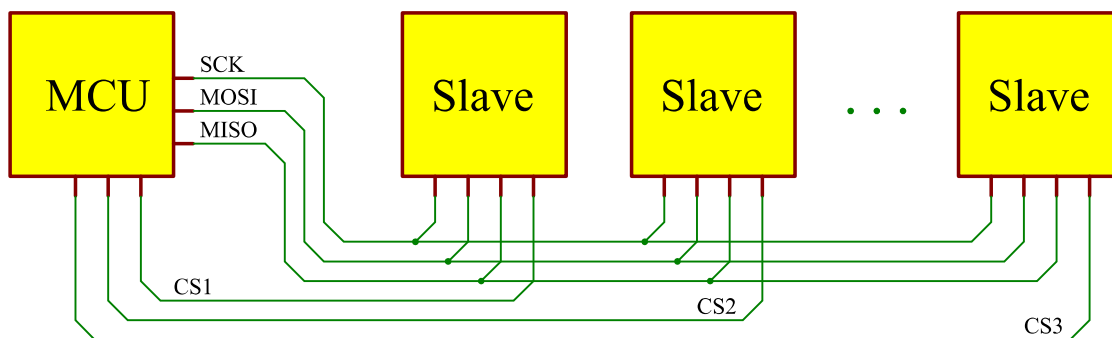


# Интерфейс SPI

О том, что интерфейсы бывают разные, мы уже говорили. Драйвер управления семисегментным индикатором или светодиодной матрицей (MAX7219), который мы выбрали, работает по SPI. Следовательно, чтобы его запустить, необходимо рассмотреть этот интерфейс подробнее.

Устройства на шине неравнозначны, то есть должно быть как минимум одно главное (англ. master) и одно подчиненное (англ. slave) устройство. Впрочем, подчиненных устройств может быть несколько. Чаще всего в роли ведущего устройства выступает МК, а ведомыми могут быть различного рода датчики или драйверы устройств. Шина SPI содержит в общем случае 4 провода.

- **SLK** (сокр. Serial Clock) — по этой линии главное устройство подает тактирующий сигнал, синхронизирующий прием или передачу данных.
- **MISO** (сокр. Master Input, Slave Output) — как понятно из названия, данные идут от главного устройства к подчиненному.
- **MOSI** (сокр. Master Output, Slave Input) — данные идут от подчиненного к главному.
- **CS** (сокр. Chip Select, иногда Slave Select и SS соответственно) — позволяет подчиненному устройству начать работать.

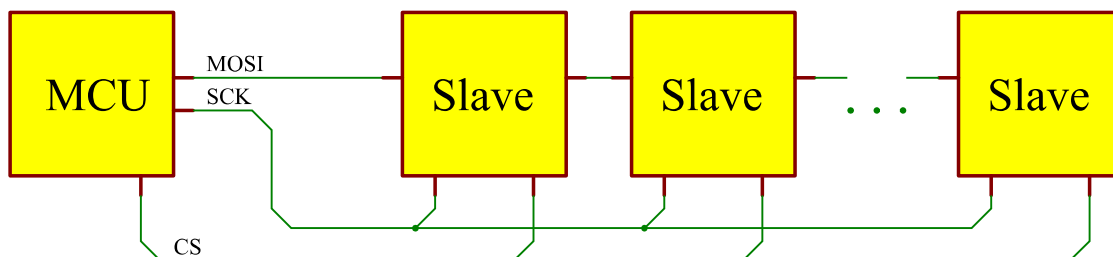


Если вам нужно только отправлять данные, то провод MISO можно просто убрать, и наоборот, если нужно только принимать данные — можно убрать провод MOSI. Также возможна работа в полудуплексном режиме (англ. half-duplex mode).

Так как мастер в цепи один, то он и задаёт тактовый сигнал. Если ничего не передаётся, то тактовый сигнал не поступает на ведомые устройства. Следовательно, что бы что-то принять от ведомого устройства, ведущее должно что-то отправить. Это может быть любой мусор, но для отладки с осциллографом лучше всего выбрать что-то легко узнаваемое, например `0xAA` ( `0b10101010` ).

