

# Barramentos série: norma SPI

**REVERSE ENGINEERING**

**José Luís Azevedo, Bernardo Cunha**

**deti** universidade de aveiro  
departamento de eletrónica,  
telecomunicações e informática

# Barramentos série: norma SPI

## Resumo

- A interface SPI (*Serial Peripheral Interface*)
- Sinalização
- Sequência de operação
- Arquiteturas de ligação
- Tipos de transferências
- Passos de configuração de um *master* SPI
- Exemplo: Sensor Digital de Temperatura com interface SPI

# Barramentos série: norma SPI

## Introdução

- SPI – sigla para "Serial Peripheral Interface"
- Interface definida inicialmente pela Motorola (Microwire da National Semiconductor é um *subset* do protocolo SPI)
- O SPI é utilizado para comunicar com uma grande variedade de dispositivos:
  - Sensores de diverso tipo: temperatura, pressão, etc.
  - Cartões de memória (MMC / SD)
  - Circuitos: memórias, ADCs, DACs, Displays LCD (e.g. telemóveis), comunicação entre corpo de máquinas fotográficas e as lentes, ...
  - Comunicação entre microcontroladores
- Ligação a muito curtas distâncias

# Barramentos série: norma SPI

## Descrição geral

- Arquitetura "Master-Slave" com ligação ponto a ponto
- Comunicação bidirecional "full-duplex"
- Comunicação síncrona (relógio explícito do *master*)
  - Relógio é gerado pelo *master* que o disponibiliza para todos os *slaves*
  - Não é exigida precisão ao relógio - os bits vão sendo transferidos a cada transição de relógio. Isto permite utilizar um oscilador de baixo custo no *master* (não é necessário um cristal de quartzo)
- Fácil de implementar por hardware ou por software
- Não são necessários "line drivers" (ou "transceivers") - circuitos de adaptação ao meio de transmissão

# Barramentos série: norma SPI

## Descrição geral

- Arquitetura "Master-Slave"
  - O sistema só pode ter um *master*
  - O *master* é o único dispositivo no sistema que pode controlar o relógio
- Um *master* pode estar ligado a vários *slaves*: para cada comunicação, apenas 1 *slave* é selecionado pelo *master*
- O *master* inicia e controla a transferência de dados

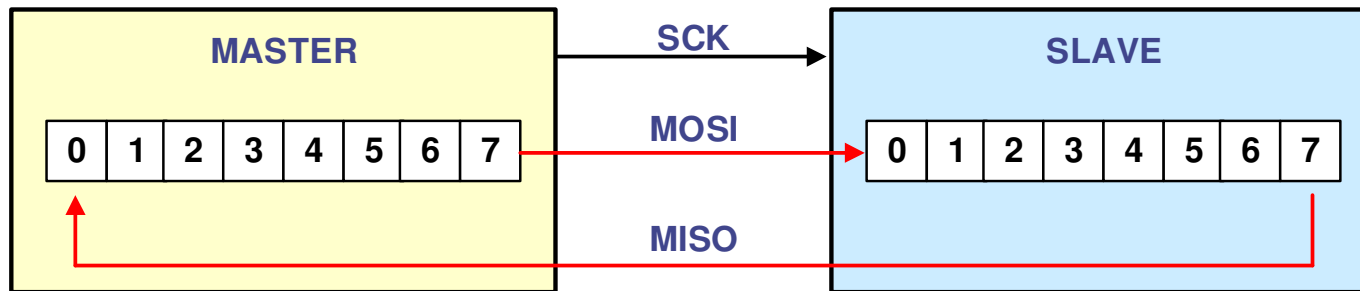
# Barramentos série: norma SPI

## Descrição geral

- Sinalização:
  - **SCK** – clock
    - Relógio gerado pelo *master* que sincroniza a transmissão/receção de dados
  - **MOSI** – Master Output Slave Input (SDO no *master*)
    - Linha do *master* para envio de dados para o *slave*
  - **MISO** – Master Input Slave Output (SDI no *master*)
    - Linha do *slave* para enviar dados para o *master*
  - **SS** – Slave select
    - Linha do *master* que seleciona o *slave* com quem vai comunicar

# Barramentos série: norma SPI

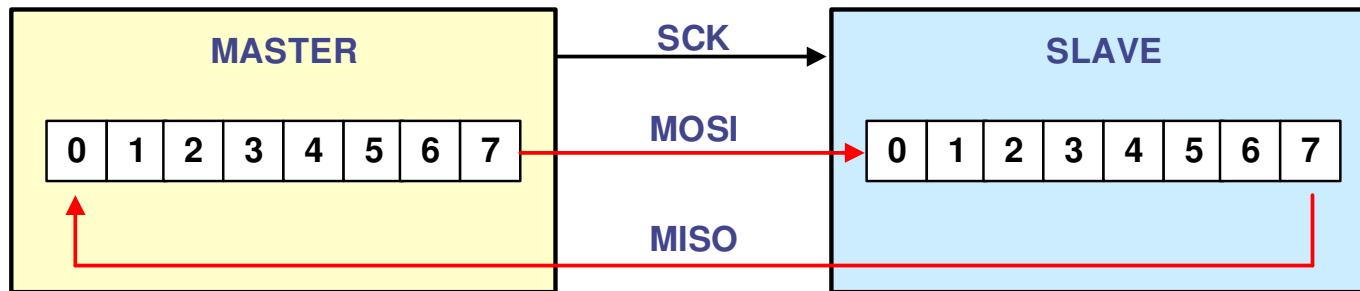
## Descrição geral – esquema de princípio



