

# SPI

SPI (serial peripheral interface), последовательный интерфейс периферийных устройств - интерфейс, обеспечивающий полнодуплексную синхронную последовательную связь между устройствами. Очень популярный интерфейс, часто используется для подключения к микроконтроллеру различных внешних периферийных устройств: датчиков, LCD-дисплеев, микросхем энергонезависимой памяти (специализированные микросхемы flash-памяти с интерфейсом SPI, часто используемые для хранения прошивок устройств; SD-карты памяти также могут работать в SPI-режиме), модулей беспроводной связи и многого другого.

Смотрите также:

"SPI в микроконтроллерах STM32"

Оглавление

SPI

[Введение](#)

[Сигнальные линии интерфейса SPI](#)

[Подключение устройств с интерфейсом SPI](#)

[Формат передачи данных SPI](#)

[Полярность и фаза тактового сигнала](#)

[Источники и дополнительная литература](#)

SPI в микроконтроллерах STM32

## Введение

Первоначально интерфейс разработан компанией Motorola, в настоящее время является стандартом де-факто (стандарт, который сложился исторически и признан в результате широкого использования, но не регулируется организациями по стандартизации).

Данный интерфейс исключительно прост, как с точки зрения реализации (допускает простую аппаратную реализацию, а при её отсутствии легко реализуется на микроконтроллере программно), так и с точки зрения использования. При этом он является достаточно эффективным, обладает рядом преимуществ по сравнению с другими интерфейсами.

Документация, предлагаемая компанией Motorola, предусматривает множество вариантов конфигурирования интерфейса. Например, можно выбирать порядок следования битов при передаче фрейма данных (обычно первым передаётся старший бит, но возможна передача от младшего бита к старшему); есть возможность выбрать полярность и фазу тактового сигнала (имеется четыре комбинации полярность/фаза); имеются дополнительные варианты использования сигнала выбора подчинённого устройства SS# (документирован режим, допускающий совместную работу нескольких ведущих устройств) и т.д. Отсутствие строгого формального стандарта приводит к дополнительной вариативности интерфейса при его реализации разными производителями. Например, обычно размер фрейма составляет 1 байт, но существуют реализации с иным размером фрейма; в некоторых случаях имеется возможность программного изменения размера фрейма - предоставляется выбор из нескольких фиксированных значений или даже поддерживается произвольный размер (как правило, в пределах от 4-х до 16-ти бит). Иногда на сигнальные линии возлагаются дополнительные функции. Например, сигнал SS# может не только выбирать устройство (ведомое устройство выбирается низким уровнем на входе SS#), но и обозначать для ведомого устройства границы между передаваемыми блоками данных (допустим, о конце блока данных сообщаем ведомому устройству снятием сигнала выбора, т.е. переходом SS# к высокому уровню). Часто исключаются незадействованные линии. Например, если некоторое простое подчинённое устройство предполагает передачу данных только в одном направлении, то может быть оставлена только одна линия данных. Если предполагается, что ведомое устройство будет единственным на шине и всё время будет находиться в выбранном состоянии, может быть исключена линия SS#.

Наличие множества вариантов конфигурации интерфейса, с одной стороны, упрощает разработку периферийных устройств, использующих SPI, так как позволяет выбрать оптимальный вариант, требующий минимальных затрат на аппаратную поддержку. С другой стороны, требуется внимательность при конфигурировании взаимодействующих устройств, так как для того, чтобы была возможность обмена информацией между ведущим и ведомым устройствами, они должны быть сконфигурированы соответственным образом.

## Сигнальные линии интерфейса SPI

SPI в базовом варианте предполагает наличие четырёх сигнальных линий, т.е., с учётом общего провода, всего требуется пять проводников для подключения устройства с SPI интерфейсом. Используются следующие сигналы:

**MOSI** (Master Output, Slave Input)

Выход ведущего устройства, вход ведомого. Используется для передачи данных от ведущего к ведомому устройству.

**MISO** (Master Input, Slave Output)

Вход ведущего устройства, выход ведомого. Используется для передачи данных от ведомого к ведущему устройству.

**SCK** (Serial Clock)

Тактовый сигнал, генерируется ведущим устройством и управляет ведомым (SCK является выходом для ведущего устройства и входом для ведомого). Тактовый сигнал определяет моменты выборки сигналов на линиях данных и управляет сдвигающими регистрами ведущего и выбранного ведомого устройства.

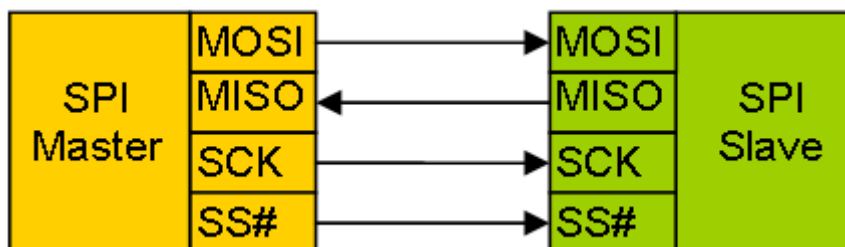
**SS#** (Slave Select)

Выбор ведомого устройства (сигнал является инверсным, активный уровень - низкий). Низкий уровень сигнала SS# (или SS) активирует ведомое устройство: его вывод MISO переходит в режим выхода, по тактовому сигналу от ведущего устройства, ведомое начинает передачу данных через выход MISO и приём через вход MOSI. В некоторых реализациях интерфейса на сигнал SS# могут быть возложены дополнительные функции, например, он может использоваться для обозначения границ блоков данных (переход сигнала к низкому уровню сигнализирует о начале нового блока, установка высокого уровня сообщает подчинённому устройству о том, что передача блока данных завершена).

