

Master - элемент, который инициализирует передачу, генерирует сигналы синхронизации, и завершает передачу.

Slave (подчиненный) - элемент, адресованный мастером

Multi-master - система с более чем одним мастером.

Арбитраж - процедура, гарантирующая, что, если больше чем один мастер одновременно пробуют управлять шиной, только один из них получит полное управление шиной и обеспечит передачу данных без ошибок.

Синхронизация - синхронизация передачи данных между элементами.

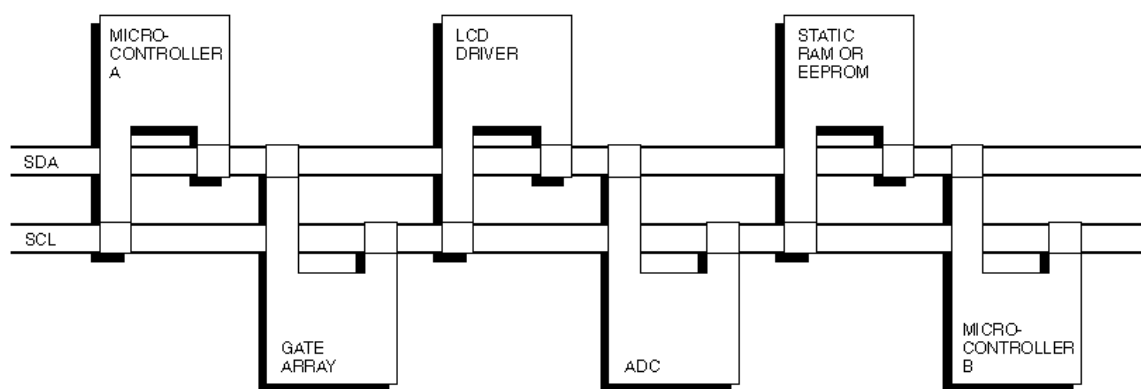


Рисунок 3. Пример конфигурации шины I²C, использующей два микроконтроллера

ОБЩИЕ ХАРАКТЕРИСТИКИ

SDA и SCL - двунаправленные линии с открытым стоком (коллектором), соединенные с положительным питающим напряжением через резистор (см. Рисунок 4). Когда шина свободна, обе линии находятся в состоянии "1". Данные могут передаваться по шине I²C со скоростью до 100 kbit/s в стандартном режиме, или до 400 kbit/s в быстром режиме. Число интерфейсов, соединенных с шиной исключительно зависит от емкости шины, максимальное значение которой - 400pF.

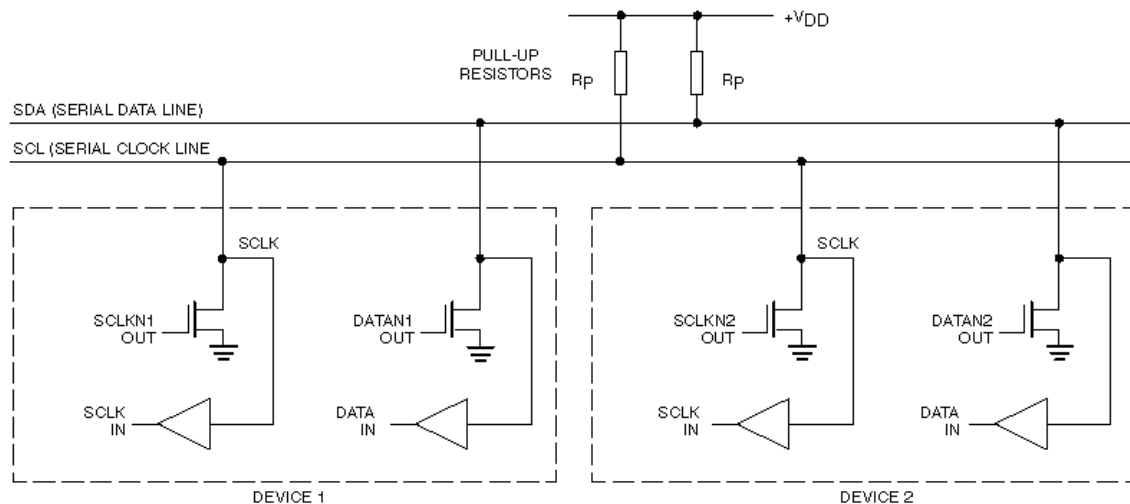


Рисунок 4. Соединение I²C устройств.

ПЕРЕДАЧА БИТА

Из-за разнообразия различных технологий (CMOS, NMOS, биполярные приборы) которые могут быть использованы с шиной I²C, уровни логического "0" и "1" не установлены и зависят от величины питающего напряжения VDD. Для передачи одного бита данных используется один импульс сигнала синхронизации, при этом уровень на линии SDA должен быть неизменным в течение высокого уровня на линии SCL, и может изменяться только при низком уровне на SCL (Рисунок 5). Исключениями служат два особых состояния - START и STOP.

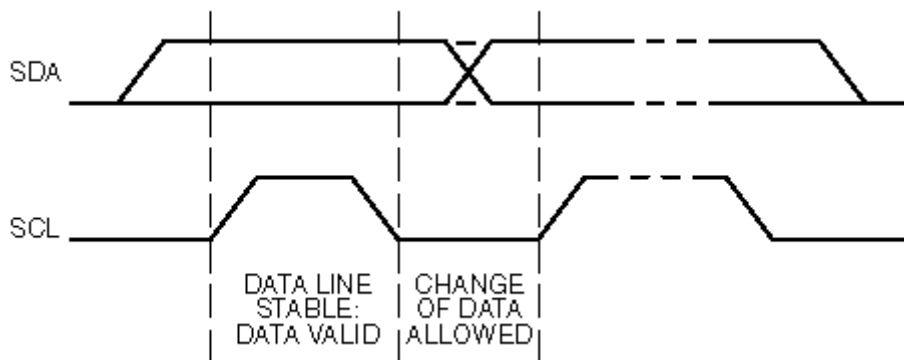


Рисунок 5. Передача бита.

START и STOP.

Существуют два особых состояния шины I²C - start и stop, которые служат для индикации начала и конца передачи и соответственно перехода шины в неактивное состояние. Следует отметить, что до тех пор, пока не установлено состояние start, сигналы на линиях SDA и SCL могут быть совершенно произвольными (Рисунок 6). Это позволяет, в частности, использовать одну линию SDA и несколько линий SLC (например, при нехватке, адресов на одной шине).

