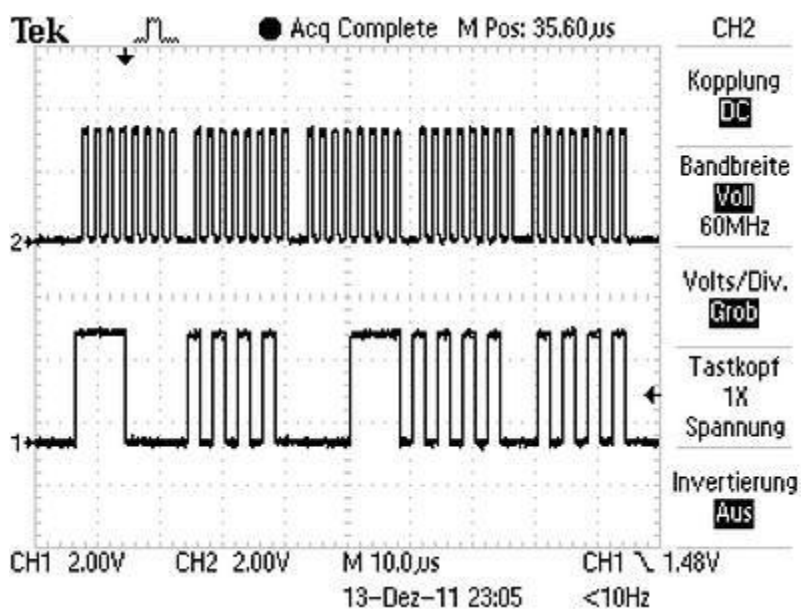


Comunicação SPI – Parte 1

Por **Francesco Sacco** - 05/05/2014



SPI

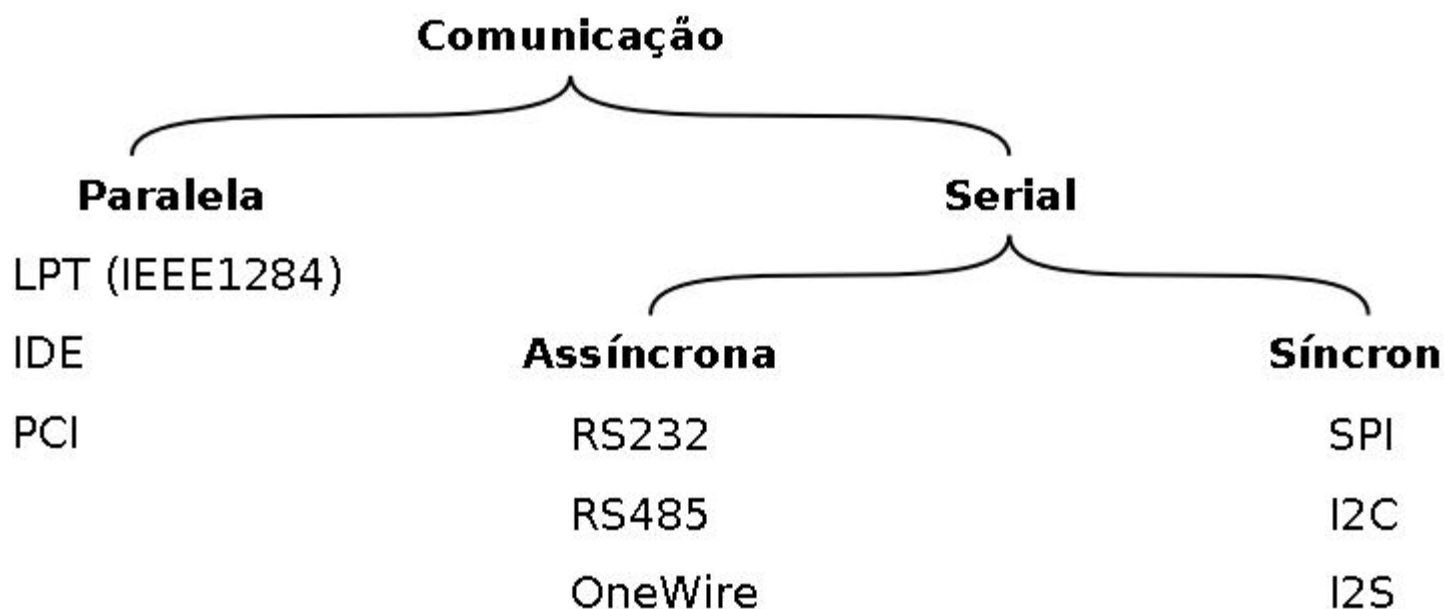
O objetivo deste artigo é definir os princípios da comunicação SPI, quais são suas vantagens e onde esses dispositivos são utilizados. Também vamos analisar os fundamentos básicos de hardware.

