

# 7919 | **Sistemas Embebidos**

2º Cuatrimestre de 2017

## **CLASE 10: PROTOCOLOS RS232, RS485 y USB**

**Prof: José H. Moyano**

**Autor original: Sebastián Escarza**

Dpto. de Cs. e Ing. de la Computación  
Universidad Nacional del Sur  
Bahía Blanca, Buenos Aires, Argentina



# **GrIDSE**

The background of the slide is a detailed, high-contrast image of a printed circuit board (PCB). It shows various electronic components such as integrated circuits, capacitors, and resistors, along with the intricate network of copper traces. The image is slightly blurred and has a monochromatic, high-contrast aesthetic. A dark blue horizontal band is superimposed over the middle of the image, containing the title text.

# **Estándar RS-232**

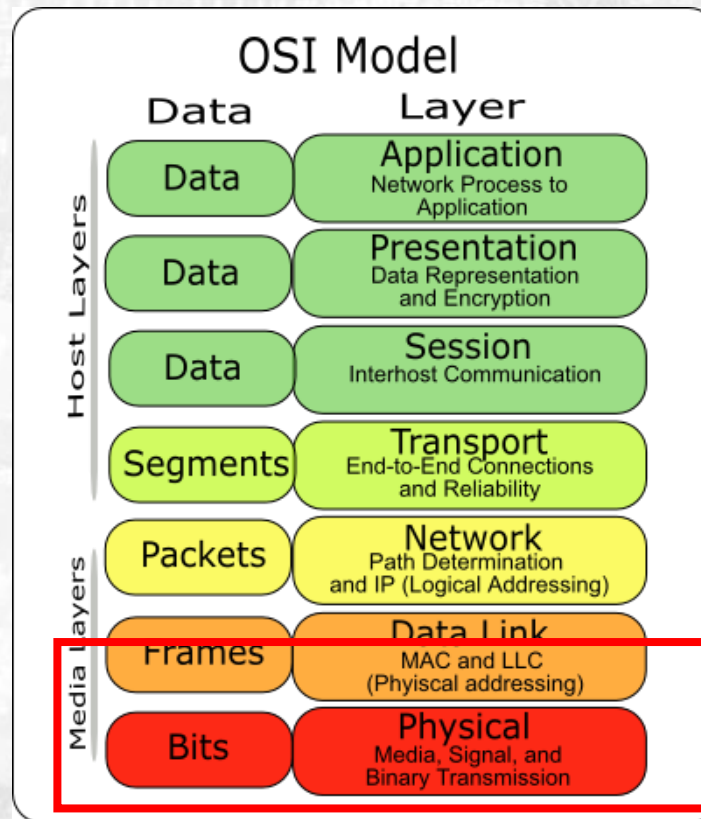
# Estándar RS-232

- **Estándar de la Electronic Industries Association (también conocido como EIA-232).**
- **Define:**
  - Características de la señal eléctrica.
  - Características de la interface mecánica (formato de los conectores y definición de pines).
  - Significado de las señales.
- **No define:**
  - Bit rates, framing de datos, protocolo de comunicación, encodings, compresión de datos, y detección de errores.
  - Cada nodo debe adoptar los mismos valores y configuración



# Estándar RS-232

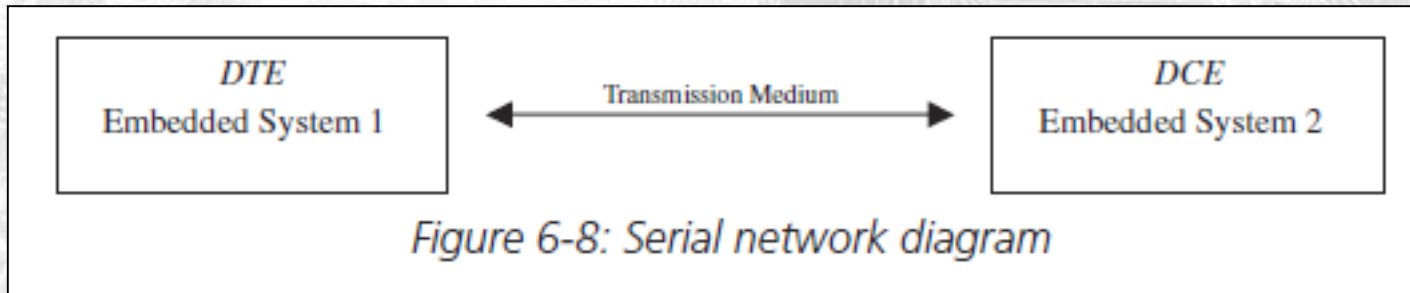
- Define capa física (y data link en firmware).