Состояние START - переход от "1" к "0" на линии SDA при "1" на линии SCL.

Состояние STOP - переход от "0" к "1" на линии SDA при "1" на линии SCL.

Эти два состояния всегда генерируются мастером.

Детектирование состояний start и stop в специализированных элементах обычно производится аппаратно. При полностью программной реализации шины I²C в микроконтроллерах без аппаратной I²C-части необходимо как минимум 2 раза проверять состояние линии SDA.

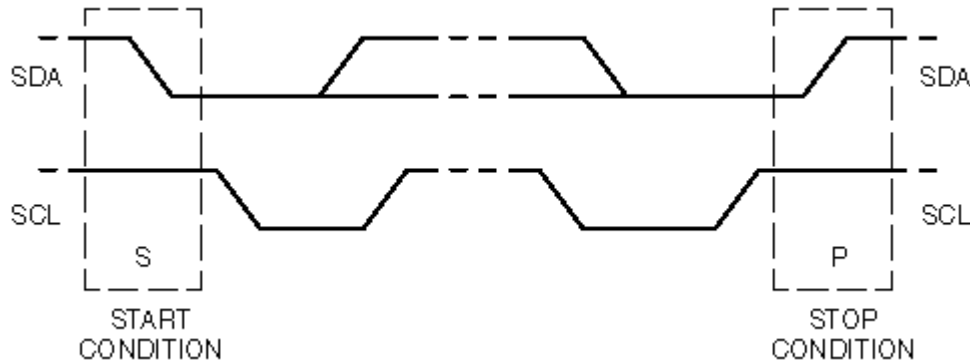


Рисунок 6. START и STOP состояния.

ПЕРЕДАЧА ДАННЫХ

Все передачи производятся 8-разрядными байтами. Число байтов, которые могут быть переданы за одну передачу не ограничено. Каждый байт должен сопровождаться битом подтверждения (ACK). Данные передаются начиная со старшего бита (MSB), (Рисунок 7).

Если приемник не может получить другой полный байт данных, он не выдает сигнал ACK, который используется передатчиком для синхронизации или сигнализации о неисправности приемника (или его отсутствии).

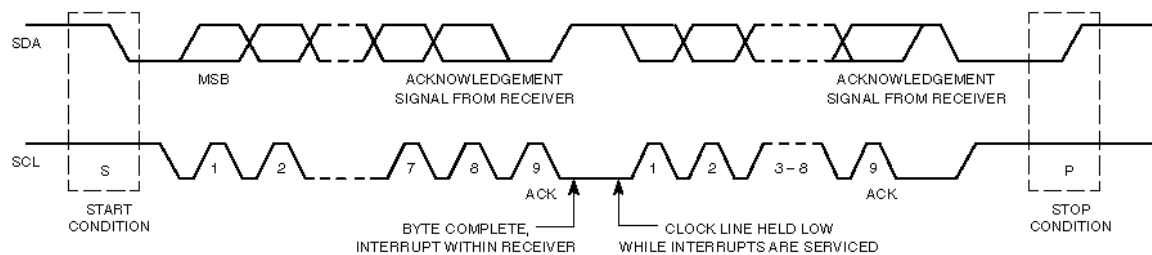


Рисунок 7. Передача данных по шине I²C.

ACKNOWLEDGE

Для подтверждения передачи байта передатчик устанавливает линию SDA в "1" в течение синхронизирующего импульса. Приемник при этом должен выставить "0" на SDA (рисунок 8). Естественно при этом должны быть выдержаны определенные временные соотношения.

Обычно, приемник, который был адресован, должен генерировать сигнал ACK после того, как каждый байт был получен, за исключением того, когда сообщение начинается с адресом CBUS.

Когда подчиненный приемник не подтверждает подчиненный адрес (например, устройство не готово, потому что это выполняет некоторую внутреннюю функцию), линия SDA данных должна быть оставлена в "1". Мастер затем может выдать состояние STOP, чтобы прервать передачу.

Если подчиненный приемник подтверждает подчиненный адрес, но, некоторое время позднее не может получить больше байты данных, мастер должен приостановить передачу.

При приеме последнего байта в серии, вместо сигнала ACK мастер может выставить состояние STOP, при этом подчиненный передатчик должен освободить линию данных.

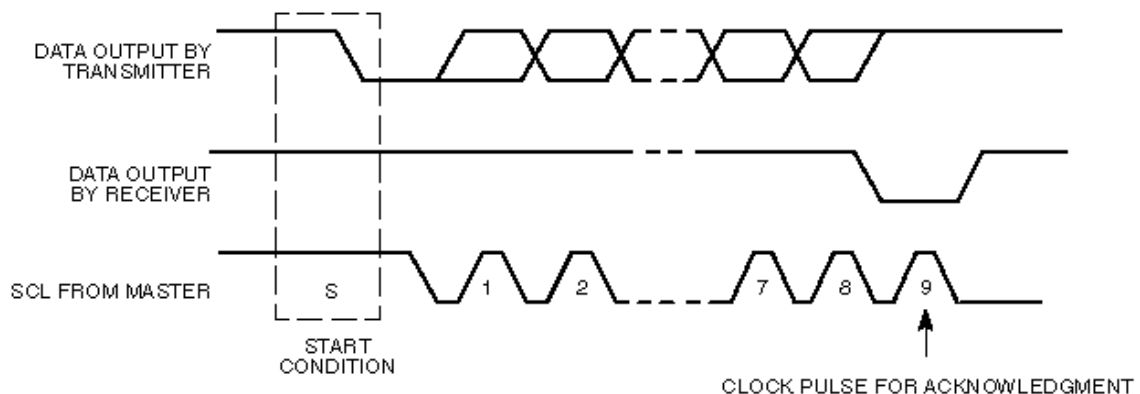


Рисунок 8. Подтверждение передачи.

АРБИТРАЖ И СИНХРОНИЗАЦИЯ

Синхронизация

Каждое мастер-устройство генерирует собственные сигналы синхронизации на линии SCL. Данные на линии SDA действительны только течение ВЫСОКОГО уровня SCL.

