

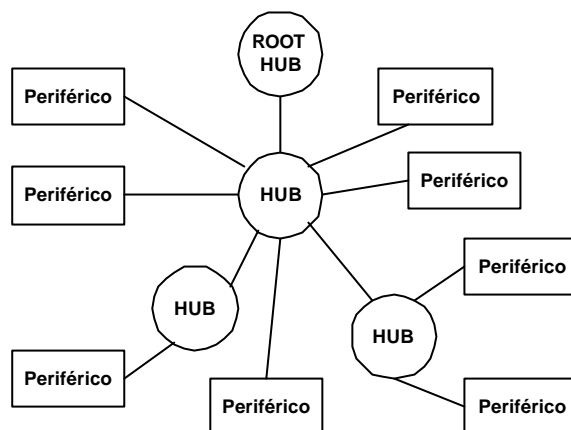
## USB – Universal Serial Bus: Introdução

- **História:** Barramento standard para periféricos desenvolvido pelos principais fabricantes de PCs e algumas indústrias de telecomunicações com vista a conferir características de *plug-and-play* à ligação dos diferentes dispositivos.
- **Objectivos:** permitir a configuração automática dos periféricos logo que ligados ao barramento (sem necessidade de arranque do computador ou de instalação especial) e facilitar a ligação física, promovendo a disponibilização de ligações nos diferentes periféricos.

## USB – Características da Camada Física

- **Ligação física:**
- A ligação física é feita em árvore (*tiered star*), partindo de um sistema central (*host*) para vários dispositivos que podem actuar como extremos ou disponibilizar acessos para outros dispositivos (operando como *hubs*).
- Entre cada dois dispositivos utiliza-se um cabo de 4 condutores com um máximo de 5 metros, sendo um par para transmissão de sinal em modo diferencial e o outro para fornecimento de potência (há pois dispositivos que podem ser alimentados a partir de um *hub*).

## USB – Ligação Física



José A. Fonseca, Dezembro de 2001

3

## USB – Características da Camada Física

### **Codificação e transmissão de dados:**

- Os dados são codificados em NRZI e transmitidos à taxa fixa de 12Mbps (v.1.x) ou a 480Mbps (v.2.0).
- Há um modo especial para transmissão de dados a 1.5Mbps para dispositivos mais lentos e simples.

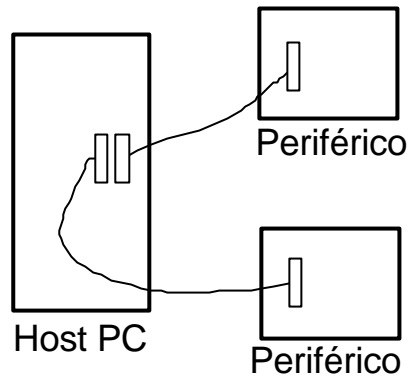
### **Gestão da energia:**

- A disponibilização de energia entre um *hub* e um dispositivo é utilizada para a detecção da ligação ou da retirada do dispositivo o que permite implementar o modo de operação *plug-and-play*.