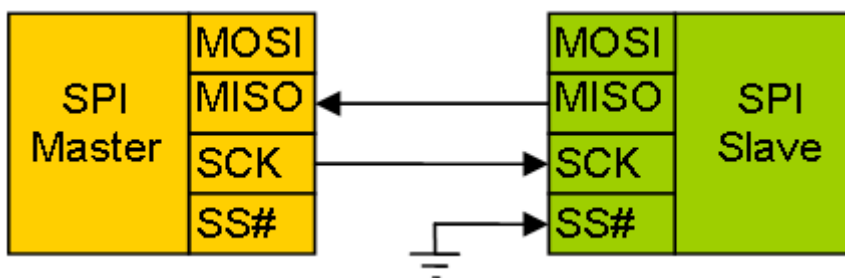
Наименования сигналов у разных производителей могут отличаться от приведённых здесь (и предлагаемых в документе Motorola), однако, по функциональному назначению легко установить соответствие между разными вариантами обозначений.

### Подключение устройств с интерфейсом SPI

Довольно распространённая ситуация - когда нам требуется подключить к ведущему устройству одно ведомое. Для этого необходимо просто соединить одноимённые выводы этих устройств ([рис. %img:con1](#)).

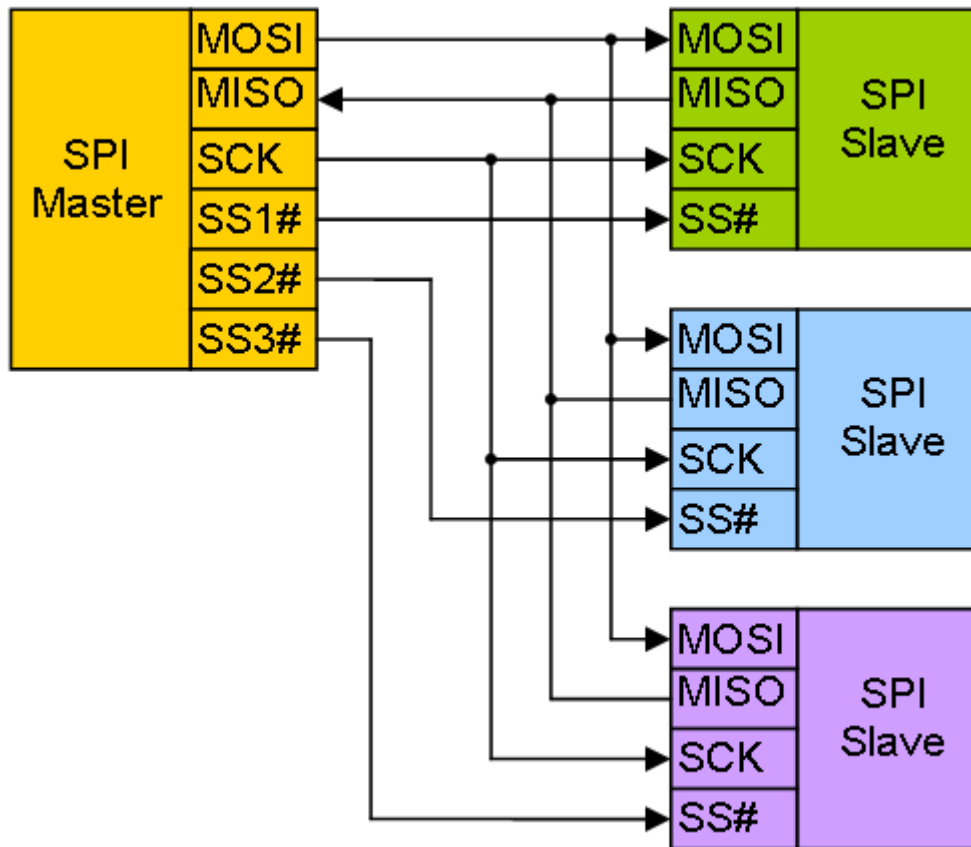


В некоторых случаях количество используемых линий может быть уменьшено. Например, когда требуется лишь односторонняя передача: ведущее устройство только передаёт данные (допустим, на некоторое исполнительное устройство без обратной связи) или только получает данные (допустим, от некоторого датчика, не требующего конфигурирования). Тогда неиспользуемую линию можем исключить. Если допускается, чтобы ведомое устройство было всё время выбрано, его вывод SS# можно просто подключить к общему проводу. В итоге получаем, что для односторонней передачи данных можно обойтись двумя линиями, одна из них - для передачи данных, другая - для тактового сигнала ([рис. %img:con2](#)).