Синхронизация осуществляется благодаря "монтажному-И" на линии SCL. Это означает, что состояние "0" на линии SCL будет длиться до тех пор, пока все мастер-устройства не освободят линию синхронизации (рисунок 9). Линия SCL будет, следовательно, задержана в уровне "0" элементом с самым длинным уровнем "0". Элементы с более коротким уровнем "0" при этом входят в состояние ожидания.

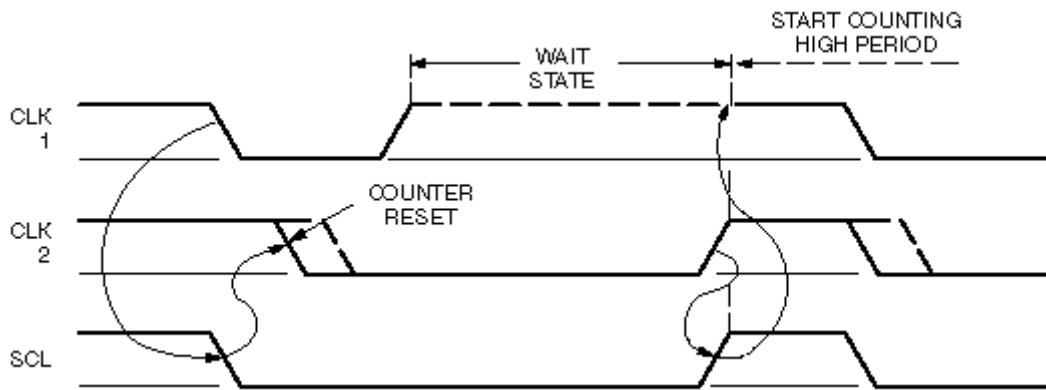


Рисунок 9. Синхронизация в течение процедуры арбитража

Арбитраж

Мастер может начинать передачу только, если шина свободна. Два или больше мастера могут генерировать состояние START практически одновременно, поэтому необходим арбитраж между ними, для того чтобы выяснить, кто же из них все-таки был первым. Для этого используется линия SDA - благодаря тому, что она, как и SCL выполнена по схеме "монтажное-И" (Рисунок 10 показывает процедуру арбитража для двух мастеров).

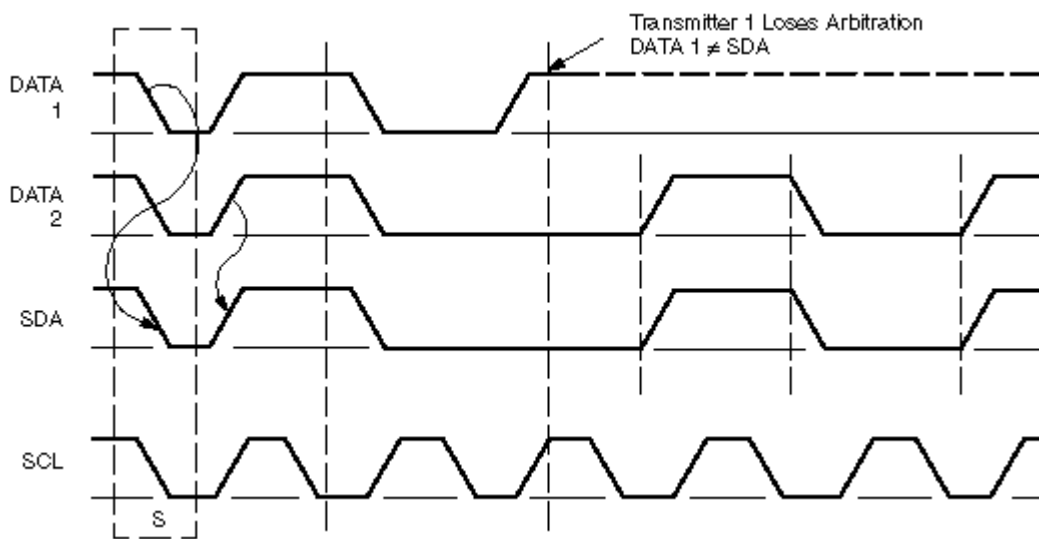


Рисунок 10. Процедура арбитража двух мастеров.

ФОРМАТЫ С 7-БИТНОЙ АДРЕСАЦИЕЙ

Передача с 7-битной адресацией показана на рисунке 11.

После выдачи состояния START следует передача адресного байта, при этом 8-й бит адреса определяет направление передачи данных ("0" - запись данных от мастера к подчиненному, "1" - чтение данных из подчиненного к мастеру). Передача данных всегда завершается состоянием STOP, сгенерированного мастером.

Возможные форматы передачи данных:

- мастер-передатчик передает на приемник-подчиненный. Направление передачи при этом не изменяется (Рисунок 12).
- мастер-приемник читает данные с передатчика-подчиненного немедленно после первого байта (Рисунок 13).
- комбинированный формат (Рисунок 14).

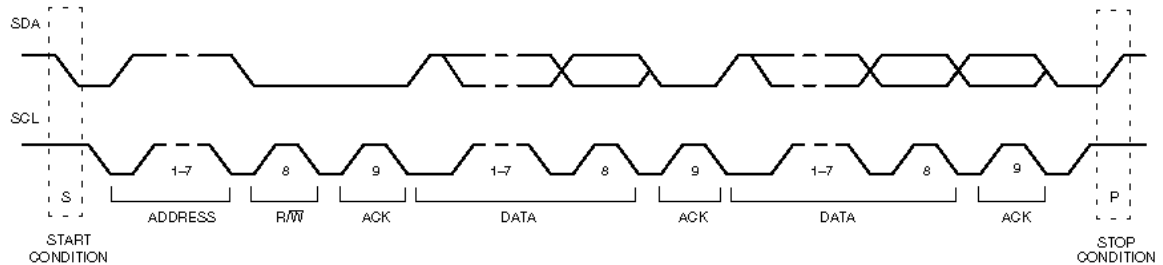
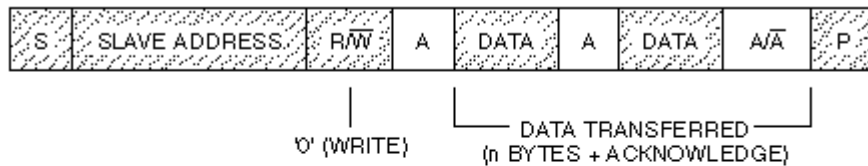


Рисунок 11. Полная схема передачи данных



- ▨ FROM MASTER TO SLAVE
- FROM SLAVE TO MASTER
- A = ACKNOWLEDGE (SDA LOW)
- \bar{A} = NOT ACKNOWLEDGE (SDA HIGH)
- S = START CONDITION
- P = STOP CONDITION

Рисунок 12. Передача от мастера к подчиненному.

