



**Vitor Amadeu Souza**

vitor@cerne-tec.com.br

## **Introdução**



Veremos agora os passos para enviar uma string de caracteres pela USB. Porém antes veremos um tutorial referente a porta USB.

## **Tutorial USB**

### ***Sistema de descrição do USB***

Um sistema USB é descrito em três diferentes áreas:

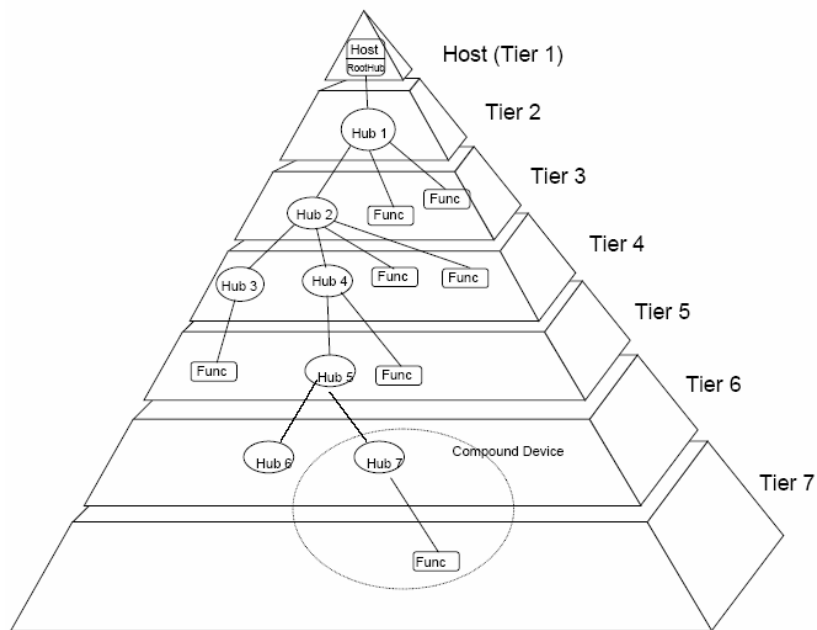
- Interconexão USB;
- USB Device;
- USB Host;

A interconexão USB é a maneira no qual os dispositivos USB estão conectados com o host. Os seguintes aspectos estão incluídos:

- Topologia do Barramento: Modo de conexão entre o device e o host;
- Relação entre camadas: Capacidade de cada tarefa ser executada na pilha USB;
- Modo de fluxo de dados: A maneira no qual os dados se movem no sistema sobre o protocolo USB;

### ***Topologia do Barramento***

A interconexão física utiliza a topologia estrela. Cada HUB (Concentrador) é o centro de cada estrela. Cada fio é ligado ponto-a-ponto entre o host e o HUB. Na figura abaixo podemos apreciar melhor este conceito:



## ***USB Host***

Somente há um Host (Hospedeiro) no barramento USB. Esta interface é chamada de Host Controller. O Host Controller pode ser implementado por hardware, firmware ou software.

## ***USB Devices***

Existem dois tipos de devices USB, vejamos:

- Hubs, que disponibilizam pontos adicionais de acesso ao USB;
- Funções, que disponibilizam capacidades adicionais ao sistema, como joystick digitais ou alto-falantes.

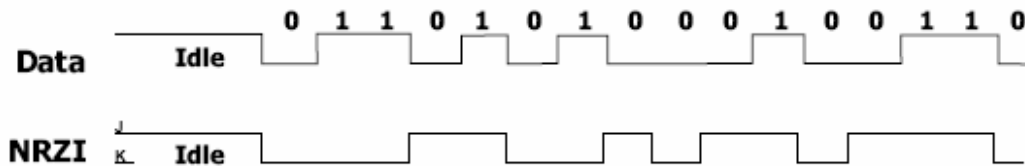
Os dispositivos USB atendem aos seguintes termos:

- Compreendem o protocolo USB;
- Respondem as operações standards (padrão) do protocolo, como configuração e reset;