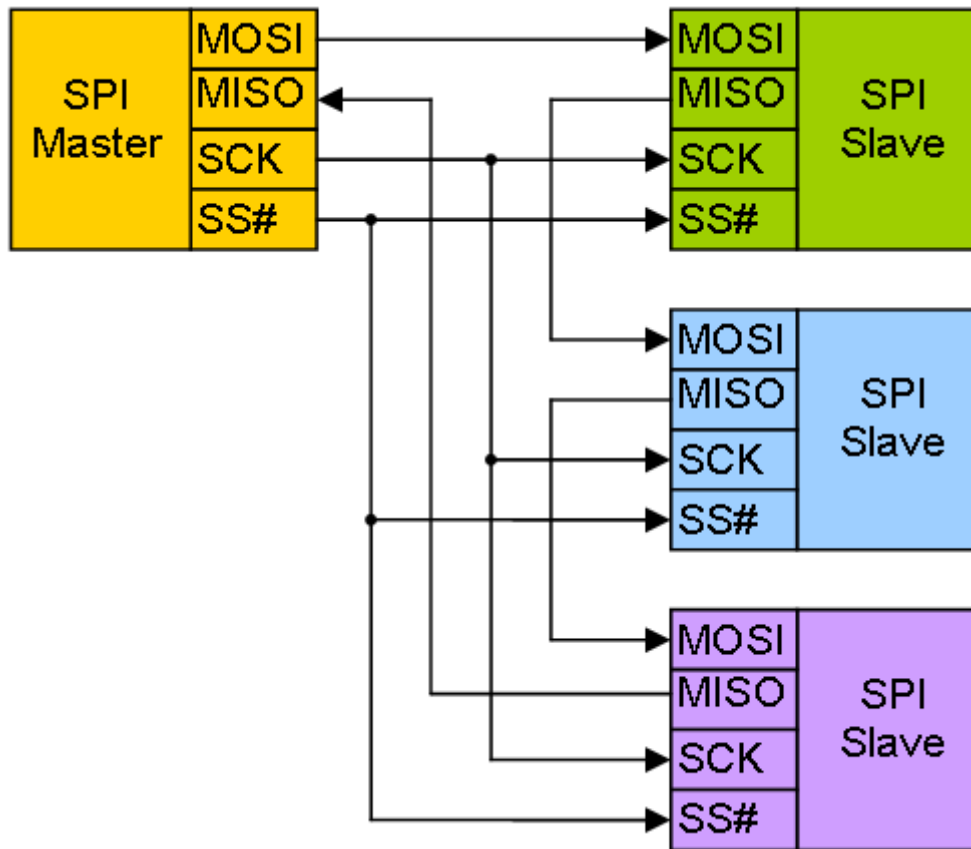
Также существует вариант полудуплексной связи по одной линии данных, когда одна линия данных поочерёдно используется для передачи от первого устройства ко второму и в обратном направлении. Подробнее об этом смотрите: "SPI в микроконтроллерах STM32".

К одному ведущему устройству можно подключить несколько ведомых. При этом, во избежание конфликта на линии MISO, в любой момент времени для обмена данными может быть выбрано не более одного ведомого устройства. Для подключения нескольких ведомых устройств используется гибридная топология: линии MOSI, MISO, CLK подключаются по шинной топологии (проще говоря, у всех устройств одноимённые выводы из перечисленных соединяются вместе). Для линий SS# используется топология "звезда"; ведущее устройство должно иметь отдельный вывод SS# для каждого ведомого устройства ([рис. %img:con3](#)).



Если, допустим, для взаимодействия с несколькими подчинёнными устройствами по SPI используется микроконтроллер, это вовсе не означает, что устройство SPI в микроконтроллере должно иметь аппаратную поддержку нескольких SS# выходов. Выходы SS# тривиальным образом реализуются программно с использованием портов ввода-вывода общего назначения.

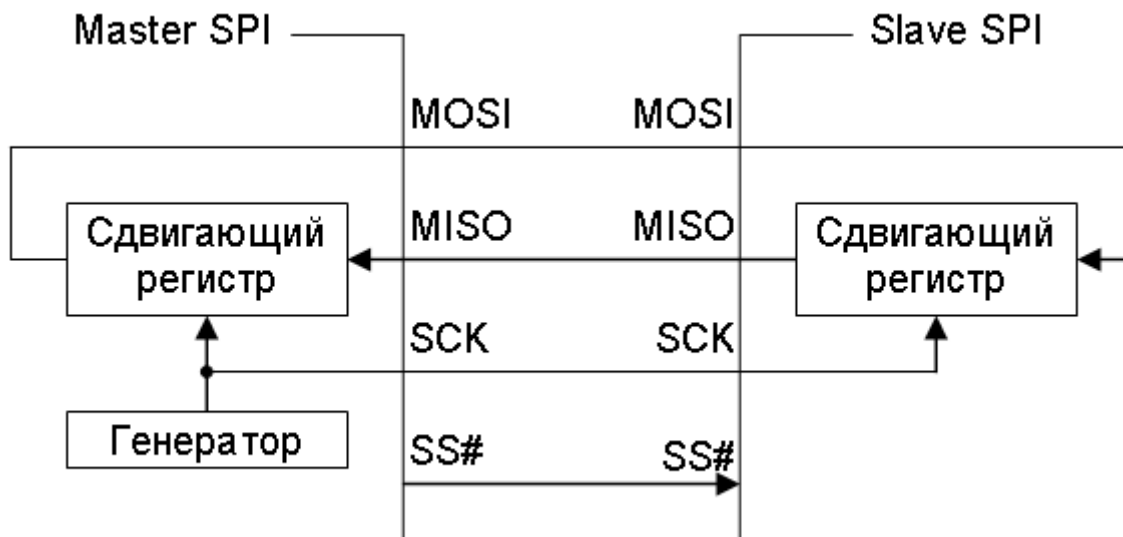
В источниках, посвящённых интерфейсу SPI, описывается также возможность подключения с использованием топологии типа "кольцо": все ведомые устройства управляются общим сигналом SS#; вывод MOSI ведущего устройства подключается к MOSI первого ведомого, вывод MISO первого ведомого - к MOSI второго ведомого и т.д.; вывод MISO последнего ведомого подключается к MISO ведущего - кольцо замыкается ([рис. %img:con4](#)). Теоретически, пакеты данных могут циркулировать по данному кольцу, при условии, что каждое из устройств рассчитано на работу в такой среде.