Часто вход CS инвертирован (это показано чертой над CS). В зависимости от устройства логический уровень на CS может быть разным. Например, «1» — позволяет работу, «0» — запрещает. Подав на CS<sub>1</sub> логическую единицу, мы скажем устройству, что будем работать именно с ним, и все остальные (CS<sub>2</sub> ... CS<sub>N</sub>) просто игнорируют то, что на выводах MOSI и SCK. Таким образом, главный ограничивающий фактор количества подчиненных устройств — это количество ножек МК. Но здесь есть одна хитрость.

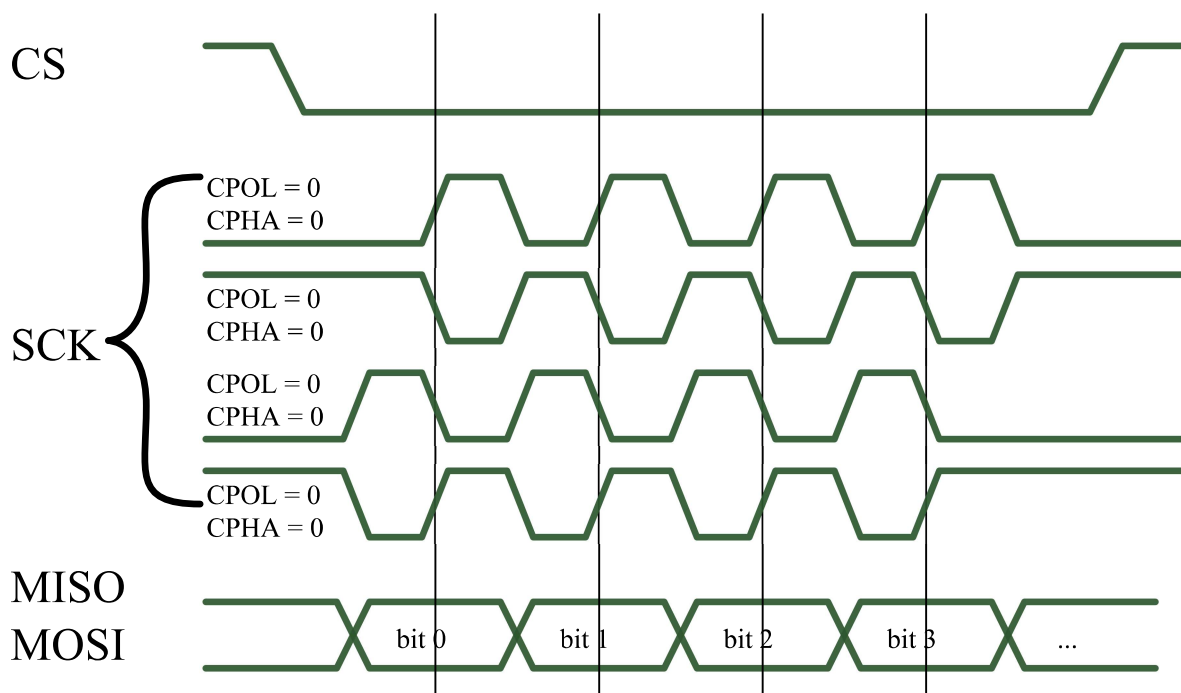
Некоторые устройства, например max7219, принимают данные и обрабатывают их только после того, когда ножка CS деактивирует устройство. Если после первой отправки отправить следующую, то данные будут вымещены новыми и уйдут через выход специальный выход. Следовательно, микросхемы можно подключить каскадно. Да, скорость передачи на шине упадет, но в общем случае потребуется только один провод CS. Такая топология называется daisy-chain.



Микросхемы max7219 в нашем курсе подключены именно таким образом.

Еще одной важной характеристикой работы данного интерфейса является скорость, измеряемая в Гц. Если скорость будет слишком большой, то подчиненное устройство попросту не сможет понять приходящие данные.

