

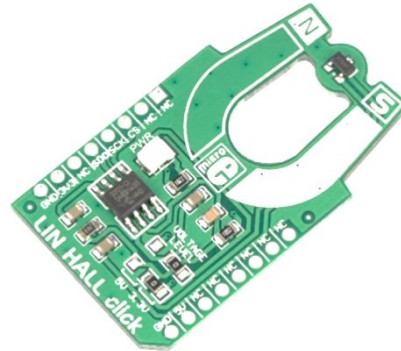
Модуль последовательного периферийного интерфейса

SPI

Последовательный периферийный интерфейс

Последовательный периферийный интерфейс – Serial Peripheral Interface или SPI-интерфейс.

Данный интерфейс используется для работы с различными периферийными устройствами, например, это могут быть различные ЦАП/АЦП, цифровые потенциометры, различные датчики, расширители портов ввода/вывода (GPIO), Flash и EEPROM память и даже более сложная периферия, такая как Ethernet и так далее.



Интерфейс был разработан компанией Motorola, но в настоящий момент используется всеми производителями. Данный интерфейс отличают простота использования и реализации, высокая скорость обмена, но малая дальность действия.

При любом обмене данными по интерфейсу SPI одно из устройств является ведущим (Master'ом), а другое ведомым (Slave'ом). Обычно (но не всегда) в роли ведущего выступает микроконтроллер.

Ведущий переводит ведомое (периферийное) устройство в активное состояние и формирует тактовый сигнал и данные. В ответ ведомое устройство передает ведущему свои данные. Передача данных в обе стороны (дуплексная) происходит синхронно с тактовым сигналом.

SPI представляет собой четырехпроводную синхронную шину, предназначенную для последовательного обмена данными между микросхемами.

SPI представляет собой соединение двух (или более) синхронных сдвиговых регистров, которые являются центральным элементом любого SPI устройства. Для соединения используется конфигурация ведущий/ведомый.

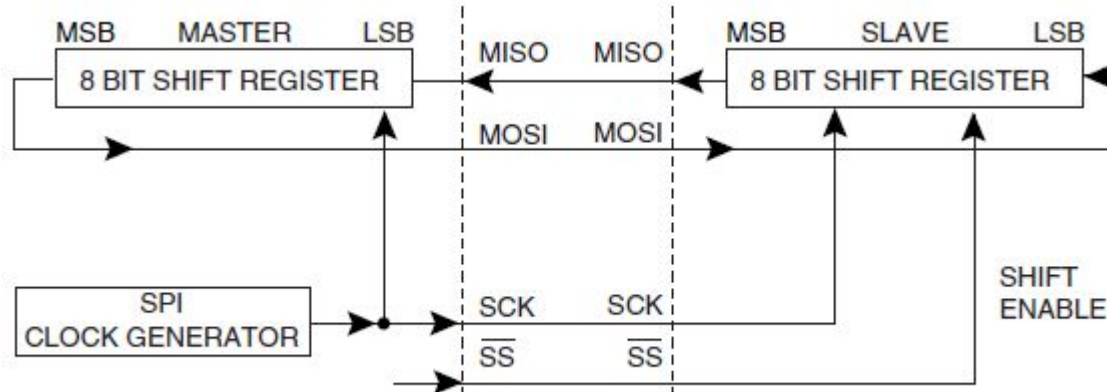


Рисунок 1 – Структурная схема интерфейса SPI

Сигналы, используемые данным интерфейсом, имеют следующее назначение:

- MOSI – Master Output / Slave Input. Выход ведущего / вход ведомого. Служит для передачи данных от ведущего устройства к ведомому.
- MISO – Master Input / Slave Output. Вход ведущего / выход ведомого. Служит для передачи данных от ведомого устройства к ведущему.
- SLK – Serial Clock. Сигнал синхронизации. Служит для передачи тактового сигнала всем ведомым устройствам.
- SS – Slave Select. Выбор ведомого. Служит для выбора ведомого устройства.

Производители микросхем часто используют другие названия для этих сигналов.

Альтернативные варианты могут быть такими:

MOSI – DO, SDO, DOUT. MISO – DI, SDI, DIN. SCK – CLK, SCLK. SS – CS, SYNC.

При подаче импульсов синхронизации на выход SCK, данные выталкиваются ведущим с его выхода MOSI, и захватываются ведомым по его входу MOSI. И обратно, данные передаются ведомым с выхода MISO и захватываются ведущим со его входа MISO.

Таким образом, если подать количество импульсов синхронизации соответствующее разрядности сдвигового регистра, то данные в обоих регистрах обменяются местами.

Отсюда следует что SPI всегда работает в полнодуплексном режиме.

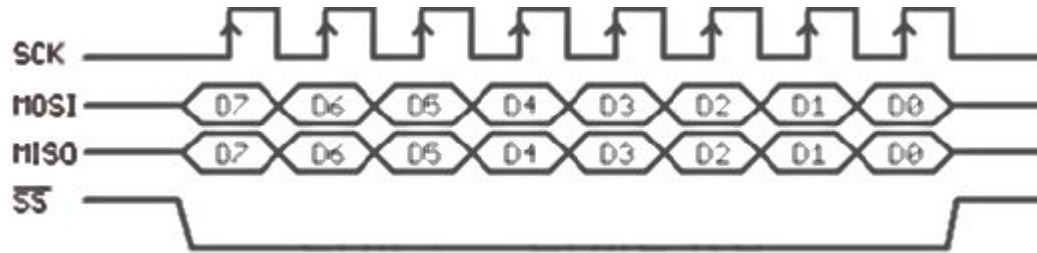


Рисунок 2 – Временная диаграмма работы интерфейса SPI

Контроллер SPI, как правило, реализуется специальным периферийным блоком в микроконтроллере. В большинстве чипов он программируется и может работать как в режиме ведущего, так и в режиме ведомого.

Схемы соединений по SPI

Типовая схема соединения устройств по SPI (рисунок 3).

