

Laboratorio
de **Sistemas**
Embebidos

SASE 2022
The 'E' in SASE is rendered in a green and yellow pixelated style.

SIMPOSIO ARGENTINO DE
SISTEMAS EMBEBIDOS

Conectividad con periféricos en sistemas embebidos: USB device

Dr. Ing. Pablo Martín Gomez (pgomez@fi.uba.ar)

“Conectividad con periféricos en sistemas embebidos: USB device”

Por Dr. Ing. Pablo Gomez e Ing. Eric Pernia, se distribuye
bajo una Licencia Creative Commons
Atribución-CompartirIgual 4.0 Internacional.



USB - Historia

- Introducido y estandarizado por un grupo de compañías (Compaq, DEC, IBM, Intel, Microsoft, NEC, HP, Lucent, Philips y Nortel) en 1995
- La idea fundamental fue la de reemplazar la gran cantidad de conectores disponibles en la PC's simplificando la conexión y configuración de dispositivos logrando grandes anchos de banda

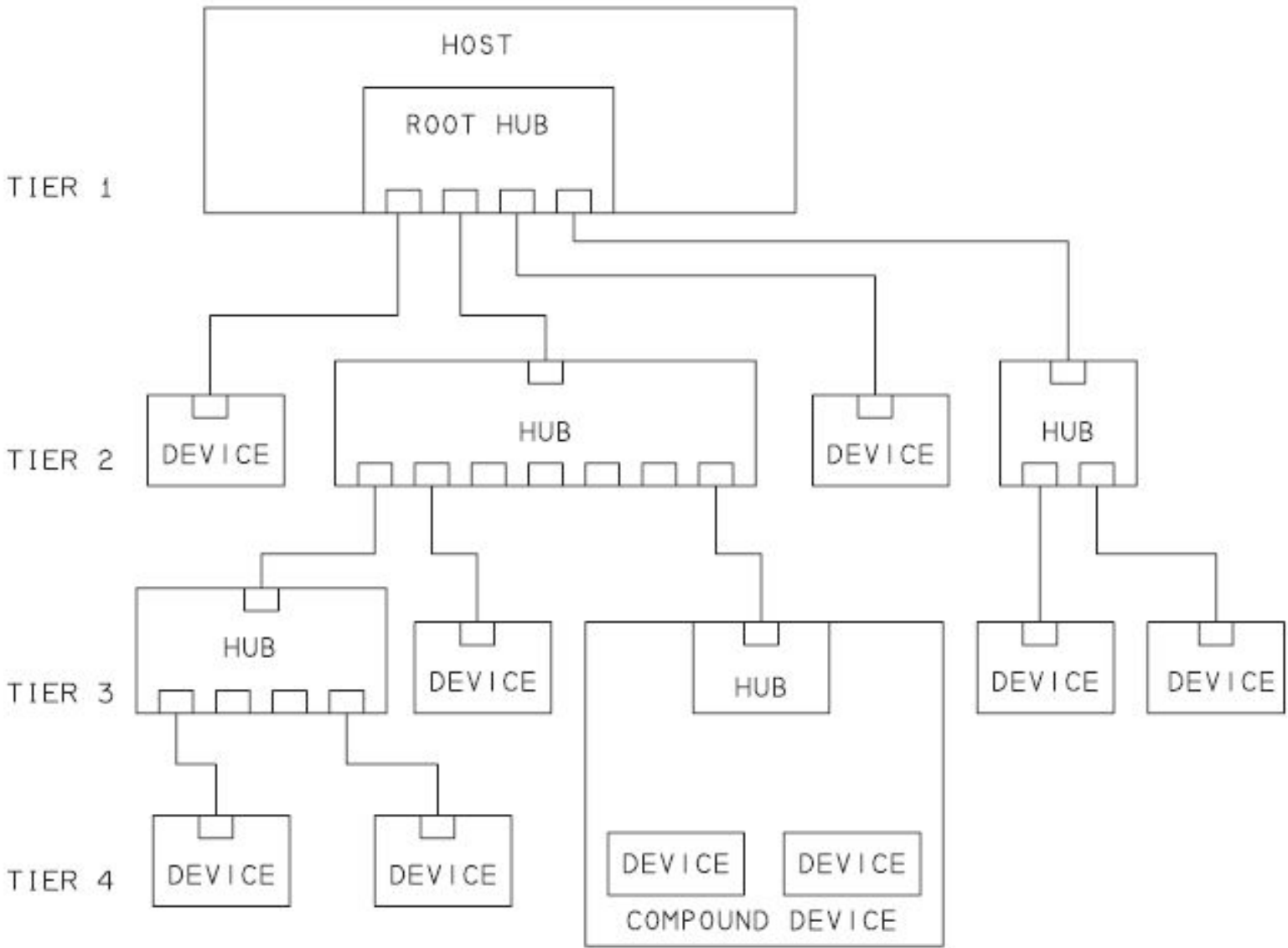


USB - Historia

- Existen 3 versiones de USB
 - USB 1.0 Enero 1996
 - Velocidades de 1.5 Mbps y 12 Mbps
 - USB 1.1 Septiembre 1998
 - Primer versión popular de USB
 - USB 2.0 Abril 2000
 - La principal mejora es la inclusión de una tasa de transferencia de alta velocidad de 480 Mbps
 - USB 3.0 Noviembre 2008
 - Tasa de transferencia de 5 Gbps

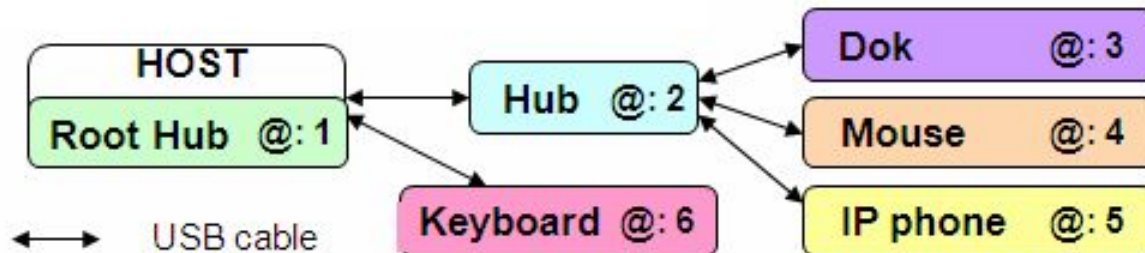
- USB significa “Universal Serial Bus”
- Controlado por “Host” (solamente uno por bus)
 - On-the-Go (Protocolo de negociación de host) – permite a dos dispositivos negociar el rol de host
- Topología estrella
 - Se pueden utilizar hubs para dividir alta y baja velocidad
- Hasta 127 dispositivos pueden ser conectados a un bus USB en cualquier momento
- Utiliza 4 líneas malladas: 2 son de alimentación (+5 V & GND) y los otros 2, un par trenzado donde las señales se transmiten en modo diferencial (3,3 V)

