Interfaz USB

Transmisión de Datos

Ing. Luis Di Pinto (ldipinto@herrera.unt.edu.ar)

http://www.microprocesadores.unt.edu.ar/transmision/

Repaso Esquema General (2da parte)

- En esta segunda parte de la materia, vamos a enfocarnos en un conjunto de interfaces seriales.
  - Algunas de las más usadas y populares.



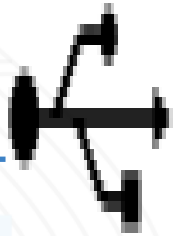
Temas que veremos

- Características generales y versiones.
- Topologías y componentes.
- Elementos de una comunicación.
  - Transferencias.
  - Transacciones.
  - Paquetes.
- Tipos de descriptores.
- Clases de dispositivos.
- Cables y conectores.
- USB Power Delivery.
- USB On The Go.

Introducción

- **Universal Serial Bus**
- Define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer alimentación eléctrica entre computadoras y periféricos.
- A diferencia de otras interfaces seriales, USB define varias capas de protocolo.

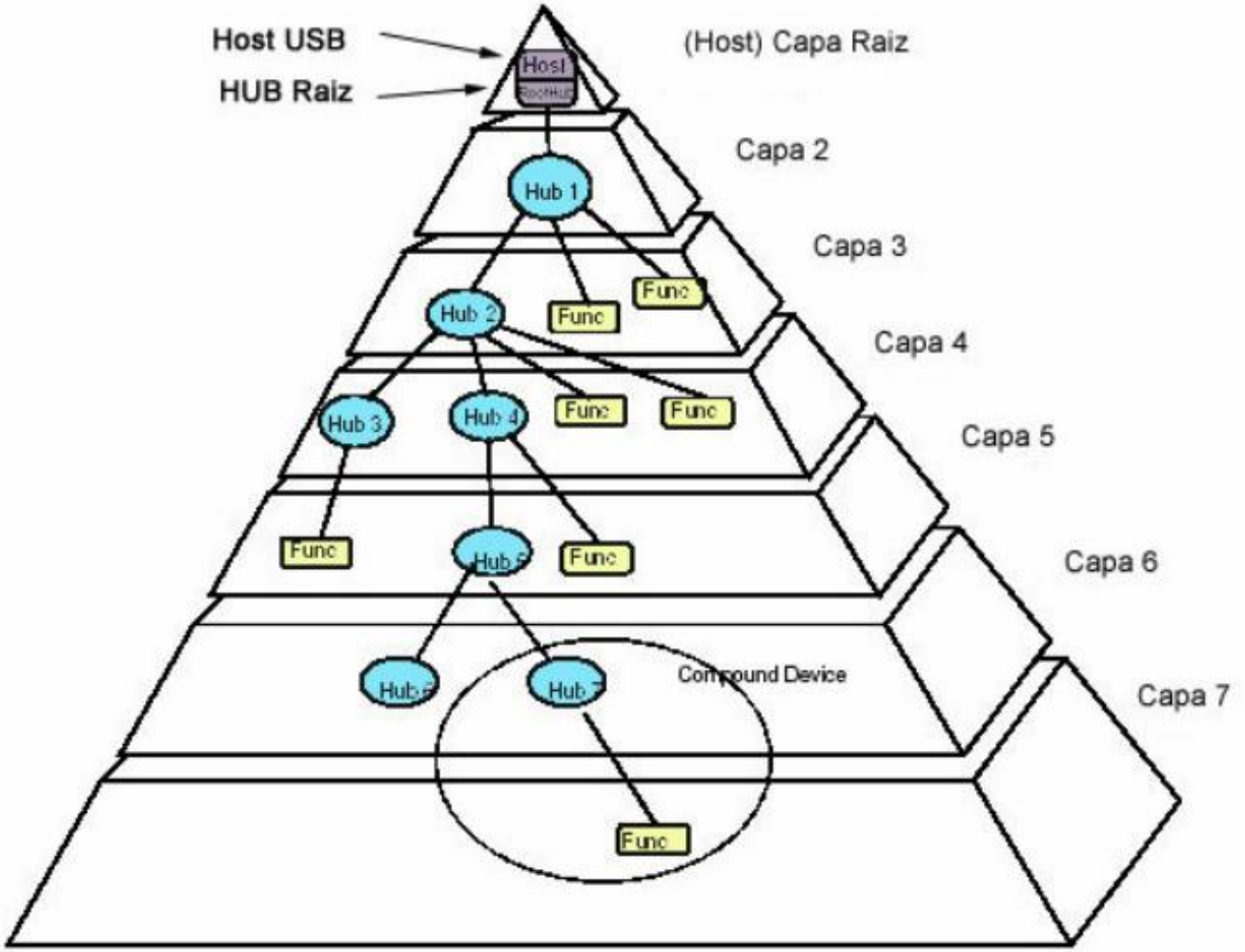
Versiones y velocidades

Rate Name	Old Name	Standard	Encoding	Data pairs	Rate	Logo
Low-Speed		USB 1.0			1.5 Mbit/s	
Full-Speed			NRZI	1 HDx	12 Mbit/s	
High-Speed		USB 2.0			480 Mbit/s	
USB 3.2 Gen 1×1	USB 3.0; USB 3.1 Gen 1	USB 3.0	8b/10b	2 FDx	5 Gbit/s	
USB 3.2 Gen 2×1	USB 3.1 Gen 2	USB 3.1	128b/132b	2 FDx	10 Gbit/s	
USB 3.2 Gen 1×2		USB 3.2	8b/10b	4 FDx ×2	10 Gbit/s	—
USB 3.2 Gen 2×2			128b/132b	4 FDx ×2	20 Gbit/s	
USB4 Gen 2×1		USB4	64b/66b[a]	2 FDx	10 Gbit/s	—
USB4 Gen 2×2			64b/66b[a]	4 FDx ×2	20 Gbit/s	
USB4 Gen 3×1			128b/132b[a]	2 FDx	20 Gbit/s	—
USB4 Gen 3×2			128b/132b[a]	4 FDx ×2	40 Gbit/s	