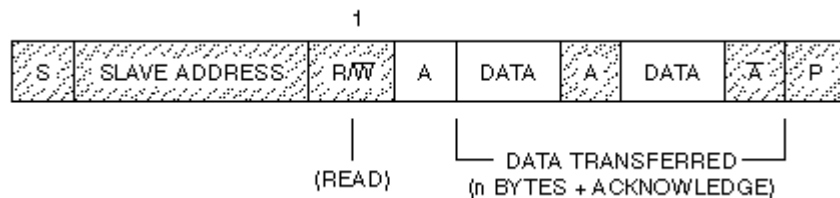
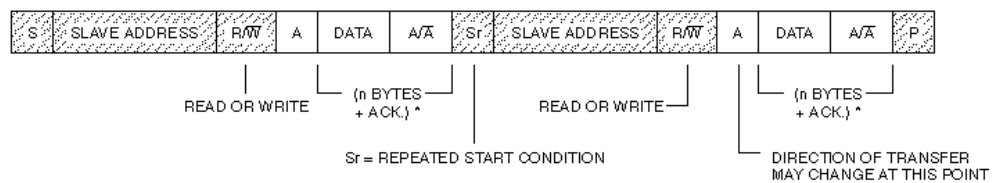


Рисунок 13. Чтение мастером данных с подчиненного устройства.



* TRANSFER DIRECTION OF DATA AND ACKNOWLEDGE BITS DEPENDS ON R/W BITS.

Рисунок 14. Комбинированный формат

Комбинированные форматы могут использоваться, например, при управлении последовательной памятью. Во время первого цикла передается адрес памяти, затем осуществляется передача данных из ячейки с данным адресом. Возможно авто инкремент или декремент адреса. Каждый байт сопровождается битом АСК.

Исключением при 7-битной адресации может быть “общий запрос” - адрес, который может адресовать все элементы. Когда этот адрес используется, все элементы должны выставлять сигнал АСК, однако, некоторые элементы могут быть сделаны так, чтобы игнорировать этот адрес. Второй байт общего адреса затем определяет действие, которое будет принято.

Формирование битов в первом байте. Первые семь битов первого байта сделают подчиненный адрес (рисунок 15). Восьмой (младший) бит - определяет направление сообщения. “0” означает, что мастер будет писать данные в выбранное подчиненное устройство.

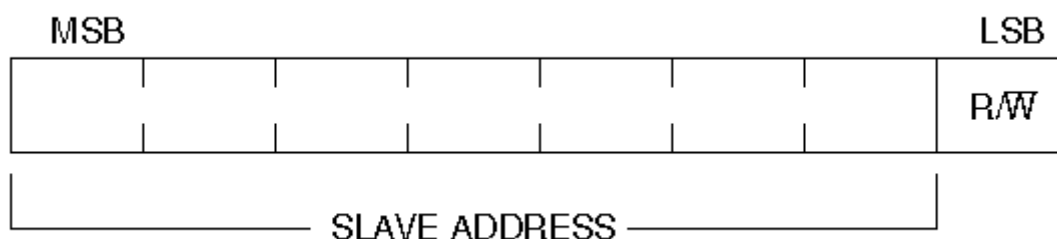


Рисунок 15. Первый байт после состояния START.

Когда адрес послан, каждый элемент в системе сравнивает первые семь битов после состояния START с собственным адресом. Если они согласуют, элемент считает себя адресованным мастером как подчиненный приемник или подчиненный передатчик, в зависимости от R/W бита.

Подчиненный адрес состоит из постоянной и переменной части. Так как вероятно, что будут иметься несколько идентичных элементов в системе, то постоянная часть у них будет одинаковой, т.к. будет являться кодом устройства, переменная же при этом будет однозначно идентифицировать конкретное устройство. Например - на шине присутствуют 8 устройств последовательной энергонезависимой памяти, все они имеют одинаковый код устройства (старшие 4 бита), при этом младшие 3 бита будут определять какое именно устройство адресуется в данный момент.

Некоторые коды устройств и адреса зарезервированы для определенных целей:

SLAVE ADDRESS		R/W бит	Описание
Код устройства	Адрес		
0000	000	0	Общий адрес
0000	000	1	START byte
0000	001	X	CBUS address
0000	010	X	Зарезервировано для других форматов шины
0000	011	X	Зарезервировано на будущее
0000	1XX	X	
1111	1XX	X	
1111	0XX	X	10-битная адресация

Общий адрес.

Он служит для адресации любого элемента на шине. Однако если какой-то элемент не нуждается в дополнительных данных, он может игнорировать этот адрес, не выдавая квитирование. Если элемент требует данных общего адреса, он подтвердит этот адрес и будет вести себя как подчиненный приемник. Второй и следующий байты будут подтверждены каждым подчиненным приемником, способным к обработке этих данных. Подчиненный, который не может обрабатывать хотя бы один из этих байтов не должен выдавать АСК.

Значение общего запроса определяется во втором байте (Рисунок 16).

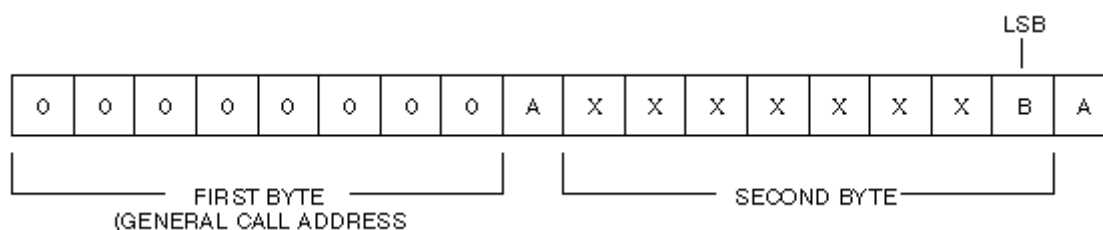


Рисунок 16 Формат общего вызова.