- Transmissão "full-duplex" baseada em dois *shift-registers* (um no *master* e outro no *slave*)
- Em cada ciclo de relógio:
  - O *master* coloca 1 bit na linha MOSI e o *slave* recebe-o
  - O *slave* coloca 1 bit na linha MISO e o *master* recebe-o

# Barramentos série: norma SPI

## Descrição geral – esquema de princípio

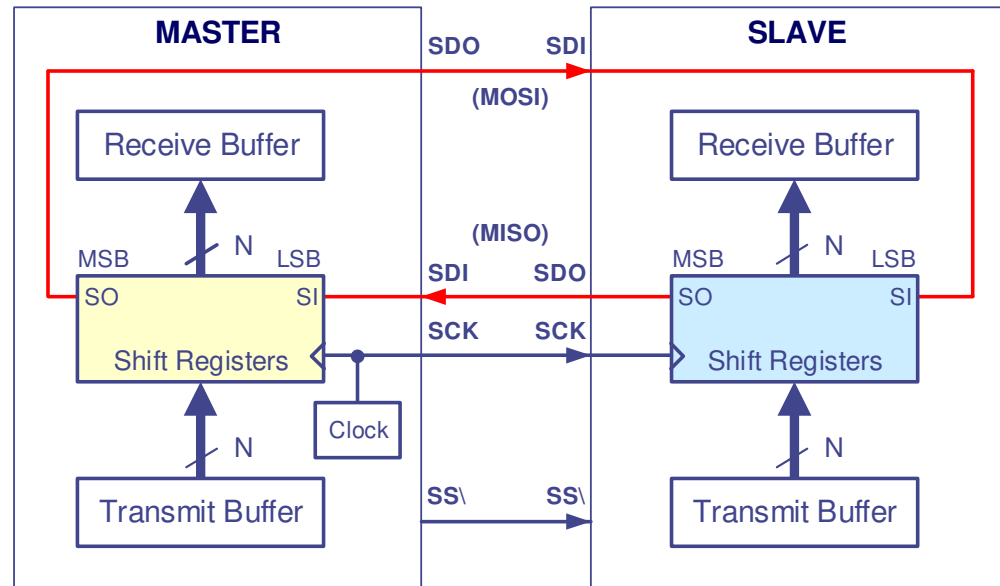


- Ao fim de N ciclos de relógio o *master* enviou uma palavra de N bits e recebeu do *slave* uma palavra com a mesma dimensão – "Data Exchange"
- Esta sequência é realizada mesmo quando é pretendida uma comunicação unidirecional



# Barramentos série: norma SPI

## Sinalização

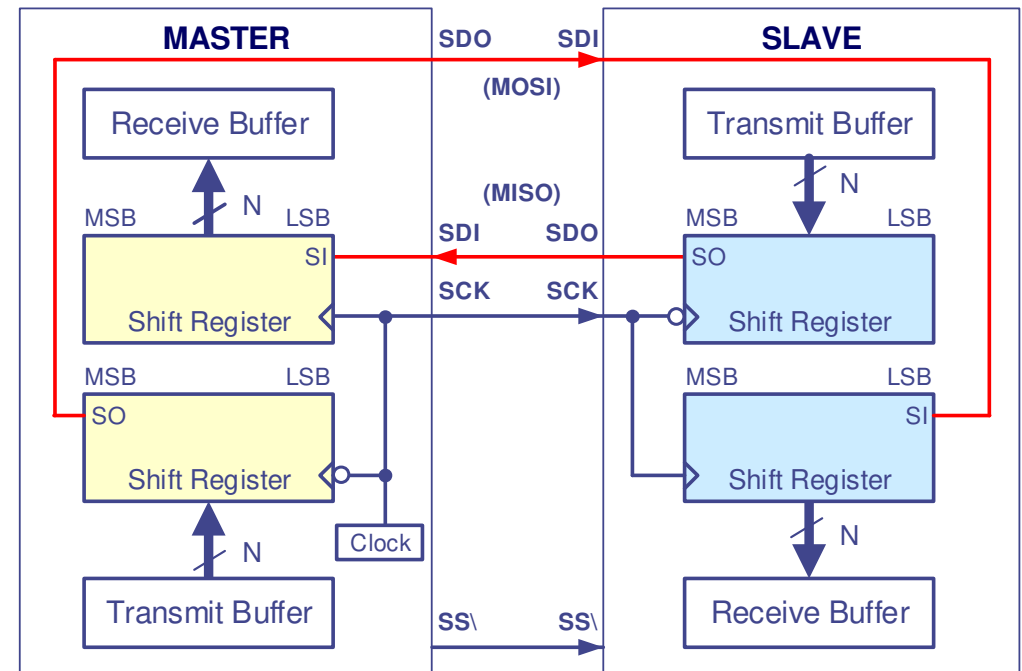


- Dados:
  - MOSI – *Master Output Slave Input* (SDO – serial data out no *master*)
  - MISO – *Master Input Slave Output* (SDI – serial data in no *master*)
- Controlo:
  - SS\ – *Slave select* (sinal ativado pelo *master* para selecionar o *slave* com quem vai comunicar)
  - SCK – *serial clock*