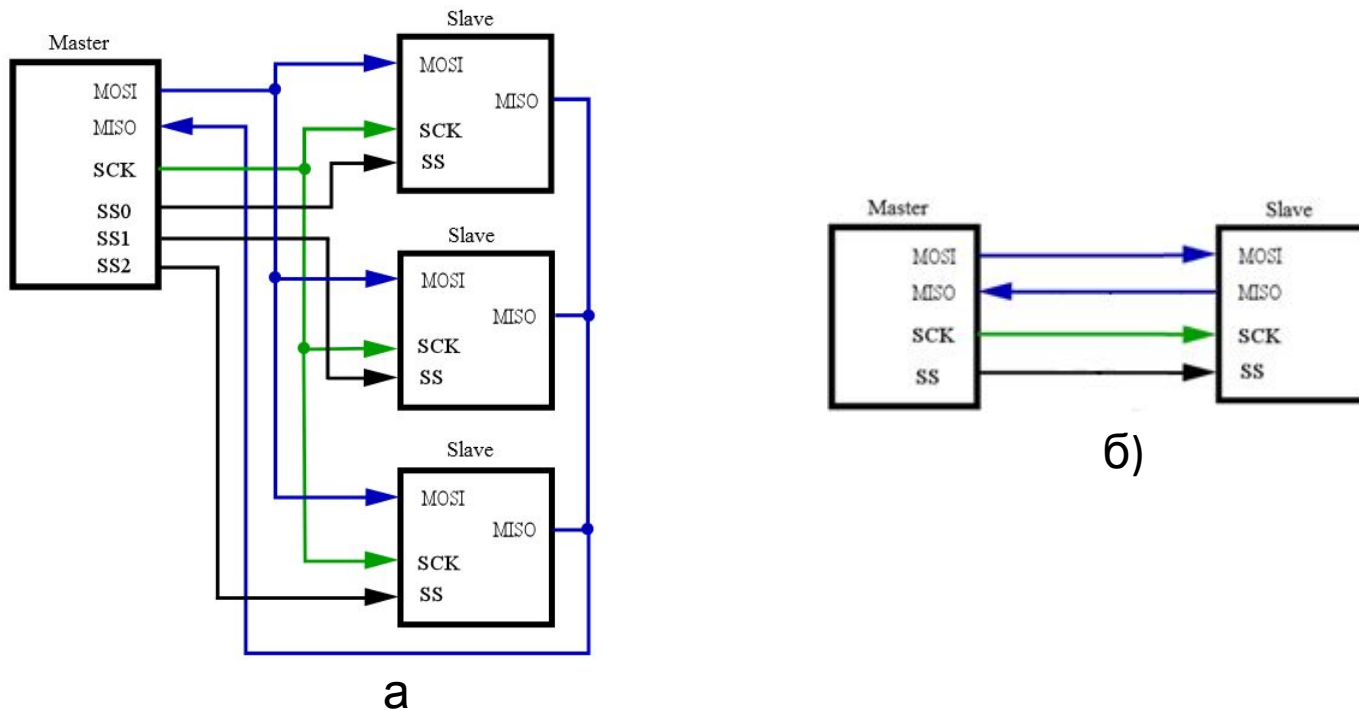


Рисунок 3 – Типовая схема соединения устройств по SPI-интерфейсу

К ведущему все ведомые подключаются параллельно, за исключением сигнала выбора ведомого (SSx). Для каждого ведомого необходим отдельный сигнал выбора ведомого (рисунок 3, а). Для сигналов выбора ведомого могут использоваться как специально предназначенные для этого выходы SPI-контроллера, так и порты ввода/вывода общего назначения (GPIO) микроконтроллера.

Частным случаем независимого подключения является вариант с одним единственным ведомым (рисунок 3, б). Крайне не рекомендуется подтягивать сигнал SS к земле, чтобы устройство всегда было в активном состоянии, так как ведомое устройство может использовать сигнал SS для инициализации или для других служебных целей.

Основное неудобство при независимом подключении ведомых (см. рисунок 3) в том, что для каждого из ведомых необходим отдельный сигнал выбора ведомого (SS).

Каскадная схема подключения, в зарубежной литературе называемая «daisy-chain» (можно перевести как «гирлянда»), лишена такого недостатка. Каскадная схема подключения устройств по SPI (рисунок 4).

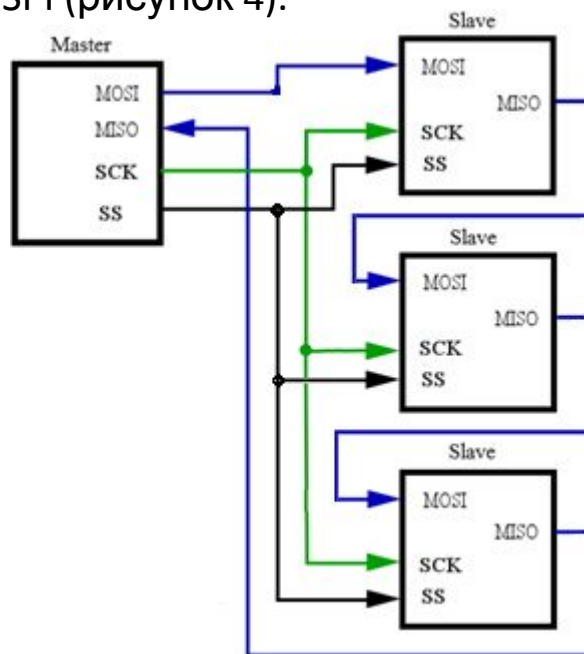


Рисунок 4 – Каскадная схема соединения устройств по SPI-интерфейсу

Как видно из рисунка 4, в каскадной схеме подключения используется общий сигнал выбора ведомого для всех ведомых. Выход каждого из ведомых соединяется со входом следующего. Выход последнего ведомого соединяется со входом ведущего, таким образом образуется замкнутая цепь. При таком подключении можно считать что последовательно соединённые устройства образуют один большой сдвиговый регистр. Соответственно, данные можно записать сразу во все устройства, предварительно собрав нужный пакет, объединяющий данные для каждого из устройств в порядке соответствующем физическому порядку соединения.

Сокращенный (однаправленный) вариант схемы подключения устройств по SPI (рисунок 5).

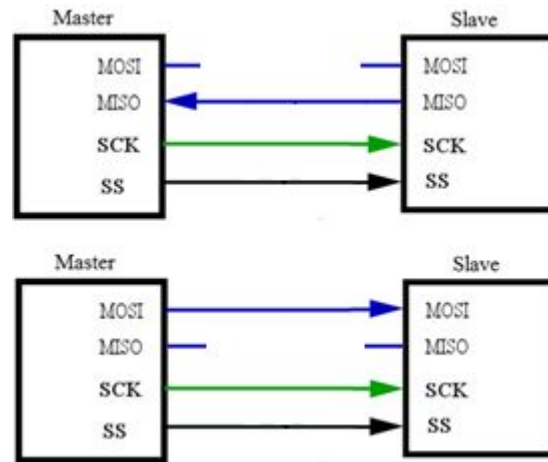


Рисунок 5 – Сокращенная схема соединения устройств по SPI-интерфейсу

Сокращенный вариант схемы подключения, когда линия MOSI или MISO не используется, то есть передача данных осуществляется только в одну сторону используются, например, при подключении к микроконтроллеру внешних микросхем ЦАП и АЦП.

Протокол обмена по

SPI

Протокол обмена по SPI аналогичен логике работы сдвигового регистра и заключается в последовательном побитном выводе/вводе данных по определенным фронтам тактового сигнала.

Установка данных и выборка осуществляется по противоположным фронтам тактового сигнала.

Спецификация SPI предусматривает 4 режима передачи данных, которые отличаются между собой соотношением фазы и полярности тактового сигнала и передаваемых данных.

Эти режимы описываются двумя параметрами:

- CPOL – clock polarity. Полярность тактового сигнала — определяет исходный уровень сигнала синхронизации
- CPHA – clock phase. Фаза тактового сигнала — определяет последовательность

SPI mode 0 — установка и выборки данных.

CPOL = 0, CPHA=0. Тактовый сигнал начинается с уровня логического нуля. Защелкивание данных выполняется по нарастающему фронту. Смена данных происходит по падающему фронту. Моменты защелкивание данных показаны на рисунках стрелочками

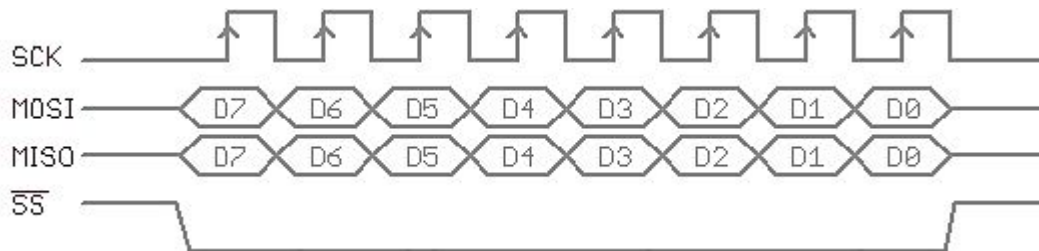


Рисунок 6 – Временная диаграмма работы SPI-интерфейса в режиме mode 0.