И наконец, режимов работы SPI существует целых 4. Они определяются по выбранному фронту. Подробнее ниже:



Режим определяется сочетанием битов настройки **CPOL** (англ. clock polarity) и **CPHA** (англ. clock phase), которые отвечают за полярность и фазу тактирующего сигнала соответственно. **CPOL** определяет, в каком состоянии будет тактирующий сигнал в момент, когда данные не передаются. Если сигнал «0», то это низкий логический уровень, если «1» — то, соответственно, высокий. Следующий бит **CPHA** определяет, по какому фронту (переднему или заднему) необходимо считать значение на линии данных.

## Регистры модуля SPI

Существует ещё один интерфейс, который похож на SPI, под названием I2S. В stm32 оба интерфейса объединены в один модуль, а значит и делят общие регистры. Всего в stm32f103c8 имеется целых три модуля SPI — SPI1 (шина APB2) и SPI2, SPI3 (шина APB1). Рассмотрим только то, что нам необходимо для работы с микросхемой max7219.

Как и у другой любой периферии начать стоит с регистров настройки CR1 и CR2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Первые два бита, CPOL и CPHA, нам уже знакомы, они отвечают за режим работы интерфейса. MSTR, как можно догадаться, задаёт роль устройства на шине. Записав 1 микроконтроллер становится ведущим, и наоборот, записав 0 делает его ведомым. Следующая группа битов BR позволяет настроить делитель частоты тактирования шины: 000 соответствует делению на 2; 001 делению на 4; и т.д. Бит SPE разрешает или запрещает работу модуля; трогать его следует в самый последний момент, когда настройка завершена. LSB\_FIRST как понятно из названия отвечает за порядок бит. Биты SSM и SSI задают работу CS, обычно её оставляют программной, для ведущего устройства. Бит DFF задаёт размер данных, 0 соответствует 8 битам, 1 16-ти. Модуль может аппаратно вычислять суммы CRC, но мы их использовать не будем (биты CRC\_EN и CRC\_NEXT). Последние два бита задают конфигурацию проводов: BIDI\_MODE задаёт двунаправленный (0) или однонаправленный (1) режим, а BIDI\_OE говорит какая из линий в однонаправленном режиме будет использоваться, MOSI (1) или MISO (0). Бит RXONLY работает влияет на работу только двухпроводной линии, при записи туда 1 SPI будет работать только на приём данных.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	Res.	Res.	SSOE	TXDMAEN	RXDMAEN
								rw	rw	rw			rw	rw	rw

Следующий регистр CR2 в основном отведён под прерывания. Всего событий от модуля SPI может быть три: буфер передачи пуст (TXEIE); буфер заполнен (RXNEIE); произошла ошибка (ERRIE). Биты TXDMAEN и RXDMAEN разрешают или запрещают запрос к модулю DMA по завершению передачи или приёма соответственно.

Ошибок передачи может быть несколько, что бы определить что именно пошло не так, следует обращаться к регистру статуса SR. Там же устанавливаются флаги других прерываний.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
								r	r	r	rc_w0	r	r	r	r

Бит BSY сигнализирует, что идёт приём или передача данных (буфер не пуст). Если данные будут приходить быстрее, чем они будут считываться, то очевидно часть данных будет потеряно. Модуль в таком случае выставит 1 в бите OVR. Флаг MODF сообщает, что произошла ошибка режима работы, ножка CS была подтянута к нулю из вне. Данный флаг удобен, когда нужно организовать работу нескольких ведущих устройств на шине. Так как данные проходят через среду, они могут быть испорчены. Флаг CRCERR сигнализирует о такой ошибке. UDR предназначен для I2S. Последние два флага отвечают за прерывания по завершению приёма (RXNE) и передачи (TXE).

Так как SPI и I<sup>2</sup>S разные интерфейсы, но повешены на один модуль, то нужно указать какой именно интерфейс мы собираемся использовать. Делается это через регистр `SPI_I2SCFGR`, а конкретно через бит `I2SMOD`. Записав туда `0` модуль будет работать в режиме SPI.

Последний регистр который нам понадобится — буфер `DR`. Он устроен хитрым образом: при записи в него он перекладывает данные в буфер отправки, а при чтении отдаёт то, что хранится в буфере приёма.

Можно приступать к написанию драйвера для микросхемы MAX7219.

---

[Назад](#) | [Оглавление](#) | [Дальше](#)