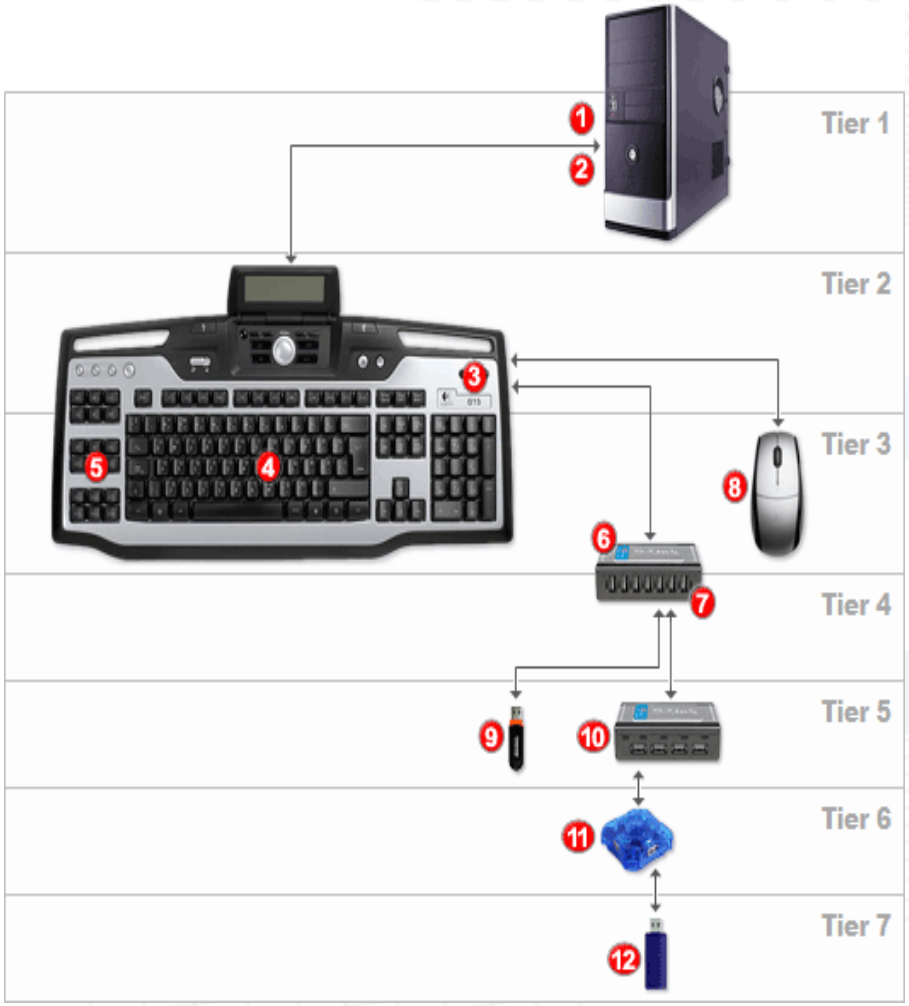
Arquitectura General

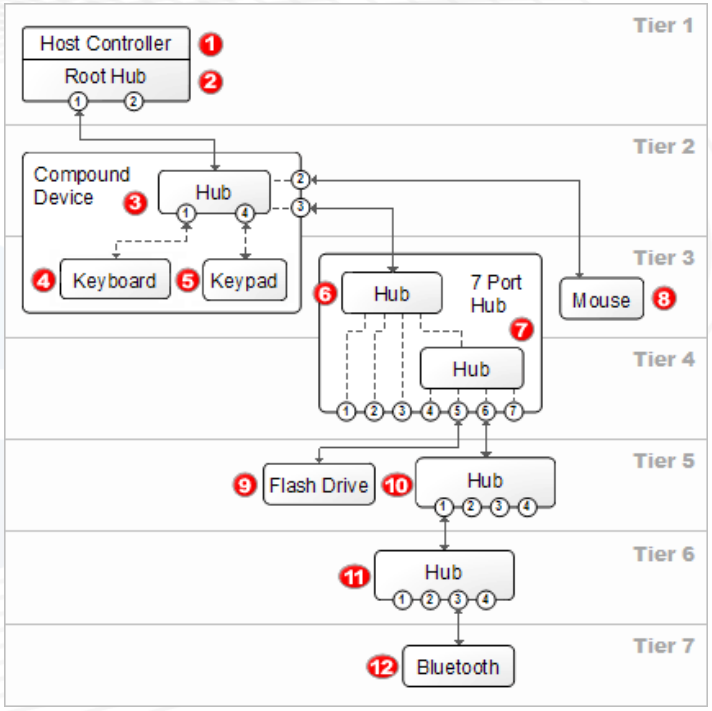
- Tres componentes:
  - Un controlador.
  - Hubs o concentradores.
  - Dispositivos periféricos.
- La interconexión USB incluye:
  - Topología de bus.
  - Relaciones entre capas.
  - Modelos de flujo de datos.

Topología del bus USB

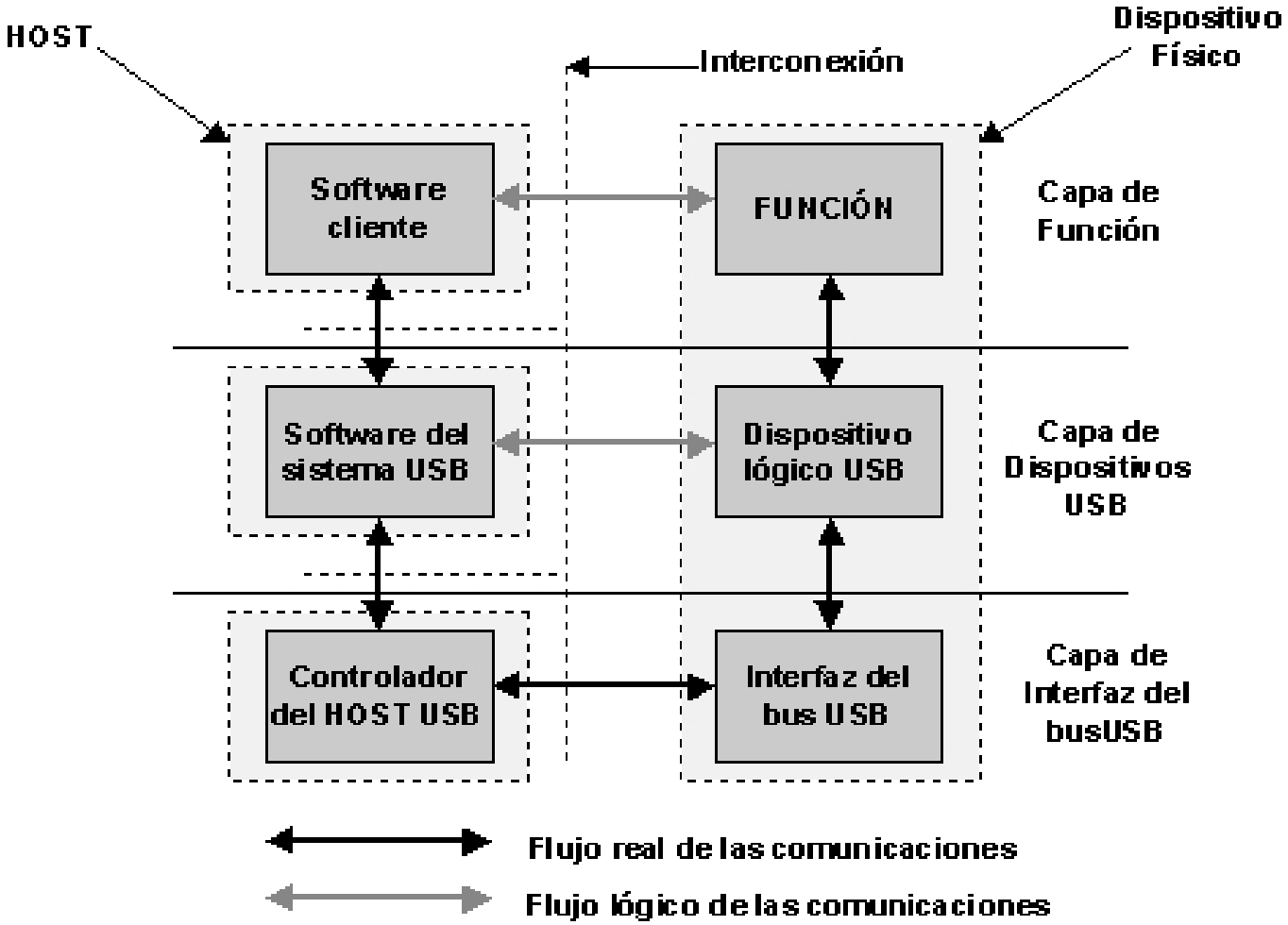


Topología del bus USB





Topología Física y Lógica



Controlador USB

- Tiene la misión de hacer de interfaz entre el micro y los diferentes dispositivos.
- Contiene el *root-hub*.
- Tienen sus propias especificaciones.
- Actualmente se encuentran tarjetas multipuerto con dos o más controladores permitiendo canales individuales.
- Su implementación es una combinación de hardware y software.

Controlador USB - Hardware

- Comunicaciones entre los periféricos USB y la CPU.
- Admisión de los periféricos dentro del bus.
- Comunicar errores.
- Asignar recursos necesarios.
- Control de flujo de datos.
- Administrar y regular los flujos de control.
- Recolectar estadísticas.
- Proveer energía eléctrica.

Controlador USB - Software

- Enumeración y configuración de los dispositivos conectados al sistema.
- Administración y control de transferencias sincrónicas y asincrónicas de información.
- Administración avanzada de suministro eléctrico a los diferentes dispositivos.
- Administración de la información del bus y los dispositivos USB.

