

BARRAMENTOS E INTERFACES DE COMUNICAÇÃO SÉRIE

José A. Fonseca

Barramentos Série - Conceitos Fundamentais

- Diz-se que se está na presença de um barramento ou interface série sempre que exista uma só linha (suporte) física para transferência de informação, sendo esta transmitida 1 bit de cada vez.
- A sequência temporal de bits transferida pode ser organizada por forma a transferir informação mais complexa que 1 bit, incluindo funções de controlo e endereçamento as quais, num barramento paralelo, implicam habitualmente a atribuição de linhas físicas específicas.

Barramentos Série - Conceitos Fundamentais

Razões da popularização de interfaces e barramentos série:

- Simplicidade de implementação de algumas interfaces (RS-232, SPI, ...).
- Nota: Muitas são mais complexas que as paralelas (p.ex. USB).
- Simplicidade de ligação de cablagem.
- Diminuição de custos de interligação.
- Possibilidade de transmissão a distâncias elevadas.

Barramentos Série - Conceitos Fundamentais

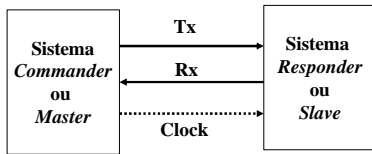
A comunicação série pode ser utilizada para:

1. Interfaces de comunicação ponto a ponto:
 - SPI, RS232
2. Ligações Multiponto (entre várias unidades) / barramentos:
 - I2C, USB, RS422, RS485
 - CAN (Controller Area Network)

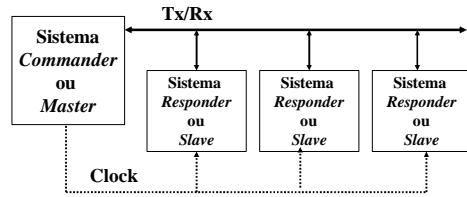
Nota 1: Nos exemplos indicados acima existem especificações físicas e especificações que contemplam todas as camadas do modelo OSI.

Nota 2: No caso 1 é vulgar assistir-se a unidirecionalidade na comunicação pelo que existem 2 caminhos separados consoante a direcção.

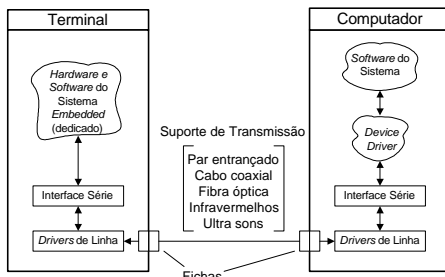
Interfaces de comunicação ponto a ponto



Barramento Série



Elementos de uma ligação ponto a ponto

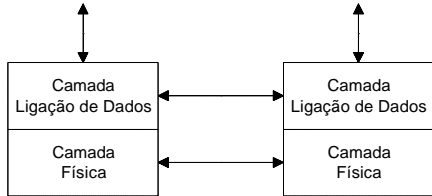


Modelo OSI (Open Systems Interconnection Basic Reference Model)



- Serviços de acesso à informação e comunicação para diferentes tipos de aplicações.
- Transporte de informação, poupando aos níveis superiores os problemas relacionados com a comunicação entre dois equipamentos distantes.

Camadas do modelo OSI sobre as quais incide em geral a especificação dos barramentos



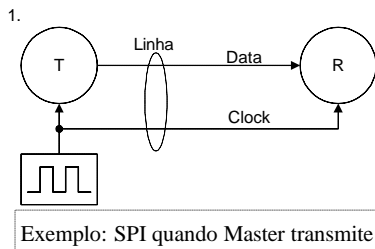
Sincronização de relógios

O sincronismo é obtido através da utilização de um relógio no transmissor e outro no receptor. Esses relógios têm de estar sincronizados.