USB - Topología física y lógica

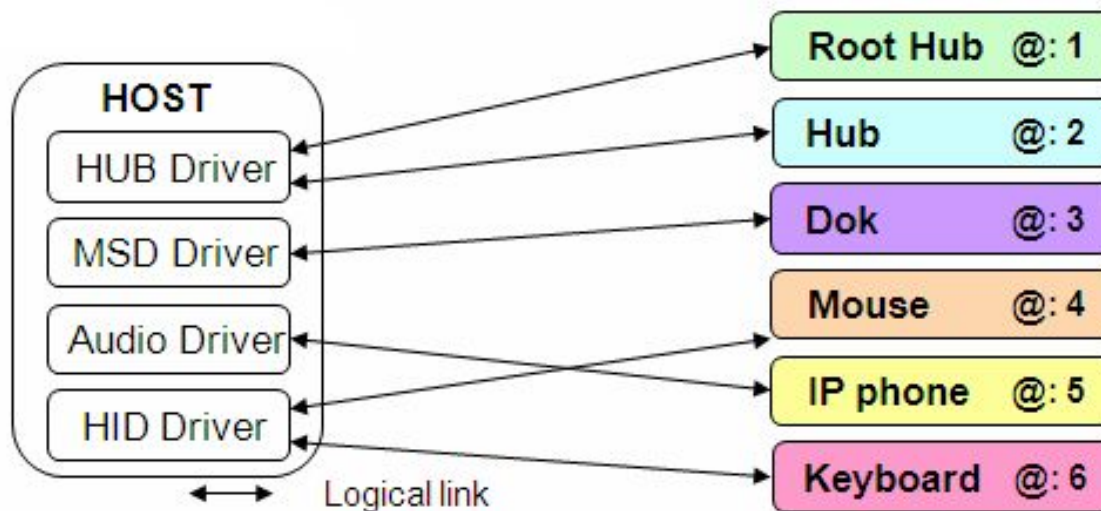


USB - Topología física y lógica

- Topología física: estrella



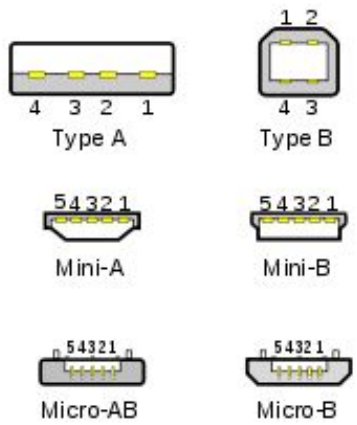
- Topología lógica: punto a punto



- Soporta plug'n'plug con drivers que son cargados dinámicamente - PID/VID (Product ID/Vendor ID)
- USB soporta diferentes modos de transferencia: **Control, Interrupción (Interrupt), Masiva (Bulk) e Isócrona (Isochronous)**
- La alimentación se transporta por el Bus
 - USB distribuye la alimentación a todos los dispositivos conectados, eliminando la necesidad de una fuente externa para dispositivos de bajo consumo

USB - Conectores

- Los conectores a cada lado del cable **no son mecánicamente intercambiables**



- » El conector tipo A siempre se conecta “aguas arriba”. En general los encontramos en hosts y hubs.
- » El conector tipo B siempre se conecta “aguas abajo”. Los encontramos en dispositivos.
- » Los conectores micro-AB pueden ser tanto conectores micro-A como micro-B. Para USB On-the-Go.

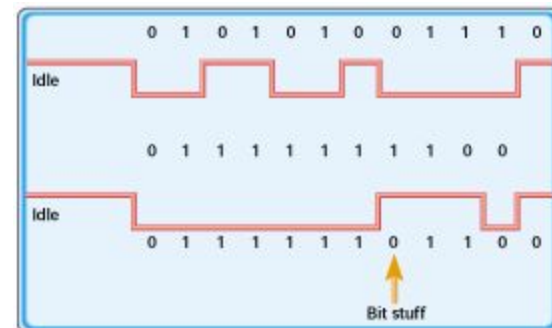
Pin	Standard A, Standard B	Mini B, Micro B
1	VBUS	VBUS
2	D-	D-
3	D+	D+
4	GND	Open or $\geq 1M\Omega$
5	Not present	GND
Shell	Shield	Shield



USB - Alimentación y niveles

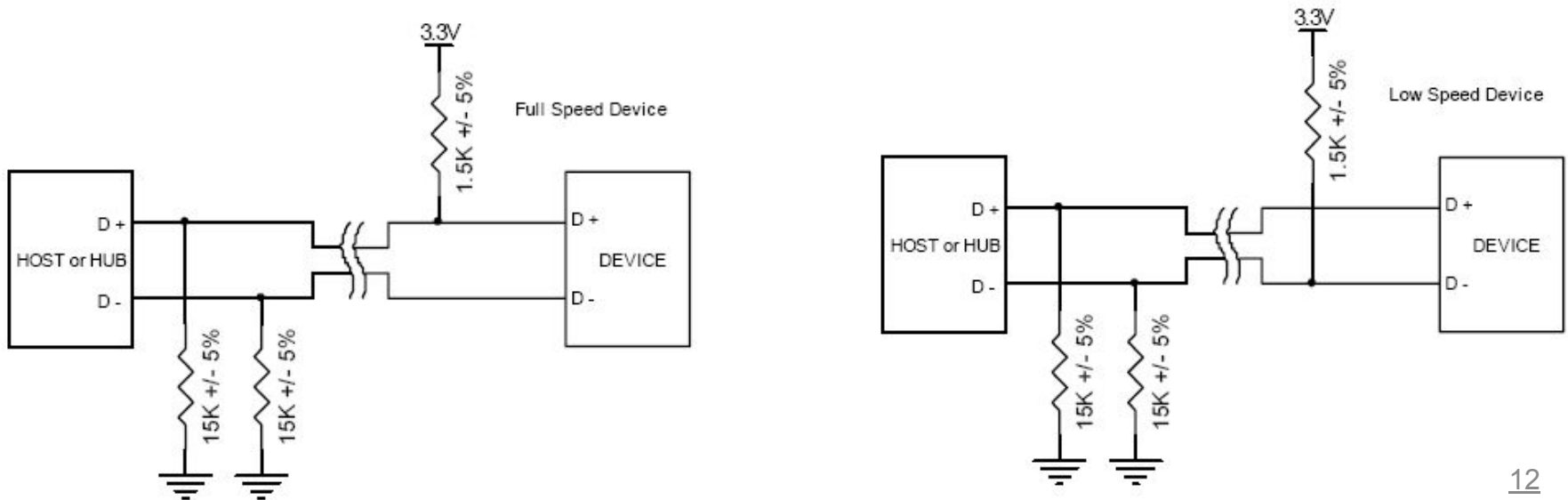
- Alimentación
 - Entrega 5 V en una de las líneas ($5\text{ V} \pm 5\%$).
 - La unidad de carga es 100 mA (USB 2.0) y 150 mA (USB 3.0).
 - La máxima carga es 500 mA (USB 2.0) y 900 mA (USB 3.0).
 - Los hubs alimentados por Bus solamente entregan 1 unidad de carga para los dispositivos.
 - Los hubs alimentados autónomamente pueden entregar la máxima carga a todos los dispositivos.

- Utiliza codificación **NRZI** para enviar los datos con un campo de sincronización para sincronizar el clock del host y el receptor
- NRZI define un 0 lógico como una transición en el valor de tensión, y un 1 manteniendo el nivel
- Se necesita Bit stuffing porque los receptores sincronizan transiciones. Si se envían muchos 1s entonces el receptor puede perder sincronismo