Tipos de Comunicação Serial

Uma vez que os equipamentos eletrônicos vêm agregando mais e mais funções, a utilização de diversos circuitos integrados torna-se cada vez mais comum. No entanto, não é mais possível estender longos barramentos de comunicação paralelos, pois tornariam as placas de circuito impresso caras e muito grandes. Logo, uma comunicação serial entre esses dispositivos se torna necessária.

Diversas tecnologias de interligação serial entre dispositivos foram desenvolvidas, podendo ser separadas em duas grandes categorias, a comunicação síncrona e a comunicação assíncrona. Dentre os métodos de comunicações mais conhecidos, destacam-se três:

- UART: Universal Asynchronous Receiver Transmitter;
- SPI: Serial Peripheral Interface;
- I2C: Inter Integrated Circuit.



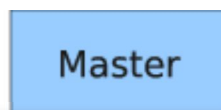
Já estamos bastante familiarizados com a comunicação serial assíncrona pelo uso do padrão UART (RS232). No entanto, existem grandes vantagens no uso de uma comunicação serial síncrona em detrimento da assíncrona.

de cada protocolo definir limites máximos de taxas, cada fabricante possui a liberdade de desenvolver dispositivos com suas velocidades. Abaixo temos um comparativo entre diversos padrões dispositivos seriais:

Tecnologia	Barramento de comunicação	Taxa máxima	Fluxo de dados
UART (RS232)	2 (sem controle de fluxo)	115.200 bps	Half ou Full Duplex
SPI	3 + n° de Slaves	2 Mbps	Full Duplex
I2C	2 (até 127 dispositivos)	400 Kbps	Half Duplex

Como um exemplo de que esses limites podem ser ultrapassados, temos AT45BD0100D c [Adesto](#). Essa memória SPI pode chegar até 66MHz (66Mbps) de taxa de transferência. Para o I2C, o 24FC64 da [Microchip](#) pode chegar a 1MHz (1Mbps).

Na comunicação serial síncrona definimos também o conceito de Mestre-Escravo. Normalmente o gerador do sinal de sincronismo é definido como o Mestre (Master) da comunicação. Para os dispositivos que utilizam do sinal de sincronismo gerado damos a definição de Escravo (Slave). A ligação mais comum desse tipo de comunicação é um Master e vários Slaves.

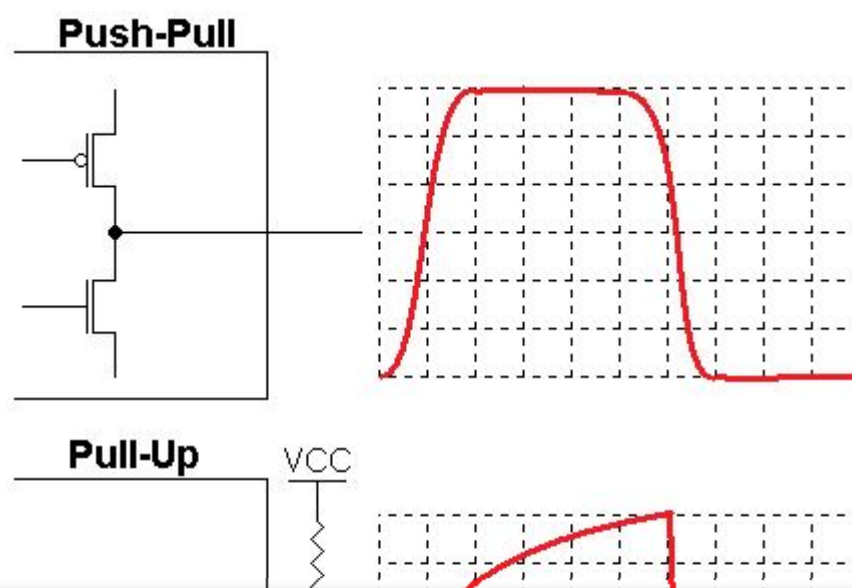


Gostou? [Junte-se à comunidade Embarcados](#)

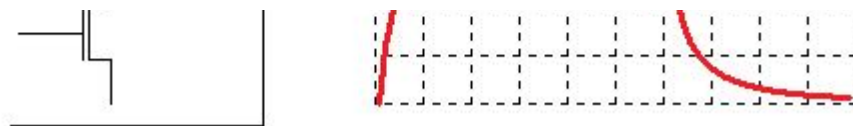


Comunicação SPI

A comunicação SPI possui algumas características básicas. Primeiramente os sinais de comunicação possuem uma direção fixa e definida. Isso significa que sempre existem dois transistores definindo o estado de um pino (Push-Pull). Essa característica é uma das grandes diferenças entre outras comunicações seriais como I2C e OneWire, que possuem um mesmo barramento de dados para os sinais de entrada e saída através do esquema de dreno-aberto (Pull-Up).