RS-232

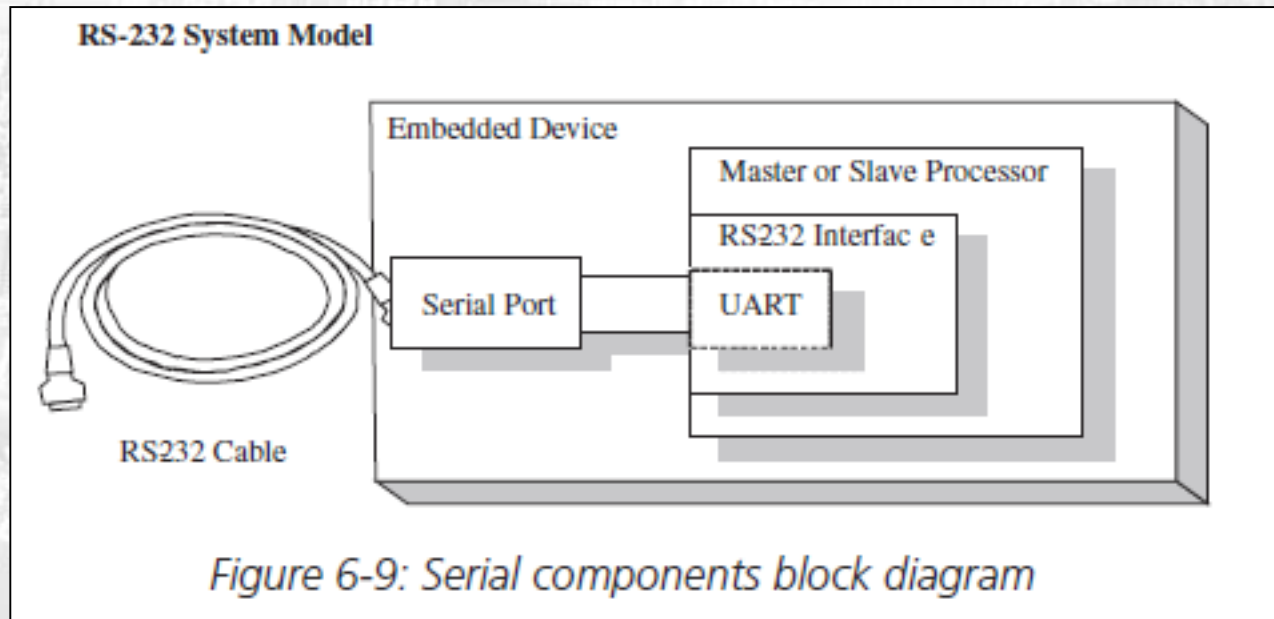
# Estándar RS-232

- Puede operar sincrónica o asincrónicamente.
- Nodos:
  - DTE: Data Terminal Equipment (nodos iniciadores).
  - DCE: Data Communication Equipment (dispositivos pasivos).



# Estándar RS-232

- La interface RS-232 debe mapear las señales al puerto serie a partir de una interface serie (por ej. vía un UART).





# Estándar RS-232

- Distintos conectores y señales...

Pin	Nombre	Significado	Descripción	Voltaje	DTE	DE
1	FG	Frame Ground/Shield			Out	In
2	BA	TxD	Transmit Data	-12	In	Out
3	BB	RxD	Receive Data	-12	Out	In
4	CA	RTS	Request To Send	+12	In	Out
5	CB	CTS	Clear To Send	+12	In	Out
6	CC	DSR	Data Set Ready	+12		
7	AB	SG	Signal Ground			
8	CF	DCD	Data Carrier Detect	+12	In	Out
9			Positive Test Voltage			
10			Negative Test Voltage			
11			Not Assigned			
12		sDCD	Secondary DCD	+12	In	Out
13		sCTS	Secondary CTS	+12	In	Out
14		sTxD	Secondary TxD	-12	Out	In
15	DB	TxC	DCE Transmit Clock		In	Out
16		sRxD	Secondary RxD	-12	In	Out
17	DD	RxC	Receive Clock		In	Out
18	LL		Local Loopback			
19		sRTS	Secondary RTS	+12	Out	In
20	CD	DTR	Data Terminal Ready	+12	Out	In
21	RL	SQ	Signal Quality	+12	In	Out
22	CE	RI	Ring Indicator	+12	In	Out
23	SEL		Speed Selector DTE		In	Out
24	DA	TCK	Speed Selector DCE		Out	In
25	TM	TM	Test Mode	+12	In	Out

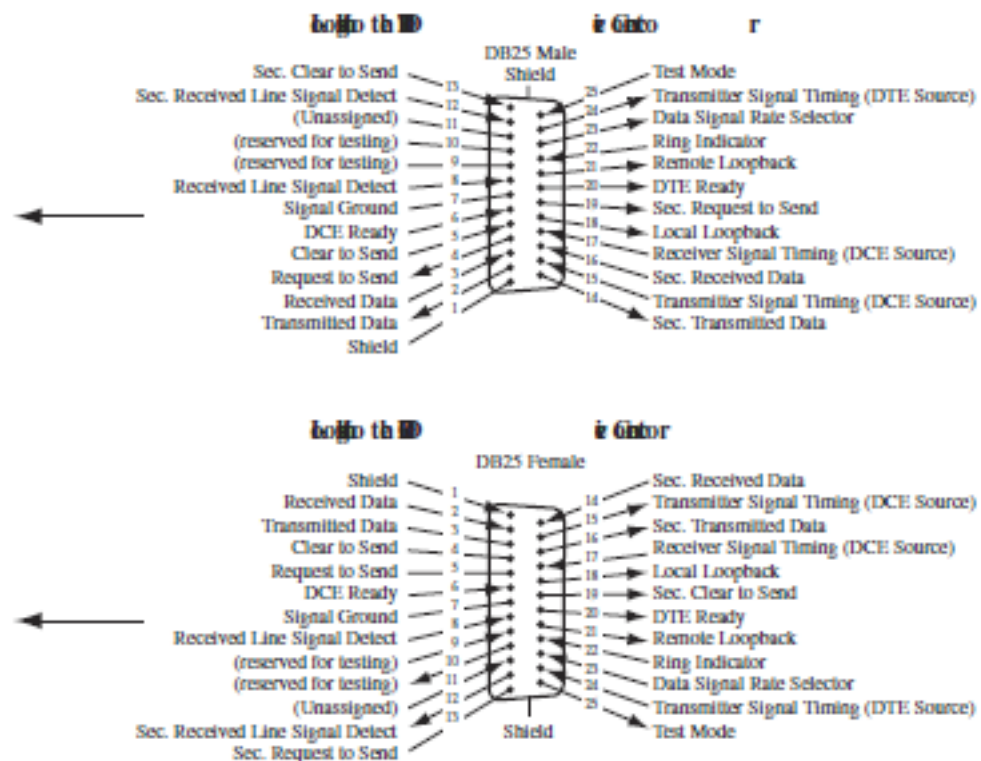
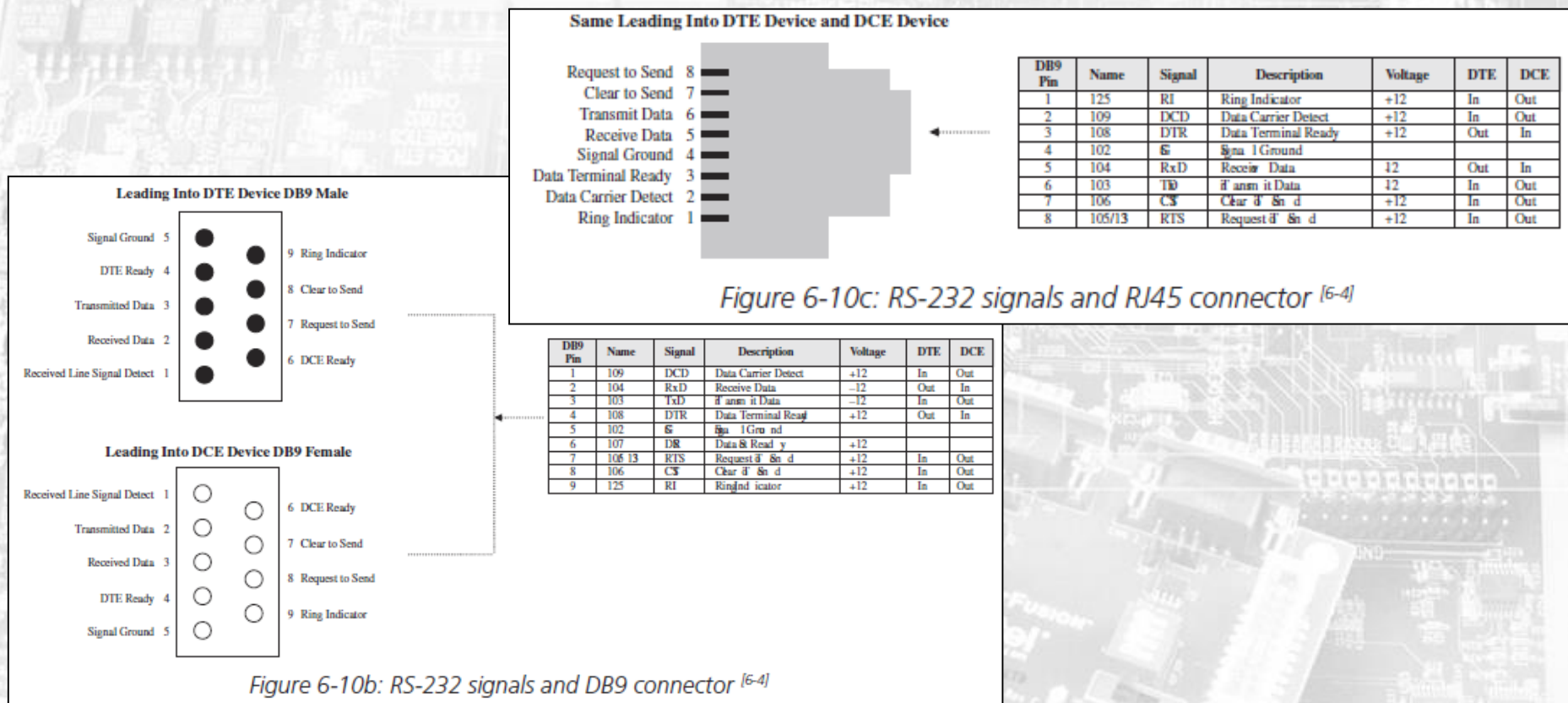