### Формат передачи данных SPI

В процессе SPI передачи используется классический способ преобразования данных из параллельной формы в последовательную с помощью регистра сдвига. Сначала по шине данных или из некоторого буферного регистра производится параллельная загрузка данных в сдвигающий регистр. Далее, по тактовому сигналу, формируемому ведущим устройством SPI, выполняется побитовый сдвиг данных в сдвигающем регистре. Данные, выдвигаемые с одной стороны регистра, формируют выходной сигнал устройства; входные данные устройства побитно вводятся с другой стороны регистра. Это одновременно происходит как в ведущем, так и в ведомом устройстве. Тактовый сигнал SCK, который формируется ведущим устройством, является общим для ведущего и ведомого; тактовый сигнал синхронизирует сдвиг данных в сдвигающих регистрах и фиксацию информации на двух линиях данных (MOSI, MISO). Можно сказать, что за счёт соединения двух устройств с помощью выводов MOSI и MISO, сдвигающие регистры обоих устройств формируют один распределённый сдвигающий регистр удвоенного размера. Причём, он имеет циклическую организацию. При типичном размере сдвигового регистра 8 бит, размер распределённого регистра составит 16 бит. В результате, после 8 импульсов тактового сигнала, произойдёт обмен содержимым между половинами этого распределённого регистра, т.е. между сдвигающими регистрами ведущего и ведомого устройства.

Интересная особенность интерфейса SPI (в базовой конфигурации), обусловленная его синхронностью, полнодуплексностью и использованием общего тактового сигнала для передачи в обоих направлениях, состоит в том, что для любого из обменивающихся устройств передача всегда сопровождается приёмом данных (и наоборот). Количество полученных бит всегда в точности равно количеству переданных. Даже если устройству в данный момент требуется только что-то принимать, оно всё равно должно и что-то передавать (это могут быть фиктивные данные, скажем, последовательность нулевых или единичных битов; или какая-либо более полезная информация, допустим, о текущем состоянии устройства). Наоборот, если предполагается только передача некоторого блока данных и никакие входные данные не ожидаются, они всё равно будут получены (возможно, фиктивные), в объёме, равном объёму переданных данных. Если входящие данные не нужны, они должны быть просто проигнорированы, отброшены.



Обычно данные передаются, начиная со старшего бита. В этом случае сигнал на выходе устройства определяется старшим битом сдвигового устройства, а данные со входа устройства помещаются в сдвигающий регистр со стороны младшего бита (как для ведущего, так и для ведомого устройства). Иногда используется обратный порядок передачи битов; в микроконтроллерах обычно имеется возможность выбора требуемого варианта.

Как уже было отмечено ранее, интерфейс SPI предполагает, что каждый обмен сопровождается отправкой и получением данных одинакового объёма (на каждый переданный бит получаем входящий бит). Обмен инициируется всегда ведущим устройством, которое выбирает нужное ему ведомое устройство сигналом  $SS\#$  (невыбранные ведомые устройства не вмешиваются в деятельность шины), после чего начинается формирование тактового сигнала. Ведомое устройство, если не хочет пропустить свою первую передачу, должно подготовиться к ней заблаговременно, выполнив запись в регистр данных своего SPI ещё до начала проявления активности ведущим устройством. Таким образом, каждая передача в SPI - это именно взаимный обмен данными, нельзя сказать, что ведомое устройство "отвечает" ведущему, так как "ответ" должен быть готов до "запроса".

Для того чтобы ведущее устройство начало обмен, необходимо выполнить запись в его регистр данных. После чего происходит параллельная загрузка из регистра данных в сдвигающий регистр (сразу, если он свободен или когда закончится передача/приём текущего фрейма, если он ещё занят). Начинается формирование тактового сигнала, сдвигающий регистр по одному биту передаёт данные, с другой стороны в этот же регистр помещаются принимаемые биты данных. Когда ведущее устройство завершит передачу фрейма, в его сдвигающем регистре будет находиться принятый фрейм, который выгружается в приёмный буфер и должен быть считан до момента завершения передачи/приёма следующего фрейма (во избежание потери получаемых данных).

