

# Интерфейсы передачи информации

То что вы видите на мониторе — интерфейс. Клавиатура с мышкой — интерфейс. И даже окно регистратуры в больнице — это тоже интерфейс.

**Интерфейс** (англ. interface) — общая граница между двумя функциональными объектами, требования к которой определяются стандартом; ... // Wikipedia

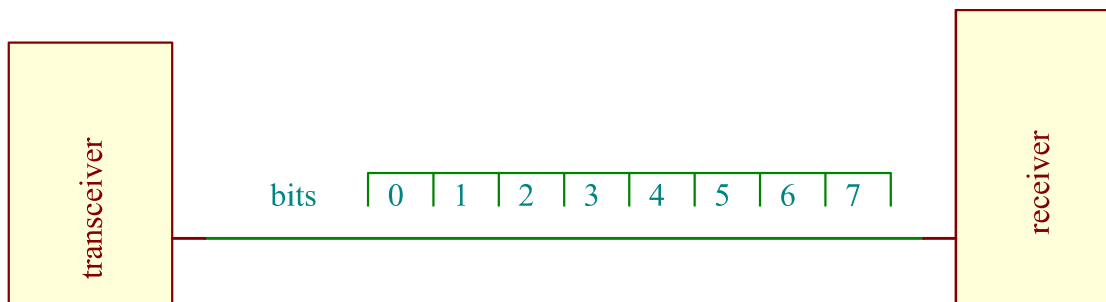
Во встраиваемых системах чаще всего под словом «интерфейс» понимают физический блок МК отвечающий за передачу данных. Они могут быть последовательными или параллельными, синхронными или асинхронными, дифференциальными или обычными.

## Параллельные и последовательные интерфейсы

Допустим нам нужно переслать 1 байт (8 бит) информации. Как это сделать? Можно выделить под каждый бит по одной ножке МК (линии), тогда для передачи потребуется 8 ножек, плюс одна, которая будет говорить принимающей стороне что передача закончена и нужно считать входной сигнал. Такой интерфейс называется параллельным (англ. parallel): группа битов передаётся одновременно за один квант времени. К таким интерфейсам относится PCI (32 линии) и её можно встретить в LCD знакогенерирующих индикаторах (например WINSATR).



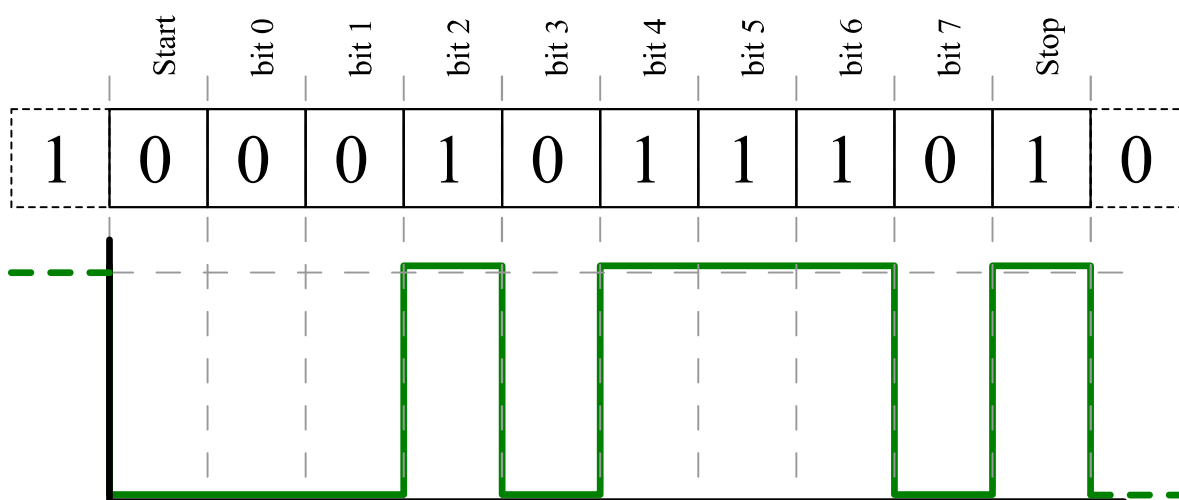
Можно поступить по другому и передавать данные по одной линии, кодируя, например, **1** как высокий уровень сигнала (3,3 В), а **0** как низкий (0 В). В таком случае потребует всего одна ножка МК для передачи и одна что бы сообщать когда этот сигнал считывать. Такие интерфейсы называют последовательными (англ. serial): N битов передаётся по одному за N-квантов времени. Примером последовательного интерфейса — USART.



Параллельный интерфейс, как не сложно догадаться, быстрее в N-раз, однако требует в N-раз больше линий. В микроконтроллерах чаще всего присутствуют только последовательные интерфейсы (периферийные блоки, которые делают всё автоматически), к ним относятся SPI, I<sup>2</sup>C, I<sup>2</sup>S, CAN, USART и USB. Некоторые из них мы рассмотрим подробнее в этом курсе.