# Barramentos série: norma SPI

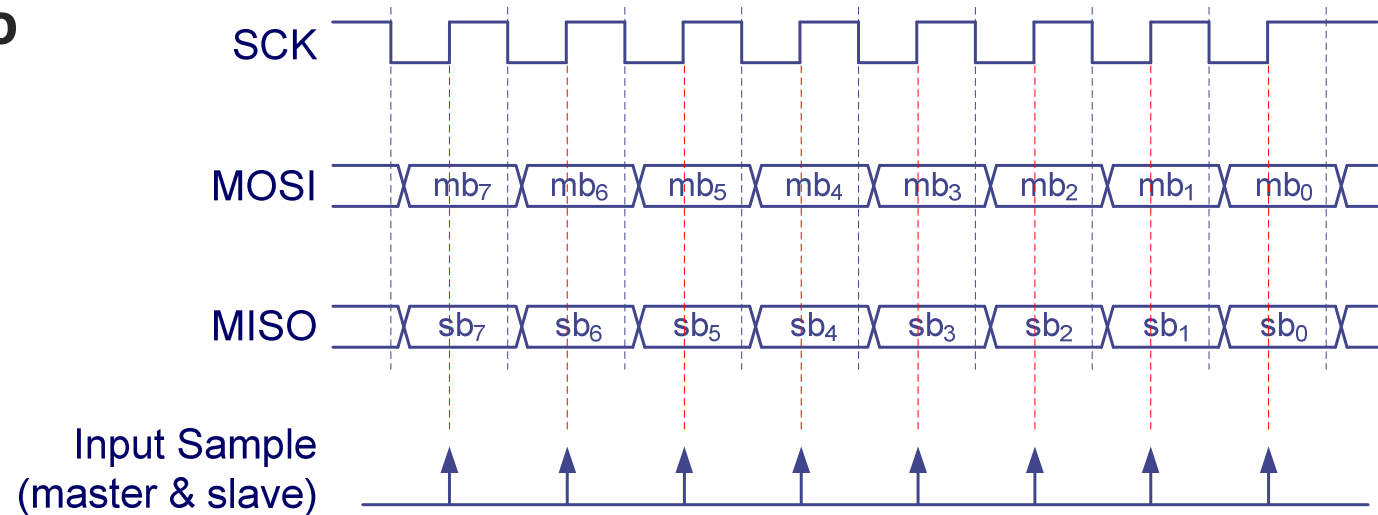
## Sinalização

- O sinal de relógio tem um "duty-cycle" de 50%
- No exemplo da figura:
  - *master* e *slave* usam a transição negativa do relógio para colocarem 1 bit na linha (*master* na linha MOSI, *slave* na linha MISO)
  - Na transição positiva seguinte, o *master* armazena o valor presente na linha MISO e o *slave* armazena o valor que se encontra na linha MOSI



# Barramentos série: norma SPI

## Operação – exemplo



- A transição negativa do relógio é usada pelo *master* e pelo *slave* para colocar na respetiva linha de saída um bit de informação
- A transição positiva seguinte é usada pelo *master* e pelo *slave* para armazenar o bit presente na respetiva linha de entrada
- Ao fim de oito ciclos de relógio:
  - o valor inicialmente armazenado no *shift-register* do *master* foi transferido para o *shift-register* do *slave*
  - o valor inicialmente armazenado no *shift-register* do *slave* foi transferido para o *shift-register* do *master*