Figure 6-10a: RS-232 signals and DB25 connector [6-4]

# Estándar RS-232

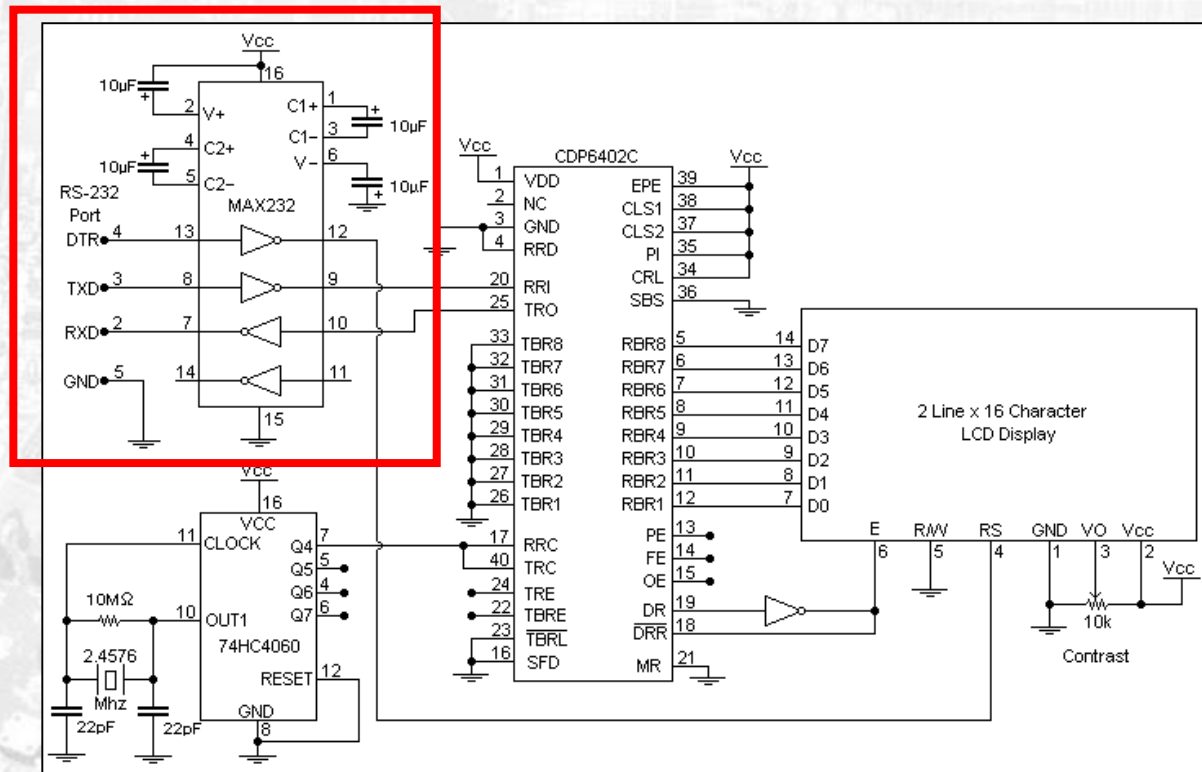
- Distintos conectores y señales...





# Estándar RS-232

- **Max232: ASIC para comunicación por RS-232.**
  - Adapta los niveles lógicos a los voltajes del estándar.



The background of the slide is a detailed, high-contrast image of a printed circuit board (PCB). It shows various electronic components such as integrated circuits, capacitors, and resistors, along with the intricate network of copper traces. The image is slightly blurred and has a monochromatic, high-contrast aesthetic. A dark blue horizontal band is superimposed over the middle of the image, containing the title text.