Host Controller Interface

- USB 1.0
  - UHCI (Universal Host Controller Interface)
- USB 1.1
  - OHCI (Open Host Controller Interface)
- USB 2.0
  - EHCI (Enhanced Host Controller Interface)
- USB 3.0
  - XHCI (eXtensible Host Controller Interface)

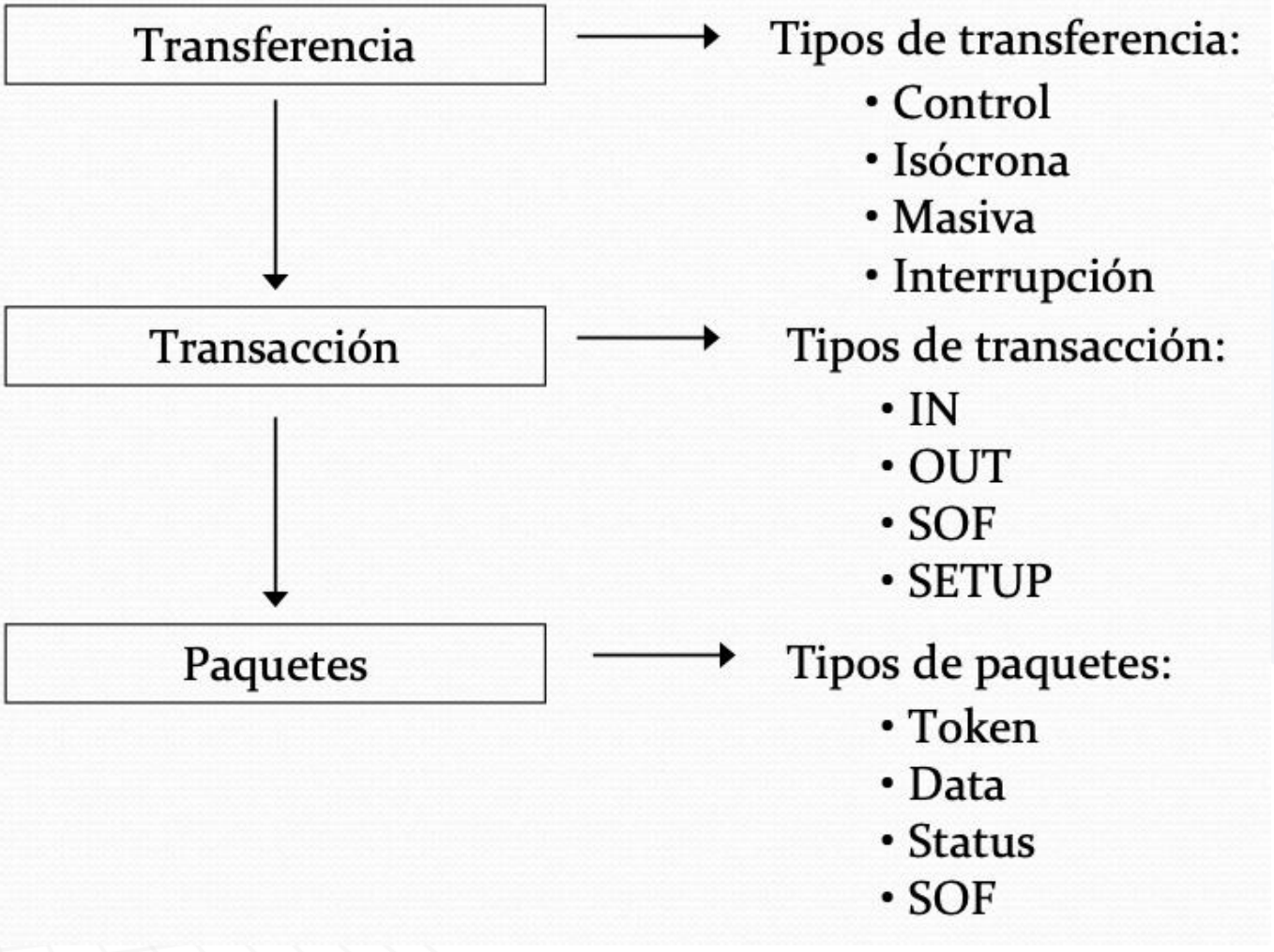
Funciones de los dispositivos USB

- Un dispositivo USB puede proveer múltiples funciones simultáneamente (*composite device*).
  - En este caso el dispositivo tiene una sola dirección.
- Un dispositivo USB puede también incluir un hub interno y estar formado por múltiples dispositivos (*compound device*)
  - En este caso el dispositivo tiene una dirección por cada dispositivo USB independiente.

USB ON-THE-GO (OTG)

- La extensión On-The-Go permite que los dispositivos puedan actuar como host y controlar a otros dispositivos.
- Define las funciones OTG A (proveedor de energía) y OTG B (consumidor de energía).
- Al inicio, el OTG A actúa como host y el OTG B como dispositivo, pero se pueden invertir las funciones usando el protocolo HNP (*Host Negotiation Protocol*).

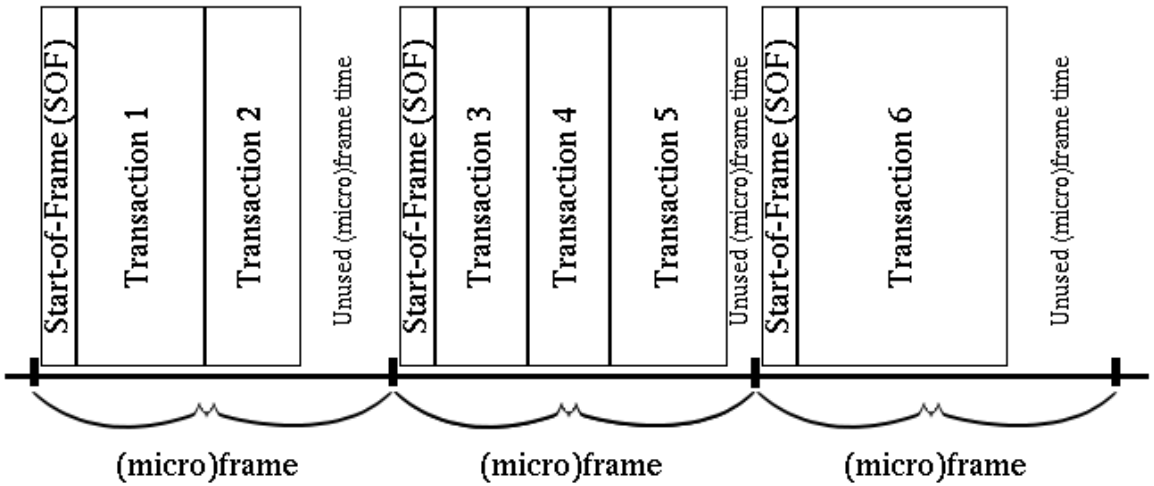
Comunicación



Nivel de Transferencias

- Este nivel configura los parámetros asociados con:
  - La dirección del flujo de datos.
  - El ancho de banda que se va a utilizar.
  - El tipo de transferencia se va a utilizar.
  - El tamaño máximo y/o mínimo de los paquetes/buffers.