# Barramentos série: norma SPI

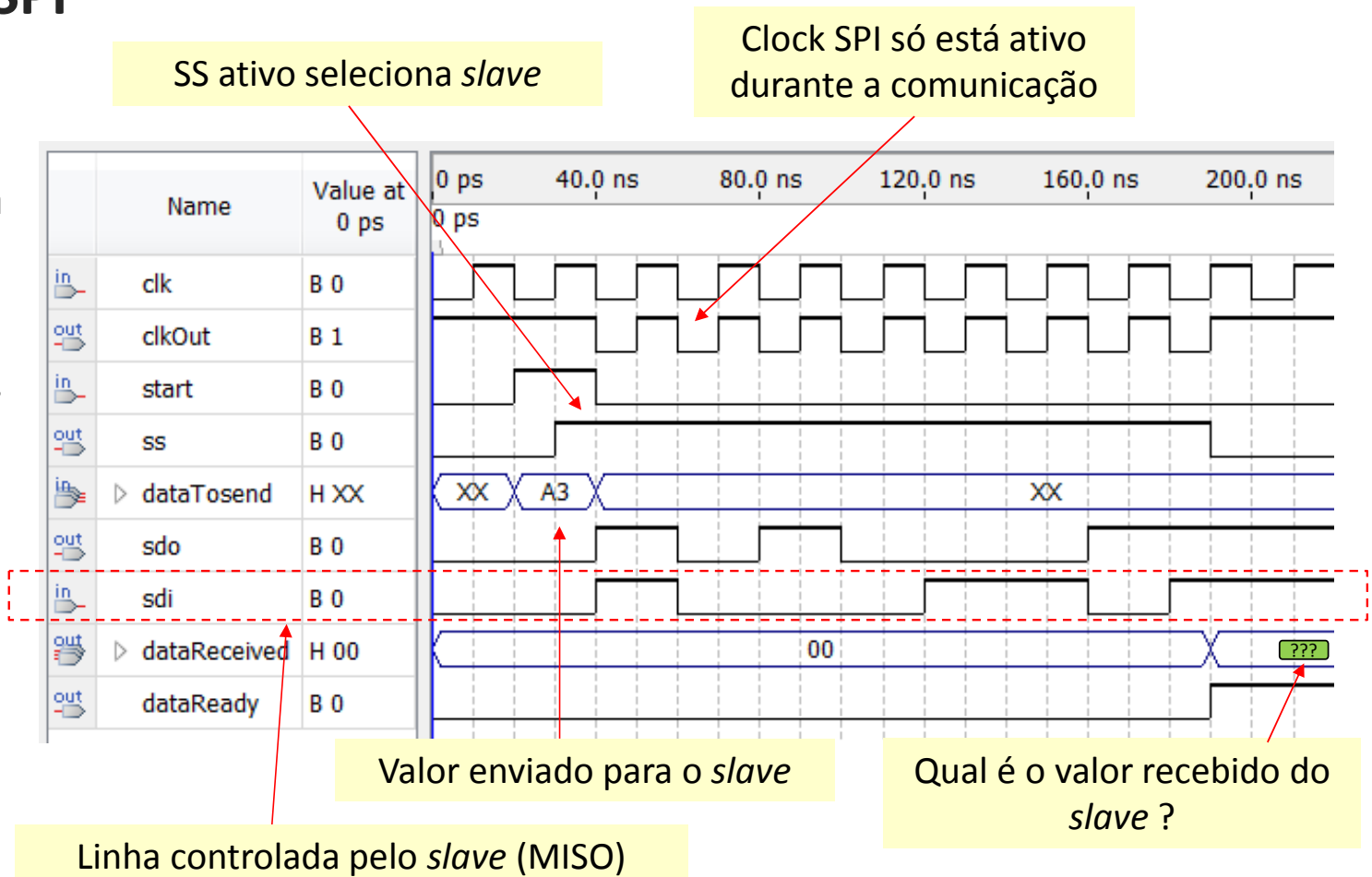
## Operação

- O *master* configura o relógio para uma frequência igual ou inferior à suportada pelo *slave* com quem vai comunicar
- O *master* ativa a linha SS\ do *slave* com que vai comunicar
- Em cada ciclo do relógio, por exemplo na transição positiva
  - O *master* coloca na linha MOSI um bit de informação que é lido pelo *slave* na transição de relógio oposta seguinte
  - O *slave* coloca na linha MISO um bit de informação que é lido pelo *master* na transição de relógio oposta seguinte
- O *master* desativa a linha SS\ e desativa o relógio (que fica estável, por exemplo, no nível lógico 1)
  - Só há relógio durante o tempo em que se processa a transferência
- No final, o *master* e o *slave* trocaram o conteúdo dos seus *shift-registers*

# Barramentos série: norma SPI

## Simulação do *master* SPI

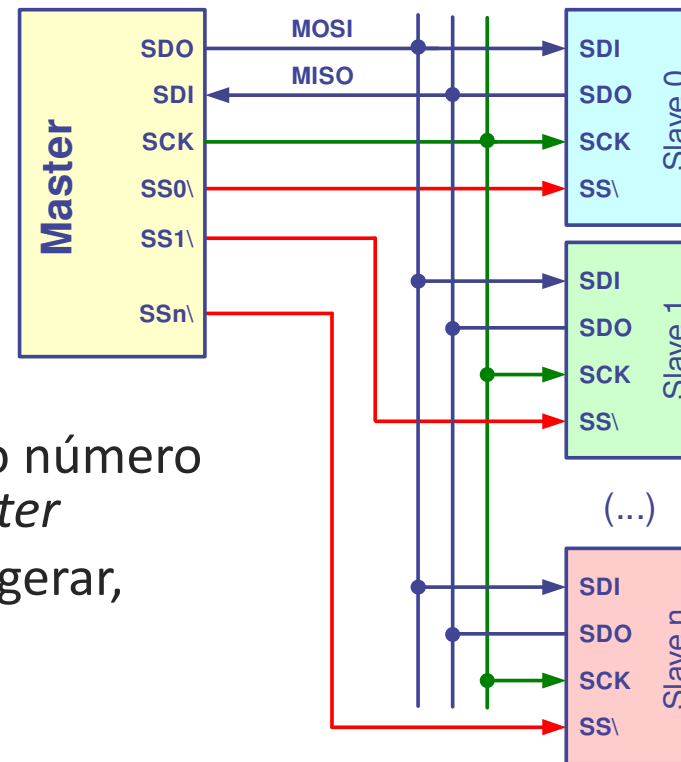
- Normalmente, o master pode ser configurado para usar, em alternativa, o nível lógico '1' ou o nível lógico '0' como nível inativo do sinal de relógio.
- Dessa forma é possível adaptar o controlador a características específicas dos módulos slave.



