Protocolos de comunicação

Protocolo de comunicação é um conjunto de regras que permite que dois ou mais dispositivos comuniquem entre si, para tal os dispositivos têm de conhecer as regras do protocolo de comunicação. O protocolo de comunicação defini quantos dispositivos podem comunicar, se necessitam de identificadores únicos ou não, se são síncronos ou assíncronos, se são unidirecionais ou bidirecionais e se funcionam com um dispositivo que “manda”, *master*, nos outros dispositivos, funcionando os últimos como *slaves*. Exemplos de protocolos de comunicação são UART, I2C, SPI e CAN, sendo estes o que vamos dar mais relevância na disciplina de Sistemas Embebidos.

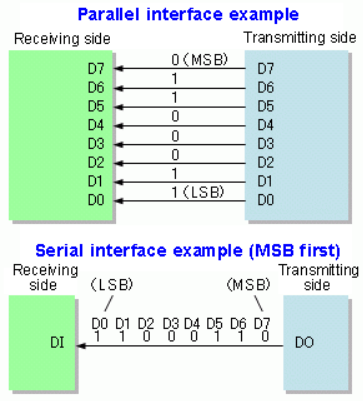
**Comunicação síncrona vs assíncrona**

**Comunicação síncrono** em regra geral é uma comunicação mais rápida, necessita de algum tipo de sincronismo (normalmente usa um *clock* para garantir que todos os dispositivos estão sincronizados).

**Comunicação assíncrona** todos os dispositivos podem comunicar quando necessitarem, não necessitam de um *clock* para comunicarem.

**Comunicação série vs paralela**

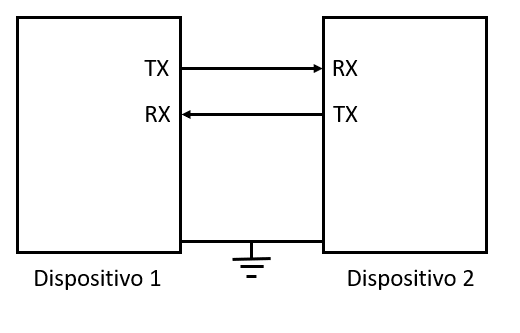
**Comunicação série** permite a trocas de dados de uma forma sequencial, ou seja, a informação flui *bit* a *bit* não necessitando de uma linha específica por *bit*. Ao contrário da **comunicação paralela** que necessita de uma linha específica para transferir a informação por *bit.*



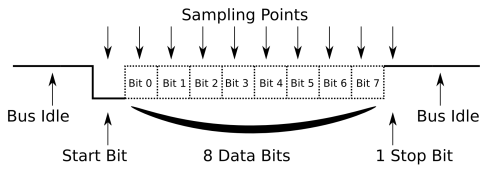
**Alguns protocolos de comunicação série:**

**UART**

*Universal Synchronous Asynchronous Receiver Transmitter,* UART, é um protocolo de comunicação do tipo série, assíncrono. A informação flui por duas linhas: TX, transferência de informação, e RX, receção de informação, permitindo assim a informação fluir nos dois sentidos em simultâneo, *full-duplex.* A velocidade de comunicação dos dispositivos é acordada entre ambos e é denominada por *baudrate.*

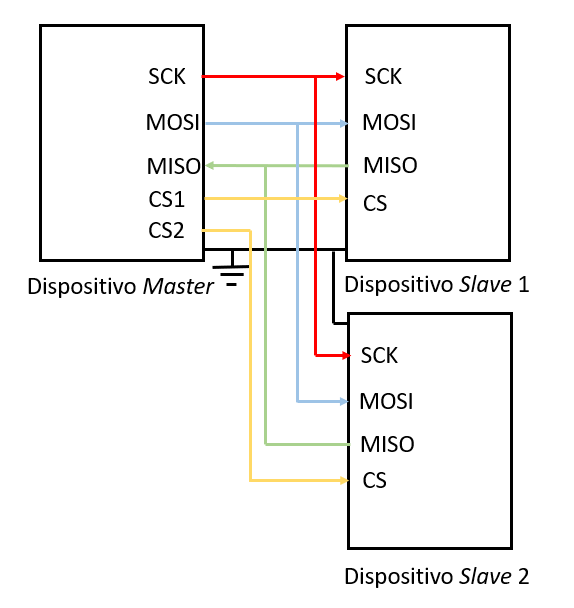


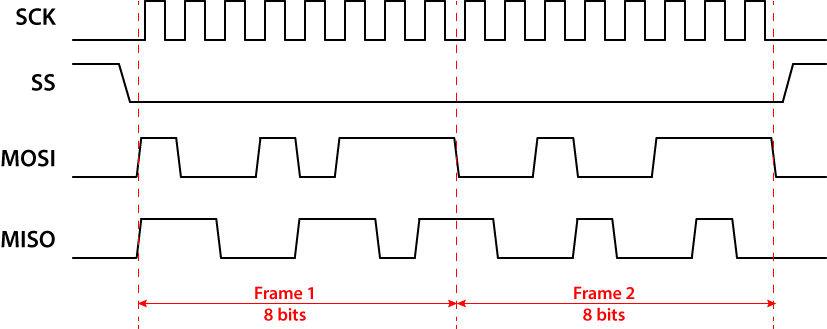
A trama de comunicação é constituída por: bit de começo, *start bit,* informação e *stop* bit.



**SPI**

*Serial Peripheral Interface*, SPI, é um protocolo de comunicação síncrono, ou seja, necessita de um sinal relógio para efetuar a comunicação. Possui um dispositivo que funcionará com *master* que é responsável por geral o *clock* que comandará os restantes dispositivos, *slaves*. Para a comunicação os dispositivos necessitam de quatro linhas: MOSI, *Master Output Slave Input*, MISO, *Master Input Slave Output*, o SCLK, *clock*, e o CS, *Chip Select,* que permite selecionar o dispositivo que o *master* pretende comunicar. Uma vez que usa duas linhas de comunicação, uma para transmitir e outra para receber informação, o SPI também é *full-duplex.*



**I2C**

*Inter-Integrated Circuit,* I2C, é um protocolo de comunicação síncrono que apenas necessita de duas linhas para o seu funcionamento, o SDA, *Serial Data,* por onde flui a informação e o SCL, *Serial Clock,* que permite sincronizar os diversos dispositivos. Como no SPI o responsável pela geração do sinal de relógio é o dispositivo *master*, no entanto os dispositivos *slaves* podem forçar o relógio ao nível lógico de zero para parem a comunicação por breves instantes, estando definida esta operação no protocolo como *clock stretching*. O I2C tens vantagens sobre o UART e o SPI, em comparação com o UART para a comunicação funcionar ambos os dispositivos necessitam de estar configurados previamente para uma determina frequência de comunicação, *baudrate,* o que faz com que ambos os dispositivos disponham de *clocks* próprios e não os partilhem entre si. Em comparação com o SPI o I2­C permite que no barramento existam mais *masters* ao contrário do SPI que apenas permite um. O I2C permite ainda conectar aos dispositivos por um endereço, ou seja, cada dispositivo *slave* possui um ID único de 7bits evitando assim a ligação física de um pino específico a cada dispositivo, como no caso do SPI.

Ao contrário do SPI e UART as linhas SDA e SCL têm resistência de *pull-up* o que faz com que os dispositivos apenas consigam “puxar” a linha ao sinal lógico de zero, permitindo assim evitar dissipação de energia desnecessária e dando visibilidade aos outros dispositivos no barramento que está a decorrer uma comunicação.