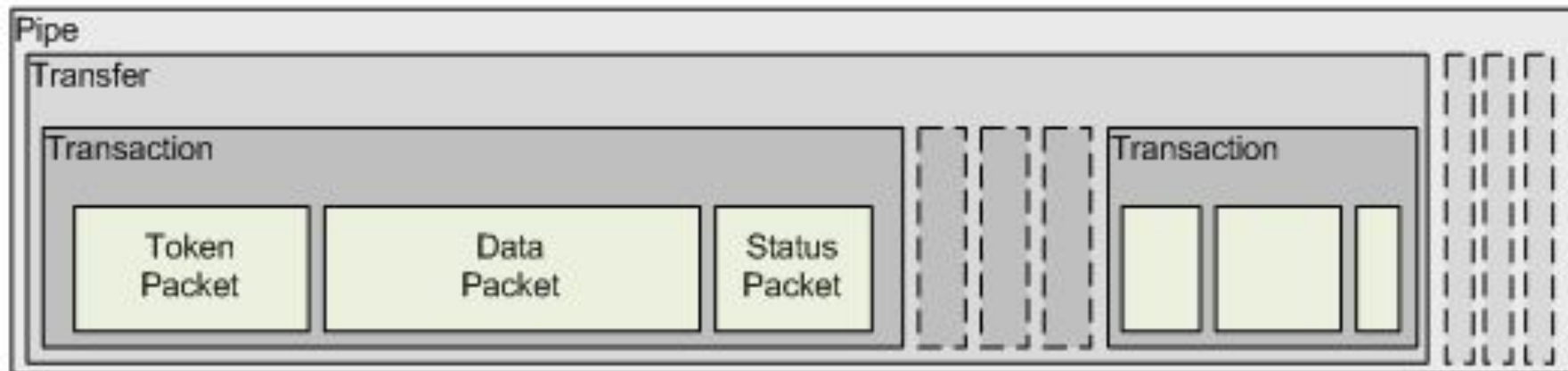


USB - Velocidad

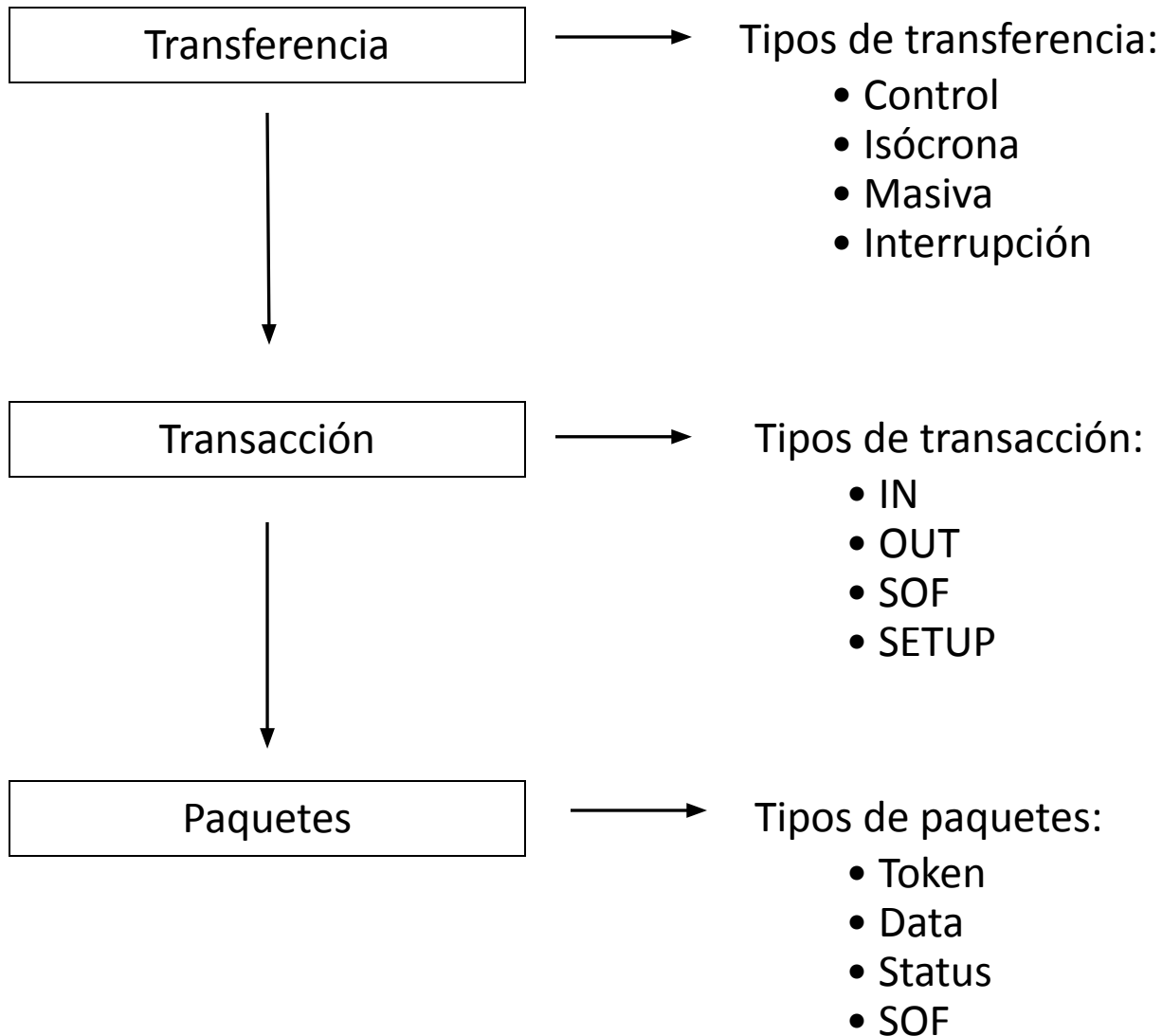
- Un dispositivo USB debe indicar su velocidad llevando D+ o D- a 3.3 volts.
- Sin resistencia de pull-up, USB asume que no hay nada conectado al Bus.
 - En el modo “high speed” el dispositivo primero se conecta en modo “full speed”, luego se remueve el resistor de pull-up para balancear la línea



- A diferencia de RS-232 o interfaces serie similares donde el formato de los datos a ser enviados no está definido, USB posee **varias capas de protocolos**

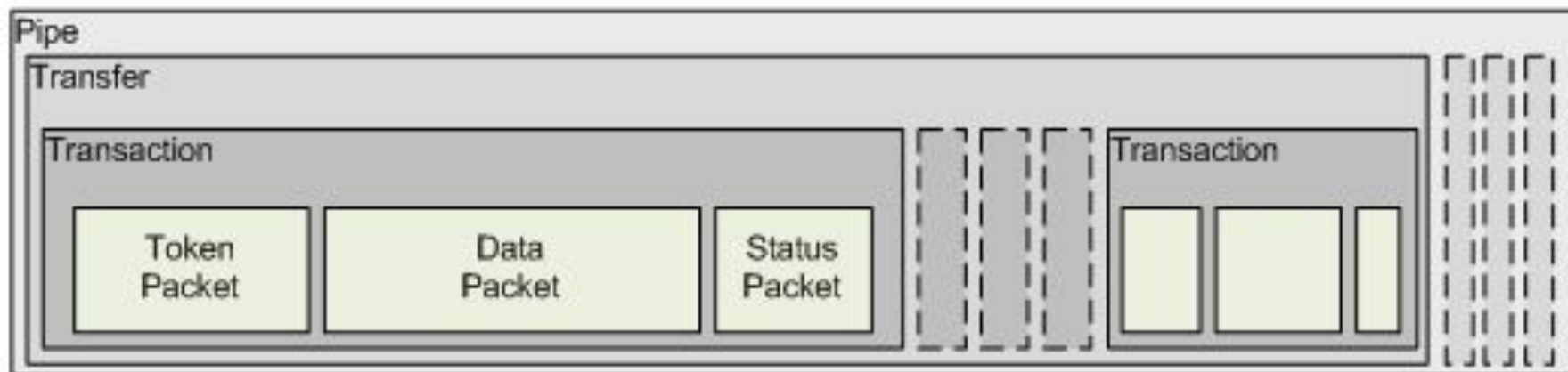


USB – Comunicación



USB – Paquetes

- 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 acknowledge en las transacciones y como una forma de corregir errores)



USB - Campos del paquete

- Sync
 - 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 que dispositivo va dirigido el paquete
 - Teniendo un tamaño de 7 bits permite soportar 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

USB - Campos del paquete - PID

Type	PID value (msb-first)	Transmitted byte (lsb-first)	Name	Description
<i>Reserved</i>	0000	0000 1111		
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)

- ENDP (Endpoint 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 2 bits seguido por una J (estado lógico, el significado depende de la velocidad) durante el tiempo de 1 bit

USB - Tipos de paquete

- Paquetes “Token”

- **In** – Informa al dispositivo USB que el “host” desea leer información
- **Out** – Informa al dispositivo USB que el “host” desea enviar información
- **Setup** – Utilizado para comenzar transferencias de control



- Paquete de datos

- Dos tipos. Cada uno capaz de transmitir de 0 a 1023 bytes de datos



USB - Tipos de paquete

- 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.