Organización de las transferencias

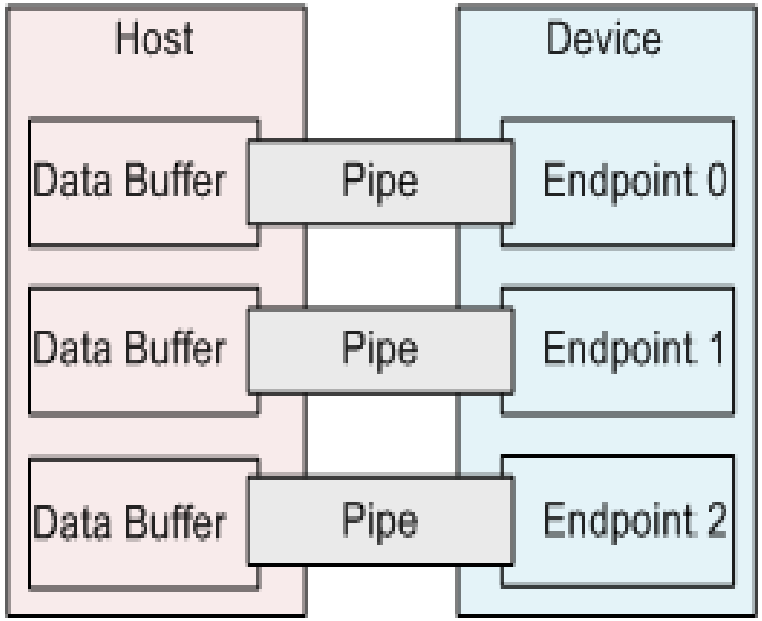


- El tráfico en el bus USB es regulado utilizando el tiempo.
- La unidad de tiempo se llama ***frame***.
  - Velocidad “Full” y “Low”: *frames* cada 1 ms.
  - Velocidad “High”: *micro-frames* cada 125 µs.
- Cada *frame* comienza con un paquete SOF.
- Cada transferencia consiste en una o más transacciones.

Elementos de una transferencia

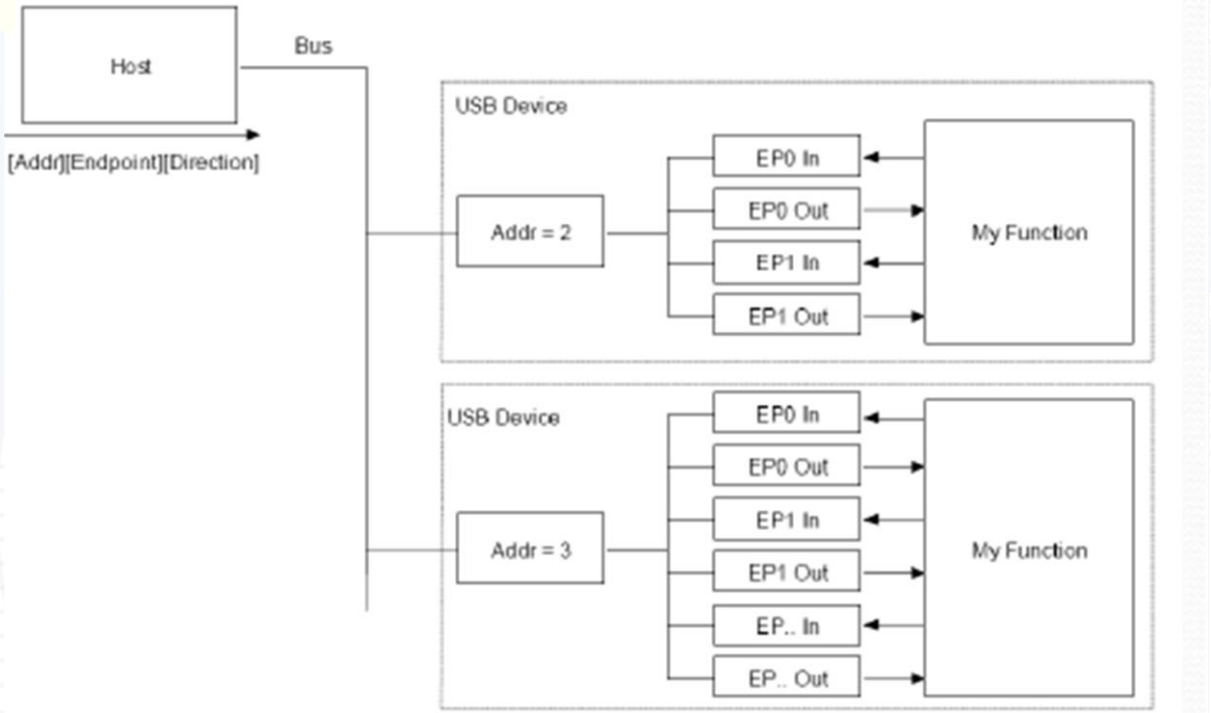
- ***Endpoints***: Es una porción de memoria singularmente identificable de una función de USB la cual provee la terminación del flujo de comunicación entre el Host y la función.

- ***Pipes***: Es un enlace virtual entre el host (la PC) y el dispositivo USB.



Funciones de los dispositivos USB

- Cada una de las funciones puede requerir diferentes tipos de transferencias:



Tipos de transferencias

- **Control:** Son comunicaciones por interrupción, no periódicas, iniciadas por el host, que se utilizan en operaciones de comando o status.
- **Masiva (*Bulk*):** Situaciones donde la latencia de la transferencia no es crítica.
- **Interrupción:** Son comunicaciones de baja frecuencia, para tamaños de paquete de datos muy pequeños, y tiempo de recuperación de datos limitado.
- **Isocrónica:** Se trata de un tipo de comunicación periódica y continua.
  - Tienen un tiempo de envío garantizado, pero no poseen control de errores.

Tipos de transferencias

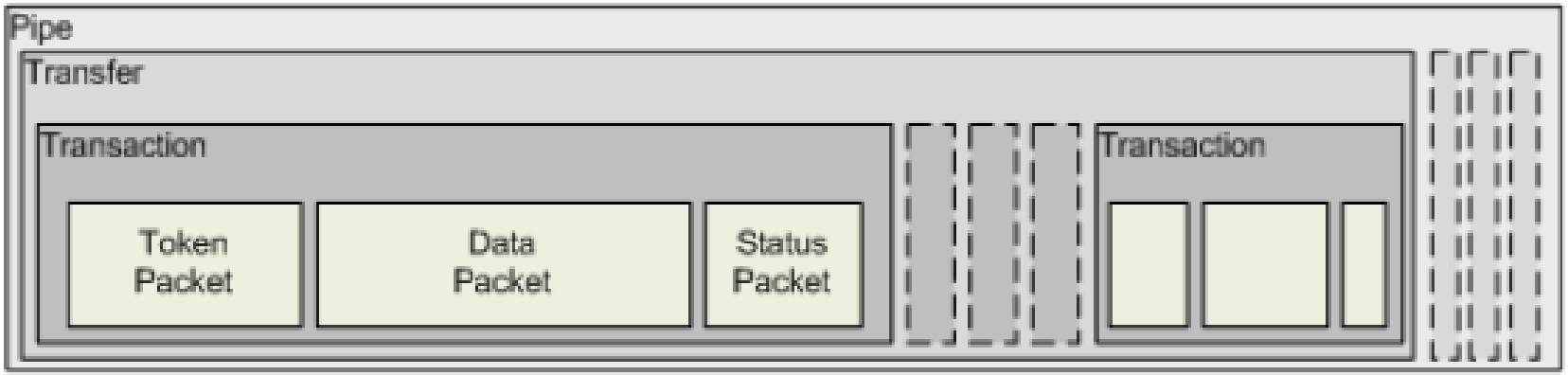
Uso típico	Configuración	Impresora	Teclado	Audio
Obligatoria	Si	No	No	No
Soportada en Low Speed	Si	No	Si	No
Corrección de errores	Si	Si	Si	No
Tipo de pipe	Message	Stream	Stream	Stream
Garantiza Velocidad de Envío	No	No	Si	Si
Garantiza Minima Latencia	No	No	Si	Si
Tamaños del Endpoint (Low Speed)	8 bytes	No aplica	8 bytes	No aplica
Tamaños del Endpoint (Full Speed)	8, 16, 32 o 64 bytes	8, 16, 32 o 64 bytes	1 a 64 Bytes	Hasta 1023 Bytes
Ancho de banda reservado por frame	10 %	Ninguno	90% (ambas combinadas)	