## ***Interface Física***

## ***Codificação de dados***

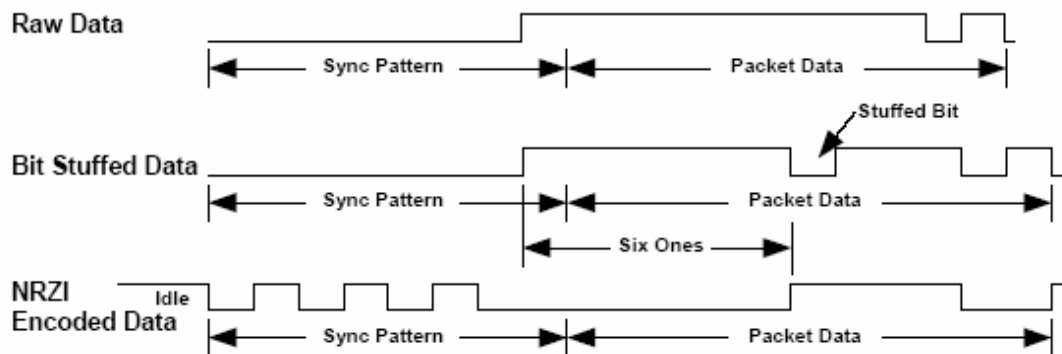
A USB implementa a codificação de dados do tipo NRZI para transmissão de dados. Na codificação do NRZI, o “1” é representado por não haver troca do nível enquanto o “0” representa uma troca. Uma string de zeros causa no NRZI uma troca de bit a cada tempo. Uma string de uns causa um período de inatividade no barramento.



### Bit de Stuffing

Para garantir sincronismo na rede, após a transmissão de 6 uns, o NRZI impõe um “0” para garantir a transição na linha e assegurar o sincronismo na comunicação.

#### Data Encoding Sequence:



### Velocidades de comunicação

As velocidades de comunicação disponíveis para o USB são as seguintes:

- High Speed – 480 Mbps podendo variar +-0,05%
- Full-Speed – 12 Mbps podendo variar +-0,25%
- Low-Speed – 1,5 Mbps podendo variar 1,5 %

### Atenuação do Cabo

Os cabos USB devem atender as seguintes exigências sobre o cabo para poderem operar normalmente:

Frequency (MHz)	Attenuation (maximum) dB/cable
0.064	0.08
0.256	0.11
0.512	0.13
0.772	0.15
1.000	0.20
4.000	0.39
8.000	0.57
12.000	0.67
24.000	0.95
48.000	1.35
96.000	1.9
200.00	3.2
400.00	5.8

### ***Distribuição de energia***

#### ***Classes de dispositivos***


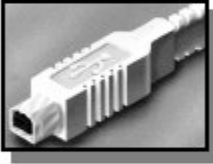


A potência exigida por cada dispositivo pode ser simplificada com a introdução do conceito de unidade de carga. A unidade de carga é definida por uma corrente de 100 mA. O número de unidade de carga de um dispositivo pode consumir é um valor absoluto independente do tempo. Um dispositivo pode ser low-power consumindo uma unidade de carga e high-power consumindo 5 unidades de carga. Por default, todos os dispositivos são low-power. A transição para high-power é feita através de controle de software. É responsabilidade do software assegurar a potência adequada para o funcionamento do dispositivo.

O USB suporta uma faixa de fontes de energia que pode ser observado abaixo:

- **Bus-powered hubs:** Estas unidades somente podem consumir uma unidade de carga na configuração e 5 após a mesma.
- **Self-powered hubs:** A energia para o funcionamento não são provenientes do Vbus do barramento.

- **Low-power bus-powered functions:** Toda a energia para o funcionamento do dispositivo vem do Vbus. Este dispositivo pode consumir no máximo uma unidade de carga.
- **High-power bus-powered functions:** Toda a energia para o funcionamento do dispositivo vem do Vbus. Este dispositivo pode consumir no máximo uma cinco unidades de carga.
- **Self-powered functions:** Pode consumir uma unidade de carga do Vbus. O restante é fornecido através de uma fonte externa.