# **Estándar RS-485**

# Estándar RS-485

- Estándar de la Electronic Industries Association (también conocido como ANSI/TIA/EIA-485).
- Utilizado para comunicaciones en serie asincrónicas a distancias de ~1200m o 10Mbit/s en ambientes ruidosos en un par trenzado
  - línea balanceada (líneas del mismo tipo con igual impedancia a lo largo, respecto de tierra y de otros circuitos)
  - señalización diferencial (señales simétricas y opuestas)
- Base para otros protocolos (CAN bus, Modbus, Profibus, etc).

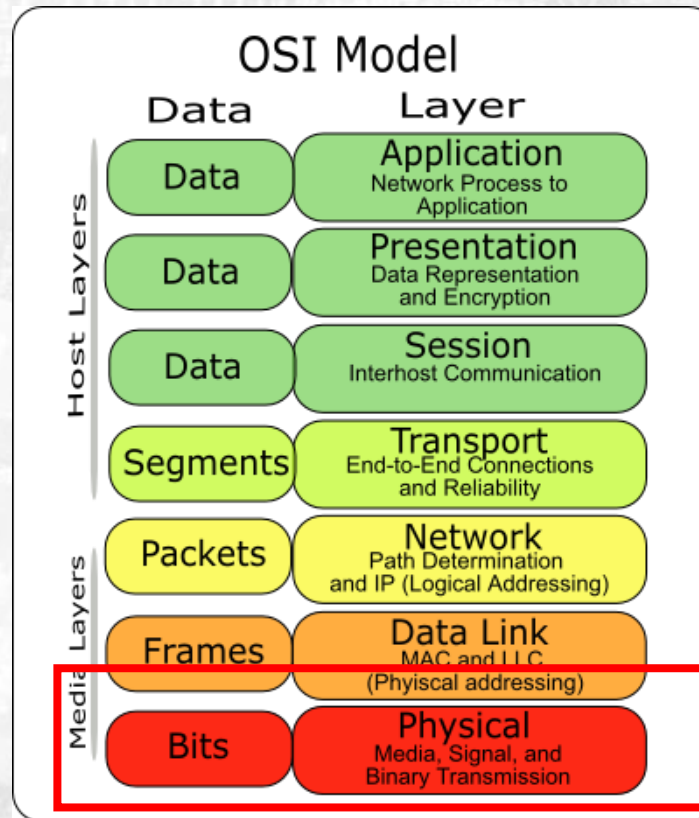


# Estándar RS-485

- **Define:**
  - Características de la señal eléctrica (rangos de tensión, carga, tensión de circuitos abiertos, etc).
  - Significado de las señales.
- **No define:**
  - Valores de resistencias de terminación, pull up y pull down (si se utilizan), ni longitudes de los cables.
  - Características de la interface mecánica (formato de los conectores y definición de pines).
  - bit rates, framing de datos, protocolo de comunicación, encodings, compresión de datos, y detección de errores.
  - Mecanismos de arbitraje de bus.

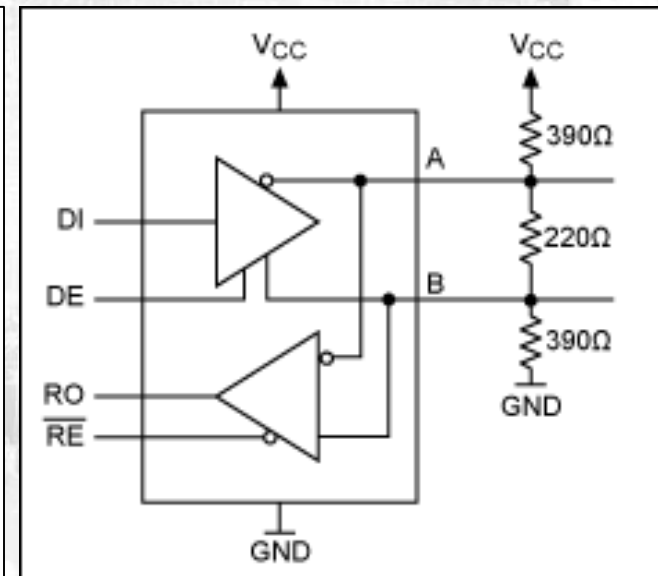
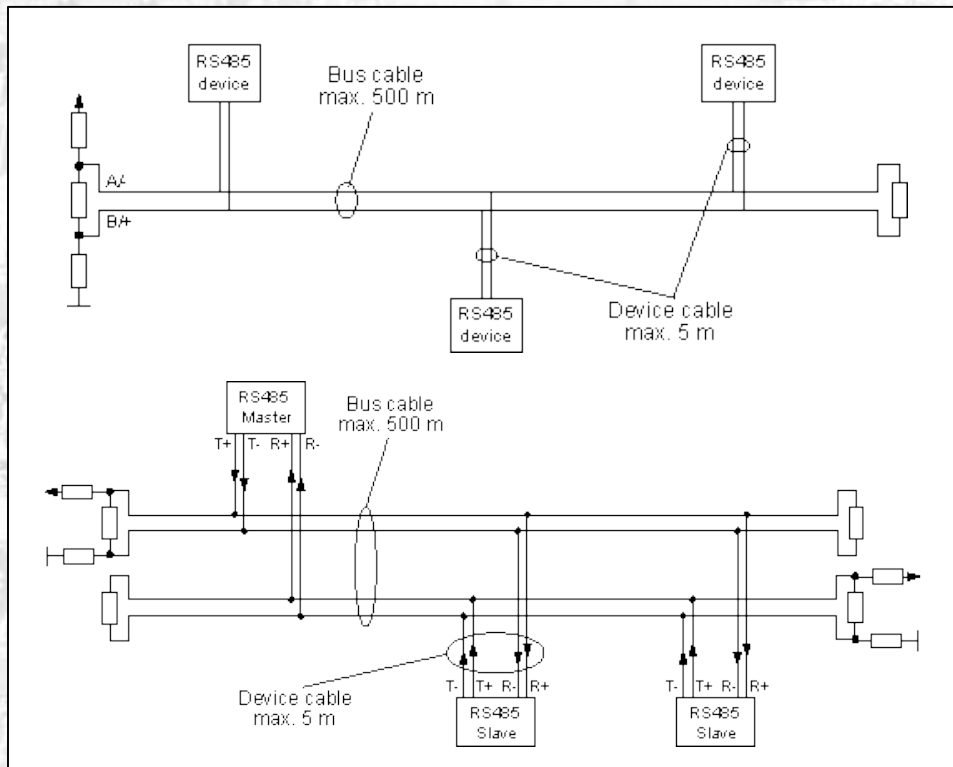
# Estándar RS-485

- Define capa física principalmente.