# Barramentos série: norma SPI

## Arquiteturas de ligação – *slaves* independentes

- Sinais de seleção ("slave select") independentes
- Em cada instante apenas um  $SSx\backslash$  está ativo, isto é, apenas 1 *slave* está selecionado
- Os sinais SDO dos *slaves* (MISO) não selecionados estão em alta impedância
- O número máximo de *slaves* está limitado pelo número de linhas de seleção disponibilizadas pelo *master*
- Alternativamente, o microcontrolador poderá gerar, através de portos digitais, todos os sinais  $SSx\backslash$  necessários para comunicar com os *slaves*, ultrapassando a limitação anterior



# Barramentos série: norma SPI

## Tipos de transferências

- O SPI funciona sempre em modo "data exchange", isto é, o processo de comunicação envolve sempre a troca do conteúdo dos *shift-registers* do *master* e do *slave*
- Cabe aos dispositivos envolvidos na comunicação usar ou descartar a informação recebida
- Podem considerar-se os seguintes cenários de transferência:
  - **Bidirecional**: são transferidos dados válidos em ambos os sentidos (master → slave e slave → master)
  - **Master → slave (operação de escrita)**: *master* transfere dados para o *slave*, e ignora/descarta os dados recebidos
  - **Slave → master (operação de leitura)**: *master* pretende ler dados do *slave*; para isso transfere para o *slave* uma palavra com informação irrelevante (por exemplo 0); o *slave* ignora/descarta os dados recebidos

# Barramentos série: norma SPI

## Configuração de um *master* SPI

- Antes de iniciar a transferência há algumas configurações que são efetuadas no *master* para adequar os parâmetros que definem a comunicação às características do *slave* com que vai comunicar:
  1. Configurar a frequência de relógio
  2. Especificar qual o flanco do relógio usado para a transmissão (a recepção é efetuada no flanco oposto). Esta configuração é feita em função das características do *slave* com o qual o *master* vai comunicar:
    - Transmissão no flanco ascendente (consequentemente, a recepção é feita no flanco descendente)
    - Transmissão no flanco descendente (consequentemente, a recepção é feita no flanco ascendente)



# Barramentos série: norma SPI

## Procedimento para identificação dos sinais

- Caso não exista informação prévia sobre quais são os sinais que estamos a observar:
  1. Identificar o sinal de seleção (CS, CE...). Deve ser o que se encontra estável quando simultaneamente os outros sinais alternam o seu valor. Identificar se é ativo a '1' (lógica positiva) ou '0' (lógica negativa)
  2. Identificar o sinal de relógio e verificar se o estado não ativo (quando não há comunicação) é '0' ou '1'
  3. Identificar o sinal MOSI. Tipicamente o master envia a primeira palavra enquanto o slave devolve uma palavra sem significado (todos os bits a '0' ou a '1' ou sinal em alta impedância)
  4. O sinal que sobra deverá ser o MISO.
  5. Medir, no sinal de relógio, qual o seu período e calcular a respetiva frequência (taxa de comunicação)
  6. Identificar os valores de cada byte transferido em ambos os sentidos e interpretar o seu significado de acordo com o manual do dispositivo.

## Barramentos série: norma SPI

### Exemplo real: Sensor Digital de Temperatura com interface SPI

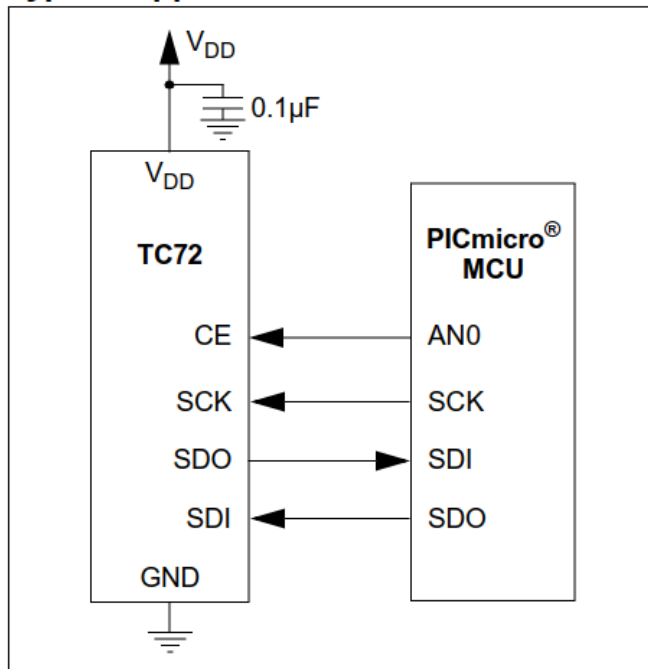
- No exemplo dos slides seguintes podemos observar algumas características funcionais da ligação entre um micro-controlador Master e um dispositivo TC72 da Microchip