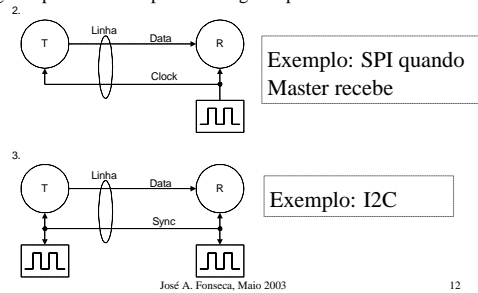
Procedimentos básicos:

1. Relógio explícito do transmissor.
2. Relógio explícito do receptor.
3. Relógio explícito auto-sincronizado.
4. Relógio implícito.
5. Relógio codificado (*self-clocking*).

Sincronização de relógios:
Relógio explícito do transmissor



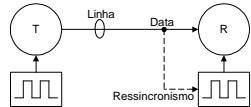
Sincronização de relógios:
Relógio explícito do receptor e relógio explícito auto-sincronizado



Sincronização de relógios:

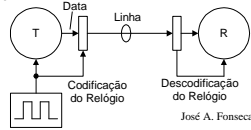
Relógio implícito e relógio codificado (*self-clocking*)

4.



Exemplo: Rs-232, CAN, HDLC

5.



Exemplo: barramentos utilizando codificação Manchester

José A. Fonseca, Maio 2003

13

Serialização da transmissão de dados

Orientada ao Bit.

- A transmissão é feita por uma sequência de bits de comprimento arbitrário (fixo, variável em degraus, variável em contínuo, limitado, ...).

Orientada ao Byte ou ao Caracter.

- A transmissão é feita byte a byte ou caracter a caracter, ou seja, não é possível transmitir um número de bits diferente do estipulado previamente.

José A. Fonseca, Maio 2003

14

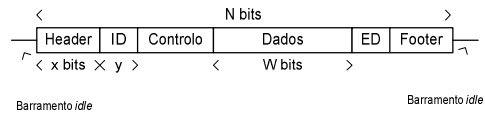
Transmissão orientada ao Bit

- Uma sequência de bits enviada por um Master entre duas situações de barramento ou interface *idle* (livre, disponível) constitui uma trama.
- As tramas dividem-se em campos com diferentes funções:
 - Sinalização de início e de fim.
 - Identificação do produtor, destinatário ou da própria informação.
 - Arbitragem de acesso ao barramento.
 - Detecção de erros de transmissão.
 - Transmissão de dados

José A. Fonseca, Maio 2003

15

Transmissão orientada ao Bit

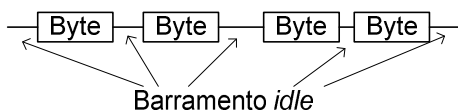


José A. Fonseca, Maio 2003

16

Transmissão orientada ao Byte

- Entre dois intervalos com o barramento *idle* é enviado um *byte* (ou uma sequência curta de bits interpretada em conjunto, quando não se enviam 8 bits - carácter).
- O envio de um byte é a operação atómica (indivisível) do barramento.
- As tramas são constituídas por sequências de bytes.



José A. Fonseca, Maio 2003

17

Transmissão orientada ao Byte

- A transacção de quantidades de informação superiores a 1 byte / carácter terá pois de ser efectuada por múltiplas transacções “atómicas”.
- É pois possível, com base em múltiplas transacções de 1 carácter, desenvolver protocolos elaborados para trocas de superiores quantidades de informação (em termos do modelo OSI tal pertence à camada ligação de dados pelo que não será aqui objecto de discussão).

José A. Fonseca, Maio 2003

18

RS-232

- A norma RS232 utiliza transacções orientadas ao carácter que pode ser de 5, 6, 7 ou 8 bits.
- A transacção “atómica” em RS-232 inclui:

Start Bit + Data Bits + (Parity Bit) + Stop Bit(s)

- **Start Bit:** Um “0” (SPACE) (Nota: A linha em repouso deverá estar a “1” - MARK).
- **Data Bits:** Habitualmente 7 ou 8 (podem ser 5 ou 6); quando são 7 é habitual a utilização do código ASCII.