Имеются два случая:

1) Когда бит В - "0" и второй байт имеет следующее значение:

- 00000110 (H '06') - аппаратный сброс.
- 00000100 (H '04') - запись переменной части адреса.
- 00000000 (H '00'). Этот код не должен использоваться как второй байт.

2) Когда бит В - "1" 2-байтовая последовательность - "hardware general call".

Это означает, что последовательность передана мастером типа клавиатуры, который не может программироваться, чтобы передать нужный подчиненный адрес. Так как мастер не знает заранее, к какому элементу сообщение должно быть передано, то он может только генерировать

общий вызов и собственный адрес — идентификация в системе (рисунок 17).

Семь битов, остающиеся во втором байте содержат адрес мастера. Этот адрес должен быть распознан другим элементом (например, микроконтроллером) соединенным с шиной.

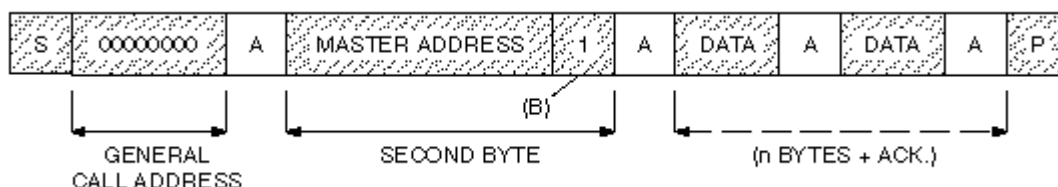


Рисунок 17. Передача данных из мастера при общем вызове.

START BYTE

Микроконтроллеры могут быть соединены с шиной I²C двумя способами:

1. Микроконтроллер со встроенным аппаратным I²C-интерфейсом может быть запрограммирован на прерывание запросами шины.
2. Не имеющий такого интерфейса должен постоянно контролировать шину программно.

Очевидно, во втором случае контроллер должен больше время отвлекаться на мониторинг шины, и меньше времени заниматься другими обязанностями.

В этом случае передаче данных может предшествовать более длительная стартовая процедура (рисунок 18). Процедура состоит из:

- состояния START (S)
- START BYTE (00000001)
- подтверждающего синхронизирующего импульса (ACK)
- повторенного состояния START (Sr).

После того, как состояние START передано мастером, передается стартовый байт,

Микроконтроллер может, следовательно, опрашивать линию SDA с низкой частотой, пока один из семи полей в стартовом байте не будет обнаружен. После обнаружения этого нуля на линии SDA, микроконтроллер может переключиться на более высокую частоту дискретизации, чтобы найти повторенное состояние START (Sr).

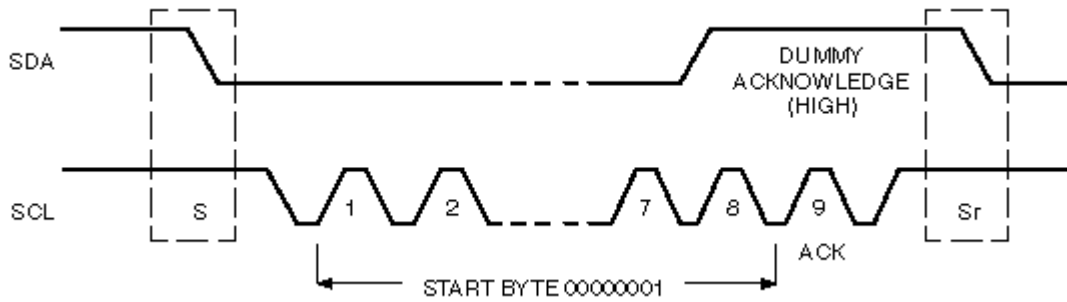


Рисунок 18. START BYTE

ПОДКЛЮЧЕНИЕ УСТРОЙСТВ С РАЗНЫМИ ПИТАНИЯМИ

Различные устройства могут использовать разные питающие напряжения и уровни сигналов. При этом элементы с фиксированными входными уровнями 1.5V и 3V могут иметь различные источники питания, и подтягивающие резисторы, подключенный к источнику +5V (рисунок 19).

Элементы с входными уровнями, зависящими от напряжения питания, должны иметь общее питание, к которому подключены подтягивающие резисторы (рисунок 20).

Если элементы с фиксированными входными уровнями, смешаны с элементами с входными уровнями, зависящими от V_{DD} , необходимо выполнять соединения как показано на рисунке 21.

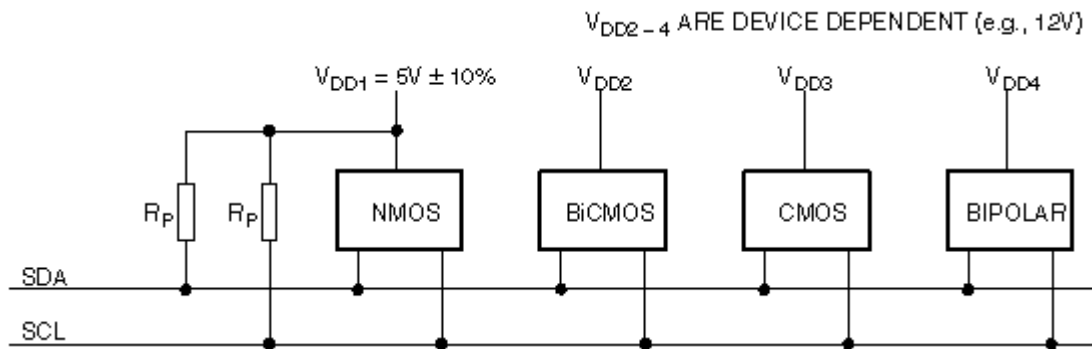


Рисунок 19. Элементы с фиксированными входными уровнями.

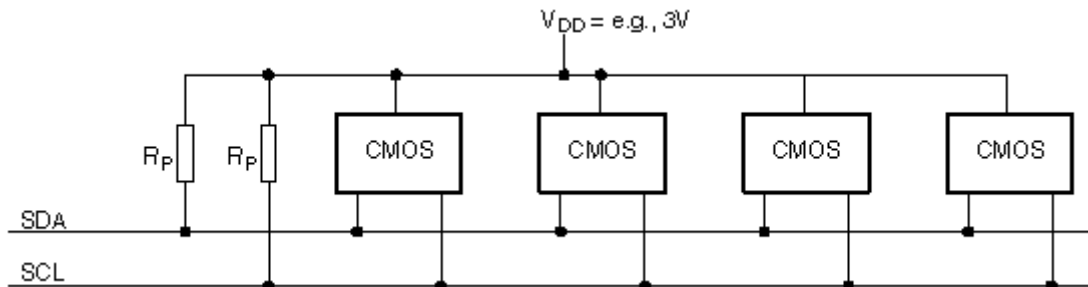


Рисунок 20. Элементы с широким диапазоном питающего напряжения.