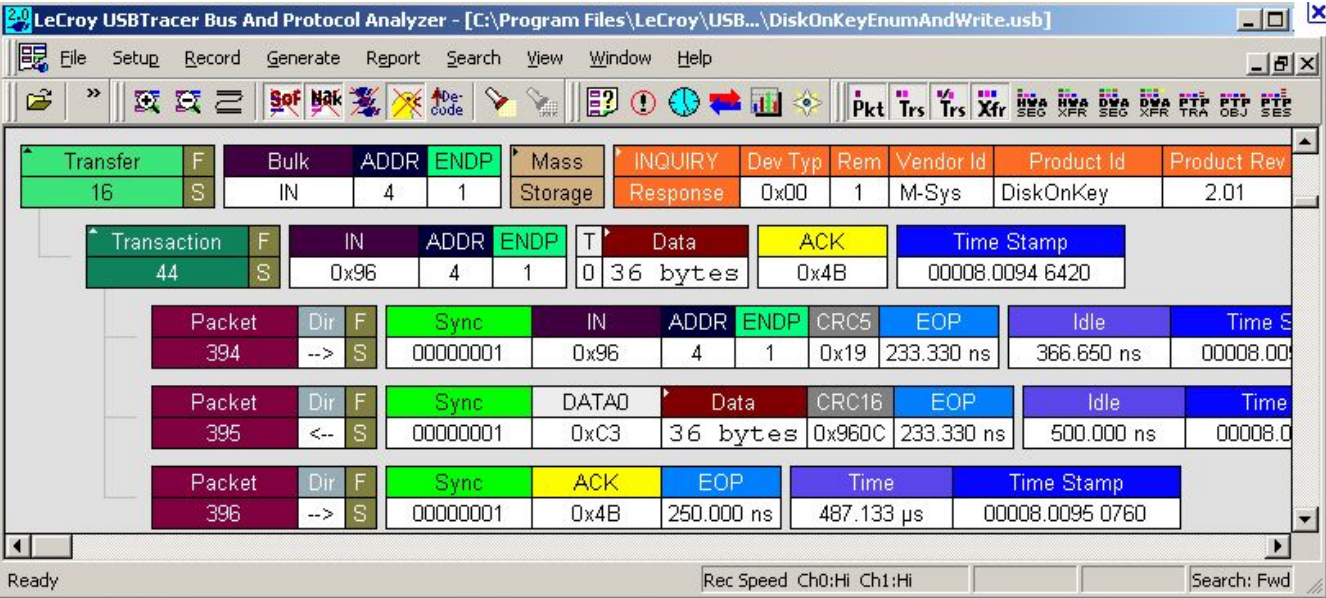


- Paquetes de comienzo de “frame” (SOF)
 - El número de frame (11 bits) es enviado por el “host” cada $1\text{mS} \pm 500\text{nS}$



USB – Analizador de protocolo

- Las transferencias consisten en una o más transacciones. Un pipe solamente soporta un tipo de transferencia
- En una transacción, transferencia desde “host” a dispositivo o viceversa. La dirección se define en el paquete “token”
- En general, el destinatario responde con un paquete de status indicando si fue exitosa



The screenshot shows the LeCroy USBTracer interface with the following data:

Transfer	F	Bulk	ADDR	ENDP	Mass	INQUIRY	Dev Typ	Rem	Vendor Id	Product Id	Product Rev
16	S	IN	4	1	Storage	Response	0x00	1	M-Sys	DiskOnKey	2.01

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
44	S	0x96	4	1	0	36 bytes	0x4B	00008.0094 6420

Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time S
394	-->	S	00000001	0x96	4	1	0x19	233.330 ns	366.650 ns	00008.00
395	<--	S	00000001	DATA0	0xC3	36 bytes	0x960C	233.330 ns	500.000 ns	00008.0
396	-->	S	00000001	ACK	0x4B	250.000 ns	487.133 µs	00008.0095 0760		



USB – Analizador de protocolo

LeCroy USBTracer Bus And Protocol Analyzer - [C:\Program Files\LeCroy\USB...\DiskOnKeyEnumAndWrite.usb]

File Setup Record Generate Report Search View Window Help

Sof Nak De-code
Pkt Trs Trs Xfr
HWA SEG HWA XFR DWA SEG DWA XFR PTP TRA PTP OBJ PTP SES

Transfer	F	Bulk	ADDR	ENDP	Mass	INQUIRY	Dev Typ	Rem	Vendor Id	Product Id	Product Rev
16	S	IN	4	1	Storage	Response	0x00	1	M-Sys	DiskOnKey	2.01

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
44	S	0x96	4	1	0	36 bytes	0x4B	00008.0094 6420

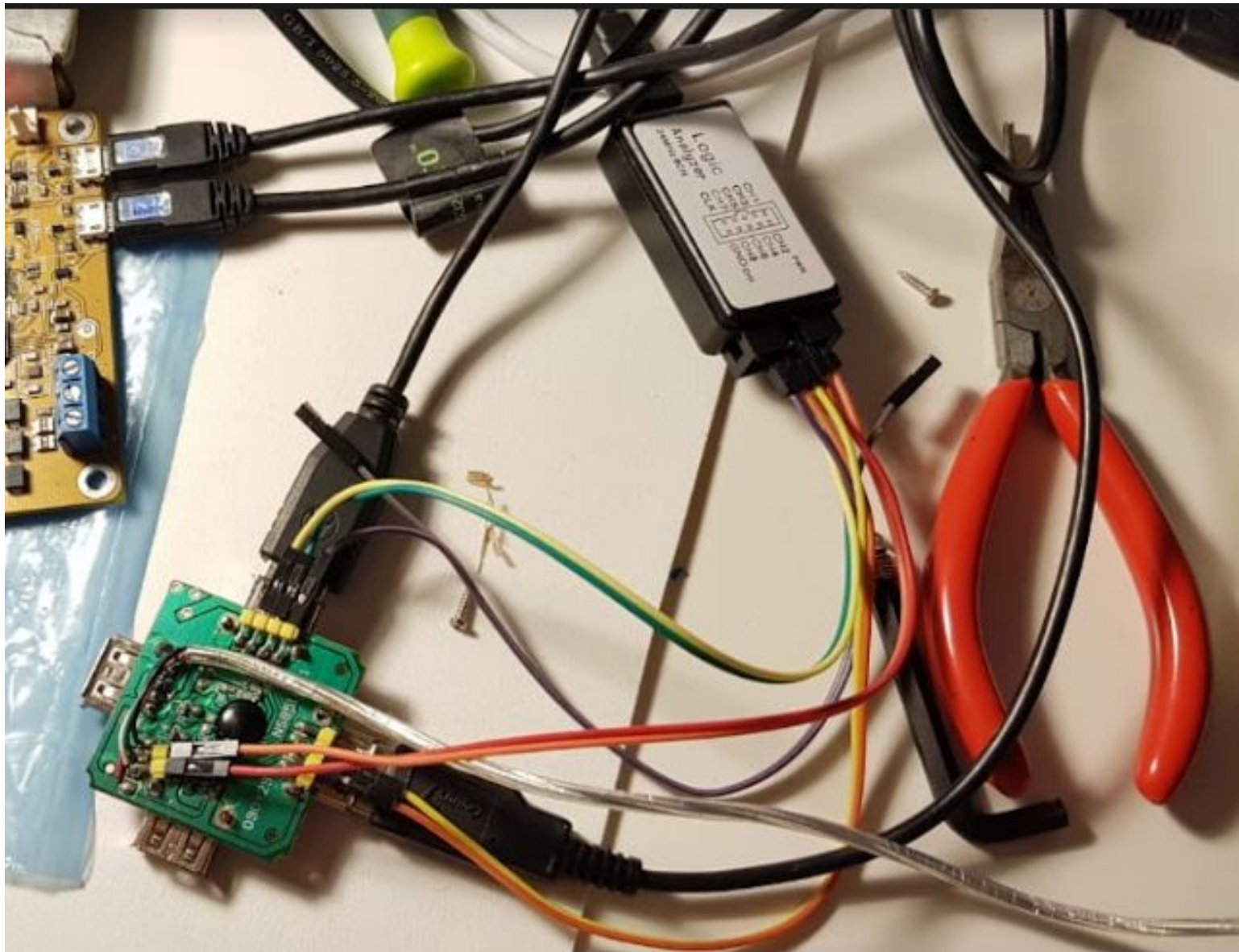
Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time S
394	-->	S	00000001	0x96	4	1	0x19	233.330 ns	366.650 ns	00008.00

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time
395	<--	S	00000001	0xC3	36 bytes	0x960C	233.330 ns	500.000 ns	00008.0