# Estándar RS-485

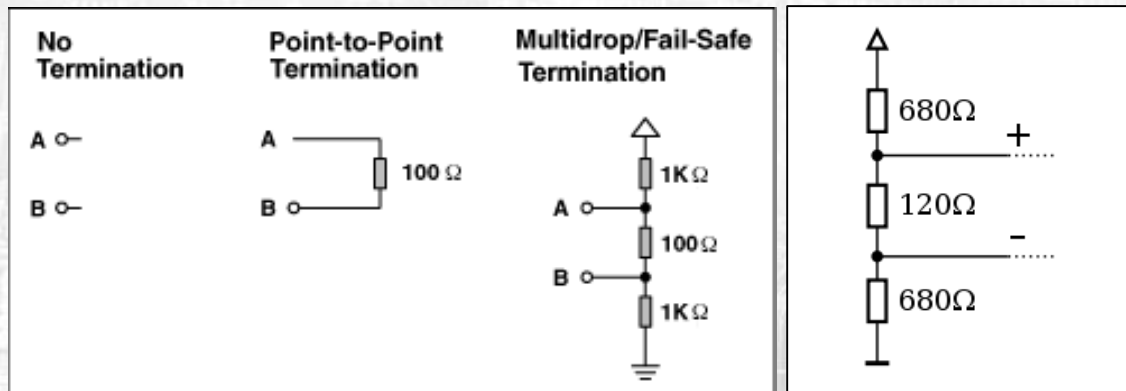
- Half duplex o Full duplex / Lógica 3-state:



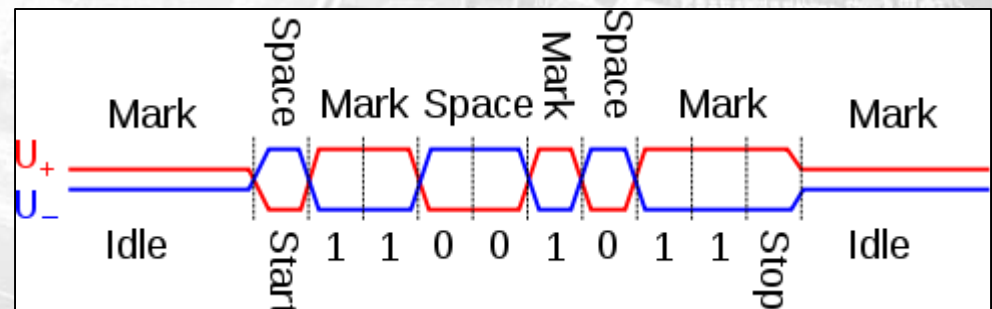


# Estándar RS-485

- Resistencias: los valores dependen de la impedancia del cable ( $\sim 120$  ohms):

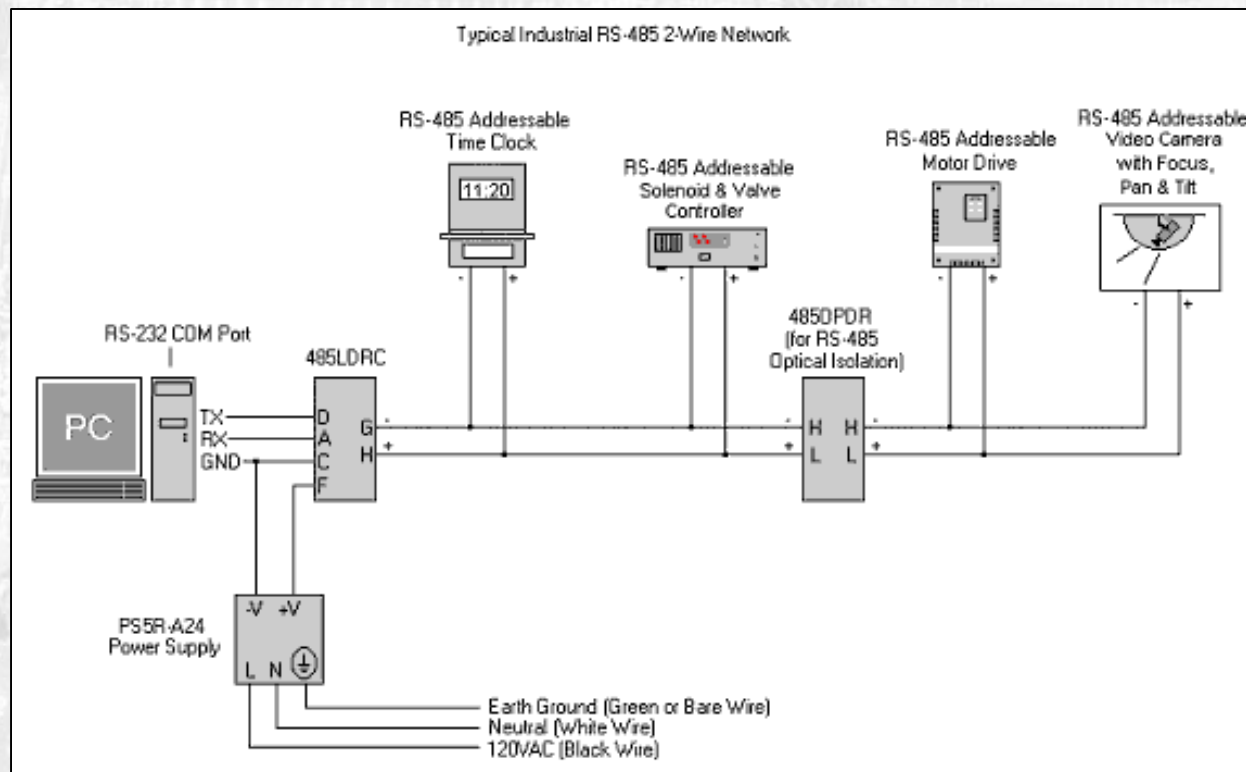


- Formato de trama:



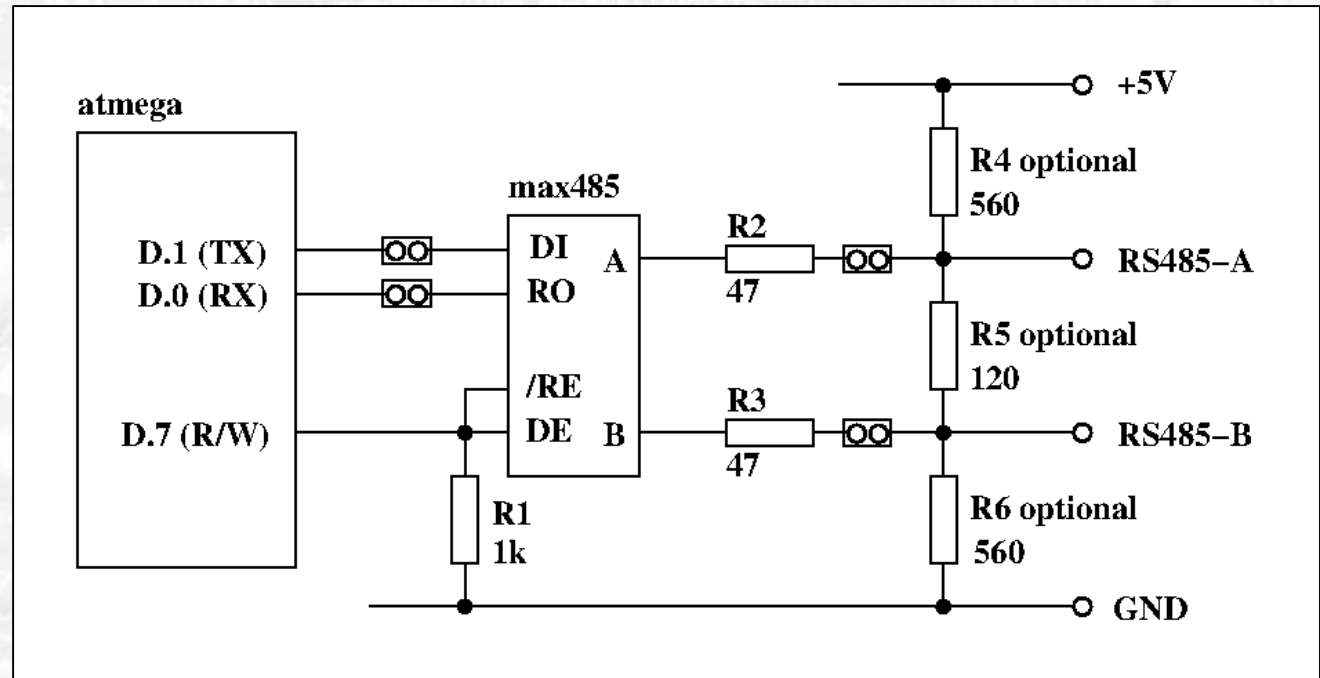
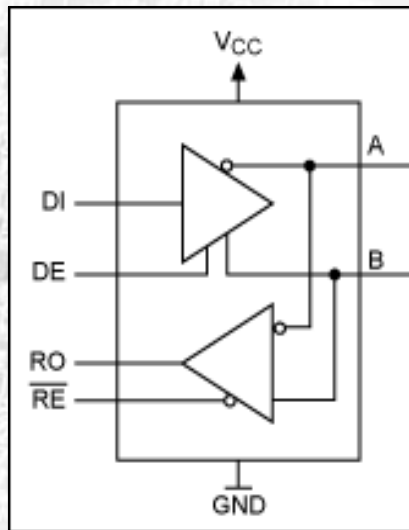
# Estándar RS-485

- Ejemplo de sistema RS-232/RS-485 (half duplex):



# Estándar RS-485

- Ej. Transceiver RS-485: Max485



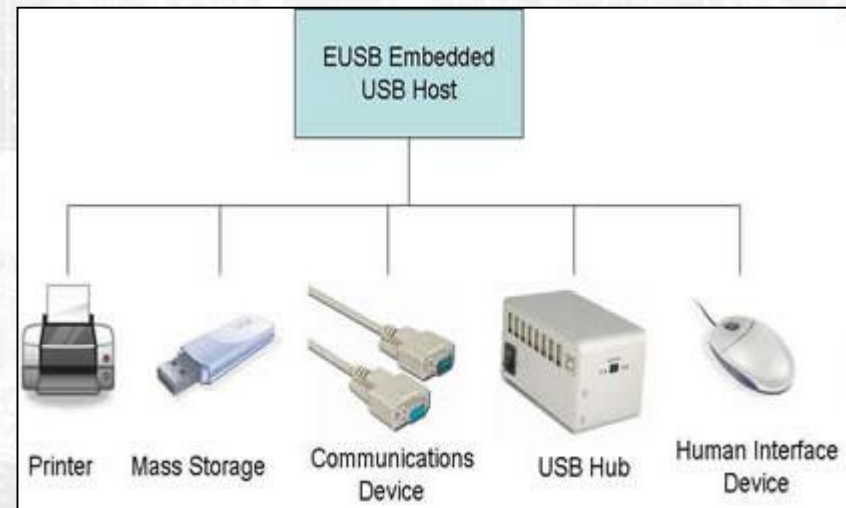
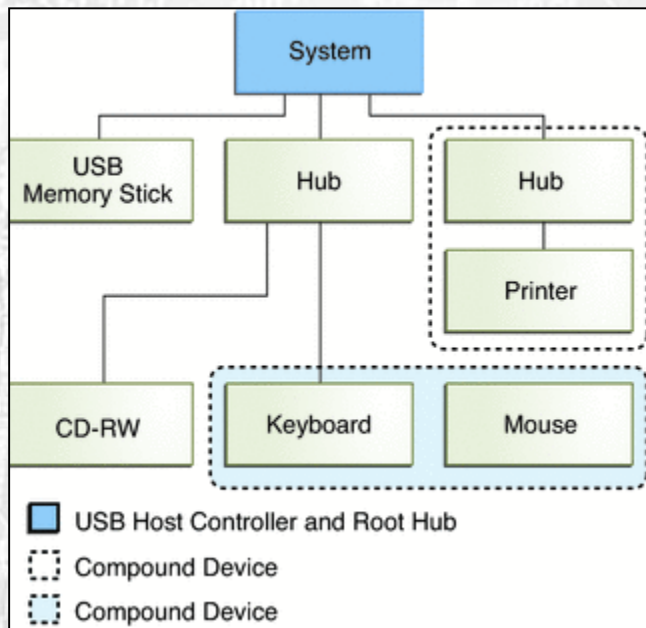


The background of the slide is a detailed, high-contrast image of a printed circuit board (PCB). It shows various electronic components such as integrated circuits, capacitors, and resistors, along with the intricate network of copper traces. The image is slightly blurred and has a monochromatic, high-contrast aesthetic. A dark blue horizontal band is superimposed over the middle of the image, containing the title text.