Transacciones

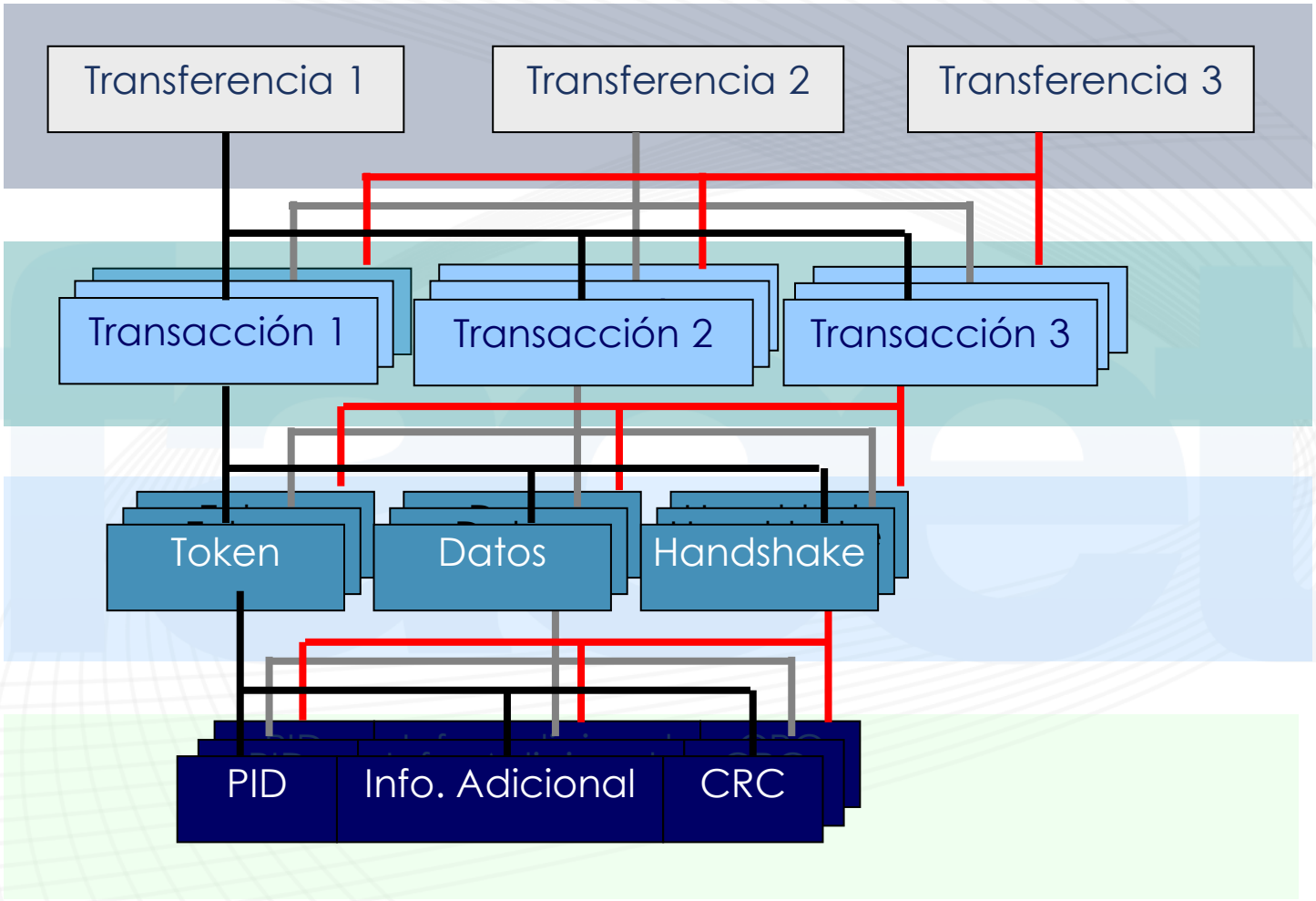
- Se define como la entrega de servicios (información) a un *endpoint*.
- Existen 3 tipos de transacciones:
  - **SETUP**: envían pedidos de control a un dispositivo.
  - **IN**: envían al host datos o información de estado desde un dispositivo.
  - **OUT**: envían a un dispositivo datos o información de estado desde el host.

Fases de una transacción

- Cada transacción USB consiste en:
  - Paquete Token (encabezado que define lo que se espera a continuación).
  - Paquete de datos (opcional - contiene el payload).
  - Paquete de Status (Usado como ACK en las transacciones y como una forma de corregir errores).



Transacciones



Campos de los paquetes USB

- SYNC (*Synchronization flags*)
  - Todos los paquetes deben comenzar con un campo de “sync” utilizado para sincronizar el clock receptor con el transmisor.
- PID (*Packet ID*)
  - Utilizado para identificar el tipo de paquete que está siendo enviado (4 bits complementados).
- ADDR (*Address field*)
  - Especifica a qué dispositivo va dirigido el paquete.
  - Tiene un tamaño de 7 bits y soporta 127 dispositivos.
  - La dirección cero no es válida ya que cualquier dispositivo al que todavía no se le ha asignado una dirección debe responder los paquetes enviados a ésta.

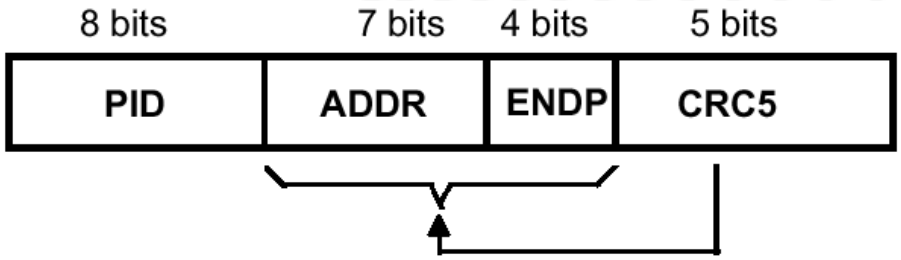
Valores del campo PID

Type	PID value	Transmitted	Name	Description
Token	1000	0001 1110	**SPLIT**	High-bandwidth (USB 2.0) split transaction
	0100	0010 1101	**PING**	Check if endpoint can accept data (USB 2.0)
Special	1100	0011 1100	**PRE**	Low-bandwidth USB preamble
			ERR	Split transaction error (USB 2.0)
Handshake	0010	0100 1011	**ACK**	Data packet accepted
	1010	0101 1010	**NAK**	Data packet not accepted; please retransmit
	0110	0110 1001	**NYET**	Data not ready yet (USB 2.0)
	1110	0111 1000	**STALL**	Transfer impossible; do error recovery
Token	0001	1000 0111	**OUT**	Address for host-to-device transfer
	1001	1001 0110	**IN**	Address for device-to-host transfer
	0101	1010 0101	**SOF**	Start of frame marker (sent each ms)
	1101	1011 0100	**SETUP**	Address for host-to-device control transfer
Data	0011	1100 0011	**DATA0**	Even-numbered data packet
	1011	1101 0010	**DATA1**	Odd-numbered data packet
	0111	1110 0001	**DATA2**	Data packet for high-bandwidth isochronous transfer (USB 2.0)
	1111	1111 0000	**MDATA**	Data packet for high-bandwidth isochronous transfer (USB 2.0)

Campos de los paquetes USB

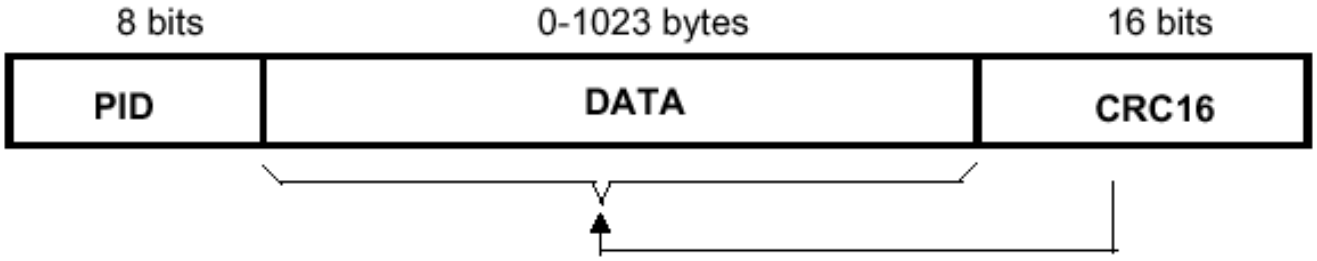
- ENDP (*End point field*)
  - Formado por 4 bits, permite 16 posibles "*endpoints*".
- CRC (*Cyclic Redundancy Check*)
  - Efectuado en los datos contenidos en el "*payload*" del paquete.
  - Todos los paquetes *token* tienen un CRC de 5 bits mientras que los de datos tienen un CRC de 16 bits.
- EOP (*End of packet*)
  - Señalizado a través de un "Single Ended Zero" (SE0 / D+ and D- se mantienen bajos) por aproximadamente el tiempo de dos bits, seguido por una J (estado lógico, el significado depende de la velocidad) durante el tiempo de un bit.