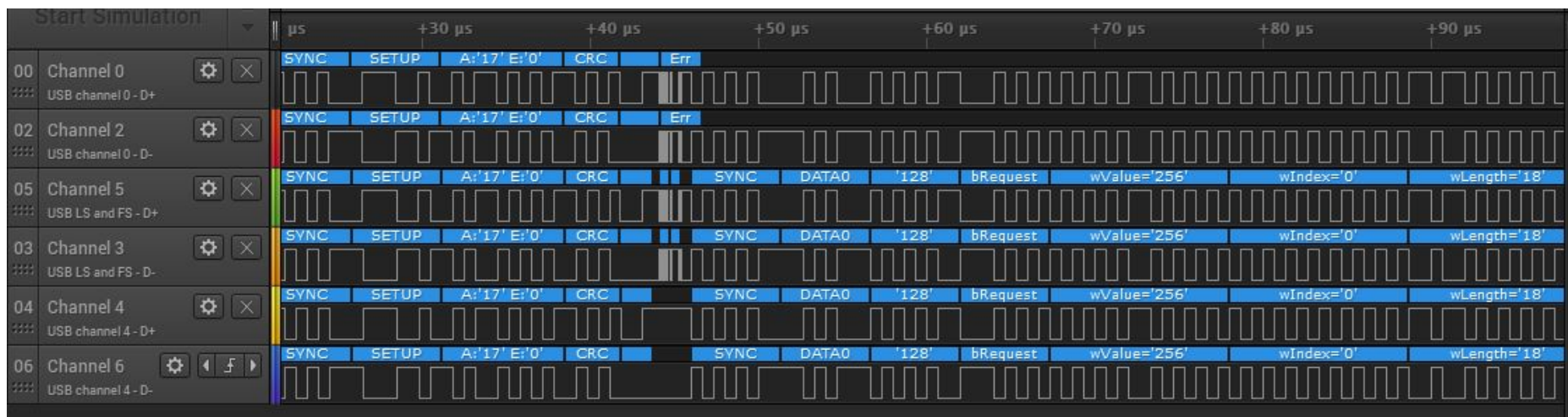
Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
396	-->	S	00000001	0x4B	250.000 ns	487.133 μs	00008.0095 0760

Ready Rec Speed Ch0:Hi Ch1:Hi Search: Fwd

USB – Analizador de protocolo

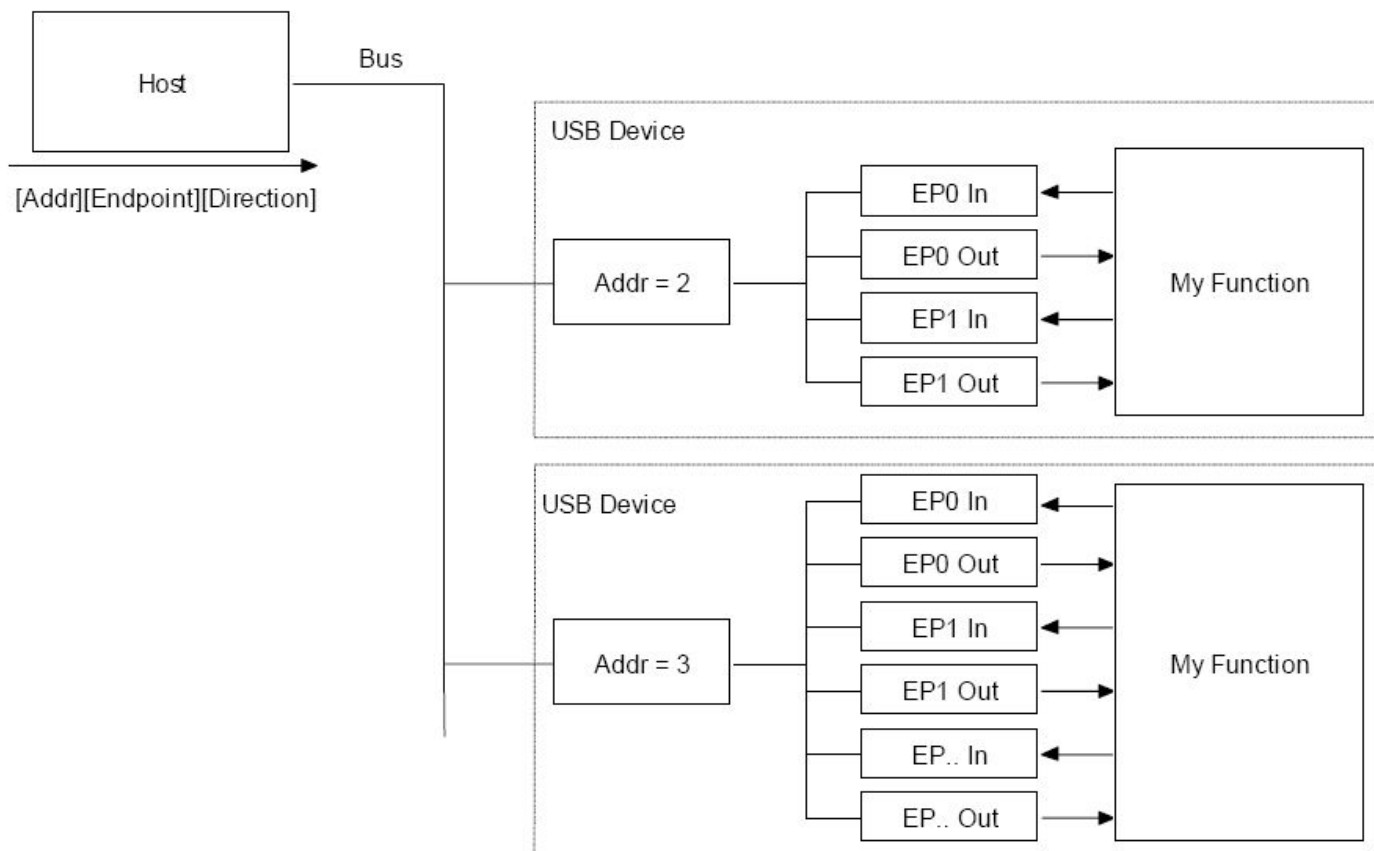


- Captura de enumeración de un mouse Genius
- Captura mouse Genius y EDU-CIAA HID
 - Ejemplos USB device EDU-CIAA



USB – Funciones

- Funciones USB
 - Pueden verse como dispositivos USB que proveen capacidades o funciones tales como impresora, escáner, lector de memorias u otro periférico

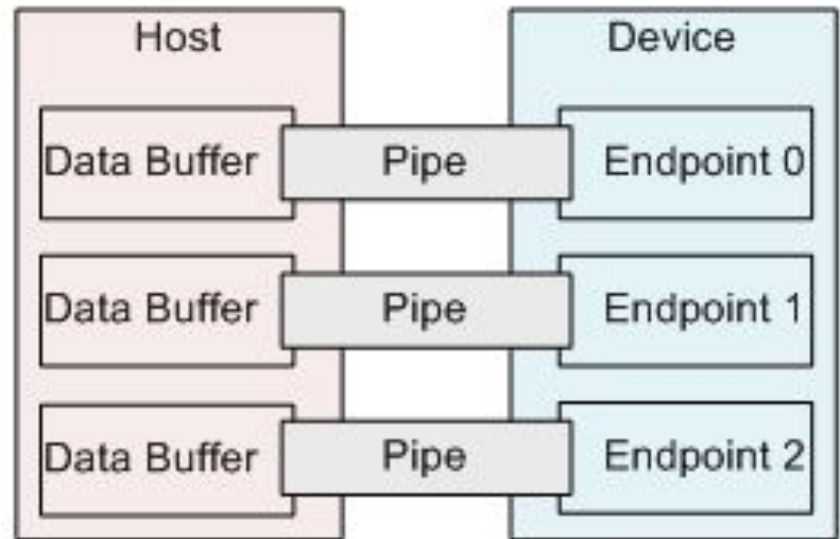


- Son las conexiones lógicas entre “host” y “endpoint(s)”
- Tienen una serie de parámetros:

- Ancho de banda asignado

- Tipo de transferencia:

- Control, Másiva (Bulk),
Isócrona o Interrupción



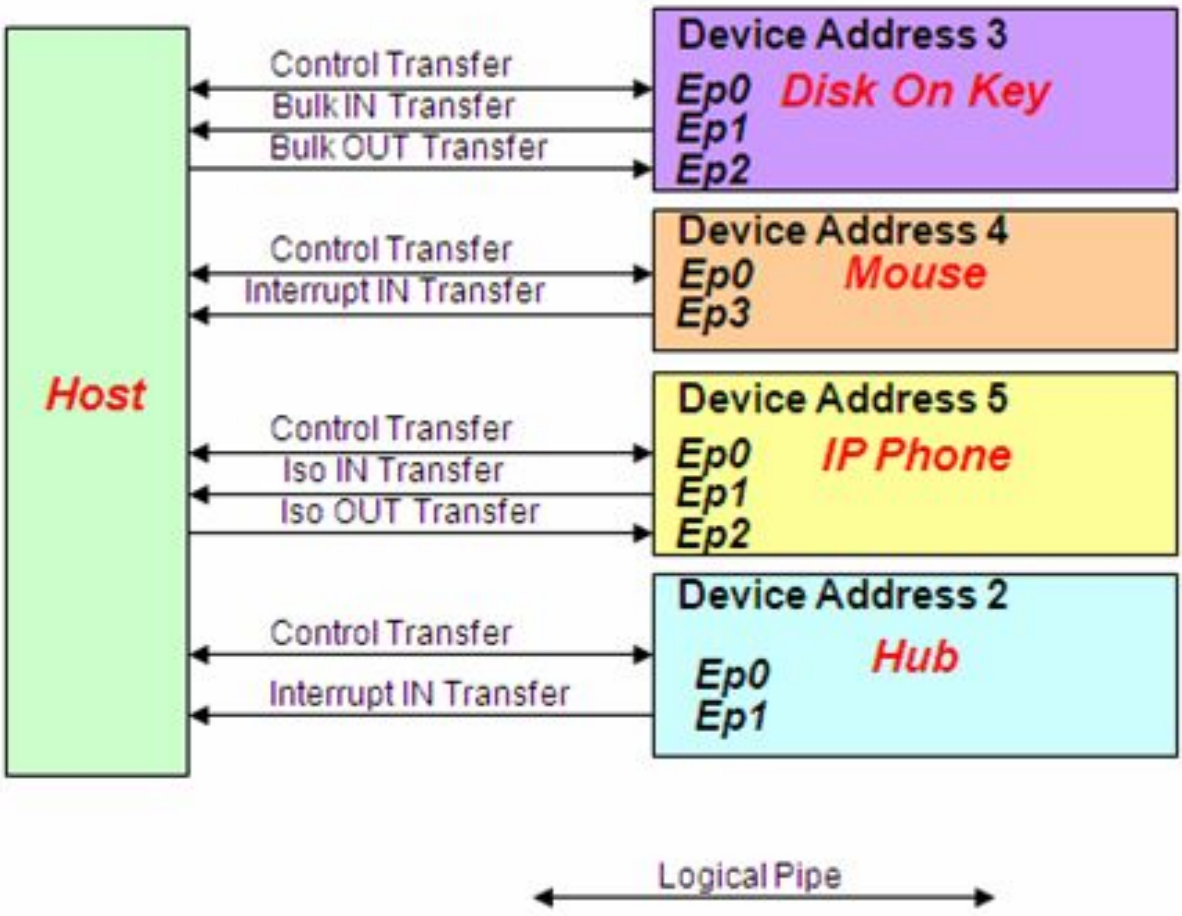
- Dirección del flujo de datos

- Tamaño máximo de paquetes/buffer

- Todos los dispositivos tienen un “default control pipe” que utiliza el endpoint cero

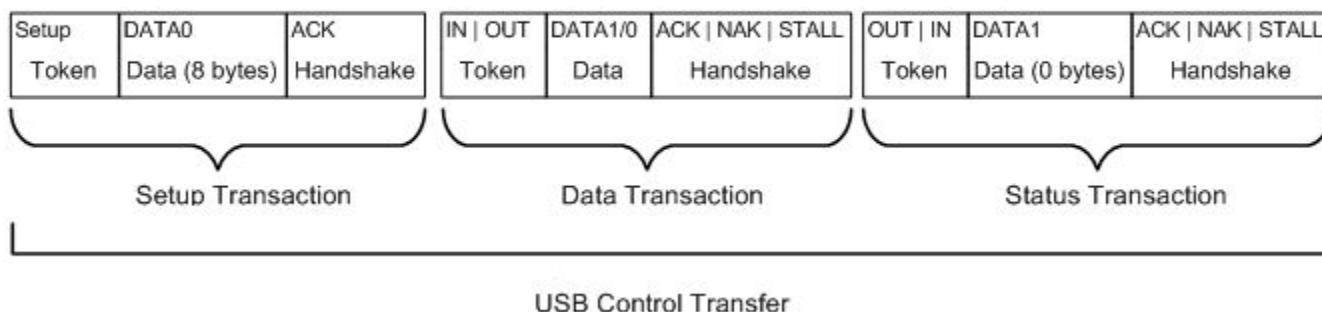
USB – Endpoints

- Pueden describirse como fuentes o sumideros de datos
- Todos los dispositivos deben tener “endpoint” cero



USB – Transferencias de control

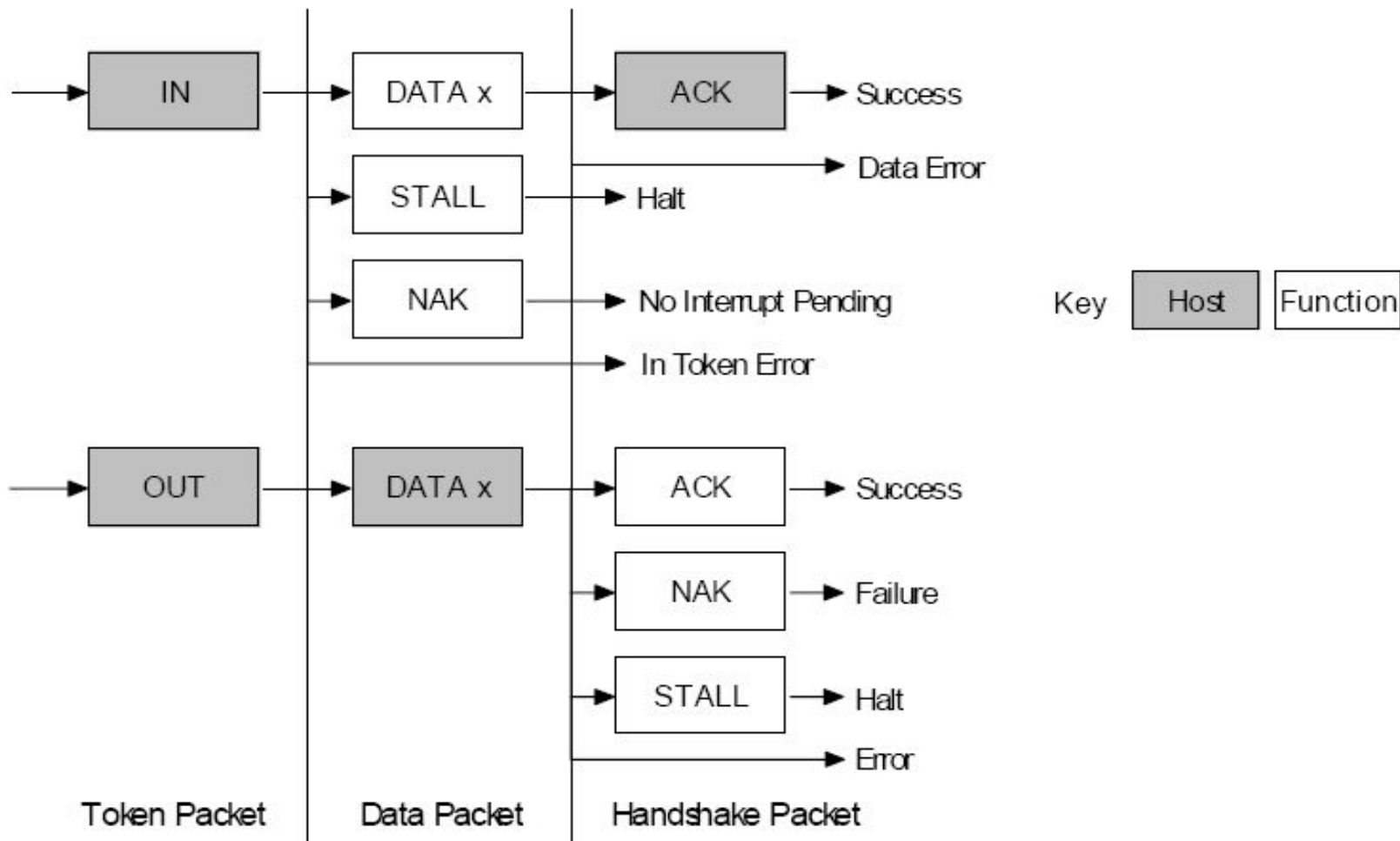
- Las transferencias de control son típicamente utilizadas para operaciones con comandos y de status
- Una transferencia de control puede tener hasta tres etapas
 - Etapa “**Setup**”: donde la petición es enviada. Contiene la dirección y el número de endpoint
 - Etapa de datos (opcional): consiste en una o multiples transferencias IN / OUT
 - Etapa de “Status”: informa el status de la totalidad de la petición. Varía en función de la dirección de la transferencia