SPI mode 1

$CPOL = 0$, $CPHA=1$. Тактовый сигнал начинается с уровня логического нуля. Смена данных происходит по нарастающему фронту. Защелкивание данных выполняется по падающему фронту.

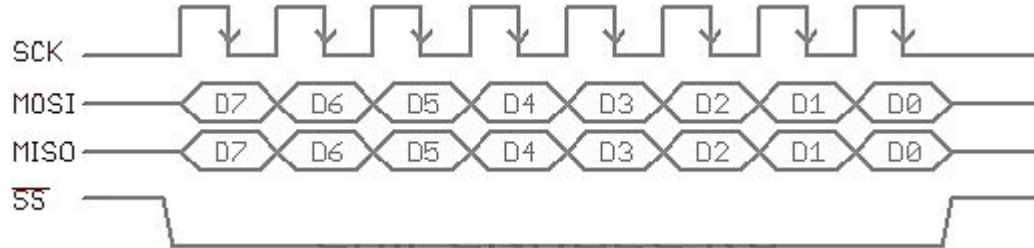


Рисунок 7 – Временная диаграмма работы SPI-интерфейса в режиме mode 1.

SPI mode 2

$CPOL = 1$, $CPHA=0$. Тактовый сигнал начинается с уровня логической единицы. Защелкивание данных выполняется по падающему фронту. Смена данных выполняется по нарастающему фронту тактового сигнала.

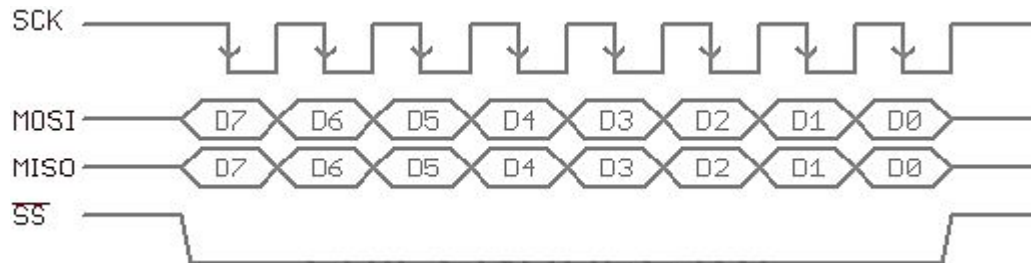


Рисунок 8 – Временная диаграмма работы SPI-интерфейса в режиме mode 2.

SPI mode 3

CPOL = 1, CPHA=1. Тактовый сигнал начинается с уровня логической единицы. Смена данных выполняется по падающему фронту тактового сигнала. Защелкивание данных выполняется по нарастающему фронту.

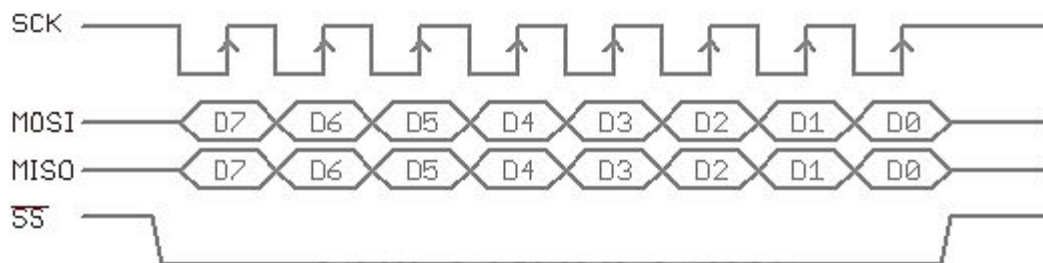


Рисунок 9 – Временная диаграмма работы SPI-интерфейса в режиме mode 3.

Современные микроконтроллеры поддерживают все четыре режима работы SPI.

Стоит отметить, что:

- передача данных по SPI может происходить как старшим битом вперед, так и младшим.
- количество байт передаваемых за время удержания сигнала выбора (SS) ничем не ограничено и определяется спецификацией используемого ведомого устройства.
- в спецификации на ведомое устройство указываются:
 - поддерживаемые режимы работы SPI,
 - максимальная частота тактового сигнала,
 - содержимое передаваемых или принимаемых данных.

Цепи SPI в лабораторном стенде

UNI-DS3

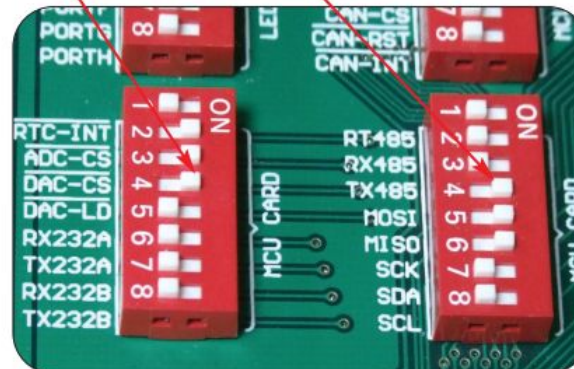
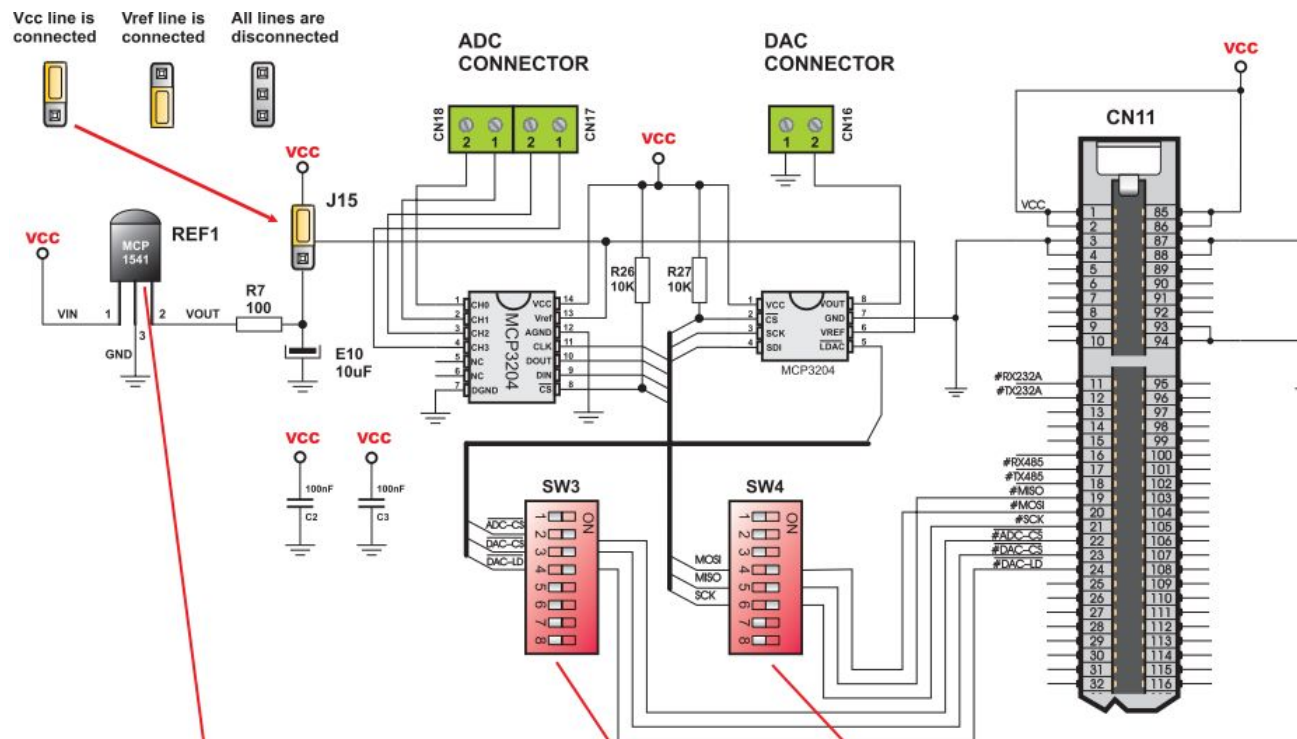


Рисунок 10 – Использование интерфейса SPI в лабораторном стенде UNI-DS3.