Tipos de paquetes USB



- **IN**: Informa al dispositivo que el host desea leer información.
- **OUT**: Informa al dispositivo que el host desea enviar información.
- **SETUP**: Utilizado para comenzar transferencias de control.

Tipos de paquetes USB



- Paquete de datos
  - Dos tipos. Cada uno capaz de transmitir de 0 a 1023 bytes de datos.
- Paquetes de comienzo de *frame* (SOF)
  - El número de frame (11 bits) es enviado por el host cada 1 ms ± 500 ns.

Tipos de paquetes USB

- Paquetes de “Status” o “Handshake”
  - ACK (*Acknowledgment*): Confirmación de que el paquete fue recibido exitosamente
  - NAK: Reporta que el dispositivo no puede enviar ni recibir datos temporalmente. También utilizado durante las transacciones de interrupción para informar al “host” que no hay datos para enviar
  - STALL: Puede significar un “control request” no soportado, una falla en el “control request” o que el endpoint falla.

Enumeración

- Antes de comenzar a trabajar con un dispositivo el host debe averiguar sus características (Tipos de transferencias, cantidad de *endpoints*, etc.).
- La **enumeración** es el intercambio de información que acompaña a estas tareas.
- La información se intercambia por descriptores normalizados.

Enumeración

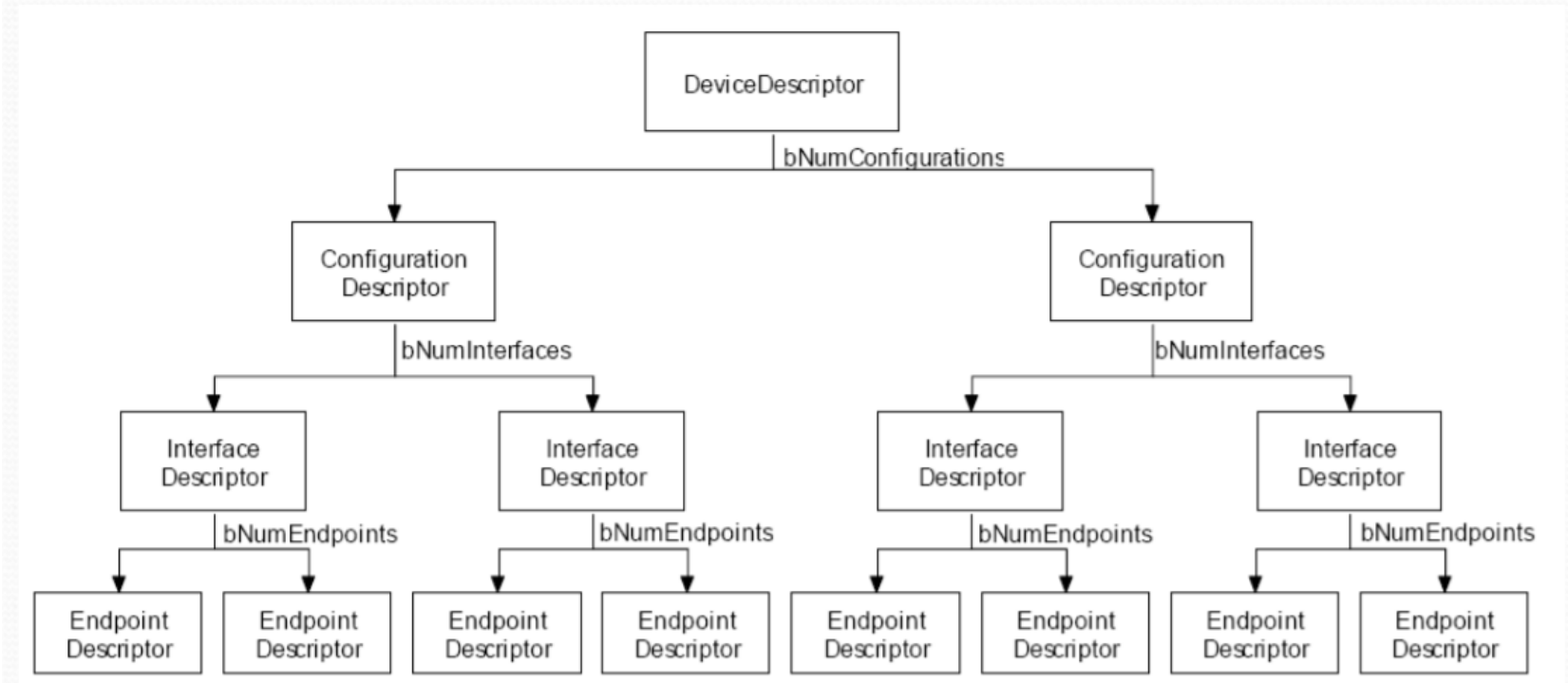
- El usuario conecta un dispositivo a un puerto USB.
- El hub detecta eléctricamente el dispositivo.
- El hub detecta cuando un dispositivo es low o full speed.
- El hub resetea el dispositivo poniéndolo en estado default (Addr=0).
- El host envía un pedido **`Get_Descriptor`** para aprender el tamaño del *endpoint* por defecto (8 primeros bytes del device descriptor).
- El host resetea nuevamente el dispositivo.
- El host asigna una dirección en el bus con el comando Set Address.
- El host solicita el Device Descriptor completo (18 bytes).
- El host aprende acerca de las capacidades del dispositivo.
- El host asigna y carga un controlador de dispositivo.
- El controlador de dispositivo del host selecciona una configuración.

Descriptores

- Todos los dispositivos USB tienen una jerarquía de descriptores que informan al host información tal como:
  - qué es el dispositivo,
  - quién lo fabricó,
  - qué versión de USB soporta,
  - de cuántas formas puede configurarse,
  - el número de endpoints y sus tipos

- Los descriptores USB más comunes son:
  - Descriptores de dispositivo (*Device descriptors*).
  - Descriptores de configuración (*Configuration Descriptors*).
  - Descriptores de interfaz (*Interface Descriptors*).
  - Descriptores de endpoint (*Endpoint Descriptors*).
  - Descriptores de cadenas (*String Descriptors*) (opcionales).

Descriptores



Descriptor de dispositivo

- Contiene información básica acerca del dispositivo.
- El host solicita un device descriptor enviando un pedido de Get_Descriptor.
- Un dispositivo puede tener un único device descriptor ya que representa la totalidad del dispositivo.
- Estos descriptores proveen información general como el fabricante, número de producto, número de serie, la clase de dispositivo y el número de configuraciones.