- El dispositivo que requiere atención debe esperar que el “host” le “encueste” antes que pueda informar que necesita atención
- Características
 - Latencia garantizada
 - Flujo del “pipe”: Unidireccional
 - Detecciones de errores y re-proceso en próximo período
- Interrupción IN
 - El “host” encuesta periódicamente al endpoint. La frecuencia con que encuesta está especificada en el **descriptor del endpoint**. Cada encuesta implica que el “host” envíe un IN Token
- Interrupción OUT
 - Cuando el “host” desea enviar al dispositivo datos de interrupción, solicita un OUT token seguido por un paquete de datos que contiene los datos de interrupción

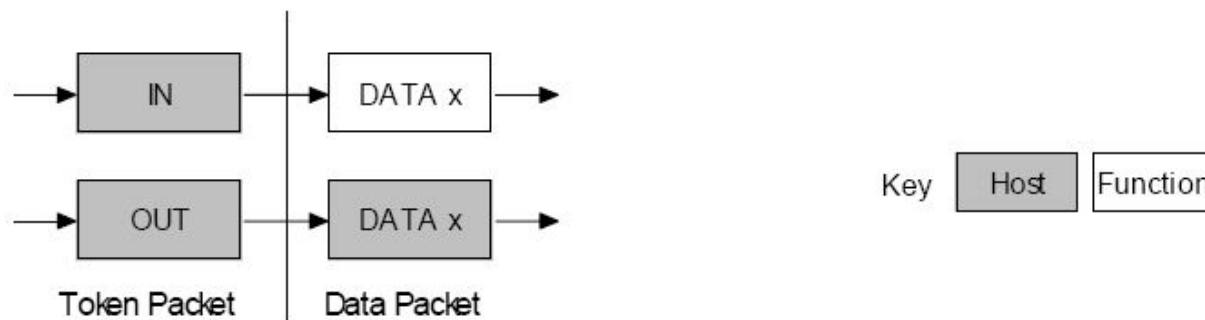
USB – Transferencia de Interrupción

- Formato de transferencia de interrupción



USB – Transferencia de Isócrona

- Las transferencias isócronas ocurren continuamente y periódicamente. Típicamente contienen información sensible al tiempo, como flujo de video o audio
- Características
 - Ancho de banda USB garantizado
 - Latencia acotada
 - Flujo del “Pipe”: Unidireccional
 - Detección de errores vía CRC, pero sin re-proceso ni garantía de entrega
 - Disponible solamente en modos “full speed” y “high speed”

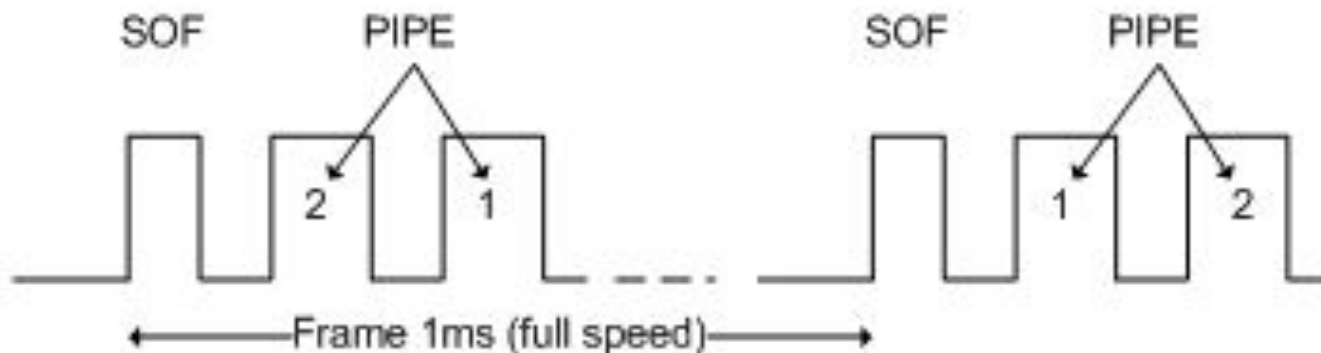


- Las transferencias isócronas **no tienen etapa de “handshaking”** y no pueden reportar errores o condiciones de STALL/HALT

USB – Transferencias masivas

- Utilizado para envío masivo de datos (Ej.: datos de impresión enviados a una impresora o datos de una imagen generados por un escáner)
- Características
 - Corrección de errores (Campo CRC16 en el “data payload”)
 - Detección de errores / mecanismos de re-transmisión
- Utiliza espacio no asignado del ancho de banda del bus después que todas las otras transacciones han sido asignadas
 - Solamente utilizado en comunicaciones no sensibles al tiempo debido a que no hay garantías respecto a la latencia
- Disponible solamente en modos “full speed” y “high speed”

- 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 μ s
- Cada “frame” comienza con un paquete SOF
- A cada “pipe” se le asigna un espacio en cada “frame”
- 10 % asignado a transferencias de control



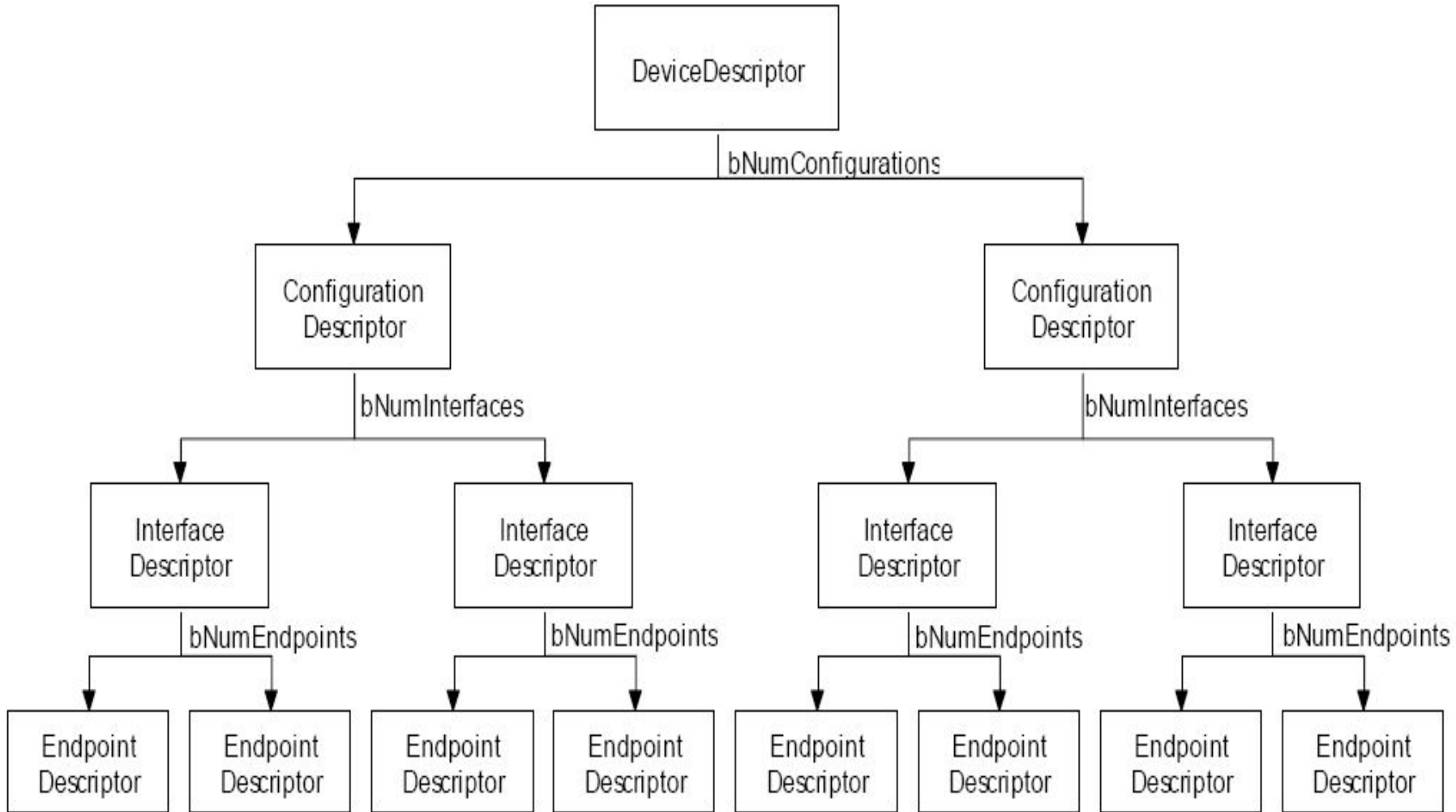
USB - “Throughput” teórico

- A medida que los dispositivos son enumerados el host va contabilizando el ancho de banda solicitado por los endpoints isócronos y de interrupción
- Pueden consumir hasta un 90 % del disponible, luego el host niega el acceso