José A. Fonseca, Dezembro de 2001

4

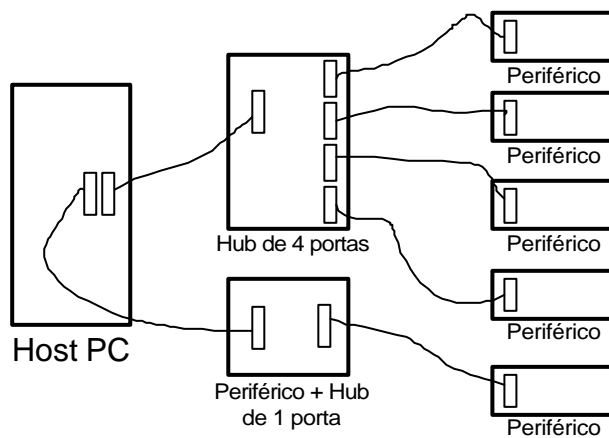
### USB – O *host* e os periféricos



José A. Fonseca, Dezembro de 2001

5

### USB – O *host*, *hubs* e os periféricos



José A. Fonseca, Dezembro de 2001

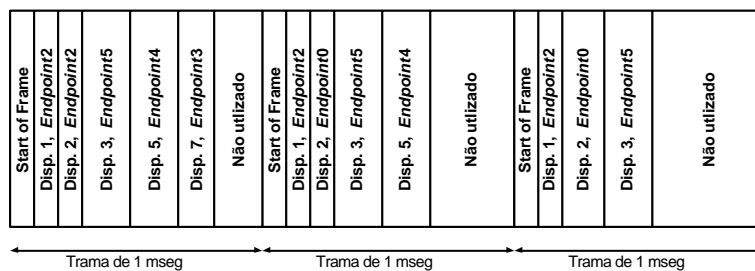
6

## USB – Camada Ligação de Dados

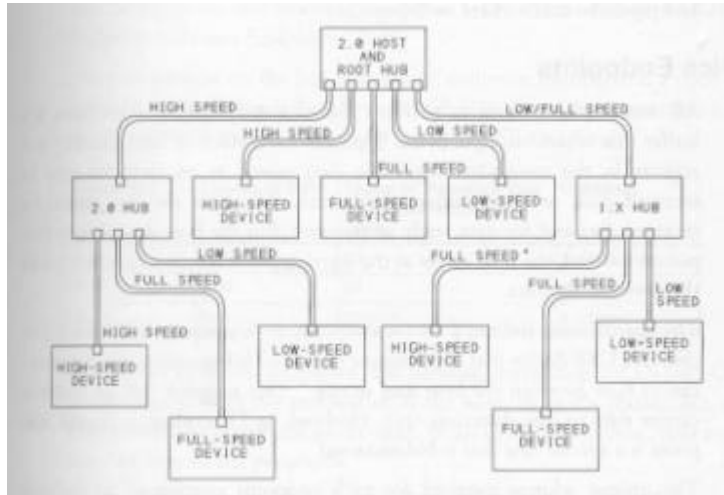
### MAC – Medium Access Control:

- A interligação é controlada pelo *host* (pode-se dizer portanto que se trata de operação em *master-slave*) que impõe uma sequência sincronizada de tramas com uma duração de 1 ms (*time-triggered*) ou de 125µs (*high speed – v2.0*).
- Cada trama começa por um campo SOF – *Start Of Frame* e termina com um intervalo mínimo com o bus *idle* designado por EOF – *End Of Frame*.

## USB – Sequência de Tramas



## USB – Interligação a velocidades diferentes



José A. Fonseca, Dezembro de 2001

9

## USB – Elementos numa transferência

**Device** (dispositivo): elemento ligado ao bus USB

**Device Endpoint:**

- *Buffer*, dentro de um dispositivo, onde se armazena informação que será transferida pelo bus (memória, registos, ...).
- Na especificação: “A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device.”
- Todos os dispositivos têm de dispôr de um *Endpoint 0*.
- Os *endpoints* podem ir de 0 a 15.
- O *host* não tem *endpoints* já que é quem controla a transferência de informação.

José A. Fonseca, Dezembro de 2001

10

## USB – Elementos numa transferência

### ***Pipes (ligação dos endpoints ao host):***

- Associação entre o software do controlador existente no *Host* e o *endpoint* do dispositivo.
- O *Host* inicia uma (um?) *pipe* na sequência do power-up ou da ligação do dispositivo; pode terminá-lo ou criar um novo na sequência de alteração da configuração ou interface com o dispositivo.
- *Default control pipe* – associada ao *endpoint* 0, obrigatório em todos os dispositivos USB

## USB – Elementos numa transferência

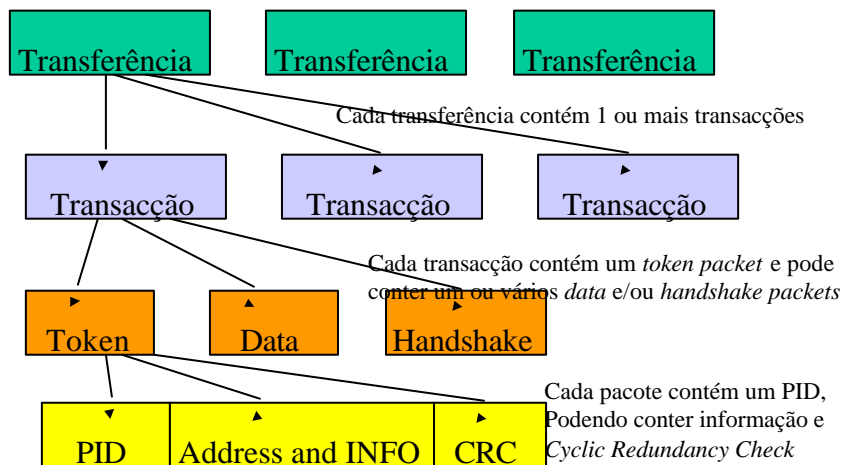
### ***Pipes (ligação dos endpoints ao host):***

- Informação associada:
  - Endereço do endpoint - Tipo de transferência
  - Nº de bytes do pacote de informação
  - Intervalo entre transferências (quando se aplica)
- A criação do pipe pode depender da existência de largura de banda suficiente para a transferência pretendida (isócronas, com taxa garantida e interrupt com latência garantida).
- Restantes são tratadas em *best effort*.