Descriptor de dispositivo

Offset (decimal)	Field	Size (bytes)	Description
0	bLength	1	Descriptor size in bytes (12h)
1	bDescriptorType	1	The constant DEVICE (01h)
2	bcdUSB	2	USB specification release number (BCD)
4	bDeviceClass	1	Class code
5	bDeviceSubclass	1	Subclass code
6	bDeviceProtocol	1	Protocol Code
7	bMaxPacketSize0	1	Maximum packet size for endpoint zero
8	idVendor	2	Vendor ID
10	idProduct	2	Product ID
12	bcdDevice	2	Device release number (BCD)
14	iManufacturer	1	Index of string descriptor for the manufacturer
15	iProduct	1	Index of string descriptor for the product
16	iSerialNumber	1	Index of string descriptor for the serial number
17	bNumConfigurations	1	Number of possible configurations

Configuraciones

- Una configuración específica:
  - Cómo se alimenta el dispositivo.
  - Cuál es el máximo consumo de potencia.
  - El número de interfaces que tiene.
- Es posible tener dos configuraciones: una para el dispositivo siendo alimentado por el bus y otra cuando lo hace externamente.
- Como este es el “encabezado” de los descriptores de interfaz, es también posible tener diferentes modos de transferencia para cada una de las configuraciones.
- La mayoría de los dispositivos son simples y solamente tienen una configuración.

Descriptor de configuración

Offset (decimal)	Field	Size (bytes)	Description
0	bLength	1	Descriptor size in bytes (09h)
1	bDescriptorType	1	The constant CONFIGURATION (02h)
2	wTotalLength	2	The number of bytes in the configuration descriptor and all of its subordinate descriptors
4	bNumInterfaces	1	Number of interfaces in the configuration
5	bConfigurationValue	1	Identifier for Set Configuration and Get Configuration requests
6	iConfiguration	1	Index of string descriptor for the configuration
7	bmAttributes	1	Self/bus power and remote wakeup settings
8	bMaxPower	1	Bus power required in units of 2 mA (USB 2.0) or 8 mA (SuperSpeed).

Descriptor de interfaz

- Una interfaz puede verse como una agrupación de *endpoints* que operan juntos para una determinada función del dispositivo.
- Cada función del dispositivo necesita por lo menos una interfaz, pero una función puede requerir más de una interfaz.
- El descriptor de interfaz detalla el número de *endpoints* usados en la interfaz, el tipo de interfaz, y la clase de dispositivo.

Descriptor de interfaz

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of Descriptor in Bytes
1	bDescriptorType	1	Constant	Interface Descriptor (0x04)
2	bInterfaceNumber	1	Number	Number of Interface
3	bAlternateSetting	1	Number	Value used to select alternative setting
4	bNumEndpoints	1	Number	Number of Endpoints used for this interface
5	bInterfaceClass	1	Class	Class Code (Assigned by USB Org)
6	bInterfaceSubClass	1	SubClass	Subclass Code (Assigned by USB Org)
7	bInterfaceProtocol	1	Protocol	Protocol Code
8	iInterface	1	Index	Index of String Descriptor Describing this interface

Descriptor de Endpoint

- Identifican el tipo de transferencia y su sentido, como otros datos específicos de un *endpoint*.
- El host utilizará la información devuelta por estos descriptores para definir los requerimientos de ancho de banda del bus.
- El *endpoint* cero siempre se asume como de control.

Descriptor de Endpoint

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of Descriptor in Bytes (7 bytes)
1	bDescriptionType	1	Constant	Endpoint Descriptor (0x05)
2	bEndpointAddress	1	Endpoint	Endpoint Address, Encoded as follows 0..3b Endpoint Number 4..6b Reserved. Set to Zero 7b Direction (Ignored for Control Endpoints) 0 = Out Endpoint, 1 = In Endpoint
3	bmAttributes	1	Bitmap	Bits 0..1 Transfer Type 00 = Control 01 = Isochronous 10 = Bulk 11 = Interrupt Bits 2..7 are reserved. If Isochronous endpoint, Bits 3..2 = Synchronisation Type (Iso Mode) 00 = No Synchonisation 01 = Asynchronous 10 = Adaptive 11 = Synchronous Bits 5..4 = Usage Type (Iso Mode) 00 = Data Endpoint 01 = Feedback Endpoint 10 = Explicit Feedback Data Endpoint 11 = Reserved
4	wMaxPacketSize	2	Number	Maximum Packet Size this endpoint is capable of sending or receiving
6	bInterval	1	Number	Interval for polling endpoint data transfers. Value in frame counts. Ignored for Bulk & Control Endpoints. Iso must equal 1 and field may range from 1 to 255 for interrupt endpoints.

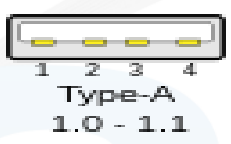
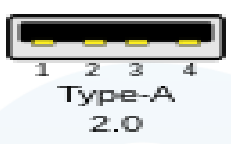
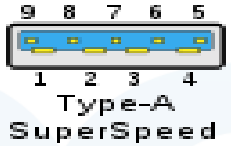
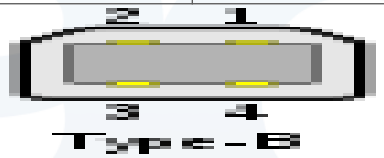
Clases de dispositivos

- Muchos dispositivos o interfaces comparten atributos o proveen servicios similares.
- El protocolo define un conjunto de clases de dispositivos genéricos que definen:
  - La cantidad y tipo de los *endpoints* requeridos u opcionales.
  - Puede definir valores para los ítems en los descriptores estándar, como también descriptores específicos, interfaces, usos de *endpoints* y pedidos de control.
- Cuando un dispositivo utiliza una clase estándar no se requieren controladores específicos.

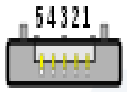
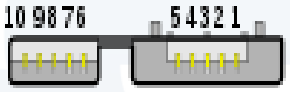
Clases de dispositivos

Class	Usage	Description	Examples, or exception
00h	Device	Unspecified	Device class is unspecified, interface descriptors are used to select drivers
01h	Interface	Audio	Speaker, microphone, sound card, MIDI
02h	Both	Communications and CDC control	UART and RS-232 serial adapter, Modem, Wi-Fi adapter, Ethernet adapter. Used together with class 0Ah *(CDC-Data*) below
03h	Interface	Human interface device (HID)	Keyboard, mouse, joystick
06h	Interface	Media (PTP/MTP)	Scanner, Camera
07h	Interface	Printer	Laser printer, inkjet printer, CNC machine
08h	Interface	USB mass storage, USB Attached SCSI	USB flash drive, memory card reader, digital audio player, digital camera, external drive
09h	Device	USB hub	High speed USB hub
0Ah	Interface	CDC-Data	Used together with class 02h (Communications and CDC Control) above
0Eh	Interface	Video	Webcam
10h	Interface	Audio/Video (AV)	Webcam, TV
11h	Device	Billboard	Describes USB-C alternate modes supported by device
FFh	Both	Vendor-specific	Indicates that a device needs vendor-specific drivers

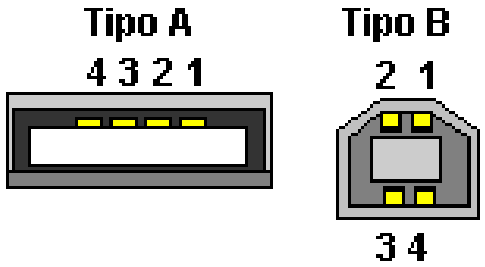
Cables y conectores

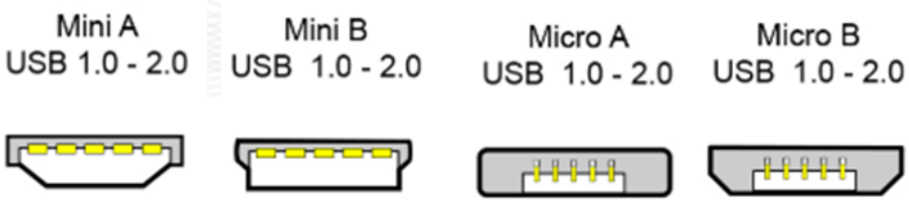
Standard	USB 1.0 1996	USB 1.1 1998	USB 2.0 2001	USB 2.0 Revised	USB 3.0 2008	USB 3.1 2013	USB 3.2 2017	USB4 2019
Max. transfer rate	12 Mbps		480 Mbps		5 Gbps	10 Gbps	20 Gbps	40 Gbps
Type A connector							Deprecated	
Type B connector							Deprecated	
Type C connector	Backwards compatibility only							
Mini-A connector	—				Deprecated			
Mini-B connector	—				Deprecated			