Аналогичные процессы происходят и на стороне ведомого устройства SPI с той лишь разницей, что оно работает по внешнему тактовому сигналу и не может само управлять процессом пересылки данных. Ему остаётся лишь своевременно считывать данные из приёмного буфера по мере его заполнения и успевать записывать новые данные в передающий буфер по мере его освобождения.

### Полярность и фаза тактового сигнала

Интерфейс SPI предполагает возможность задать полярность и фазу тактового сигнала с помощью конфигурационных регистров, в которых с этой целью выделены два бита. В документации Motorola данные биты именуются как CPOL и CPHA (имена могут быть иными у других производителей; простые периферийные устройства могут иметь фиксированную настройку). Два бита дают четыре возможных конфигурации. Естественно, для того, чтобы ведущее и ведомое устройство могли обмениваться информацией, параметры тактирования должны быть заданы одинаковыми.

Под полярностью тактового сигнала понимается его пассивный уровень. Переключения на линии SCK происходят только в процессе передачи данных, в остальное время сигнал удерживается на пассивном уровне. Если CPOL бит равен нулю, это будет нулевой уровень; если CPOL = 1 - единичный уровень. Сигнал при CPOL = 1 является инверсным по отношению к сигналу при CPOL = 0, принципиальных отличий в процессе передачи данных нет.

Бит CPHA определяет фазу тактового сигнала и даёт возможность сделать выбор между двумя существенно отличающимися форматами передачи данных. Проиллюстрируем эти форматы передачи с помощью диаграмм.

Поясним некоторые обозначения на предлагаемых далее диаграммах.

$t_L$  (leading time before the first SCK edge) - промежуток времени от момента перехода  $SS\#$  к низкому уровню до первого (ближайшего) фронта тактового сигнала;

$t_T$  (trailing time after the last SCK edge) - промежуток времени от последнего фронта тактового сигнала до момента переключения  $SS\#$  на высокий уровень;

$t_I$  (idling time between transfers) - промежуток времени между двумя передачами, пока сигнал  $SS\#$  имеет высокий уровень, а ведомое устройство находится в состоянии ожидания.

Так как в SPI используется общий тактовый сигнал, который управляет как ведущим устройством, так и ведомым, то какие-либо жёсткие требования к временным характеристикам или регулярности сигнала отсутствуют - лишь бы не были превышены скоростные возможности ведомого устройства. Требуется, чтобы величины  $t_L$ ,  $t_T$ ,  $t_I$ , а также длительность

импульса тактового сигнала были не меньше половины периода, соответствующего максимально допустимой тактовой частоте для ведомого устройства. В то же время, максимальные длительности промежутков времени не ограничиваются. Ведущее устройство может откладывать моменты переключения сигналов на произвольное время по своему усмотрению.

На диаграммах на линейке SAMPLE отмечены моменты фиксации входных сигналов в ведущем и ведомом устройстве.