José A. Fonseca, Maio 2003

19

RS-232

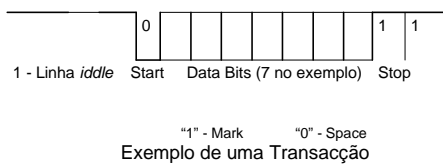
Start Bit + Data Bits + (Parity Bit) + Stop Bit(s)

- **Parity Bit:** É opcional; quando existe pode ser definido como paridade PAR ou paridade ÍMPAR.
- **Stop Bit(s):** É obrigatório utilizar pelo menos 1; podem ser, no entanto, 1,5 ou 2 bits; coincidem com o estado de linha inactiva (*idle*); proporcionam a existência de um intervalo de tempo mínimo entre dois caracteres consecutivos enviados sobre a linha (quase não são necessários actualmente face à rapidez de funcionamento dos dispositivos utilizados).

José A. Fonseca, Maio 2003

20

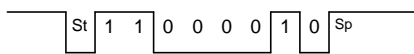
RS-232 – Transmissão de um caracter



RS-232 - Codificação

- NRZ-L (Non Return to Zero - Level)
- Aos bits "0" e "1" correspondem dois níveis de tensão adequados. Em RS-232 atribui-se:
 - +3 a + 12 Volts ao "0" (também designado por ON ou SPACE)
 - 3 a -12 Volts ao "1" (também designado por OFF ou MARK)

RS-232 – Exemplo de transmissão



Exemplo: Transmissão do caractere 43_H em código ASCII com paridade ímpar.

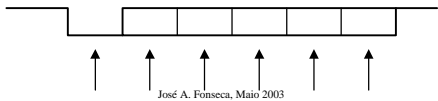
Nota: Transformado em tensões, o aspecto é exactamente invertido devido à codificação do "1" (-12V) e do "0" (+12V).

RS-232 *Baud Rate*

- Baud rate ou taxa de transmissão é o nº de bits por segundo passíveis de serem transmitidos numa dada ligação.
- Exemplos: 4.800, 9.600, 19.200, 38.400, 57.600, 115.200
- Exemplo: o caractere anterior, considerando apenas 1 stop bit (10 bits total) demoraria 174µseg a ser transmitido a 57.600 *baud*.

RS-232 Recuperação no receptor

- Além do nº de bits de informação, da paridade e do nº de stop bits, o receptor tem de conhecer o valor da taxa de transmissão.
- O receptor deve então sincronizar-se pelo flanco negativo (transição de Mark → Space) do Start Bit e fazer as leituras a meio do intervalo reservado ao Bit.

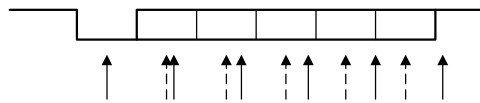


José A. Fonseca, Maio 2003

25

RS-232 Recuperação no receptor

- Para a recuperação o receptor utiliza um relógio com frequência igual ao da transmissão.
- O facto de existir um erro de fase entre os dois relógios e de as diferenças de frequência darem origem a erros cumulativos pode originar erros na leitura.



José A. Fonseca, Maio 2003

26

RS-232 Recuperação no receptor

- É portanto habitual utilizarem-se relógios cuja frequência é:

$$f_{CLK} = N \cdot \frac{1}{T_{BIT}}$$

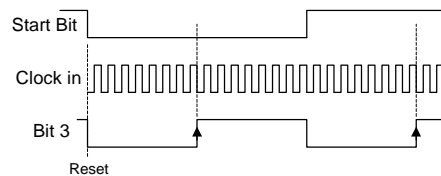
- Valores típicos de N: 16 ou 64
- Utilizando um relógio com N=16 é possível obter um erro de fase cujo valor máximo é T/16.

José A. Fonseca, Maio 2003

27

RS-232 Recuperação no receptor

- Exemplo com um contador de 4 bits ao qual é feito o Reset quando da detecção do Start Bit e o qual usa um relógio com N=16:



José A. Fonseca, Maio 2003

28

RS-232 Outras características

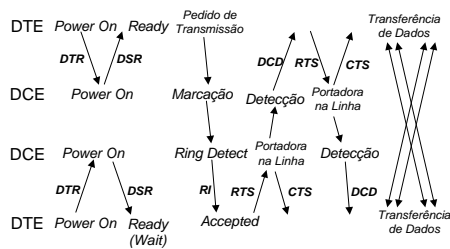
No standard RS232 são definidas essencialmente:

- Características eléctricas (tensão de saída, corrente, resistência de entrada dos receptores, polaridade, ...)
- Características físicas (fichas, pinos das fichas, distâncias)
- Taxa de transmissão (desactualizada no standard)
- *Handshaking* entre sistemas interligados.
- Modos de teste
- ...
- Nota: TRATA-SE de uma ligação PONTO a PONTO.