Интерфейсная шина

I2C



Двухпроводный последовательный

интерфейс

Двухпроводный последовательный интерфейс I²C (Inter-Integrated Circuit) — последовательная шина данных для связи интегральных схем. Протокол передачи данных по шине I²C гарантирует надежную и качественную прием/передачу данных. Интерфейс разработан корпорацией Philips в начале 1980-х как простая шина внутренней связи для создания управляющей электроники. Существует 3 версии интерфейса в основном различающиеся по скорости передачи данных и режимом разрядной адресации:

- режим медленной передачи (low-speed) 100 кбит/с, режим быстрой передачи (fast-speed) со скоростью до 400 кбит/с и быстросействующий режим (Hs-mode) со скоростью передачи до 3,4 Мбит/с. Также имеются режимы 7-и и 10-и битной адресации.

По всему миру интерфейс I²C стал международным стандартом. Было разработано более 1000 интегральных схем, лицензия на официальное использование спецификации приобретена более чем 50 фирмами. Полная гамма микросхем с интерфейсом I²C, выпускаемых фирмой, насчитывает в настоящее время более 150 наименований, выполненных с применением как перспективной КМОП-технологии, так и с уже ставшей традиционной — биполярной.

Важнейшим критерием, определяющим возможность использования той или иной коммуникационной шины, является спектр её технических характеристик.

Шина I²C относится к классу **двунаправленных асинхронных шин с последовательной передачей данных** и, как следствие, обладает достаточно низкой пропускной способностью. Поэтому её почти не используют в составе персональных компьютеров, а только как вспомогательную для идентификации установленных устройств.

Наибольшее применение шина I²C нашла для согласования работы устройств, наполняющих изделия бытовой и встроенной техники, где она вполне годится.

Основные технические характеристики шины I²C по спецификации 1.0 приведены в таблице

Таблица 1 – Основные технические характеристики шины I²C
(спецификация 1.0)

Наименование параметра	Значение параметра
Скорость обмена low-speed	не более 100 кбит/с
Скорость обмена fast-speed	не более 400 кбит/с
Число адресуемых устройств (7 бит)	до 128
Суммарная длина линий SCL и SDA	не более 4 м
Суммарная паразитная емкость относительно общего провода	не более 400 пФ
Входная емкость на каждый вывод абонента	не более 10 пФ

В таблице 2 представлены основные термины, связанные с шиной I²C.

Таблица 2 – Основные термины, связанные с шиной I²C

Термин	Описание
Передачик	Устройство, передающее данные по шине I ² C.
Приемник	Устройство, принимающее данные с шины I ² C.
Ведущий	Устройство, инициирующее передачу данных и формирующее тактовый сигнал.
Ведомый	Устройство, к которому обращается ведущий.
Конкуренция	Несколько ведущих на шине могут пытаться передать данные без разрушения текущего сообщения.
Арбитраж	Процедура, гарантирующая, что только один ведущий управляет шиной.
Синхронизация	Процедура синхронизации тактовых сигналов от двух или более устройств.

При передаче данных одно устройство является "Ведущим" (Master), которое инициирует передачу данных и формирует сигналы синхронизации. Другое устройство "Ведомое" (Slave), которое может начать передачу данных только по команде ведущего шины (рисунки 11, 12).

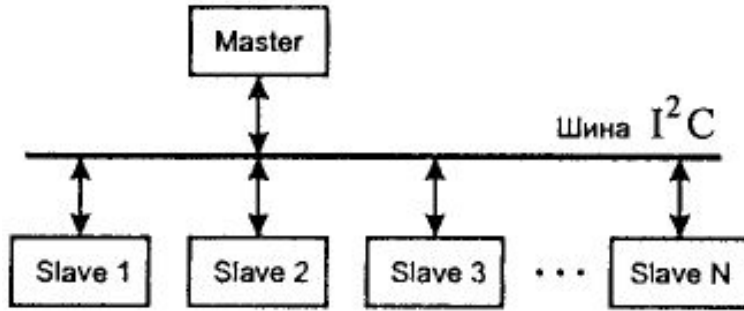


Рисунок 11 – Master-организация шины

Спецификация I²C в принципе поддерживает режим multi-master (рисунок 13), когда к одной шине подключено несколько master-устройств. Но этот режим используется в аппаратуре нечасто из-за сложности доступа к шине, так как на шине может совершать операции только одно master-устройство, остальные «мастера» обязаны отключаться. В противном случае возникает ситуация, называемая **шинным конфликтом**. Из-за которого информация может попросту не дойти до адресата и нарушится работа устройства.

Для того чтобы исключить шинные конфликты, в режиме multi-master должны содержаться процедуры **арбитража** и **синхронизации**, устанавливающие порядок работы master-

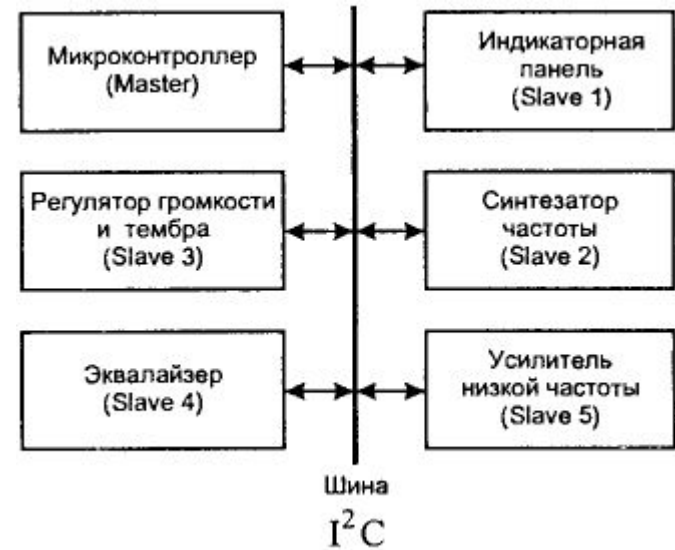


Рисунок 12 – Пример взаимодействия устройств на шине I²C

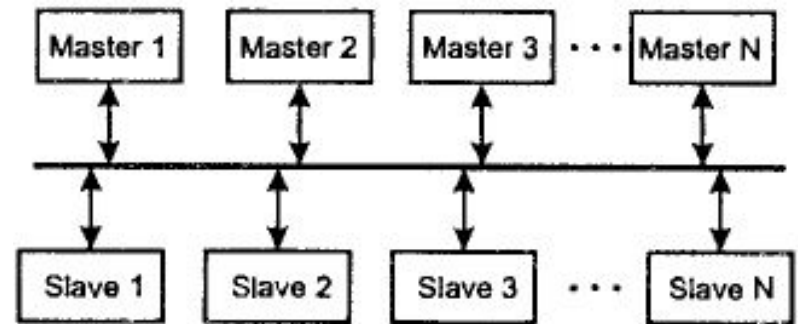


Рисунок 13 – Multi-master организация шины I²C

Физический уровень.