### Conectores disponíveis

Series "A" Connectors	Series "B" Connectors
<p>♦ Series "A" plugs are always oriented <b>upstream</b> towards the <i>Host System</i></p>  <p>"A" Plugs (From the USB Device)</p>	<p>♦ Series "B" plugs are always oriented <b>downstream</b> towards the USB Device</p>  <p>"B" Plugs (From the Host System)</p>
<p>"A" Receptacles (Downstream Output from the USB Host or Hub)</p> 	<p>"B" Receptacles (Upstream Input to the USB Device or Hub)</p> 

### Camada de Protocolo

#### Transferência dos Bits

*Todos os bits são transmitidos primeiramente pelo bit LSB.*

### ***Campo de sincronismo***

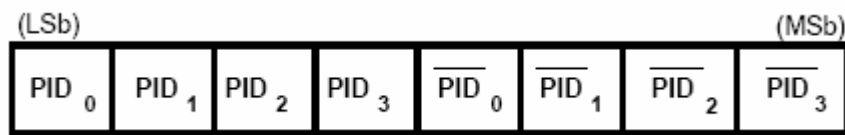
Todos os pacotes são inicializados pelo campo de SYNC. A função deste campo é que o circuito de entrada do receptor ajuste o seu clock com o do transmissor. No modo full / low-speed são gerados 8 bits enquanto que no high-speed 32.

### ***Formato dos Pacotes***

Todos os pacotes iniciam com o campo de Start e finalizam com o campo de End. O Start-of-Packet (SOP) é uma parte do campo de SYNC e o End-of-Packet (EOP) sinaliza o fim do pacote.

### ***Campo de Identificador de Pacote***

O campo de identificação do pacote (PID) segue imediatamente após o campo de SYNC. Um PID consiste de quatro pacotes seguidos de quatro bits de checagem de campo, observe:

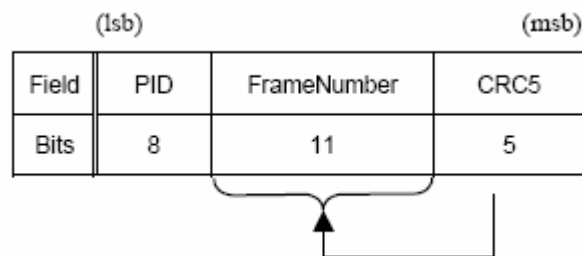
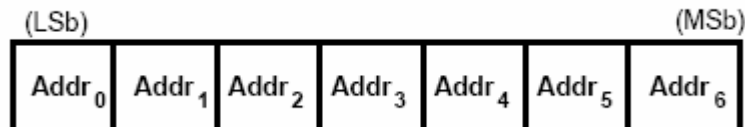


Os seguintes PIDs estão disponíveis:

PID Type	PID Name	PID<3:0>*	Description
Token	OUT	0001B	Address + endpoint number in host-to-function transaction
	IN	1001B	Address + endpoint number in function-to-host transaction
	SOF	0101B	Start-of-Frame marker and frame number
	SETUP	1101B	Address + endpoint number in host-to-function transaction for SETUP to a control pipe
Data	DATA0	0011B	Data packet PID even
	DATA1	1011B	Data packet PID odd
	DATA2	0111B	Data packet PID high-speed, high bandwidth isochronous transaction in a microframe (see Section 5.9.2 for more information)
	MDATA	1111B	Data packet PID high-speed for split and high bandwidth isochronous transactions (see Sections 5.9.2, 11.20, and 11.21 for more information)
Handshake	ACK	0010B	Receiver accepts error-free data packet
	NAK	1010B	Receiving device cannot accept data or transmitting device cannot send data
	STALL	1110B	Endpoint is halted or a control pipe request is not supported
	NYET	0110B	No response yet from receiver (see Sections 8.5.1 and 11.17-11.21)
Special	PRE	1100B	(Token) Host-issued preamble. Enables downstream bus traffic to low-speed devices.
	ERR	1100B	(Handshake) Split Transaction Error Handshake (reuses PRE value)
	SPLIT	1000B	(Token) High-speed Split Transaction Token (see Section 8.4.2)
	PING	0100B	(Token) High-speed flow control probe for a bulk/control endpoint (see Section 8.5.1)
	Reserved	0000B	Reserved PID

