

SPI (*Serial Peripheral Interface*)

Prof. Rodrigo A. Romano

Escola de Engenharia Mauá

Introdução

- O protocolo SPI, originalmente desenvolvido pela Motorola[®], descreve uma interface de comunicação serial síncrona, capaz de transmitir e enviar dados simultaneamente (*full-duplex*).
- O protocolo SPI é caracterizado pela capacidade de troca de dados em alta velocidade (até 70MHz) de forma simples e eficiente.
- A interface de comunicação é composta basicamente por 4 vias: SCK (sinal de *clock*), SDI (entrada de dados), SDO (saída de dados) e SS (seleção de escravo).
- Diversos microcontroladores disponíveis no mercado possuem esta interface que além da comunicação entre μ Cs, também é usada na comunicação com conversores A/D e D/A, LCDs, sensores, memórias, entre outros.

Conexão mestre-escravo

- Uma interface de comunicação SPI deve ter, obrigatoriamente, um dispositivo mestre responsável pela geração do sinal de sincronismo e um escravo.

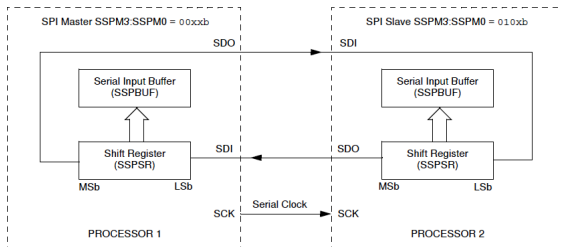


Figura: Conexão mestre-escravo

- Também é possível interligar uma série de dispositivos escravos em um barramento SPI. Nesse caso, o mestre é responsável pela ativação de um dentre os escravos interligados no barramento.

Funcionamento do protocolo

- No protocolo SPI os dispositivos não possuem endereço, portanto, para que um pino de seleção **SS** é usado para habilitar um determinado dispositivo escravo. A maior parte dos elementos com interface SPI são acionados em nível baixo, no entanto, não há uma padronização.
- A comunicação é iniciada pelo seleção do dispositivo escravo. Em seguida, o mestre gera o sinal de *clock*¹ para sincronismo, enquanto os bits de dados são sequencialmente trocados entre mestre e escravo.
- Encerrada a troca de informações a seleção do escravo é desfeita.

¹A polaridade e a fase do sinal de *clock* são configurados através dos bits CPOL e CPHA, respectivamente.

Funções da biblioteca spi.h

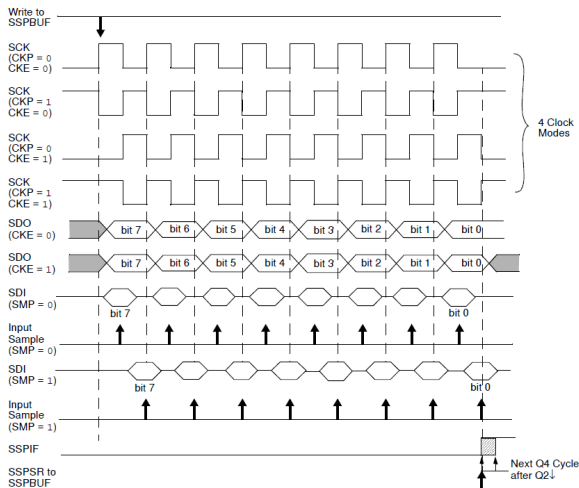
- Configuração da interface SPI

```
void OpenSPI(<modo de comunicação>,  
             <config. clock>,  
             <config. leitura>);
```

Opções de configuração:

- <modo de comunicação>: SPI_FOSC_4, SPI_FOSC_16, SPI_FOSC_64, SPI_FOSC_TMR2, SLV_SSON ou SLV_SSOFF
- <config. clock>: MODE_00, MODE_01, MODE_10 ou MODE_11
- <config. leitura>: SMPEND ou SMPMID
- Transmissão de um byte
char status = WriteSPI(**char** dado);
- Recepção de byte
char dado = ReadSPI(**void**);

Configurações da interface SPI (modo mestre)



| <i>config. clock</i> | CKP | CKE |
|----------------------|-----|-----|
| MODE_00 | 0 | 1 |
| MODE_01 | 0 | 0 |
| MODE_10 | 1 | 1 |
| MODE_11 | 1 | 0 |

Exercícios I

- 1 Desenvolver um firmware que envie os bytes 0x30 e 0xF0 através da interface SPI, no modo de *clock* 00 a cada acionamento de uma determinada tecla.
- 2 Utilize o osciloscópio para visualizar o comportamento da via de dados SDO (pino RC5) e de *clock* (pino RC3).
- 3 Alterar o firmware elaborado anteriormente para enviar uma palavra de 16 bits com “0011” nos 4bits mais significativos e, nos bits restantes, o valor de uma variável, cujo valor é limitado entre 0 e 4000 e pode ter seu valor alterado através das teclas RB2 e RB3.