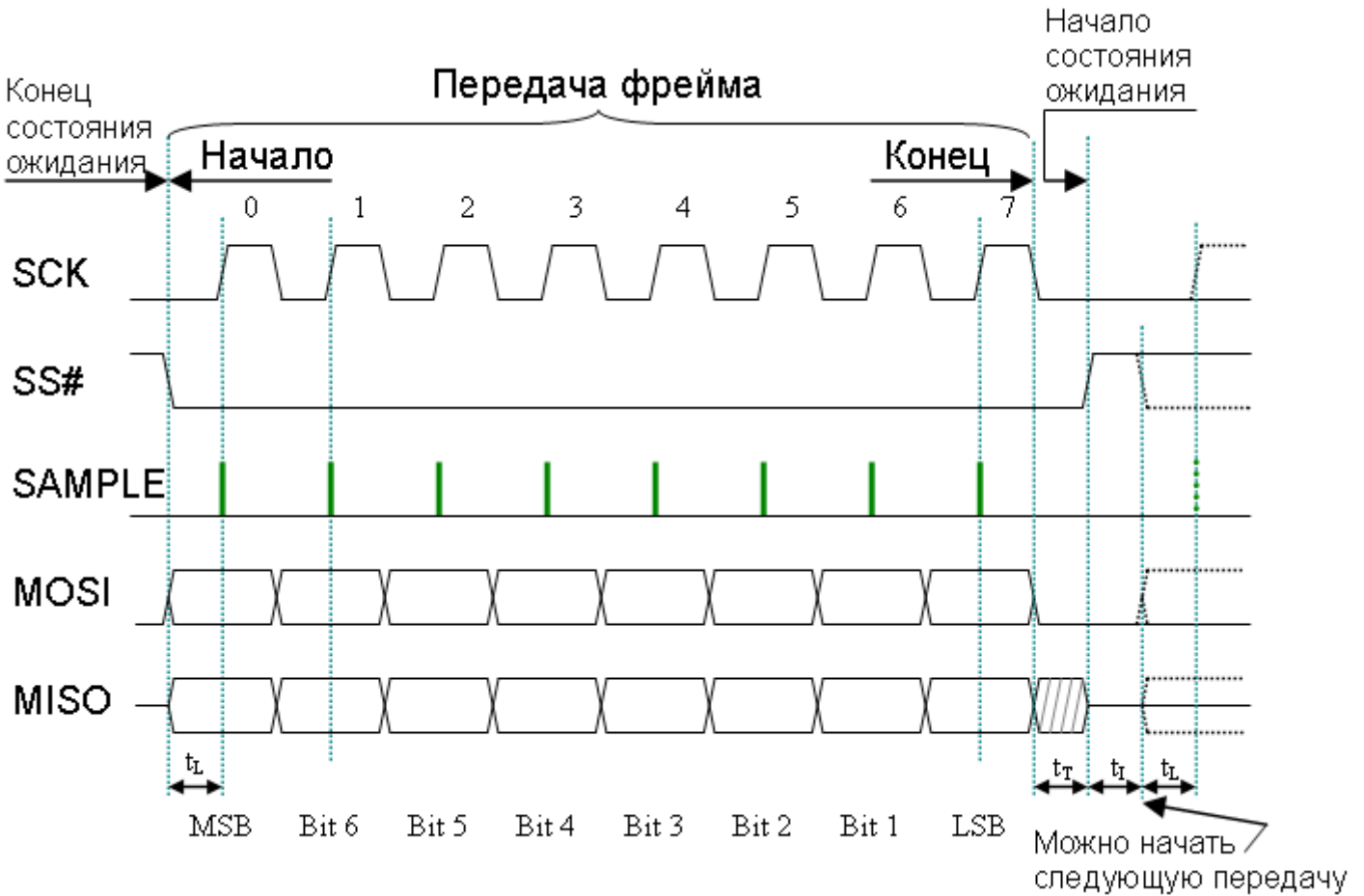


Диаграмма на [рис. %img:cpp\\_00](#) изображает передачу данных по интерфейсу SPI при CPOL = 0 и CPHA = 0. Режимы с CPHA = 0 хорошо подходят для периферийных устройств, в которых первый бит передаваемых данных доступен сразу же после выбора устройства. По первому же фронту тактового сигнала происходит фиксация первого бита, передаваемого из ведущего устройства в ведомое, и первого бита, передаваемого из ведомого устройства в ведущее.

Пока сигнал SS#, формируемый ведущим устройством имеет высокий уровень, ведомое устройство находится в состоянии ожидания - ждёт активации низким уровнем SS#. В состоянии ожидания ведомое устройство игнорирует сигналы на линиях MOSI и SCK; вывод MISO, предназначенный для передачи данных ведущему устройству, находится в высокоимпедансном состоянии. Переключение SS# с высокого на низкий уровень приводит к выбору подчинённого устройства: теперь оно начинает реагировать на сигналы SCK и MOSI, а вывод MISO переходит из высокоимпедансного состояния в состояние выхода. Устройства сразу же выставляют биты данных на своих выходах: ведущее на MOSI, а ведомое - на выходе MISO.

Далее, для осуществления процесса передачи данных, ведущее устройство начинает формировать импульсы тактового сигнала (между моментом переключения SS# и первым тактовым импульсом делается пауза длительностью  $t_L$ ).

По первому фронту тактового сигнала (в данной конфигурации - по нарастающему), происходит фиксация данных, имеющихся на входе ведущего и на входе выбранного ведомого устройства.

По спадающему фронту тактового сигнала происходит сдвиг данных в сдвигающих регистрах ведущего и ведомого устройств, в результате чего зафиксированные ранее биты попадают в регистр, а на выходах устройств выставляются следующие биты данных. Новые биты на входах будут зафиксированы по следующему нарастающему фронту тактового сигнала. Ведущее устройство формирует всего восемь тактовых импульсов, в результате чего происходит обмен данными: байт из сдвигового регистра ведущего устройства оказывается записан в регистр ведомого, а байт ведомого - в сдвигающий регистр ведущего. После этого данные из сдвигающих регистров копируются в регистры данных устройств, откуда они могут быть прочитаны программно. Кроме того, устанавливаются флаги, сигнализирующие о завершении передачи и о получении новых данных.

После успешной передачи, выбор ведомого устройства может быть снят. Через промежуток времени, не меньший 1/2 периода тактового сигнала, ведомое устройство может быть выбрано снова для передачи следующего фрейма.

Следующая передача может быть начата без снятия выбора ведомого устройства (сигнал SS# остаётся на низком уровне). В таком случае, ведомое устройство будет передавать не данные из своего регистра данных, а данные, полученные во время последней передачи (и находящиеся перед началом следующей передачи в сдвиговом регистре). В то же время, ведущее устройство, если используется функция формирования выходного сигнала SS#, всегда снимает выбор ведомого устройства

между последовательными передачами, переводя его в состояние ожидания на время, не менее 1/2 периода тактового сигнала. Это по документации Motorola. В действительности - зависит от реализации, так, в STM32F100xx ведомое устройство не переводится в состояние ожидания между передачами, а если это необходимо, "Руководство..." рекомендует формировать сигнал SS# программно, используя какой-либо порт ввода-вывода общего назначения.

Диаграмма на [рис. %img:cpp\\_10](#) изображает передачу данных по интерфейсу SPI при CPOL = 1 и CPHA = 0.