<https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>

# Barramentos série: norma SPI

## Exemplo real: Sensor Digital de Temperatura com interface SPI

Typical Application



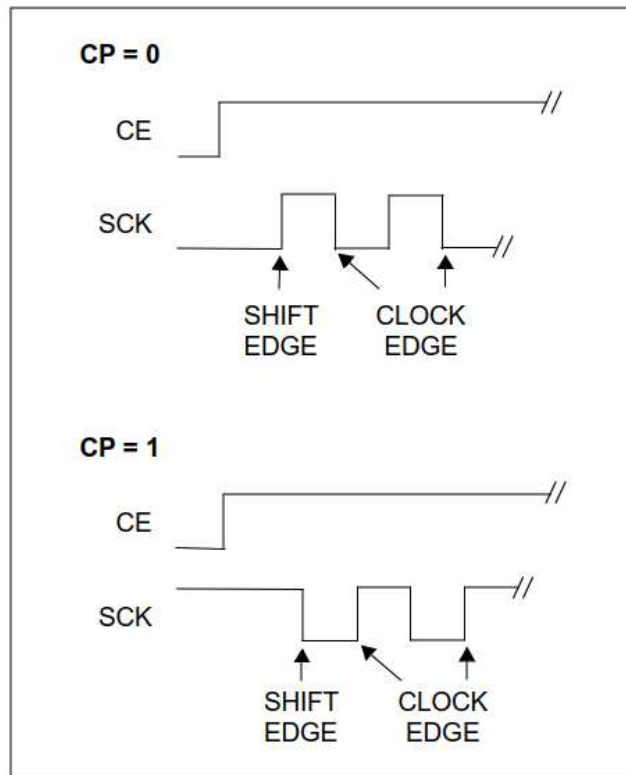
Main characteristics:

- Temperature-to-Digital Converter
- SPI compatible interface
- 10-Bit Resolution (0.25°C/Bit)
- ±2°C (max.) Accuracy from -40°C to +85°C
- ±3°C (max.) Accuracy from -55°C to +125°C
- 2.65V to 5.5V Operating Range
- Low Power Consumption:
  - 250 µA (typ.) Continuous Temperature Conversion Mode
  - 1 µA (max.) Shutdown Mode

from <https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>

# Barramentos série: norma SPI

## Exemplo real: Sensor Digital de Temperatura com interface SPI



**FIGURE 3-2:** Serial Clock Polarity (CP) Operation.

Deteção automática do estado não ativo do relógio.  
Esta deteção é feita na transição ascendente do sinal CE

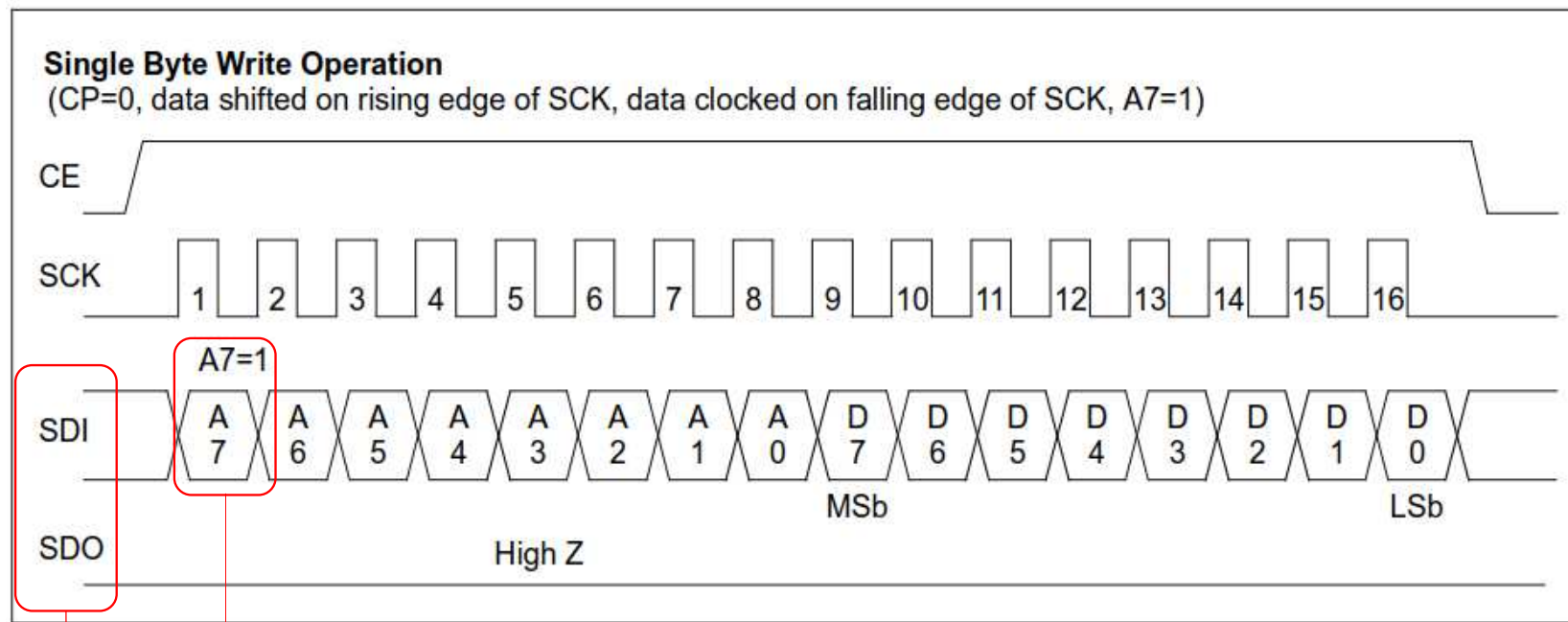
Se o estado não ativo do relógio for 0 (i.e., CP=0), a transição ascendente do relógio é usada para fazer o shift-out do próximo bit. A transição descendente é usada para validar e armazenar o próximo bit.