### USB – Tipos de transacções

Tipo de transacção	Fonte da Informação	Tipo de Transferência	Conteúdo
IN	<i>Device</i>	Todas	Dados Genéricos
OUT	<i>Host</i>	Todas	Dados Genéricos
Setup	<i>Host</i>	Controlo	Pedido / configuração

### USB – Transferências de dados



## USB – Tipos de Pacotes: *TOKEN*

(Identificação do tipo de transacção)

Nome do PID	Valor	Usado em	Fonte	Speed	Descrição
OUT	0001	Todas	<i>Host</i>	Todas	Endereço de <i>endpoint</i> para OUTs
IN	1001	Todas	<i>Host</i>	Todas	Endereço de <i>endpoint</i> para INs
SOF	0101		<i>Host</i>	Todas	Indicação de início de trama e nº da trama
SETUP	1101	<i>control</i>	<i>Host</i>	Todas	Endereço de <i>endpoint</i> para <i>setup</i>

José A. Fonseca, Dezembro de 2001

15

## USB – Tipos de Pacotes: *DATA*

(Transferência de informação ou *status*)

Nome do PID	Valor	Usado em	Fonte	Speed	Descrição
DATA0	0011	Todas	<i>H ou D</i>	Todas	<i>Toggle</i> , sequência de informação
DATA1	1011	Todas	<i>H ou D</i>	Todas	<i>Toggle</i> , sequência de informação
DATA2	0111	Isócron.	<i>H ou D</i>	<i>High</i>	Sequência de informação
MDATA	1111	Isócron, <i>Interr.</i>	<i>H ou D</i>	<i>High</i>	Sequência de informação

José A. Fonseca, Dezembro de 2001

16

## USB – Tipos de Pacotes: *HANDSHAKE*

(Transferência de *status*)

Nome do PID	Valor	Usado em	Fonte	Speed	Descrição
ACK	0010	Todas	<i>H ou D</i>	Todas	Aceitação de pacote sem erros
NACK	1010	Con., <i>Bulk</i> , <i>Interr.</i>	<i>Device</i>	Todas	Receptor não pode aceitar dados ou emissor não pode enviá-los
STALL	1110	Con., <i>Bulk</i> , <i>Interr.</i>	<i>Device</i>	Todas	Pedido de controlo não suportado ou <i>endpoint</i> em <i>Halt</i>
NYET	0110	Con.(W), <i>Bulk (Out)</i> , <i>Split</i>	<i>Device</i>	<i>High</i>	Receptor não pode receber pacote seguinte ou <i>hub</i> não tem <i>split data</i>

José A. Fonseca, Dezembro de 2001

17

## USB – Tipos de Pacotes: *SPECIAL*

Nome do PID	Valor	Usado em	Fonte	Speed	Descrição
PRE	1100	Controlo, <i>Interr.</i>	<i>Host</i>	<i>Full</i>	Indica que o próximo pacote é <i>low speed</i>
ERR	1100	Todas	<i>D. ou Hub</i>	<i>High</i>	Erro em transacção <i>split</i>
SPLIT	1000	Todas	<i>Host</i>	<i>High</i>	Precede <i>Token</i> para indicar transacção <i>split</i>
PING	0100	Con.(W), <i>Bulk (Out)</i>	<i>Host</i>	<i>High</i>	Verificação de <i>Busy</i> após NYET

José A. Fonseca, Dezembro de 2001

18

## USB – Elementos numa transferência

- Cada trama está dividida em slots nos quais se podem processar quatro tipos diferentes de transferências de informação:
  - Isócronas,**
  - Interrupção,**
  - Controlo,**
  - Bulk.***
- O processamento de um daqueles tipos de transferência de informação implica a transmissão de um a três pacotes (*token, data, handshake e special*).

## USB – Transferências Isócronas

- Permitem a transferência de informação com uma largura de banda garantida constante cujo valor é negociado durante o processo de arranque do periférico (*enumeration*).
- Não existem garantias de transmissão sem erro pelo que o dispositivo tem de ser tolerante à sua ocorrência. É, no entanto, possível detectar erros devido a um mecanismo de CRC (não há é repetições já que o receptor não dispõe de nenhum processo para comunicar ao transmissor que houve erro).

## USB – Transferências Isócronas

- Quando existe “aberta” uma transferência isócrona, há transmissão de informação em todas as tramas de 1ms
- A quantidade de informação transmitida é constante em cada trama, podendo no máximo ser 1.023 bytes.
- Exemplo: transmissão audio ou video.