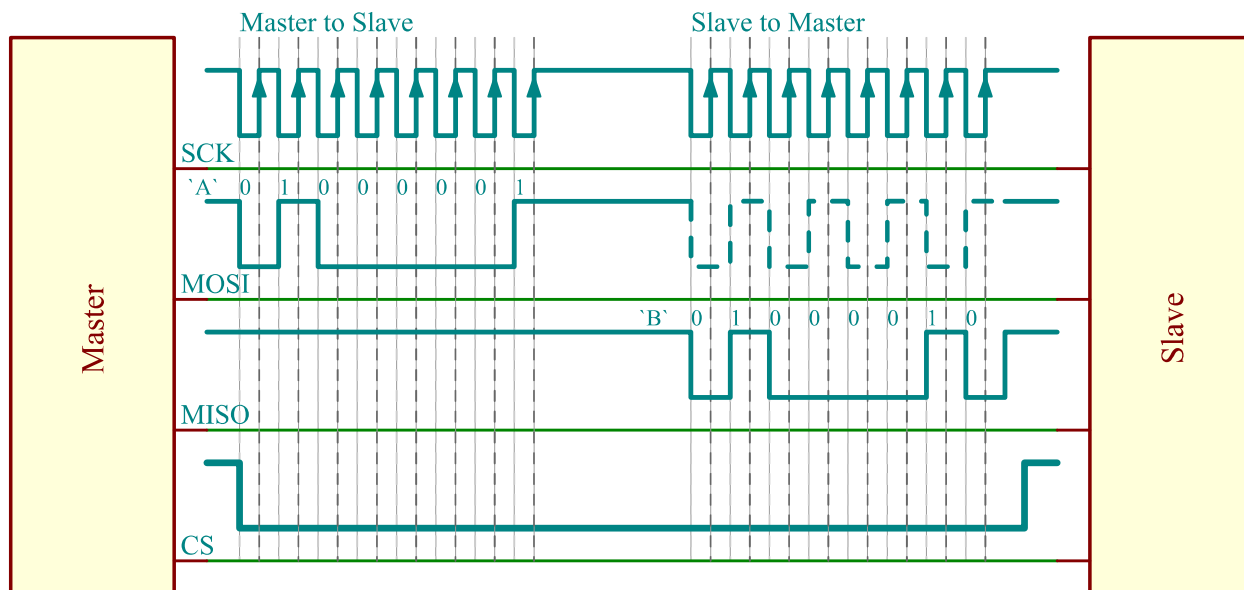
## Синхронные и асинхронные интерфейсы

Примеры которые мы привели выше были синхронными, они так называются потому что используют отдельный вывод, который сообщает принимающей стороне когда считывать данные. К ним относится SPI и USART. Однако, USART (с англ. Universal Synchronous/Asynchronous Receiver/Transmitter) как не сложно догадаться может работать в асинхронном режиме. Если время передачи бита известно, то сообщать когда именно нужно считывать бит нет необходимости. Однако нужно как-то различать между собой принятые байты. С этой целью в последовательность высоких и низких уровней на шине внедряют задержки (старт и стоп-биты). Они отличаются по длительности от обычных 0 и 1, благодаря чему всегда можно определить где начало, а где конец посылки. Асинхронная версия USART называется UART.



Кодировка битов может быть осуществлена не только как 1 — высокий уровень и 0 — низкий уровень. В интерфейсе 1-Wire 1 и 0 кодируется одновременно и низким и высоким уровнем, отличается лишь их заполнение временного отрезка. Вы увидите это дальше в курсе, когда мы будем разбираться с датчиком температуры.

В интерфейсе SPI напротив, считывание сигнала происходит только по команде тактирующей линии.

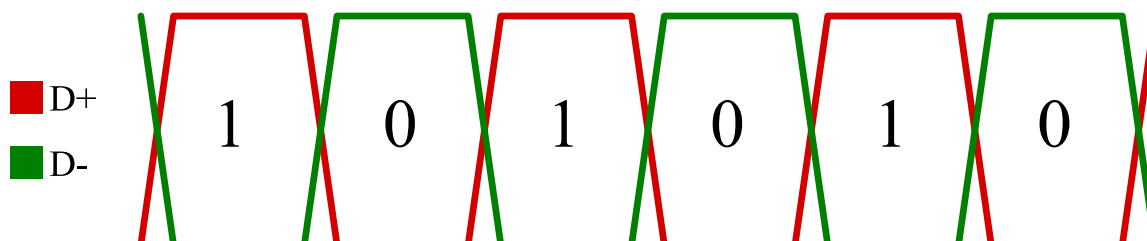


У вас наверняка возникает философский вопрос, — «Зачем есть капусту, когда есть картошка?» Может показаться, что линия с тактовым сигналом излишня. Да. Но это не всегда так. Линия с тактовым сигналом делает интерфейс независимым от стабильности тактирующих систем узлов. Если погрешность будет слишком велика, то у устройств с асинхронным интерфейсом может произойти рассогласование. В случае с синхронным, тактовый сигнал задаётся передающим устройством (не совсем корректно, но пока пусть будет так), поэтому после первого бита линию можно повесить, условно, на полчаса без каких либо последствий и продолжить передачу после этого.