Данные передаются по двум проводам — провод данных SDA и провод синхронизации SCL. Выходы устройства формирующие сигналы данных (SDA) и синхронизации (SCL) должны быть типа «открытый коллектор», чтобы выполнять требования "монтажного И" на шине. Для формирования высокого уровня сигнала на линиях к ним подключаются подтягивающие резисторы.

Число устройств, которые могут быть подключены к шине I²C, ограничивается только максимальной емкостью шины (400 пФ) и способностью адресации этих устройств.

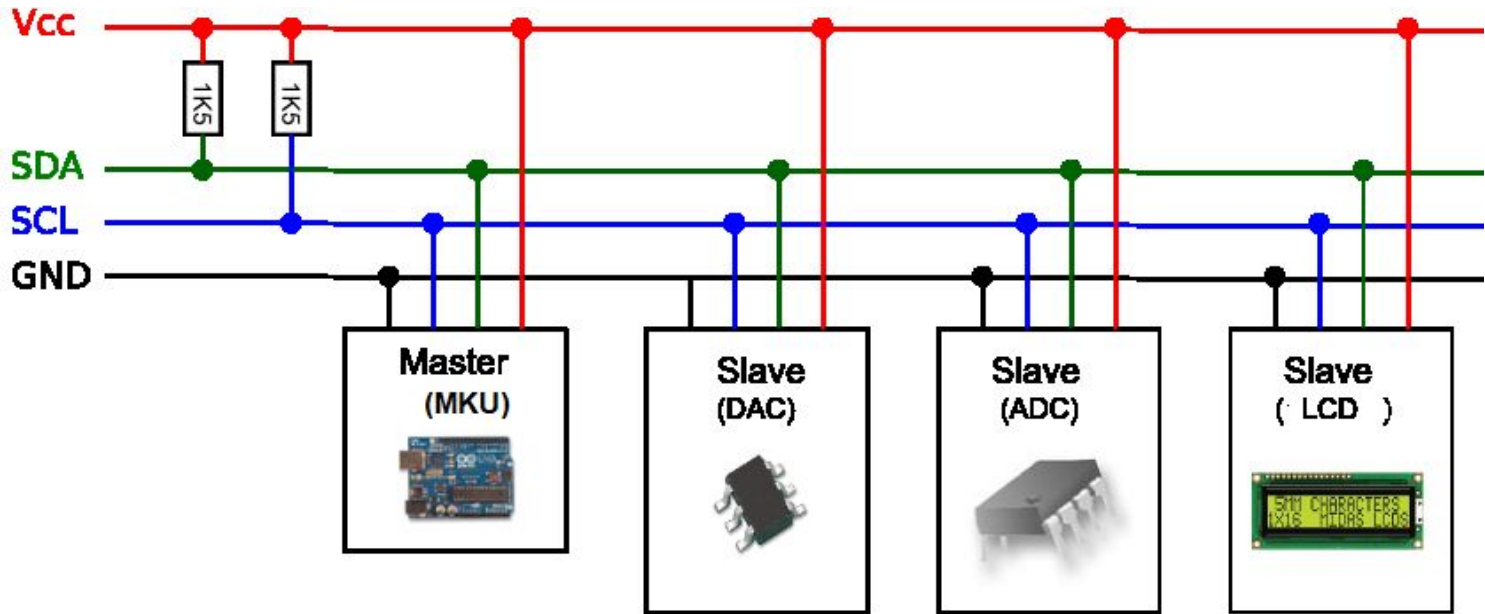


Рисунок 14 – Схема подключения устройств в шине I²C

Такты синхронизации генерирует ведущий (master), ведомый (slave) выдает сигнал подтверждения при приеме байта.

Всего на одной двухпроводной шине может быть до 127 устройств.

Совместимо

Совместимость является важной ^{СТЬ} технической характеристикой шины. Ранее разработанные элементы, обладающие только возможностями низкоскоростного обмена, должны без проблем связываться с высокоскоростными, и наоборот. К шине I²C могут быть подключены интерфейсы трех типов: low-speed, fast-speed, Hs-mode. Обмен данными может быть осуществлен со скоростью, доступной самому медленному интерфейсу. В таблице 3 приведены возможные предельные скорости обмена по совмещенной шине.

Таблица 3. Скорость обмена данными в совмещенных шинах

Направление передачи	Конфигурация шины I ² C			
	Hs + fast + low	Hs + fast	Hs + low	fast + low
Hs – Hs	0...3,4 Мбит/с	0...3,4 Мбит/с	0...3,4 Мбит/с	–
Hs – fast	0...100 кбит/с	0...400 кбит/с	–	–
Hs – low	0...100 кбит/с	–	0...100 кбит/с	–
fast – low	0...100 кбит/с	–	–	0...100 кбит/с
fast – fast	0...100 кбит/с	0...400 кбит/с	–	0...100 кбит/с
low – low	0...100 кбит/с	–	0...100 кбит/с	0...100 кбит/с

Шина I²C также позволяет совмещать устройства с разными напряжениями питания как показано на рисунке 15.

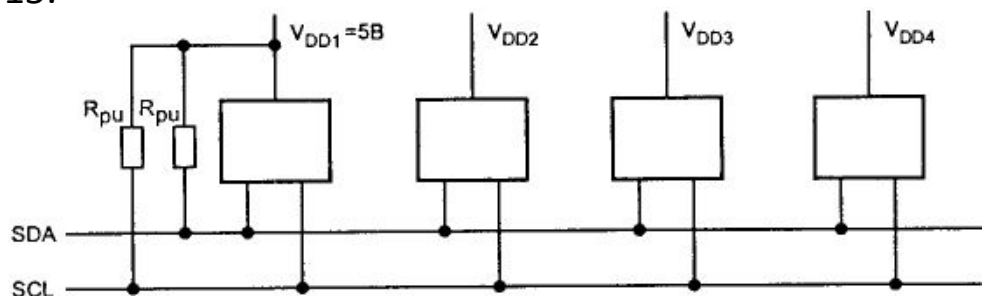


Рисунок 15 – Соединение элементов с разными напряжениями питания

Протокол обмена

данными

Передача любого бита по шине происходит при условии **стробирования** данных SDA по линии SCL. Master-устройство выставляет бит данных «0» или «1» на линию SDA.

Slave-устройство фиксирует этот бит только тогда, когда на линии SCL произойдет перепад сигнала из низкого уровня в высокий (так называемый положительный перепад).