## USB – Transferências de Interrupção

- Permitem que o *host* aceda com um período pré-estabelecido a um dispositivo, sendo esse período definido durante o processo de arranque do periférico (trata-se, com efeito de *polling*).
- Não tem portanto de existir uma transferência por interrupção em todas as tramas (a periodicidade pode ser superior a 1ms embora sempre múltipla).
- A informação flui sempre do dispositivo para o *host* e, no máximo, podem ser transferidos 64 bytes.

## USB – Transferências de Controlo

- São utilizadas pelo *host* para o envio de mensagens de comando e controlo para um dispositivo.
- Em princípio a informação a transmitir é pequena mas também pode ser necessário transmitir informação com alguma dimensão utilizando-se nesse caso mecanismos idênticos ao das transferências ***bulk***.

## USB – Transferências ***Bulk***

- Permitem a transmissão de dados com garantias de que não existirão erros.
- Para essa garantia há que ser possível a repetição de dados que sofreram erros e portanto não são dadas garantias de largura de banda.
- As transferências deste tipo utilizam a parte da trama que se encontra disponível após a transmissão das transferências dos três tipos anteriores.

### **USB –Enumeration**

- Troca de informação inicial que permite ao host reconhecer o dispositivo e atribuir-lhe um *device driver*.
- Inclui:
  - Atribuição de um endereço.
  - Leitura de informações (por ex. *Vendor*).
  - Atribuição e carga de Device Driver
  - Selecção de configuração.

### **USB –Enumeration**

- No arranque do sistema o *Host* acede ao *root hub* para descobrir se há dispositivos a ele ligados.
- Identifica também os dispositivos ligados aos diferentes hubs descobertos no sistema.
- O Host repete o procedimento periodicamente, durante o funcionamento do sistema.
- Passa-se à *Enumeration* de cada um dos dispositivos encontrados.

## USB – Passos na *Enumeration*

- Durante a Enumeration o dispositivo passa por quatro dos seis estados, definidos pela especificação USB:
  - *Powered*
  - *Default*
  - *Address*
  - *Configured*
- Os restantes são:
  - *Attached*
  - *Suspended*

## USB – Passos na *Enumeration*

1. O dispositivo é ligado a um *Hub* ou o sistema arrancou; o dispositivo recebe potência do *hub* (*Powered state*).
2. O *hub* detecta o dispositivo pela potência fornecida.
3. O *hub* usa a sua *pipe* de interrupção para informar o *host* de que alguma coisa aconteceu. O *host* usa um pedido de *Get\_Port\_Status*, recebendo a indicação de que um novo dispositivo se ligou ao USB.
4. O *hub* determina pela ligação eléctrica se o dispositivo é *low* ou *full speed*.
5. O *hub* faz reset ao dispositivo (linhas D+ e D- a “0”, só possível devido à ligação individual entre os dois).

## USB – Passos na *Enumeration*

6. Através do *hub* e de uma sequência de sinais nas linhas D, o *host* verifica se o dispositivo é *high speed*.
7. O *host* verifica se o dispositivo já foi libertado pelo *hub* do estado de *reset*, fazendo pedidos de *Get\_Port\_Status* até isso acontecer. O dispositivo está pronto a comunicar com o *host* por transferências de controlo via uma *pipe* associada ao *endpoint 0*. Utiliza o endereço 00H (*Default state*).
8. O *host* envia um pedido de *Get\_Descriptor* para o endereço 0, *endpoint 0*, para determinar o # de bytes do pacote suportado pelo dispositivo.

## USB – Passos na *Enumeration*

9. O *host* atribui um endereço específico ao dispositivo enviando-lhe um *Set\_Address*. O dispositivo deverá passar a utilizar este endereço em todas as comunicações. O endereço é apenas válido até o dispositivo ser desligado e pode ser diferente na próxima *enumeration* (*Address state*).
10. O *host* identifica as capacidades do dispositivo a partir da resposta a um *Get\_Descriptor*: tamanho de pacote, # de configurações, *vendor ID*, *product ID*, ...
11. O *host* atribui um *device driver*, através da informação de *Vendor* e *Product ID* e carrega-o.

## USB – Passos na *Enumeration*

11. O *device driver* selecciona uma configuração através de um *Set\_Configuration* (muitos dispositivos só dispõem de uma). O dispositivo está pronto para utilização (*Configured state*).

- **Os restantes estados:**
- *Attached state*: quando o dispositivo não pode receber potência do *hub* (*overcurrent*, p.ex.) fica fisicamente ligado mas não pode comunicar.
- *Suspend state*: quando o dispositivo não detectou actividade no bus por, pelo menos, 3mseg.