## Дифференциальные интерфейсы

USB является дифференциальным последовательным интерфейсом. При этом он довольно требователен к стабильности частоты и именно по этой причине в нашем устройстве не реализована на физическом уровне возможность подключения часов к компьютеру; требуется внешний высокочастотный кварцевый резонатор <sup>1</sup>.

Дифференциальным он называется потому, что сигнал передаётся сразу по двум линиям, причём если на одной линии выставлена <sup>1</sup>, то на второй должен быть выставлен <sup>0</sup> и наоборот.



Опять же вопрос, зачем два провода, когда можно использовать один? UART вроде хорошо работал на одной линии... На самом деле нет. UART хорошо работает на низких скоростях. При повышении частоты передачи начинают влиять паразитные ёмкости и индуктивности и сигнал просто теряется. Дифф-пара позволяет понизить напряжение, уменьшить электро-магнитное излучение, уменьшить потребление и

повысить устойчивость приёма сигнала.

Выбирать дифференциальную пару, конечно, нужно не только в тех случаях, где требуется высокая скорость передачи, но и там, где «шумная» среда. В электронике автомобиля используется другой интерфейс, под названием CAN. Скорость передачи данных там не высокая, за-то она обеспечивает хороший приём в условиях жёсткой эксплуатации.

К слову, проводной интернет (Ethernet) так же построен на дифференциальной паре.

## Режим работы

Некоторые интерфейсы могут поддерживать несколько режимов работы, другие напротив способны работать только в одном определённом.

Если устройство использует два канала, один для приёма и второй для передачи, и при этом обмен данными может происходить одновременно, то такой режим работы называют полным дуплексом (англ. full duplex). Ваш мобильный телефон работает в таком режиме: вы можете слушать человека и говорить ему что-либо одновременно. Если используется два канала, но передача и приём может производиться только одним устройством в один момент времени, то такой режим работы называется полудуплексным (англ. half-duplex). И наконец, если используется один канал, следовательно общение может проходить только разделённое во времени, то такой режим называется симплексным (англ. simplex).

## Свойство шины

Каждый интерфейс предъявляет свои требования к шине данных. В таких интерфейсах как UART может быть только два устройства (принимающее и отправляющее). При этом с точки зрения иерархии нет никакой разницы, оба устройства равнозначны. В интерфейсе SPI устройств может быть несколько, но только одно (на самом деле не всегда, но это исключение) является главным, т.е. ведущим или мастером (англ. master). Все остальные устройства являются ведомыми или подчинёнными (англ. slave). При этом SPI требует подводить к каждому устройству линию выбора (англ. chip select). Общение ведётся только с тем устройством, на линии которого присутствует сигнал активации.

Все ножки вышеописанных интерфейсов настраиваются как двухтактный выход. Но не все интерфейсы можно использовать подобным образом. Интерфейсы 1-Wire и I<sup>2</sup>C вместо линии выбора используют специальную команду с адресом устройства. Выгода очевидна — на линию можно повесить сколько угодно (есть ограничения) устройств, не вводя новые линии. Однако что будет, если два устройства захотят использовать шину одновременно? Непременно сложится ситуация, когда одно устройство подтянет линию к земле, а другое к питанию. Такое поведение, мягко говоря, не желательно и называется коротким замыканием (англ. short circuit).

То самое чувство, когда микросхема испустила дух, выпустила магический дым.

Во избежание таких ситуаций, интерфейс требует, что бы: линия была подтянута внешним резистором к питанию, а устройства были настроены как вход с открытым стоком. Когда устройству нужно передать **1** оно ничего не делает с линией, а когда нужно передать **0**, оно подтягивает линию к земле. В таком случае, если произойдёт коллизия и два устройства одновременно начнут работать с линией, магический дым останется внутри микросхем.

## Сравнение некоторых интерфейсов

Название	Количество линий, шт	Длина линии, м	Скорость, бит\с
<a href="#">1-Wire</a>	Simplex	до 300 м	15,4 Кбит/с, максимум 125 Кбит/с
<a href="#">SPI</a>	Duplex	до 5 м	> 100 МГц
I <sup>2</sup> C	Simplex	3 м	425 КБ/с
<a href="#">UART</a>	Duplex	5 м	> 11 520 байт/с
CAN	Simplex	до 5000	10 кбит/с - 1 Мбит/с
USB 2.0	Duplex	5	60 МБ/с

Все данные в таблице — ориентировочные, многое зависит от скорости передачи, напряжения, среды распространения и т.д.

---

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

---

1. Стандарт USB требует стабильность в 0.25% (2500 ppm), в то время как точность внутренней RC-цепочки (HSI) при 25°C лежит в пределах от -1.1% до +1.8% от номинала. См. Datasheet, Table 24. HSI oscillator characteristics [↗](#)