Gostou? [Junte-se à comunidade Embarcados](#)



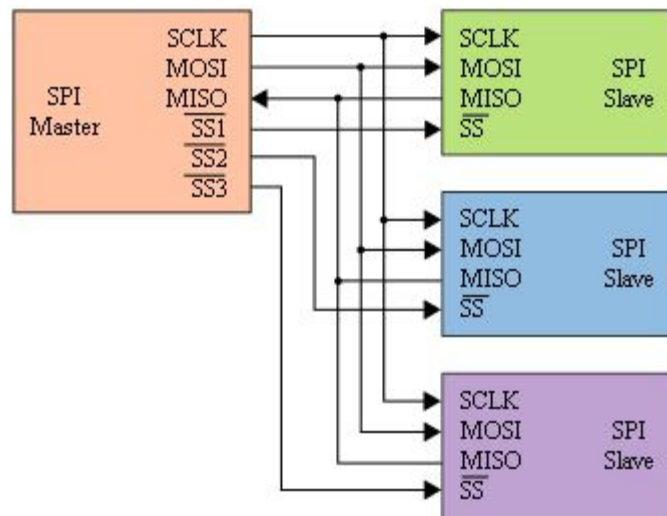
Apesar de utilizar dois sinais de comunicação de dados em vez de um, é possível atingir velocidades maiores de comunicação. Isso porque há pouca deformação do sinal.

Outra característica é que toda troca de dados acontece sempre em ambas as direções. E outras palavras, cada bit trocado entre o Master e um Slave trás um bit do Slave para o Master. Dessa forma, definimos que a comunicação é sempre full-duplex.

Os pinos básicos de comunicação entre dispositivos SPI e o esquema padrão de ligação são dados conforme abaixo:

Pino	Nome Padrão	Significado	Nomes Alternativos
Do Master para o Slave	MOSI	Master Output Slave Input	SDO, DO, SO
Do Slave para o Master	MISO	Master Input Slave Output	SDI, DI, SI
Clock	SCLK	Serial Clock	SCK, CLK
Seleção de Slave	SS	Slave Select	CS, nSS, nCS

Gostou? [Junte-se à comunidade Embarcados](#)

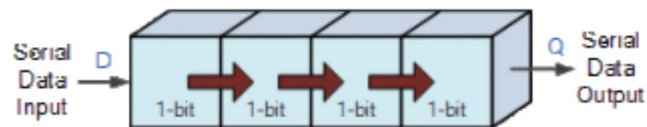


O sinal de SS funciona como Seleção de Escravo (Slave Select). É um sinal ativo em nível baixo, o que significa que o dispositivo é selecionado quando este pino se encontra em nível baixo. No entanto, muitos dispositivos utilizam este sinal como sincronismo de frame. Dessa forma, é um sinal importante que deve ser respeitado.