Процесс передачи отличается от предыдущего рассмотренного (CPOL = 0, CPHA = 0) только инверсным тактовым сигналом, никаких принципиальных отличий нет и подробно рассматривать данный вариант не будем.

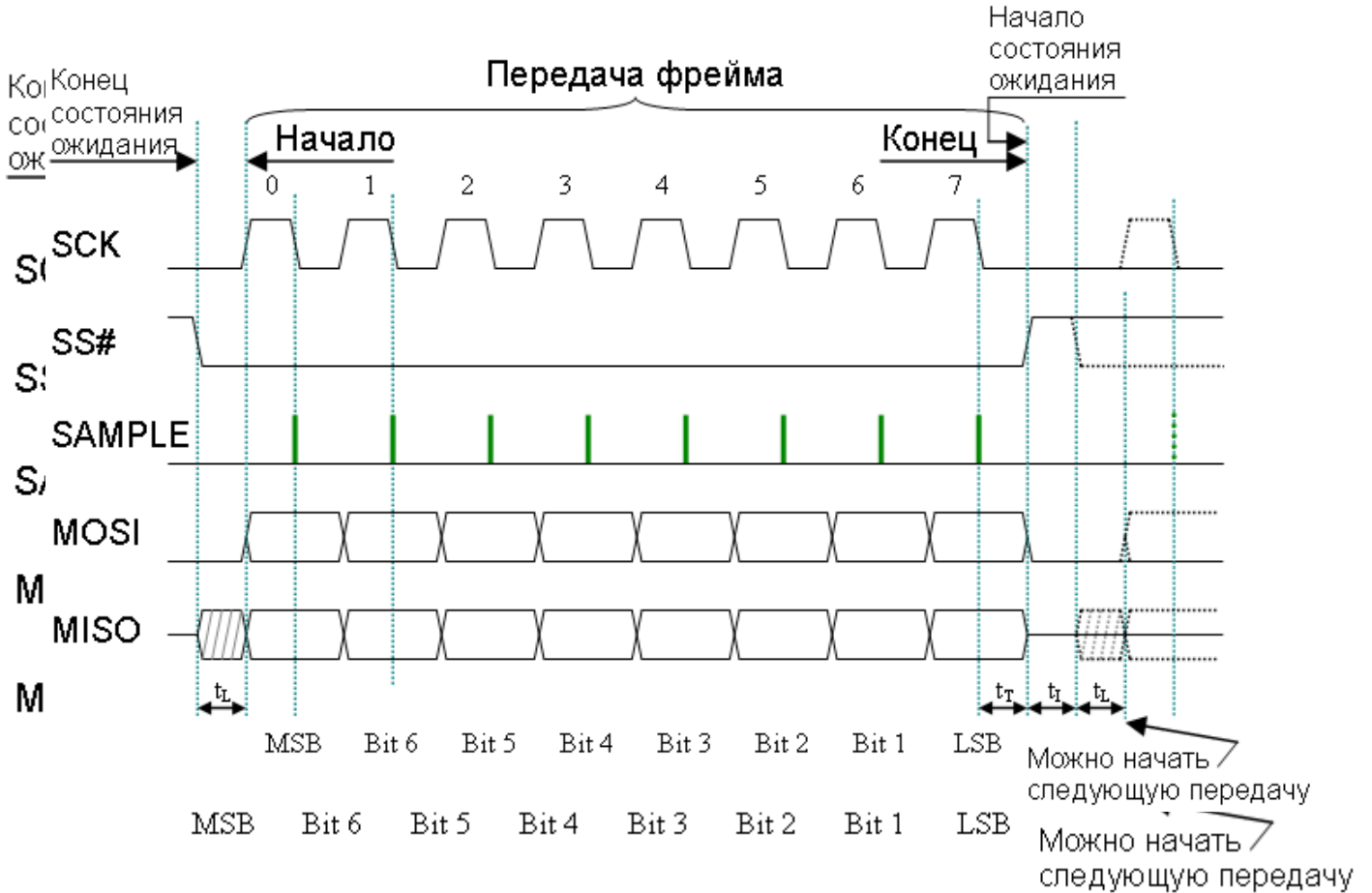


Диаграмма на [рис. %img:cpp\\_01](#) изображает передачу данных по интерфейсу SPI при CPOL = 0 и CPHA = 1. Режимы с CPHA = 1 подходят для тех периферийных устройств, в которых выходные данные становятся доступными только после первого тактового сигнала, по второму фронту происходит фиксация данных.

Через половину периода тактового сигнала после выбора ведомого устройства низким уровнем SS# формируется первый импульс тактового сигнала. По первому фронту тактового сигнала, ведомое и ведущее устройство выставляют на своих выходах первый бит передаваемых данных. С приходом следующего фронта тактового сигнала, происходит фиксация данных на входах ведущего и ведомого устройства. С приходом третьего фронта (первый фронт следующего тактового импульса), происходит сдвиг данных в сдвигающих регистрах обменивающихся информацией устройств, в результате чего зафиксированные во время предыдущего фронта данные попадают в регистры, а на выходах устройств выставляются следующие биты данных.