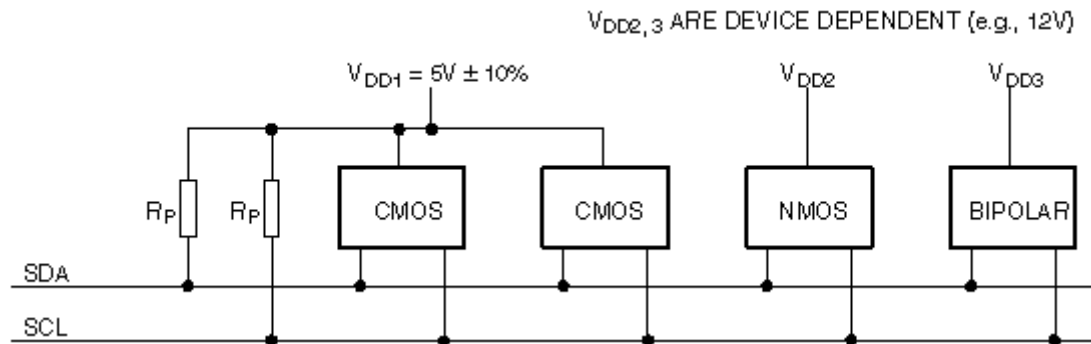


Рисунок 21. Элементы с входными уровнями, связанными с VDD смешанные с элементами с фиксированными входными уровнями.

Для защиты от высоковольтных помех могут применяться последовательные резисторы (рисунок 22).

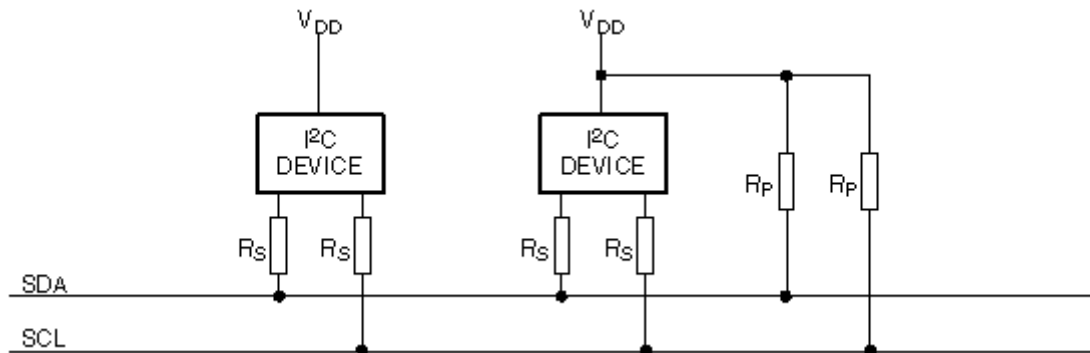


Рисунок 22. Добавочные резисторы для защиты против высоковольтных выбросов

РАСШИРЕНИЕ ШИНЫ I²C

10 лет спецификация шины I²C оставалась неизменной - со скоростью передачи данных до 100 kbit/s и 7-разрядной адресацией. Это было принято во всем мире как фактический стандарт и сотни различных типов I²C-совместимых микросхем выпускаются фирмой Philips и другими. В настоящее время спецификация шины I²C расширена:

- быстрый режим, который позволяет четырехкратное увеличение скорости передачи информации до 400 kbit/s
- 10-разрядная адресация, которая позволяет использование до 1024 дополнительных адресов.

Все новые приборы с шиной I²C выпускаются совместимыми с быстрым режимом. Т.е. они должны быть способны получать и/или передавать со скоростью до 400 kbit/s. Минимальное требование - синхронизация со скоростью 400 kbit/s, затем они могут продлевать "0" на SCL, чтобы замедлить передачу. Элементы с быстрым режимом должны быть обратно-совместимы, что означает, что они должны быть способны работать со скоростями от 0 до 400 kbit/s совместно со старыми элементами, рассчитанными на скорость от 0 до 100 kbit/s.

10-разрядная АДРЕСАЦИЯ

При использовании 10-разрядной адресации используется код устройства 1111XXX для первых семи бит первого байта, который не воздействует на элементы с обычной 7-разрядной адресацией, поэтому элементы с 7-разрядными и 10-разрядными адресами могут быть объединены в одну шину.

Хотя имеются восемь возможных комбинаций первого байта адреса, используются только четыре комбинации 11110XX. Остальные зарезервированы для дальнейшего расширения спецификации шины I²C.

Функционирование элементов с 10-разрядной адресацией аналогично обычным - устройство, получив код в первом байте адреса, проверяет код второго байта адреса, при совпадении которого с собственным выдает подтверждение и выполняет соответствующие действия до появления состояния STOP.

Соответствующие диаграммы показаны на рисунках 23-27.

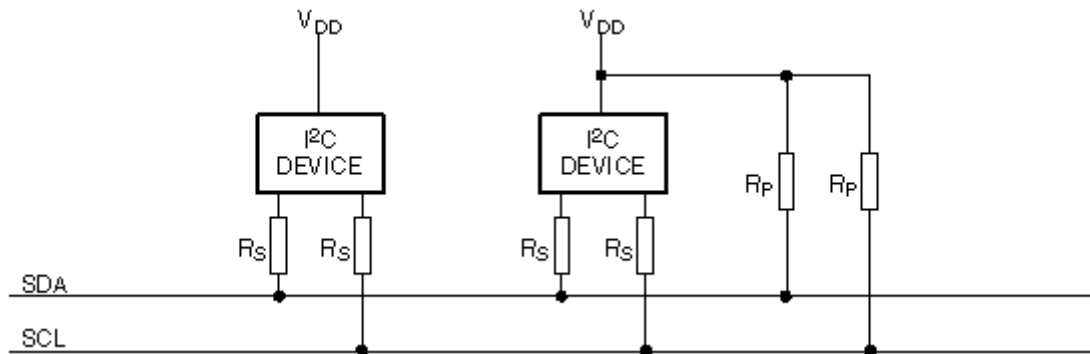


Рисунок 23. Передача мастер-передатчик - подчиненный-примник с 10-разрядным адресом