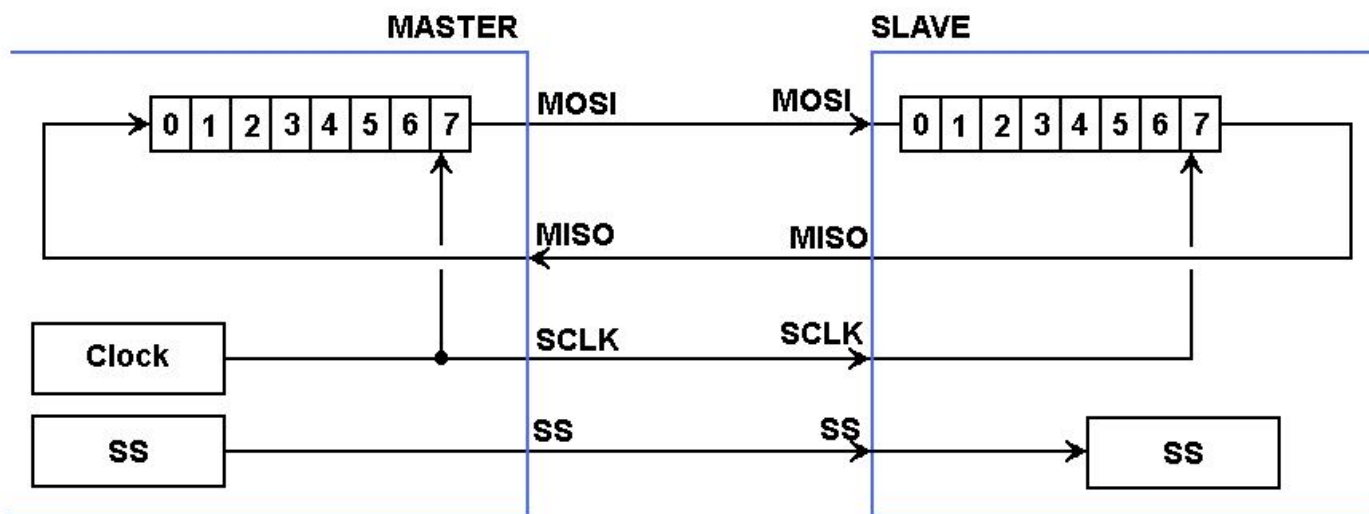
O Dispositivo

O princípio básico de um dispositivo SPI é o Shift-Register. Esse tipo de dispositivo faz a conversão de um registrador paralelo para sinais seriais de acordo com clock. Cada borda recebida no terminal de clock do dispositivo significa um bit transferido. Da mesma forma esse tipo de dispositivo é capaz de receber dados vindos de maneira serial e convertê-los

Gostou? [Junte-se à comunidade Embarcados](#)



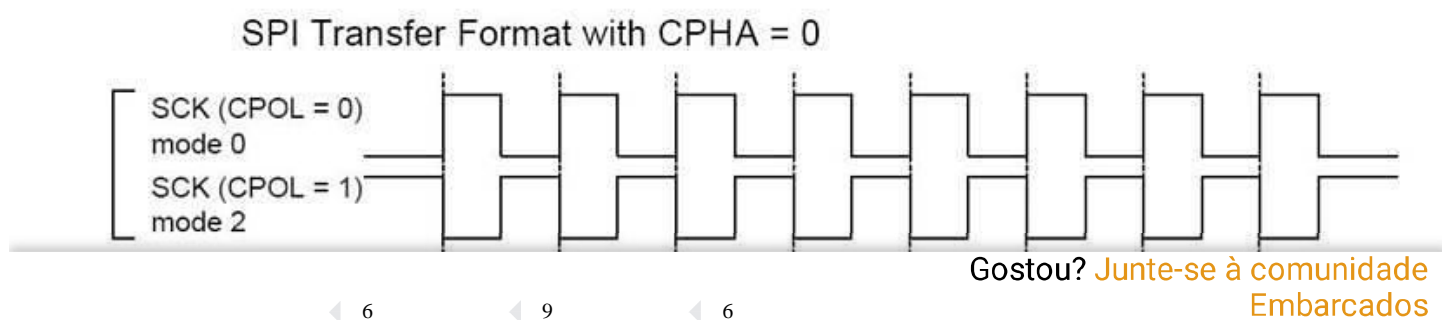
A SPI não apenas é capaz de fazer essa conversão serial/paralelo, como também possui um gerador de clock, o controle para a troca do frame e o slave-select. Dessa forma, tornando ele um dispositivo de comunicação completo. Na figura abaixo podemos observar como ele se dispõe.