Процесс повторяется, пока не будут переданы все биты фрейма. После восьми тактовых импульсов данные из регистра ведущего устройства оказываются в регистре ведомого и наоборот, данные из регистра ведомого - в регистре ведущего. Затем из сдвигающих регистров происходит параллельная загрузка информации в регистры данных устройств, и устанавливаются флаги завершения передачи.

В данном режиме при последовательной передаче фреймов не требуется переводить ведомое устройство в состояние ожидания между передачей соседних фреймов - сигнал SS# может всё время оставаться на низком уровне; ведущее устройство может начать следующую передачу сразу же после завершения предыдущей.

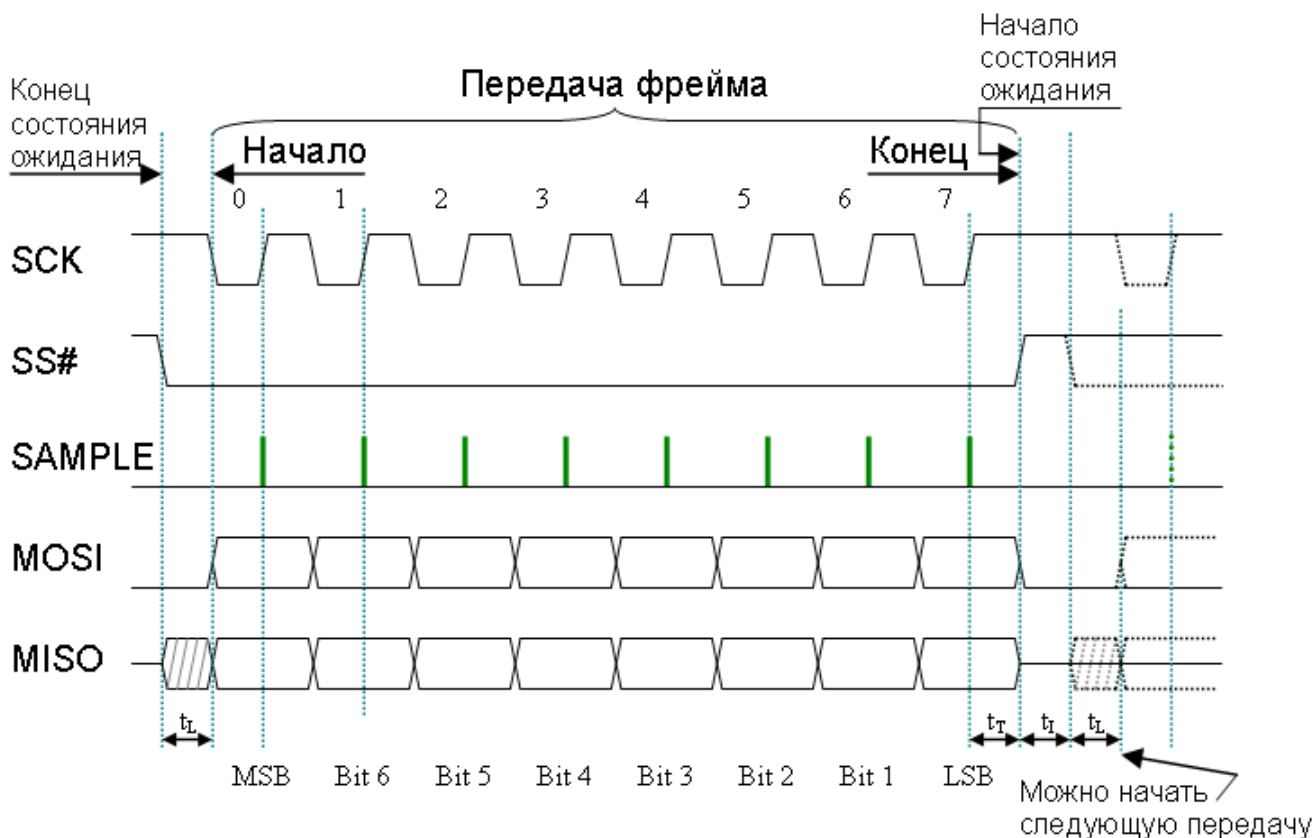


Диаграмма на [рис. %img:cpp\\_11](#) изображает передачу данных по интерфейсу SPI при CPOL = 1 и CPHA = 1. Этот режим отличается от CPOL = 0, CPHA = 1 только инвертированным тактовым сигналом.

Далее:

"SPI в микроконтроллерах STM32"

#### Источники и дополнительная литература

1. "SPI Block Guide V03.06"; Motorola, Inc.; 21 JAN 2000, Revised: 04 FEB 2003.
2. "SPI Block Guide V04.01"; Motorola, Inc. (Freescale Semiconductor, Inc., <http://www.freescale.com>); 21 JAN 2000, Revised: 14 JUL 2004.
3. "ARM PrimeCell Synchronous Serial Port (PL022)"; ARM Limited. Company; <http://www.arm.com>
4. "UM10398, LPC111x/LPC11C1x User manual"; NXP Semiconductors; Rev. 1 — 21 July 2010.
5. "RM0041. Reference manual. STM32F100xx advanced ARM<sup>(R)</sup>-based 32-bit MCUs"; STMicroelectronics; Doc ID16188 Rev 5; June 2016

author: hamper; date: 2020-10-06