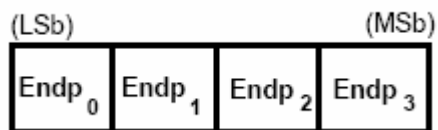
### ***Campo de Endereço***

A função deste campo é especificar o dispositivo em uma rede. Assim que o device é energizado, ele recebe o endereço 0 e aguarda o Host Controller endereçar o mesmo. O endereço 0 não pode ser usado em função disto.



### ***Campo de Endpoint***

Este campo é usado para permitir mais flexibilidade no endereçamento nos momentos em que há as transações de IN, OUT e SETUP. Full-speed e High-Speed suportam até 16 endpoints enquanto o low-speed suportam no máximo 3.



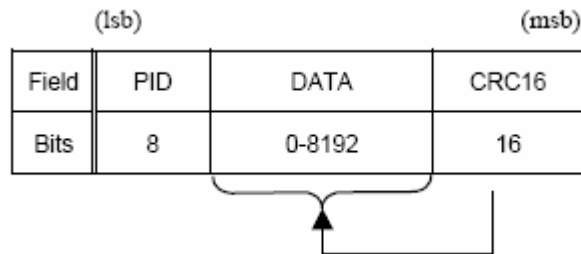
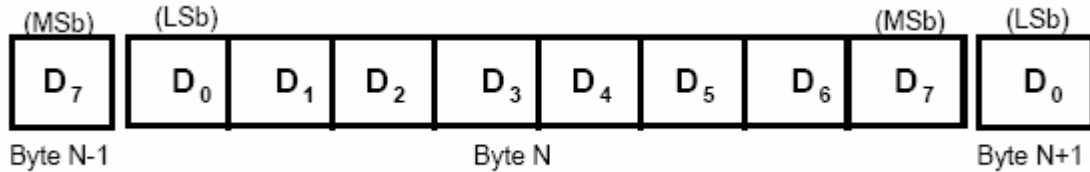
### ***Frame de Número de Campo***

É uma sequência de 11 bits que é incrementado pelo host a cada frame trafegado pela rede. Quando o valor máximo é atingido (0x7FFF) este valor retorna a 0.



## ***Campo de Dados***

O campo de dados pode variar na faixa de 0 a 1024 bytes e tem um número integral de bytes. Abaixo está apresentado o formato da transmissão dos bytes:



## ***Campo de CRC (Cyclic Redundant Checks)***

É usado para garantir a integridade na comunicação de dados. Desta forma, podem se garantir que o pacote transferido pelo Host será recebido pelo device.

## ***USB Device Framework***

### ***Alimentação***

É o momento no qual o dispositivo é alimentado. O mesmo pode ser alimentado através de fonte própria (self-powered) ou através do próprio barramento (bus-powered).

## Default

Após o dispositivo ser energizado, o mesmo não pode responder a qualquer transação do barramento até que o mesmo seja resetado pelo barramento. Após a condição de reset, o dispositivo é endereçado para o seu endereço default (0).