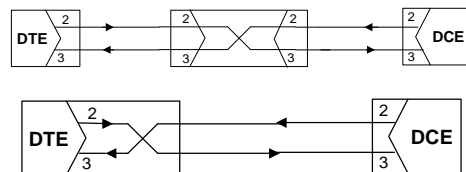
RS-232 e as portas série de PCs

- Restam ainda procedimentos para ligar MODEMS
- Há que ter cuidado com o *Handshaking*:
 - CTS (Clear to Send) vs RTS (Request to Send)
 - DTR (Data Terminal Ready) vs DSR (Data Set Ready)
 - Estes são pares de linhas físicas que permitem controlar o fluxo de dados entre os equipamentos interligados.
 - Para os desactivar há que utilizar um **Null Modem**.
 - Em alternativa utiliza-se handshaking por software com os sinais XON e XOFF. XON (X de Tx – Transmitter) liga o transmissor, XOFF desliga-o.
 - XON – CTRL-Q (11H) XOFF – CTRL-S (13H)

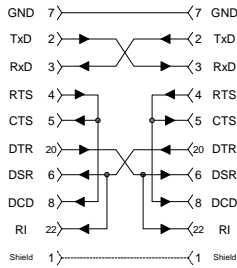
RS-232 - Ligação de Modems



RS-232 - Null Modems

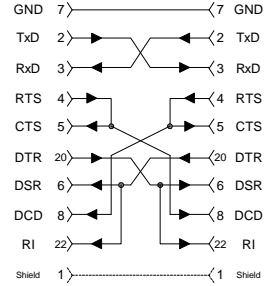


RS-232 - Null Modems



Nota: DB25; há que alterar para DB9.

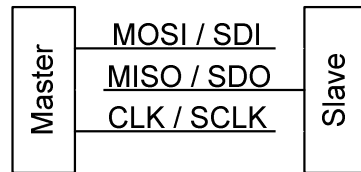
RS-232 - Null Modems



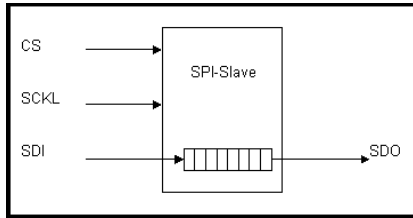
SPI – Serial Peripheral Interface

- Definida inicialmente pela Motorola.
- Também conhecida por Microwire (National Semiconductor).
- Arquitectura Master-Slave com ligação ponto a ponto entre ambos e relógio gerado pelo Master.
- Utiliza um esquema simples baseado em shift-registers, sendo cada bit transferido para o SR por um ciclo de clock gerado pelo Master (os ciclos não têm de ser de duração idêntica – fácil de gerar por hardware ou por software).

SPI – Serial Peripheral Interface



SPI – Serial Peripheral Interface

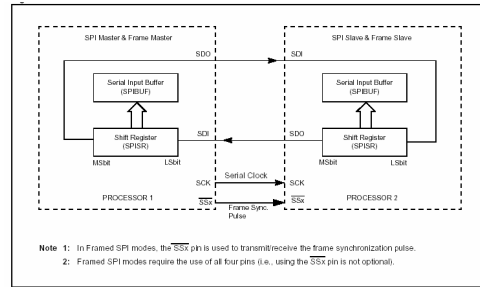


Estrutura de um Slave (<http://www.mct.net/faq/spi.html>)