Se o estado não ativo do relógio for 1 (i.e., CP=1), a operação é simétrica. A transição descendente do relógio é usada para fazer o shift-out do próximo bit. A transição ascendente é usada para validar e armazenar o próximo bit.

from <https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>

# Barramentos série: norma SPI

## Exemplo real: Sensor Digital de Temperatura com interface SPI



Frequência  
**máxima** do  
relógio:  
≈7.5MHz

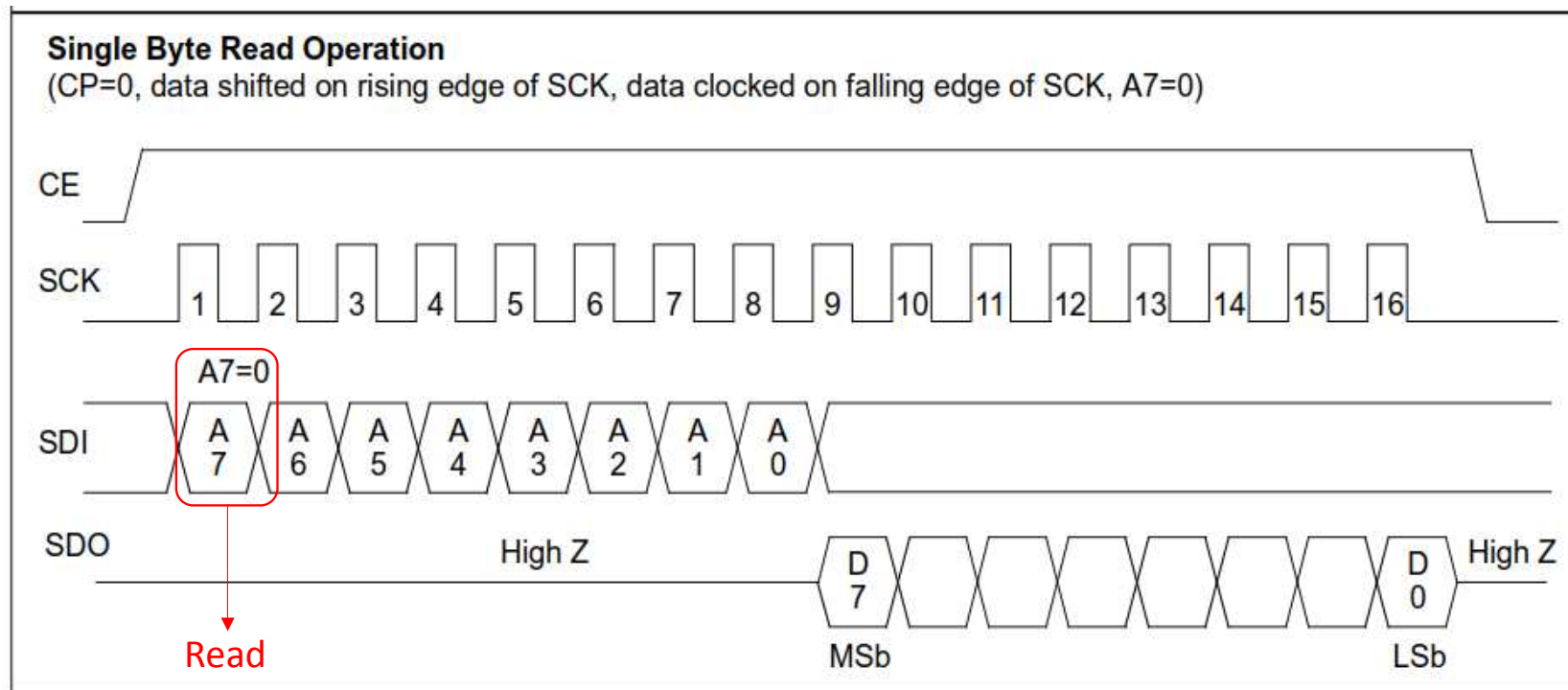
Lado do slave

Write

from <https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>

# Barramentos série: norma SPI

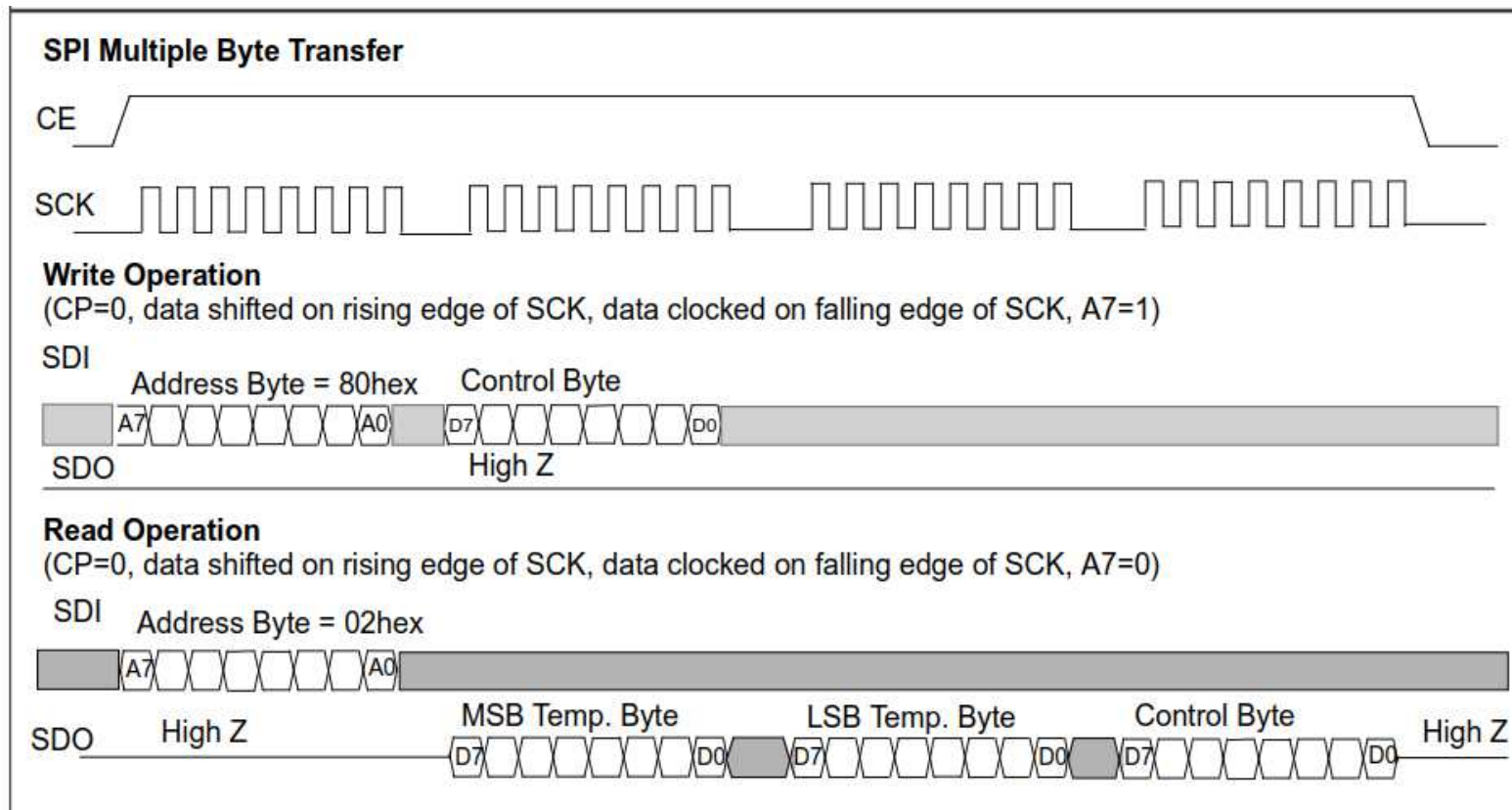
## Exemplo real: Sensor Digital de Temperatura com interface SPI



from <https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>

# Barramentos série: norma SPI

## Exemplo real: Sensor Digital de Temperatura com interface SPI



**FIGURE 3-3:** Serial Interface Timing Diagrams (CP=0).

# Barramentos série: norma SPI

## Exemplo real: Sensor Digital de Temperatura com interface SPI

**TABLE 4-1: REGISTERS FOR TC72**

Register	Read Address	Write Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR
Control	00hex	80hex	0	0	0	One-Shot (OS)	0	1	0	Shutdown (SHDN)	05hex
LSB Temperature	01hex	N/A	T1	T0	0	0	0	0	0	0	00hex