Tipo de transferencia	Low-Speed (1.5 Mbps / 187 kBps)	Full-Speed (12 Mbps / 1,5 MBps)	High-Speed (480 Mbps / 60 MBps)
Control	24 kBps	832 kBps	15.872 kBps
Interrupción	0,8 kBps	64 kBps	24.576 kBps
Masiva	-	1.216 kBps	53.248 kBps
Isócrona	-	1.023 kBps	24.576 kBps

- Todos los dispositivos USB tienen una jerarquía de descriptores que definen al “host” información tal como:
 - que es el dispositivo
 - quien lo fabricó
 - que versión de USB soporta
 - de cuantas 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 String (String Descriptors)
 - Proporciona información humanamente legible y son opcionales

USB – Descriptores



USB – Descriptores de dispositivo

- El descriptor de dispositivo del dispositivo USB representan a la totalidad del mismo por lo tanto, **sólo puede tener uno**
- Contienen
 - la versión de USB soportada
 - el máximo tamaño de paquete para el **endpoint 0**
 - identificación de proveedor y producto
 - el número de posibles configuraciones que el dispositivo puede tener
- Ejemplo:
 - bDeviceClass, bDeviceSubClass y bDeviceProtocol son utilizados por el sistema operativo para encontrar un driver para el dispositivo
 - Generalmente solo bDeviceClass es especificado en este nivel
 - Se suelen especificar los demás parámetros a nivel de interfaz. Esto permite que un mismo dispositivo soporte múltiples clases



USB – Descriptores 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

- Un dispositivo USB puede tener diferentes configuraciones. De todas formas, la mayoría de los dispositivos son simples y solamente tienen una
- Especifica
 - como se alimenta el dispositivo
 - cual es el máximo consumo de potencia
 - el número de interfaces que tiene
- Por lo tanto, 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 para cada una de las configuraciones, diferentes **modos de transferencia**



USB – Descriptores 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).

USB – Descriptores de interfaz

- Pueden ser vistos como “headers” de los endpoints en grupos funcionales que desarrollan una misma función en el dispositivo

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





















USB – Descriptores de endpoint

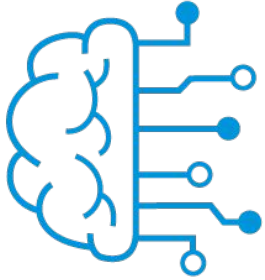
Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of Descriptor in Bytes (7 bytes)
1	bDescriptionType	1	Constant	Endpoint Descriptor (0x05)
2	bEndpointAddress	1	Endpoint	Endpoint Address, Encoded as follows 0..3b Endpoint Number 4..6b Reserved. Set to Zero 7b Direction (Ignored for Control Endpoints) 0 = Out Endpoint, 1 = In Endpoint
3	bmAttributes	1	Bitmap	<div style="border: 2px solid red; padding: 5px;"> <p>Bits 0..1 Transfer Type</p> <p>00 = Control</p> <p>01 = Isochronous</p> <p>10 = Bulk</p> <p>11 = Interrupt</p> </div> <p>Bits 2..7 are reserved. If Isochronous endpoint, Bits 3..2 = Synchronisation Type (Iso Mode)</p> <p>00 = No Synchronisation</p> <p>01 = Asynchronous</p> <p>10 = Adaptive</p> <p>11 = Synchronous</p> <p>Bits 5..4 = Usage Type (Iso Mode)</p> <p>00 = Data Endpoint</p> <p>01 = Feedback Endpoint</p> <p>10 = Explicit Feedback Data Endpoint</p> <p>11 = Reserved</p>
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.

- El endpoint cero siempre se asume como de control
- El “host” utilizará la información devuelta por estos descriptores para definir los requerimientos de ancho de banda del bus

1. El sistema tiene un nuevo dispositivo.
2. El hub detecta un dispositivo.
3. El host es notificado del nuevo dispositivo.
4. El hub detecta si el dispositivo es low o full speed.
5. El hub resetea el dispositivo.
6. El host es notificado si el dispositivo soporta high speed.
7. El hub establece un camino de señal entre dispositivo y bus.
8. El host envía un pedido de Get Descriptor para conocer el máximo tamaño de paquete del default pipe.
9. El host asigna una dirección al dispositivo.
10. El host aprende las habilidades del dispositivo.
11. El host asigna y carga los drivers del dispositivo.
12. El driver del host selecciona una configuración.

USB – Enumeración

Item	Device	Payload
 Reset (2.3 s)		
 Suspended (114.0 ms)		
 Reset (10.0 ms)		
 High speed Detection Handshake		
+  GetDescriptor (Device)	0 (5)	8 bytes (12 01 00 02 FF 00 00 08)
 Reset (10.0 ms)		
 High speed Detection Handshake		
+  SetAddress (5)	0 (5)	No data
+  GetDescriptor (Device)	5	18 bytes (12 01 00 02 FF 00 00 08 ...)
+  GetDescriptor (Configuration)	5	9 bytes (09 02 2E 00 01 01 00 A0 32)
+  GetDescriptor (Configuration)	5	46 bytes (09 02 2E 00 01 01 00 A0 ...)
+  GetDescriptor (String lang IDs)	5	4 bytes (04 03 09 04)
+  GetDescriptor (String iProduct)	5	24 bytes (18 03 57 00 69 00 6E 00 ...)
+  GetDescriptor (String lang IDs)	5	4 bytes (04 03 09 04)
+  GetDescriptor (String iProduct)	5	24 bytes (18 03 57 00 69 00 6E 00 ...)
+  GetDescriptor (Device)	5	18 bytes (12 01 00 02 FF 00 00 08 ...)
+  GetDescriptor (Configuration)	5	9 bytes (09 02 2E 00 01 01 00 A0 32)
+  GetDescriptor (Configuration)	5	46 bytes (09 02 2E 00 01 01 00 A0 ...)
+  GetStatus (Device)	5	2 bytes (00 00)
+  SetConfiguration (1)	5	No data



Laboratorio
de **Sistemas**
Embebidos

SASE **E** 2022

SIMPOSIO ARGENTINO DE
SISTEMAS EMBEBIDOS

¡Muchas gracias!

**Conectividad con periféricos en sistemas
embebidos: USB device**

Dr. Ing. Pablo Gomez (pgomez@fi.uba.ar)

- <http://www.beyondlogic.org/usbnutshell/usb-in-a-nutshell.pdf> USB in a nutshell
- Jan Axelson – USB COMPLETE – 4th edition
- <http://www.usb.org>
- <http://www.usblyzer.com/brief-usb-overview-and-history.htm>
- <http://arstechnica.com/old/content/2007/09/intel-announces-demonstrates-usb-3-0.ars>
- <http://arstechnica.com/hardware/news/2008/11/usb-3-0-specification-finalized-devices-in-2010.ars>
- http://techon.nikkeibp.co.jp/english/NEWS_EN/20090310/166949/
- http://www.reghardware.co.uk/2008/01/09/ces_usb_3_revealed/
- <http://arstechnica.com/old/content/2007/09/intel-announces-demonstrates-usb-3-0.ars>
- <http://en.wikipedia.org/wiki/Usb>
- http://news.cnet.com/8301-17938_105-9780794-1.html
- <http://thefutureofthings.com/news/5739/25gb-in-70-seconds-with-usb-3-0.html>
- <http://www.intel.com/pressroom/archive/releases/20080813corp.htm>
- <http://www.at91.com/repFichier/Document-123/USB-tutorial.ppt>
- http://www.computer-solutions.co.uk/info/Embedded_tutorials/usb_tutorial.htm