José A. Fonseca, Maio 2003

37

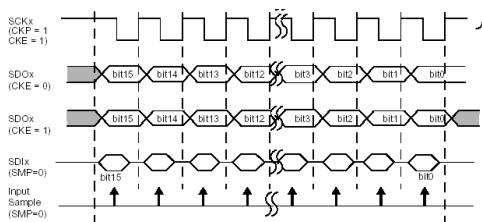
SPI – Arquitecturas de ligação



José A. Fonseca, Maio 2003

38

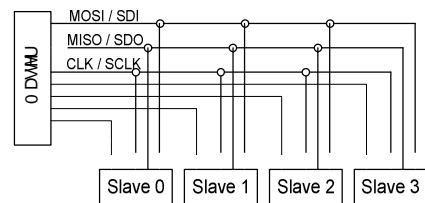
SPI – Exemplo de transferência



José A. Fonseca, Maio 2003

39

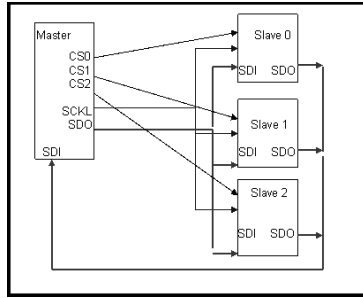
SPI – Arquitecturas de ligação



José A. Fonseca, Maio 2003

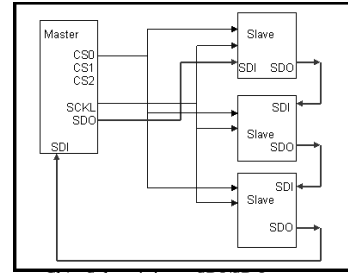
40

SPI – Arquitecturas de ligação



*Chips Selects
Independentes
com SDOs em
alta impedância*

SPI – Arquitecturas de ligação



Chip Select único e SDI/SDO em cascata

SPI – Aplicações

- Conversores (ADCs, por vezes sem SDI e DACs).
- Memórias (EEPROM, Flash)
- Real-Time Clocks.
- Sensores
- Dispositivos variados: LCDs (por vezes sem SDO), UARTs, controladores CAN, USB, ...

I2C – Dados gerais

- Desenvolvido pela Philips, 1º standard em 1992, actualmente na versão 2.1 de 2000.
- De acordo com Philips: "simple bidireccional 2-wire bus for efficient inter-IC control".
- Linhas do I2C:
 - SDA (*Serial Data*), bidireccional.
 - SCL (*Serial Clock*).
- Taxa de transmissão actual: 0 a 3,4 Mbit/seg (*High Speed Mode*); 0 a 400 kbit/seg (*Fast*); 0 a 100 kbit/seg (*Standard*)

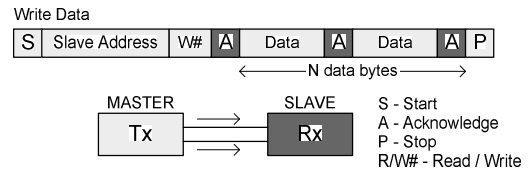
I2C – Dados gerais

- Desenvolvido pela Philips, 1º standard em 1992, actualmente na versão 2.1 de 2000.
- De acordo com Philips: “simple bidireccional 2-wire bus for efficient inter-IC control”.
- Linhas do I2C:
 - SDA (*Serial Data*), bidireccional.
 - SCL (*Serial Clock*).
- Taxa de transmissão actual: 0 a 3,4 Mbit/seg (*High Speed Mode*); 0 a 400 kbit/seg (*Fast*); 0 a 100 kbit/seg (*Standard*)

José A. Fonseca, Maio 2003

45

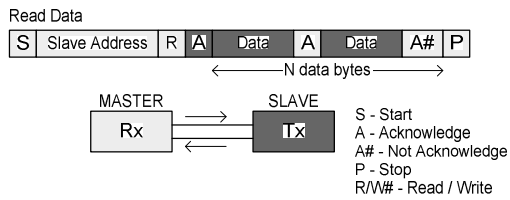
I2C – Operação de escrita



José A. Fonseca, Maio 2003

46

I2C – Operação de leitura



José A. Fonseca, Maio 2003

47

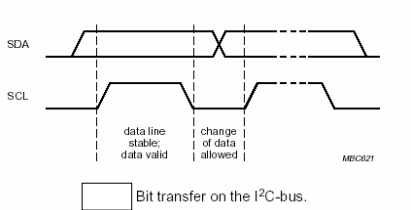
I2C – Detalhes das transferências

- Dados têm de se manter estáveis durante o estado “high” do *clock*.
- Assim, transição de 1-0 na linha SDA quando *clock* está “high” marca um START.
- Transição de 0-1 na linha SDA quando *clock* está “high” marca um STOP.
- START e STOP são sempre gerados pelo Master.
- As linhas funcionam numa ligação “wired AND”, permitindo usar bit recessivo (1) e dominante (0) para várias sinalizações.