# **Estándar USB**

# Estándar USB

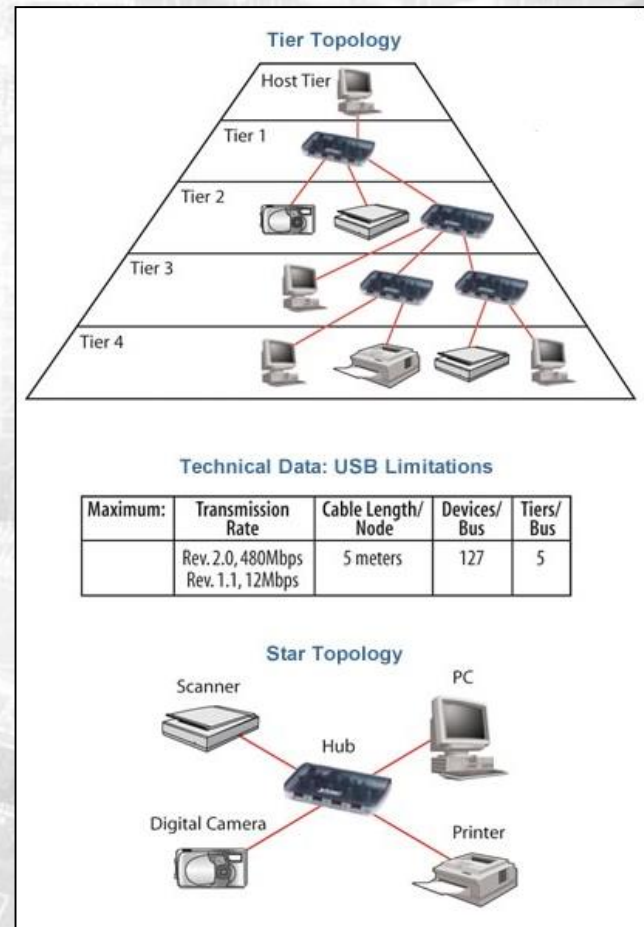
- **Universal Serial Bus (USB):** especificación para comunicar dispositivos con un controlador host.



- **Surgió para reemplazar puertos serie y paralelos.**

# Estándar USB

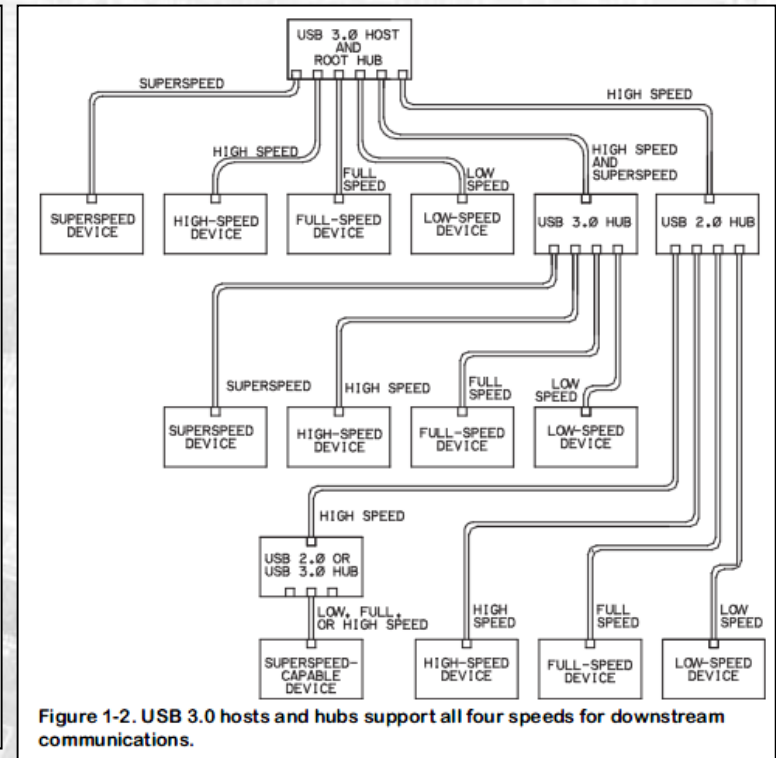
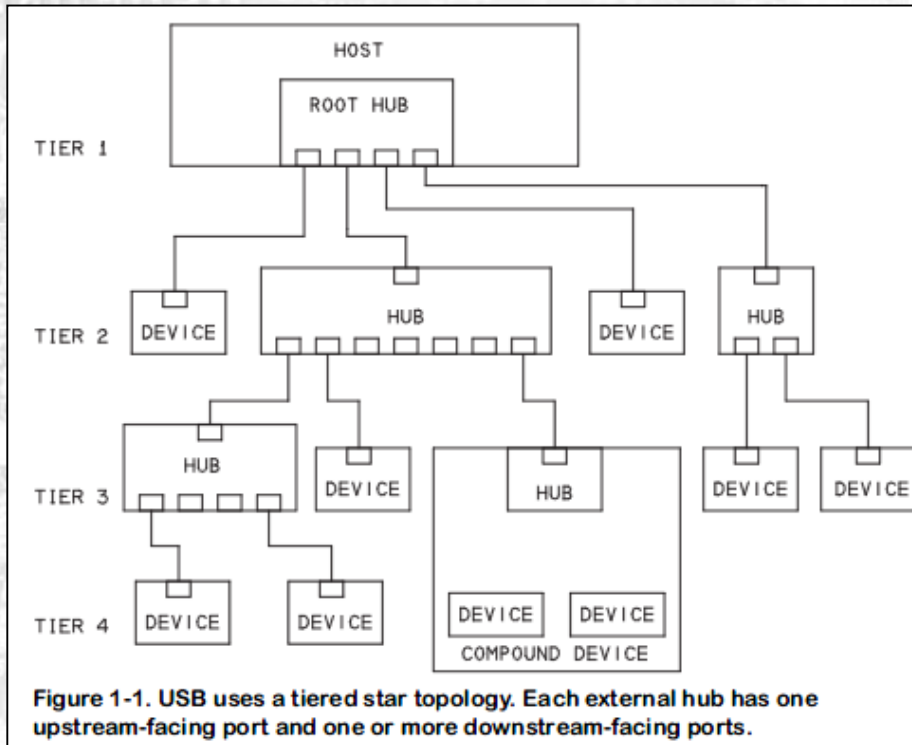
- **USB:**
  - Topología en estrella escalonada.
  - Múltiples hubs definen niveles (hasta 5).
  - Cada host provee controladores de host y c/u admite hasta 127 dispositivos.
  - Host oficia de master (controla las transferencias).