Cables y conectores

Standard	USB 1.0 1996	USB 1.1 1998	USB 2.0 2001	USB 2.0 Revised	USB 3.0 2008	USB 3.1 2013	USB 3.2 2017	USB4 2019
Max. transfer rate	12 Mbps		480 Mbps		5 Gbps	10 Gbps	20 Gbps	40 Gbps
Mini-AB connector	—				Deprecated			
Micro-A connector	—					Deprecated		
Micro-B connector	—					Deprecated		
Micro-AB connector	—					Deprecated		

Características eléctricas del conector



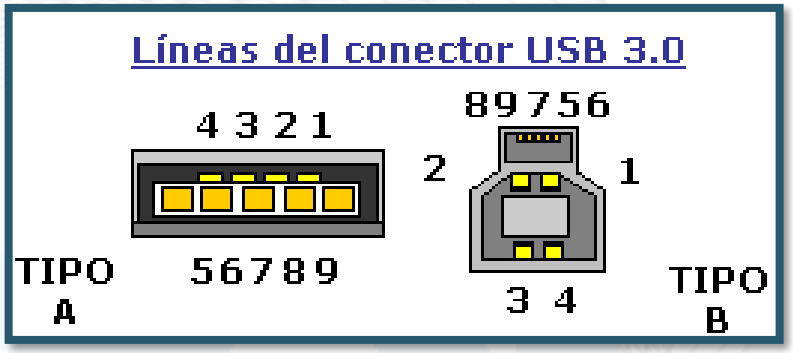


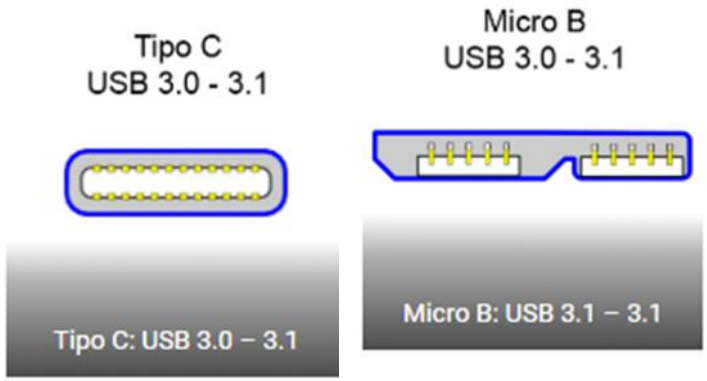
1) Vbus (+ 5 Volts)
2) D- (- datos)
3) D+ (+ datos)
4) GND (tierra)

1) Vbus (+ 5 Volts, alimentación)
2) D- (- datos)
3) D+ (+ datos)
4) ID: Distinción de Micro-A y Micro-B
   - Tipo A: conectado a masa
   - Tipo B: no conectado
5) GND (tierra)

Características eléctricas del conector

1) Vbus (+ 5 Volts, alimentación)
2) D- (- datos)
3) D+ (+ datos)
4) GND (tierra)
5) StdA_SSRX- (Recibe datos)
6) StdA_SSRX+ (Recibe datos)
7) GND_DRAIN (tierra-drenado)
8) StdA_SSTX- (Envía datos)
9) StdA_SSTX+ (Envía datos)





Consumos de corriente

- El estándar USB provee alimentación a algunos dispositivos desde su primera versión.
- En las primeras versiones los dispositivos se alimentan con 5V y pueden consumir 100 mA sin pedir autorización.
- El consumo de corriente se puede aumentar hasta 500 mA si el host lo autoriza.
- Después de USB 2.0 se agregó la posibilidad de consumir hasta 1,5 A.
  - Para mejorar la capacidad de cargar baterías utilizando el puerto USB.

Consumos de corriente

- Con el estándar USB 3 los dispositivos pueden consumir hasta 150 mA sin pedir autorización, y hasta 900 mA si el host lo autoriza.
- Posterior al estándar USB 4 se agrego la función de *Power Delivery*, lo que permite aumentar aun más las capacidades de carga de baterías usando un puerto USB.

USB Power Delivery

<table>
<tr><th colspan="2" rowspan="2">Source output power rating (W)</th><th colspan="7">Current, at: (A)</th></tr>
<tr><th>+5 V</th><th>+9 V</th><th>+15 V</th><th>+20 V</th><th>+28 V</th><th>+36 V</th><th>+48 V</th></tr>
<tr><td rowspan="5">Standard Power Range</td><td>0.5–15</td><td>0.1–3.0</td><td>—</td><td rowspan="2">—</td><td rowspan="3">—</td><td rowspan="4">—</td><td rowspan="5">—</td></tr>
<tr><td>15–27</td><td rowspan="8">3.0 (15 W)</td><td>1.67–3.0</td></tr>
<tr><td>27–45</td><td rowspan="7">3.0 (27 W)</td><td>1.8–3.0</td></tr>
<tr><td>45–60</td><td rowspan="6">3.0 (45 W)</td><td>2.25–3.0</td></tr>
<tr><td>60–100</td><td>3.0–5.0</td></tr>
<tr><td rowspan="3">Extended Power Range</td><td>100–140</td><td rowspan="3">3.0 (60 W), 5.0 (100 W)</td><td>3.57–5.0</td></tr>
<tr><td>140–180</td><td rowspan="2">5.0 (140 W)</td><td>3.89–5.0</td></tr>
<tr><td>180–240</td><td>5.0 (180 W)</td><td>3.75–5.0</td></tr>
</table>

Resumen

- Vimos solamente una parte de todo lo que es el estándar USB.
- Define múltiples versiones, con diferentes velocidades.
- Topología: un único host, múltiples dispositivos conectados jerárquicamente, utilizando hubs para expandir los puertos.
- Las transferencias están compuestas por transacciones, que a su vez están compuestas por paquetes.
  - Cada transferencia se realiza en un frame, de duración fija.
  - Cuatro tipos de transferencias: masivas, interrupciones, isocrónicas y de control.
  - Las transferencias se definen entre pipes, que vinculan endpoints que representan funciones de los dispositivos.

Resumen

- Tres tipos de transacciones: entrada, salida y configuración.
- Cuatro tipos de paquetes: SOF, Token, Datos y Status.
  - Formados por diversos campos.
- La inicialización se denomina enumeración, y consiste en el intercambio de diversos parámetros denominados descriptores.
  - Cuatro tipos de descriptores más comunes: de dispositivo, de configuración, de interfaz y de endpoint.
- Clases de dispositivos.
- Características eléctricas y conectores.
- Entrega de energía y consumo de corriente.
- USB On-The-Go para permitir que un dispositivo pueda asumir el rol de host momentáneamente.

Agradecimientos

- Las diapositivas de este tema fueron basadas en las realizadas por el Ing. Esteban Volentini y el Ing. Nicolás Majorel Padilla.

¿Qué aprendimos?

- Nociones generales de comunicaciones y de teoría de la información.
- Codificaciones de fuente
  - Métodos de compresión de datos con y sin pérdidas.
- Codificaciones de canal
  - Codificaciones de línea
  - Métodos de detección y corrección de errores.
- Tipos de modulaciones más frecuentes.
- Interfaces seriales genéricas y UART/USART.
- Normas seriales RS-232, RS-422 y RS-485.
- Interfaces SPI, I2C, CAN y USB.