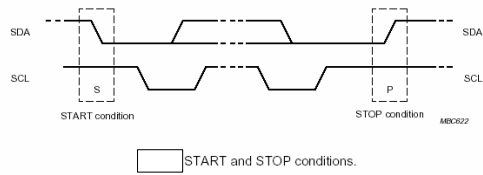
José A. Fonseca, Maio 2003

48

I2C – Transferência de um bit



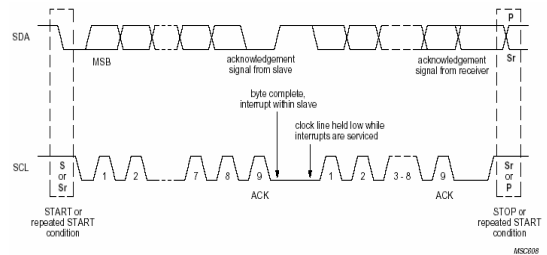
I2C – Start e Stop



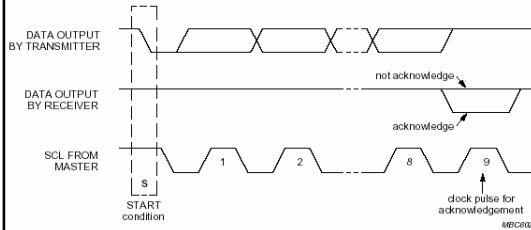
I2C – Detalhes das transferências

- A transferência baseia-se em bytes (8 bits) sendo transmitido o MSbit primeiro.
- Após o 8º bit (o LSbit), o receptor tem de gerar um *Acknowledge* sob a forma de um bit dominante.
- A sincronização é semi-síncrona, podendo os *slaves* forçar o alargamento da transferência mantendo a zero (bit dominante) a linha de SCL.
- O *Not Acknowledge* durante uma operação de *read* (*master* é o *receiver*) sinaliza ao *slave* o fim da transferência.

I2C – Transferência de dados



I2C – Acknowledge por bit dominante



José A. Fonseca, Maio 2003

53

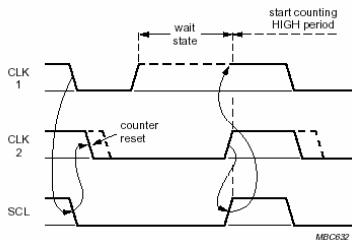
I2C – Detalhes das transferências

- O I2C é **Multi MASTER** e os *masters* devem ter os *clocks* sincronizados.
- Na sincronização de relógios: a largura do “0” é determinada pelo Master com tempo de *low* superior.
- A largura do “1” (*high*) é determinada pelo Master com tempo de *high* inferior.
- A sincronização é feita durante o processo de arbitragem.
- A arbitragem é feita por bit dominante / recessivo: os masters vão escrevendo bits na linha SDA e perdem quando lêem um dominante tendo escrito um recessivo.

José A. Fonseca, Maio 2003

54

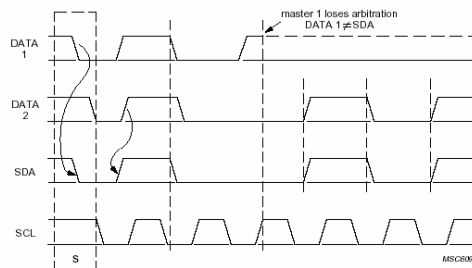
I2C – Sincronização de relógios durante arbitragem



José A. Fonseca, Maio 2003

55

I2C – Arbitragem entre dois Masters



José A. Fonseca, Maio 2003

56