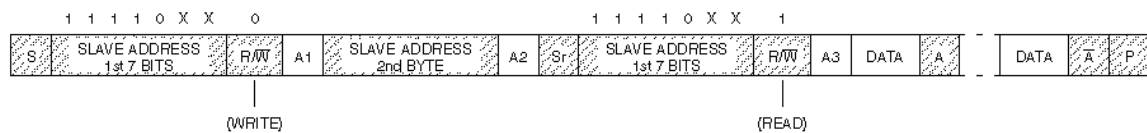


Рисунок 24. Передача мастер-приемник - подчиненный-передатчик с 10-разрядным адресом

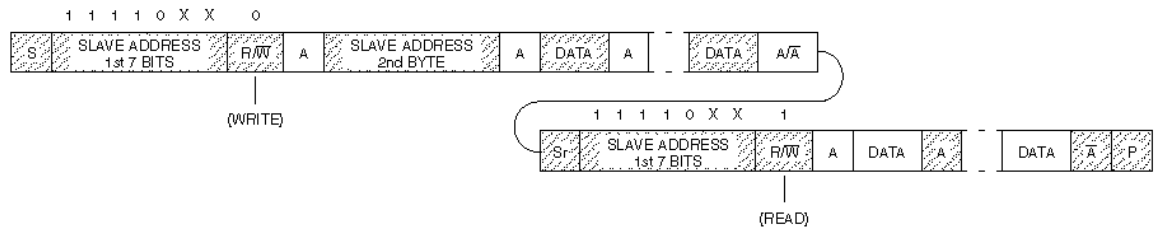


Рисунок 25. Комбинированный формат. Мастер адресует подчиненного с 10-разрядным адресом, затем передает данные к нему и читает данные из него.

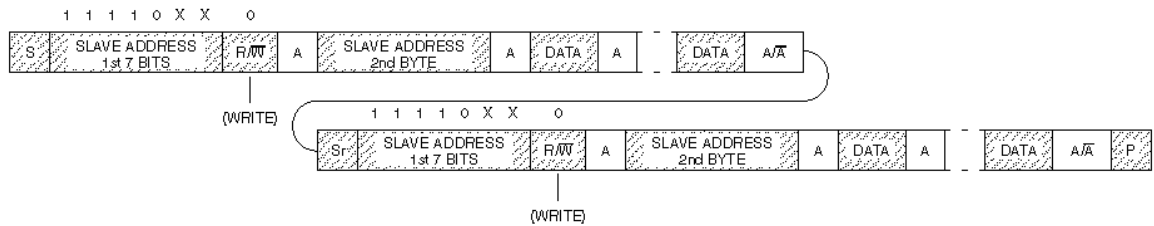


Рисунок 26. Комбинированный формат. Мастер передает данные двум подчиненным с 10-разрядными адресами

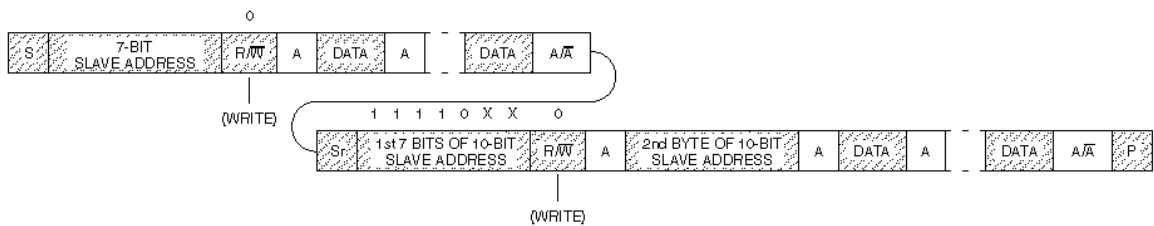


Рисунок 27. Комбинированный формат. Мастер передает данные двум подчиненным - одному с 7-разрядным адресом, и одному с 10-разрядным адресом

ПРИМЕР РАБОТЫ С I2C УСТРОЙСТВОМ

Наиболее просто поэкспериментировать с I²C устройством можно подключив его к принтерному порту персонального компьютера. Для этого необходимо всего один маломощный транзистор и пара резисторов (Рисунок 28).

Данный пример предназначен для работы с последовательной энергонезависимой памятью, но при незначительной модификации программы (полный исходный текст которой на Турбо Паскале приведен ниже). Данная схема и программа Дмитрием Орловым и являются freeware.

```
program I2C_LPT;
{ Copyright Dmitry Orlov, 2:461/27 }

uses CRT;
var
  LPT1: Word;
  A: Word;
  Mode: (mError, mRead, mWrite);
  FileName: String[80];
  Buf : Array[0..2047] of byte;
  F: File;

procedure Pause; assembler;
asm
  push ax
  @@0:
  jmp @@1
  @@1:
  in al, dx
  pop ax
end;

procedure Start; assembler;
asm
  mov dx, lpt1
  mov al, 00000110b {Vcc = 1, SCL = 1, SDA = 1}
  out dx, al
  call Pause
  mov al, 00000111b {Vcc = 1, SCL = 1, SDA = 0}
  out dx, al
  call Pause
  mov al, 00000011b {Vcc = 1, SCL = 0, SDA = 0}
  out dx, al
  call Pause
end;

procedure Stop; assembler;
asm
  mov dx, lpt1
  mov al, 00000111b {Vcc = 1, SCL = 1, SDA = 0}
  out dx, al
  call Pause
  mov al, 00000110b {Vcc = 1, SCL = 1, SDA = 1}
  out dx, al
  call Pause
end;

function PutByte(B: Byte): boolean; assembler;
asm
  mov dx, lpt1
```

```

mov ah, B
mov cx, 8
@@0:
rcl ah,1
jc @@1
mov al, 00000011b {Vcc = 1, SCL = 0, SDA = 0}
out dx, al
call Pause
mov al, 00000111b {Vcc = 1, SCL = 1, SDA = 0}
out dx, al
call Pause
mov al, 00000011b {Vcc = 1, SCL = 0, SDA = 0}
out dx, al

jmp @@2
@@1:
mov al, 00000010b {Vcc = 1, SCL = 0, SDA = 1}
out dx, al
call Pause
mov al, 00000110b {Vcc = 1, SCL = 1, SDA = 1}
out dx, al
call Pause
mov al, 00000010b {Vcc = 1, SCL = 0, SDA = 1}
out dx, al
@@2:
loop @@0
call Pause
mov al, 00000110b {Vcc = 1, SCL = 1, SDA = 1}
out dx, al
call Pause
inc dx
in al, dx
dec dx
and al, 01000000b
mov ah, 1
je @@3
mov ah, 0
@@3:
mov al, 00000010b {Vcc = 1, SCL = 0, SDA = 1}
out dx, al
mov al, ah
end;