Отсюда следует первое правило организации протокола шины: **смена информации на линии SDA может быть произведена только при нулевом состоянии линии SCL.**

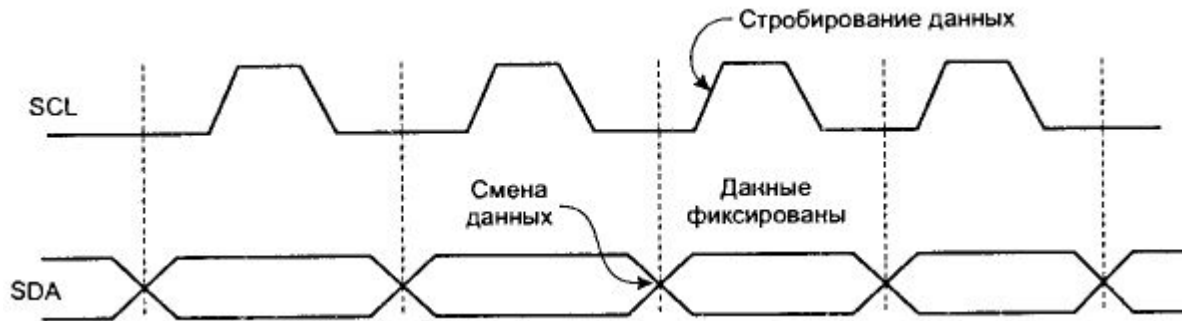


Рисунок 16 – Передача бита по шине I2C

В исходном состоянии оба сигнала SDA и SCL находятся в высоком состоянии.

Для распознавания начала и конца передачи в спецификацию шины были введены специальные условия СТАРТ (Start) и СТОП (Stop). В фирменной документации условие Start имеет условное сокращение «S», условие Stop — «P».

Состояние СТАРТ и СТОП

Процедура обмена начинается с того, что ведущий формирует состояние СТАРТ — ведущий генерирует переход сигнала линии SDA из ВЫСОКОГО состояния в НИЗКОЕ при ВЫСОКОМ уровне на линии SCL (рисунок 17). Этот переход воспринимается всеми устройствами, подключенными к шине как признак начала процедуры обмена.

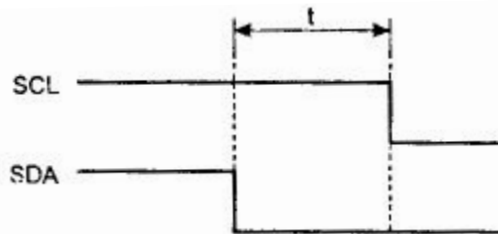


Рисунок 17 – Условие START

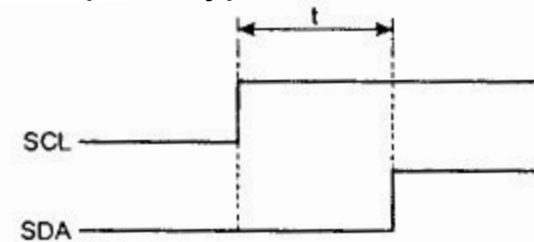


Рисунок 18 – Условие STOP

Условие СТОП возникает при положительном перепаде линии SDA при единичном состоянии линии SCL (рисунок 18).

Состояния СТАРТ и СТОП всегда должны генерироваться master-устройствами.

Укрупненно информационный пакет, передаваемый по шине I²C, выглядит так, как показано на рисунок 19.



Рисунок 19 – Информационный пакет данных на шине I²C

Передача

данных

После отработки состояния Start передатчик последовательно выставляет на линии SDA данные, начиная со старшего бита (MSB) и заканчивая младшим (LSB). Передача данных по шине производится по 8 битов. Данные стробируются по линии SCL импульсами 1...8. На 9-м такте следует сигнал подтверждения (acknowledge). Сигнал подтверждения свидетельствует о том, что данные нормально приняты и обработаны (рисунок 20). В фирменной документации состояние acknowledge условно именуется буквой «А», а в другой литературе ACK.

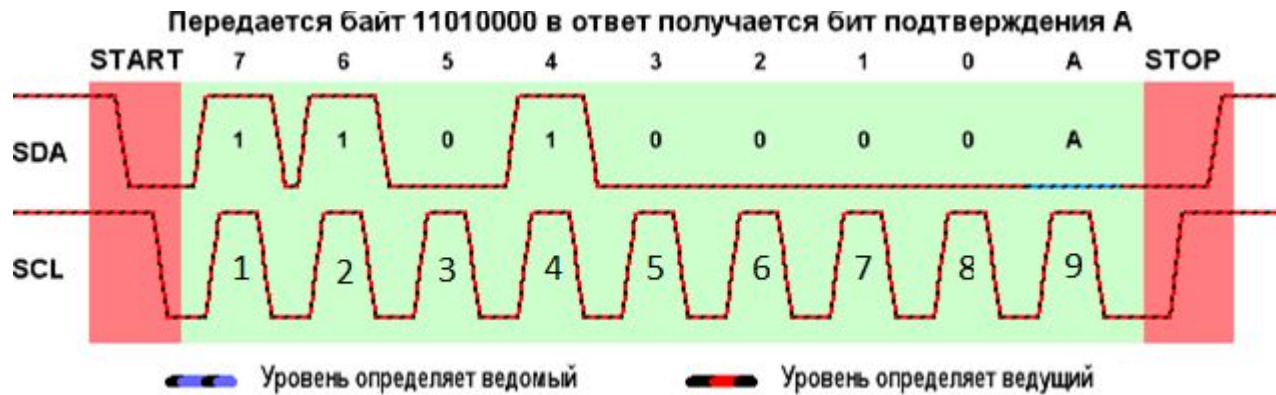


Рисунок 20 – Передача байта по шине I²C с приемом сигнала подтверждения

Когда ведущий шины принимает данные, то на каждый принимаемый байт формируется бит подтверждения, если принятый байт не последний.

Для сообщения ведомому о том, что ведущий прекращает принимать данные по приему последнего байта ACK не формируется. Ведомый отпускает SDA, чтобы ведущий смог передать бит STOP. Ведущий может формировать бит STOP на месте бита подтверждения.

В случае неподтверждения нормального приема (сигнал АСК имеет высокий уровень) передатчику желательно выполнить условие Stop и повторить передачу (рисунок 21).

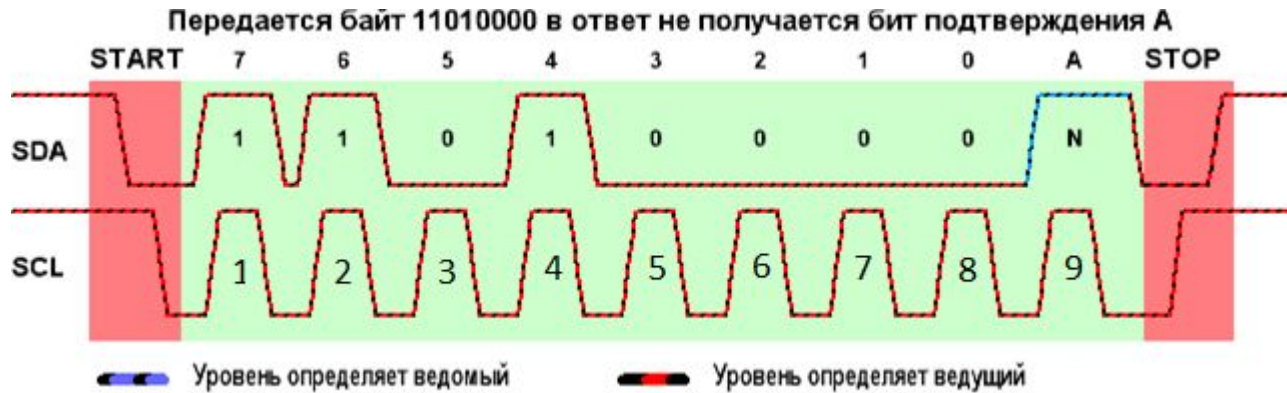


Рисунок 21 – Передача байта по шине I²C с отсутствием сигнала подтверждения

Также может возникнуть ситуация связанная с задержкой обработки данных у slave-абонента при получении сигнала подтверждения (АСК), как показано на рисунок 22.