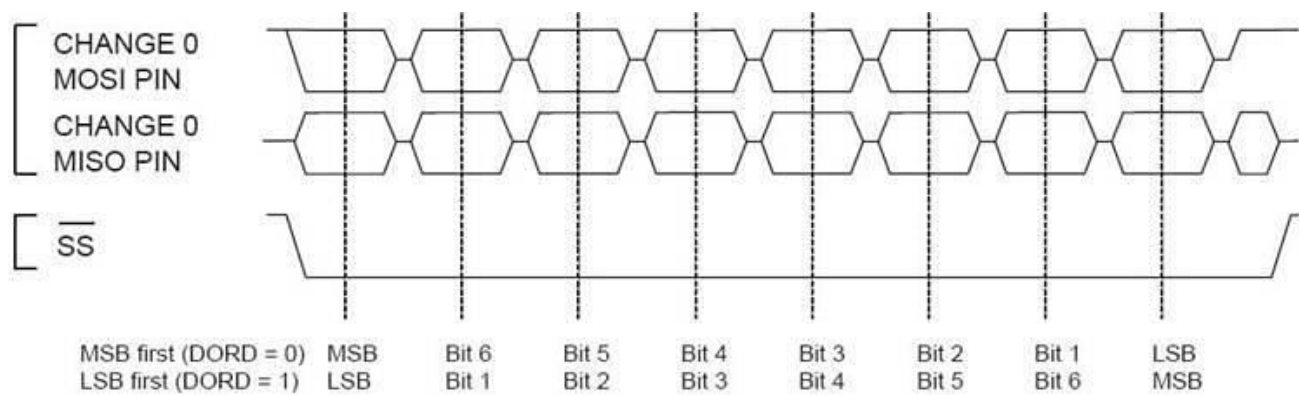


Por padrão a comunicação SPI permite a configuração das bordas de comunicação do clo através de sua polaridade e fase. A configuração da polaridade se dá através de CPOL (Clock Polarity), e a configuração da fase se dá através de CPHA (Clock Phase). Seus modos possíveis são:

Modo	CPOL	CPHA	Borda de troca	Transição	Nível em IDLE
0	0	0	Subida	Meio do bit	1
1	0	1	Descida	Começo do bit	0
2	1	0	Descida	Meio do bit	0
3	1	1	Subida	Começo do bit	1

Outra característica fundamental na comunicação é a definição da posição do bit mais significativo (MSB). Através de DORD é possível definir que o bit mais significativo será o primeiro (DORD = 0) ou o último (DORD = 1) bit trocado. É possível observar todas essas características através das duas imagens abaixo:





cpol_2

Conclusão

Pudemos observar as principais características da comunicação SPI, além de um comparativo de hardware com outros tipos de comunicação. Observamos diagramas do funcionamento do dispositivo e entendemos como se dá a forma de onda.

No próximo artigo vamos entender quais são as principais características de software par sua comunicação, utilizando o Arduino como hardware base. Também vamos estudar a estrutura de software com um dispositivo de mercado.

Gostou? [Junte-se à comunidade Embarcados](#)

Leia também a sequência desse artigo em [Comunicação SPI - Parte 2](#), onde abordamos mais detalhes de comunicação e os fundamentos básicos de software, e [Comunicação SPI Parte 3](#), onde apresentamos um exemplo de comunicação entre um microcontrolador AT89S8253 e uma memória 25LC256.

Referência

<http://gse.ufsc.br/~bezerra/disciplinas/ProgPerif/sem04.2/Seminarios/RicardoPlaner/Apresentacao.htm>

<http://www.youtube.com/watch?v=QMNck8nKw-k>

<http://www.ni.com/example/31163/en/>

http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

http://home.kpn.nl/thomas_7/1wire/1wire_isolating.html

<http://www.adestotech.com/sites/default/files/datasheets/doc3639.pdf>

<http://ww1.microchip.com/downloads/en/DeviceDoc/21189T.pdf>



Comunicação SPI – Parte 1 por *Francesco Sacco*. Esta obra está sob a licença [Creative Commons Atribuição-Compartilhado 4.0 Internacional](#).

Gostou? [Junte-se à comunidade Embarcados](#)

Francesco Sacco

<http://br.linkedin.com/in/saccofrancesco/>

Engenheiro Eletricista formado PUC-SP e Técnico em Eletrônica pela ETE Getúlio Vargas, é especialista em projetos eletrônicos analógicos e digitais, especialmente no condicionamento de sinais analógicos e integração entre dispositivos digitais. Nos últimos anos, vem atuando no desenvolvimento de software embarcado para núcleos ARM Cortex e na construção de hardwares para sistemas de potência.

- Pingback: [Comunicação SPI - Parte 3 - Microcontrolador AT89S8253 + EEPROM 25LC256 - Embarcados - Sua fonte de informações sobre Sistemas Embarcados\(\)](#)
- Pingback: [Blog do Software Livre Comunicação SPI em Linux » Blog do Software Livre\(\)](#)
- Pingback: [Raspberry PI e Python interagindo com módulo RFID\(\)](#)
- Pingback: [Arduino: RS-232/RS-485 - Hardware\(\)](#)
- Pingback: [Comunicação SPI em Linux - Embarcados - Sua fonte de informações sobre Sistemas Embarcados\(\)](#)

Artimar Desktop

Gostou? [Junte-se à comunidade Embarcados](#)