```

function GetByte: byte; **assembler;**

asm

```

mov dx, lpt1
mov cx, 8
@@0:
mov al, 00000110b {Vcc = 1, SCL = 1, SDA = 1}
out dx, al
call Pause
inc dx
in al, dx
dec dx
and al, 01000000b
shl al,1
shl al,1
rcl ah, 1
mov al, 00000010b {Vcc = 1, SCL = 0, SDA = 1}
out dx, al
call Pause
loop @@0
mov al, 00000110b {Vcc = 1, SCL = 1, SDA = 1}
out dx, al
call Pause

```

```

mov al, 00000010b {Vcc = 1, SCL = 0, SDA = 1}
out dx, al
call Pause
mov al, ah
end;

```

```

procedure WriteByte(Addr:Word; B : Byte);
begin
  Start;
  PutByte($A0 or ((Hi(Addr) shl 1) and $0e));
  PutByte(Lo(Addr));
  PutByte(B);
  Stop;
  Delay(5);
end;

```

```

function ReadByte(Addr:Word): Byte;
begin
  Start;
  PutByte($A0 or ((Hi(Addr) shl 1) and $0e));
  PutByte(Lo(Addr));
  Start;
  PutByte($A1 or ((Hi(Addr) shl 1) and $0e));
  ReadByte := GetByte;
  Stop;
end;

```

```

procedure ComStr;
var
  PCStr : ^String;
  i: Byte;

```

```

begin
  PCStr := Ptr(PrefixSeg, $80);
  for i := 0 to length(PCStr^) -1 do
    if PCStr^[i] in ['- ', '/'] then
      begin
        case PCStr^[i+1] of
          'r', 'R' : Mode := mRead;
          'w', 'W' : Mode := mWrite;
          else Mode := mError;
        end;
        Delete(PCStr^, i, 2);
        Break;
      end;
    if ParamCount = 0 then Mode := mError
    else FileName := ParamStr(1);
  end;

```

```

begin
  LPT1 := $378;
  asm
  mov dx, lpt1
  mov al, 00000110b {Vcc = 1}
  out dx, al
  end;

```

```

ComStr;
Assign(F, FileName);
case Mode of
  mRead:
    begin
      for A:= 0 to 2047 do Buf[A] := ReadByte(A);
      Rewrite(F, 1);
    end;

```

```
    if IOResult = 0 then
      begin
        BlockWrite(F, Buf, SizeOf(Buf));
        Close(F);
      end;
    end;
mWrite:
  begin
    Reset(F, 1);
    if IOResult = 0 then
      begin
        BlockRead(F, Buf, SizeOf(Buf));
        Close(F);
        for A:= 0 to 2047 do WriteByte(A, Buf[A]);
      end;
    end;
mError:
  begin
    WriteLn('24c16 fileName -r|-w');
    WriteLn(' -r Read');
    WriteLn(' -w Write');
  end;
end;
end.
```

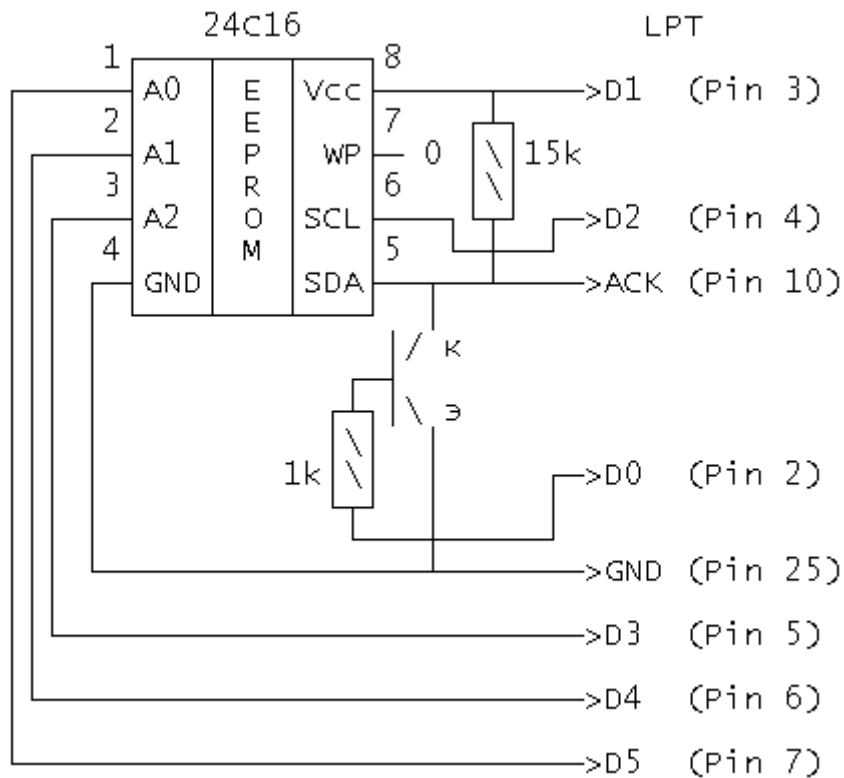


Рисунок 28. Подключение I²C устройства к LPT порту.

Разумеется, в данной статье невозможно описать полностью все нюансы функционирования шины I²C, но, по крайней мере, можно получить о ней общее представление. Для более полного изучения можно посоветовать:

- в интернете
- описания элементов с шиной I²C, например даташиты I²C-EEPROM серии 24XXX фирмы микросхем фирмы Philips и другие.

Харьков, 1997.