MSB Temperature	02hex	N/A	T9	T8	T7	T6	T5	T4	T3	T2	00hex
Manufacturer ID	03hex	N/A	0	1	0	1	0	1	0	0	54hex

**TABLE 4-2: CONTROL REGISTER TEMPERATURE CONVERSION MODE SELECTION**

Operational Mode	One-Shot (OS) Bit 4	Shutdown (SHDN) Bit 0
Continuous Temperature Conversion	0	0
Shutdown	0	1
Continuous Temperature Conversion (One-Shot Command is ignored if SHDN = '0')	1	0
One-Shot	1	1

Tempo  
aproximado  
de conversão:  
≈150ms

from <https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>



# Barramentos série: norma SPI

## Exemplo real: Sensor Digital de Temperatura com interface SPI

**TABLE 3-1: TC72 TEMPERATURE OUTPUT DATA**

Temperature	Binary MSB / LSB	Hex
+125°C	0111 1101/0000 0000	7D00
+25°C	0001 1001/0000 0000	1900
+0.5°C	0000 0000/1000 0000	0080
+0.25°C	0000 0000/0100 0000	0040
0°C	0000 0000/0000 0000	0000
-0.25°C	1111 1111/1100 0000	FFC0
-25°C	1110 0111/0000 0000	E700
-55°C	1100 1001/0000 0000	C900

**TABLE 3-2: TEMPERATURE REGISTER**

D7	D6	D5	D4	D3	D2	D1	D0	Address/ Register
Sign	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>3</sup>	2 <sup>1</sup>	2 <sup>0</sup>	02H Temp. MSB
2 <sup>-1</sup>	2 <sup>-2</sup>	0	0	0	0	0	0	01H Temp. LSB

Resultado da conversão é apresentado em complemento para 2 (com duas casas fracionárias)

from <https://ww1.microchip.com/downloads/en/DeviceDoc/21743A.pdf>