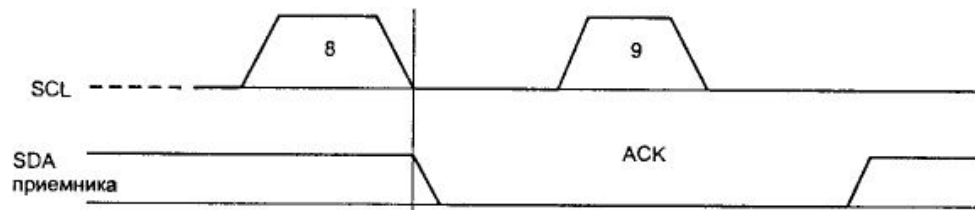


Рисунок 22 – Задержка состояния

Задержка освобождения линии SDA не свидетельствует о неправильном обмене информацией, поэтому master-абоненту достаточно дождаться окончания АСК и продолжить передачу.

Если ведомому необходимо задержать передачу данных, то он может удерживать SCL в низком логическом уровне. Передача данных продолжится, когда ведомый отпустит SCL. Это позволяет ведомому подготовить новые данные для передачи.

Форматы передачи данных на

формат (протокол) передачи **данных** на **шине I²C** — набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами. Эти соглашения задают единообразный способ передачи сообщений и обработки ошибок при взаимодействии программного обеспечения разнесённой в пространстве аппаратуры, соединённой тем или иным интерфейсом.

Каждое устройство на шине I²C должно иметь свой **уникальный адрес**, но которому к нему можно обратиться.

Адрес устройства всегда располагается в определенном месте (в определенных ячейках внутренней памяти).

Адрес может быть **назначен**, а может быть **жёстко определен**.

На шине I²C адреса устройств, называемые SLAVE-адресами, **жёстко определены** при изготовлении микросхем и не подлежат переназначению, то есть модификации. Иногда, для некоторых микросхем переназначить адрес можно (это делается с помощью 2—3 бит, которые нужно предварительно **определить аппаратно**, то есть установить переключки на соответствующие выводы).

Значение SLAVE-адреса можно узнать из документации на соответствующую микросхему.

Так как все абоненты шины обмениваются данными только по линиям SDA и SCL, то в момент начала передачи **все SLAVE-абоненты «слушают» линию** на предмет опознавания своего SLAVE-адреса. Опознавший свой адрес абонент продолжает прием данных и выдачу сигналов ACK, остальные только следят за моментом выдачи состояния STOP.

Для адресации устройств на шине I²C используется два формата адреса:

- 7-разрядный формат с битом чтения/записи R/W;
- 10-разрядный формат (передаются два байта)

Адресация устройств на шине

I²C

Возможны
передачи:

три
формата

- MASTER транслирует данные на SLAVE;
- MASTER читает данные от SLAVE;
- комбинированный формат передачи(записи)/чтения.



S - Start
R/W - Чтение/запись
ACK - Подтверждение

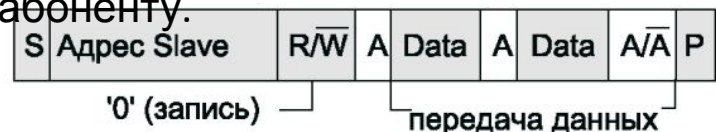
Рисунок 23 – Формат протокола шины
I²C

7-и разрядный формат

Формат кадра при 7-битной адресации, предполагает передачу SLAVE-адреса 7 битами, а восьмой бит должен содержать признак операции «чтение/запись» (R/W).

При 7-битной адресации на шине может присутствовать только 128 устройств с уникальными адресами. Фактически устройств меньше, так как некоторые адреса зарезервированы под некоторые служебные функции.

На рисунке 24 представлен формат передачи данных от MASTER-устройства к SLAVE-абоненту.



- От ведущего к ведомому
- От ведомого к ведущему
- A = подтверждение (SDA низкий)
- \bar{A} = нет подтверждения (SDA высокий)
- S = Start
- P = Stop

Рисунок 24 - MASTER транслирует данные на SLAVE

Обратите внимание: после чтения информационного байта DATA MASTER-абонент обязан подтвердить получение байта сигналом ACK.

Комбинированные форматы используются для работы с последовательной памятью (для сокращения времени доступа к данным).

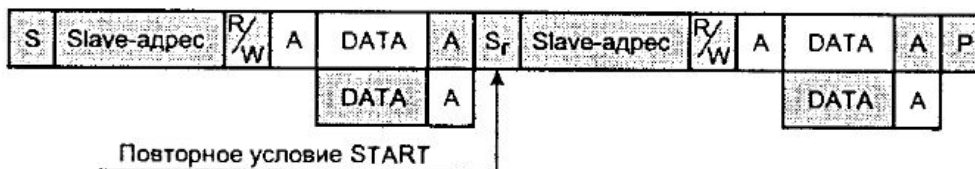


Рисунок 26 – Комбинированный формат

Единственное новшество, введенное в комбинированный формат — повторное условие START (*repeated Start condition*), обозначаемое на рисунке 4 сокращением «Sr». После него можно обратиться к другому устройству не освобождая шину.



Рисунок 25 - MASTER читает данные от SLAVE

Общий вызов может адресовать все устройства, подключенные к шине I²C.

При получении общего вызова устройства должны подтвердить прием выдачей сигнала АСК. Устройства, которые не нуждаются в сведениях, передаваемых при общем вызове, могут игнорировать этот адрес, не выставляя сигнал АСК.

Если же устройство выполнено так, что обязательно требует данных общего вызова, оно ведет себя точно так же, как обыкновенный SLAVE-абонент. Второй байт, следующий за общим вызовом, обычно содержит информационную часть. На рисунке 6 представлен формат обще

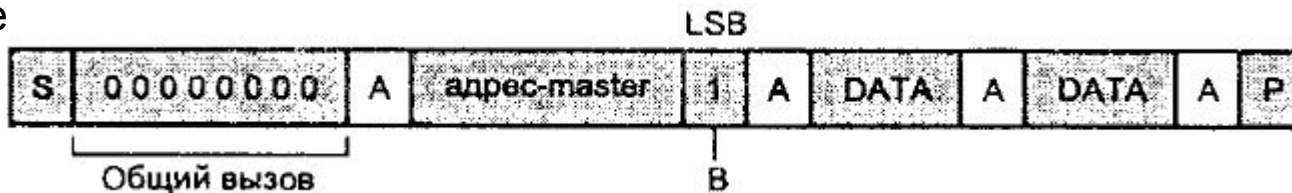


Рисунок 27 – Структура кадра общего вызова

Обратите внимание — информационная часть содержит бит В, который в сочетании с битами 7...1 (обозначенными на рис. 6 символом «X») регламентирует назначение информации, передаваемой в данном случае. Регламентируются следующие комбинации:

- 0000 011 0 (06h) — аппаратный сброс. При получении этой комбинации все абоненты, реагирующие на общий вызов, производят внутренний сброс (рестарт), но аппаратный сброс этих абонентов не должен блокировать шину;
- 0000 010 0 (04h) — запись программируемой части адреса SLAVE-абонента с помощью аппаратных средств. Все абоненты, ответившие на этот адрес, будут заблокированы и не смогут быть сброшены;
- 0000 000 0 (00h) — этот код запрещается использовать;
- XXXX XXX 1 — используется, когда на шине присутствует много совмещенных MASTER/SLAVE-абонентов. В таких случаях часто необходимо генерировать запрос об адресе устройства, которому должна быть передана информация. К примеру, устройство «А», получив от устройства «Б» данные, обработав их, передает информацию устройству «В» вместе с собственным адресом (обычно адреса «MASTER/SLAVE» в совмещенных абонентах совпадают).