Quando o processo de reset está completo, o dispositivo USB opera de acordo com a sua velocidade (low/full/high). A velocidade é selecionada através da terminação de resistores do dispositivo, conforme apresentado abaixo:

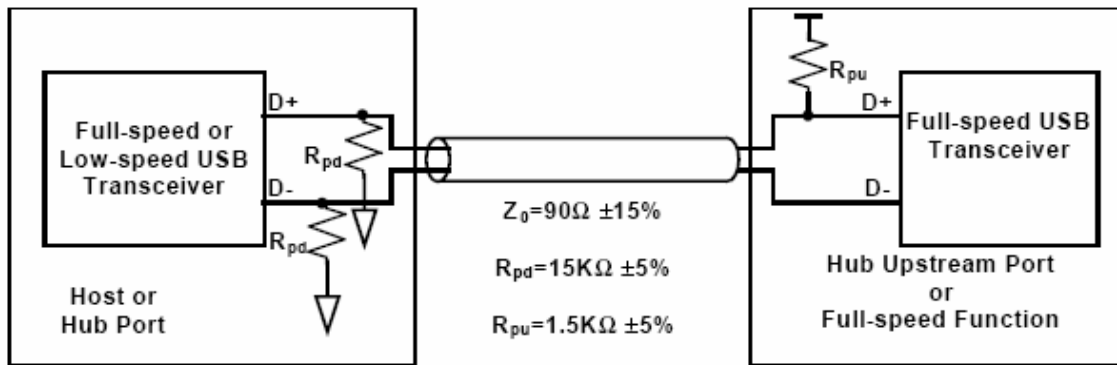
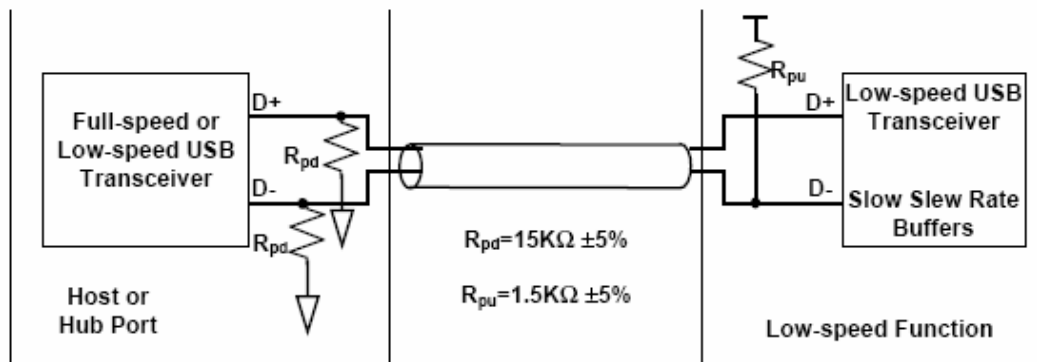


Figure 7-20. Full-speed Device Cable and Resistor Connections



Para o modo High-Speed, o mesmo inicia operando em Full-Speed e através de protocolo de software, altera para o modo High-Speed.

## Endereçamento

Todos os dispositivos USB tem um endereço único após a condição de reset. Cada device recebe o seu endereço pelo host controller. O dispositivo mantém este endereço até que o mesmo entre em condição de suspensão.

## ***Configuração***

Este é o estado em que o dispositivo é configurado em algum modo de comunicação.

## ***Suspensão***

Para conservar energia, o dispositivo USB pode automaticamente entrar neste estado quando o dispositivo detectar que não há tráfego na linha por um determinado período de tempo.

## ***Enumeração***

Quando um dispositivo USB é colocado ou removido da linha, o host usa o processo de enumeração para identificar e gerenciar o estado do dispositivo.

## ***Modos de Comunicação***

### ***Control Transfers***

Controle de dados é usado pelo sistema de software USB para configurar os dispositivos quando os mesmos são conectados ao host. Outros drivers de software podem usar este modo para implementações específicas.

### ***Bulk Transfers***

Consiste no envio de dados em massa, comumente utilizado para impressoras e scanners. Esta transferência é seqüencial.

### ***Interrupt Transfers***

Tem uma latência limitada de comunicação entre o dispositivo e o host por um determinado tempo síncrono.

### ***Isochronous Transfers***

É uma transferência contínua em tempo real. É muito usado para aplicações que envolvam voz por exemplo.