10-и разрядный формат

При использовании 10-и разрядного формата передается два байта.



Рисунок 28 – 10-и битный формат протокола I²C

В первом байте передается: пять битов, определяющих, что это 10-разрядный адрес; два старших бита адреса; бит записи/чтения. Во втором байте передается 8 младших бит адреса (рисунок 28).

Адресация с помощью 10 разрядов аналогична 7-и разрядной адресации.

Устройство, получив служебный код в первом байте и опознав возможность приема 10-разрядного адреса, подтверждает это и принимает второй байт.

При совпадении принятого адреса с содержащимся внутри устройства собственным адресом выдается подтверждение ACK и ведется прием данных в обычном режиме до появления состояния STOP.

На рисунке 29 показан формат передачи данных от MASTER-абонента к SLAVE-устройству.



Рисунок 29 – Передача данных от MASTER-абонента к SALVE-устройству

На рисунке 30 приведен формат передачи данных от SLAVE-абонента к MASTER-устройству.

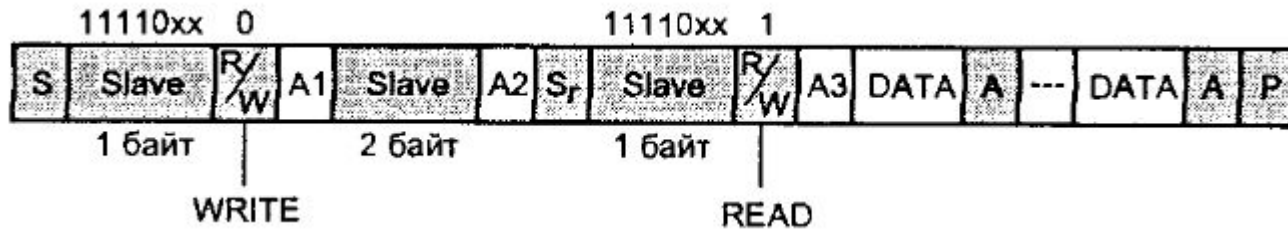


Рисунок 30 – Передача данных от SLAVE-абонента к MASTER-устройству

Вначале MASTER-устройство первым байтом адресует все SLAVE-абоненты соответствующим кодом, и они подтверждают его получение сигналом ACK (A1).

Затем вторым байтом адресуется конкретное устройство с выдачей сигнала ACK (A2).

После выполнения условия «повторный START» (Sr) адресованное SLAVE-устройство сохраняет возможность обращаться к нему, поэтому достаточно повторить первый байт адреса, но уже с другим значением бита R/W и получить ACK (A3).

Комбинированный 10-и разрядный режим

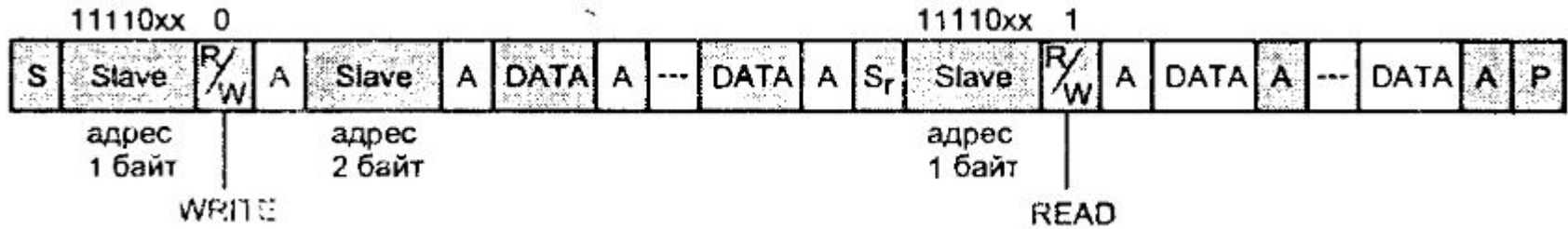


Рисунок 31 – Комбинированный формат. MASTER-абонент адресует SLAVE-устройство с 10-разрядным адресом, затем передает данные и читает данные

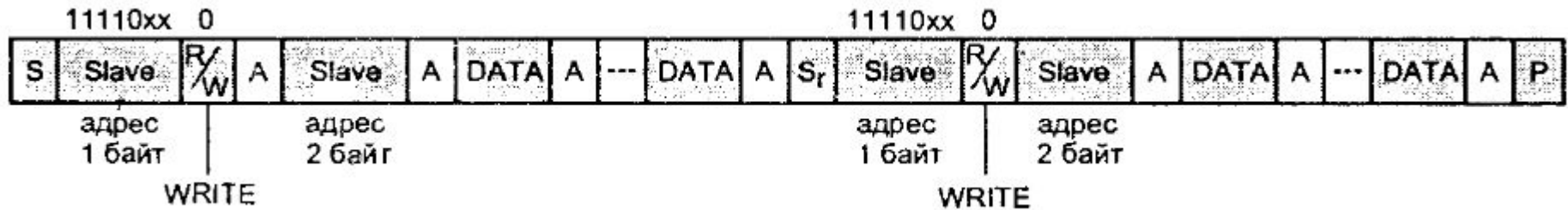


Рисунок 32 – Комбинированный формат. MASTER-устройство передает данные двум SLAVE-абонентам с 10-разрядным адресом

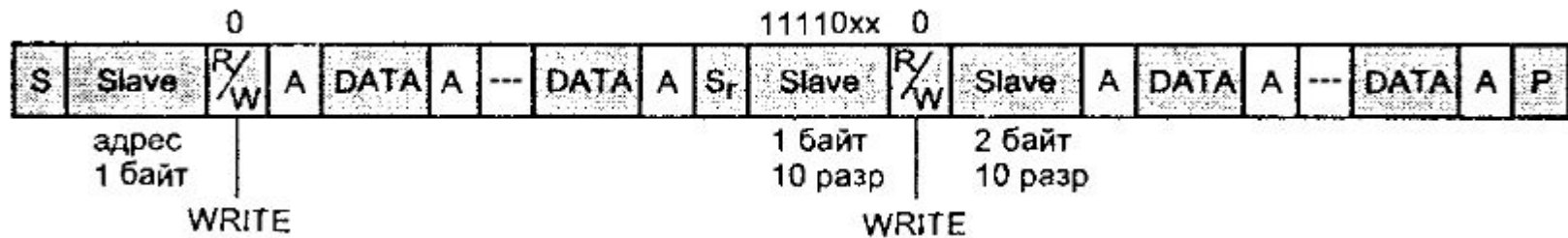


Рисунок 33 – Комбинированный формат. MASTER-абонент передает данные двум SLAVE-устройствам: одному с 7-разрядным адресом, а другому с 10-разрядным адресом

Функциональная схема блока шины I²C (TWI) микроконтроллера AVR