# Estándar USB

- **Universal Serial Bus (USB): topología y diferentes velocidades (USB 1.1, 2.0 y 3.0).**



# Estándar USB

- **USB: Los dispositivos se componen de subdispositivos lógicos (uso de descriptores)**
  - implementan una o varias configuraciones
  - implementan una o varias funciones (interface)
  - dirección propia y comunicación con endpoints vía pipes.

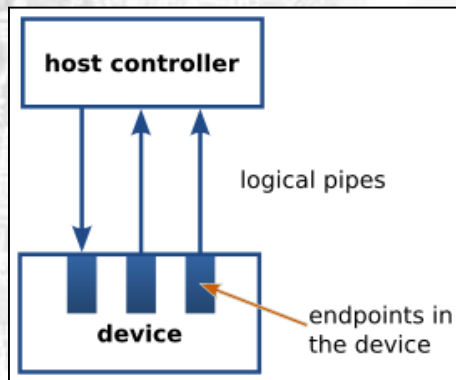
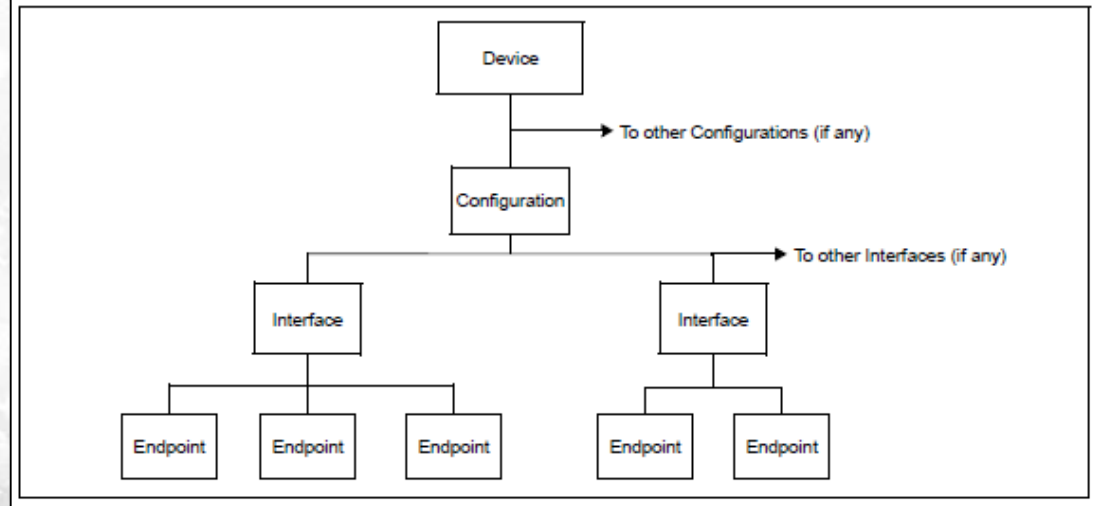


FIGURE 17-13: USB LAYERS



# Estándar USB

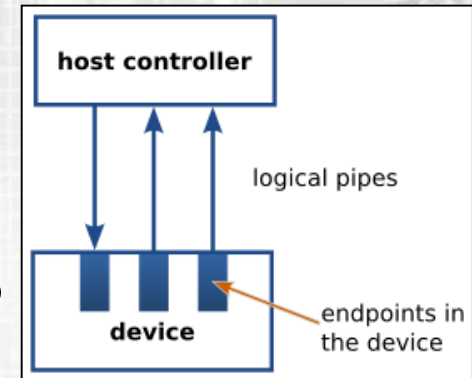
- **Diversas clases de dispositivos: para identificar la funcionalidad y cargar el driver adecuado.**

Class	Usage	Description	Examples
00h	Device	Unspecified	(Interface descriptors are used for determining the required drivers.)
01h	Interface	Audio	Speaker, microphone, sound card, MIDI
02h	Both	Com.and CDC Ctrl	Ethernet adapter, modem
03h	Interface	Human interf. device (HID)	Keyboard, mouse, joystick
05h	Interface	Physical Interf. Device (PID)	Force feedback joystick
06h	Interface	Image (PTP / MTP)	Webcam, scanner
07h	Interface	Printer	Laser printer, inkjet printer, CNC machine
08h	Interface	Mass storage	USB flash drive, card reader, digital audio player, digital cam, ext drive
09h	Device	USB hub	Full bandwidth hub
0Ah	Interface	CDC-Data	(This class is used together with class 02h)
0Bh	Interface	Smart Card	USB smart card reader
0Dh	Interface	Content security	Fingerprint reader
0Eh	Interface	Video	Webcam
0Fh	Interface	Personal Healthcare	Pulse monitor (watch)
DCh	Both	Diagnostic Device	USB compliance testing device
E0h	Interface	Wireless Controller	Wi-Fi adapter, Bluetooth adapter
EFh	Both	Miscellaneous	ActiveSync device
FEh	Interface	Application-specific	IrDA Bridge, Test & Measurement Class (USBTMC), USB DFU
FFh	Both	Vendor-specific	(This class code indicates that the device needs vendor specific drivers.)



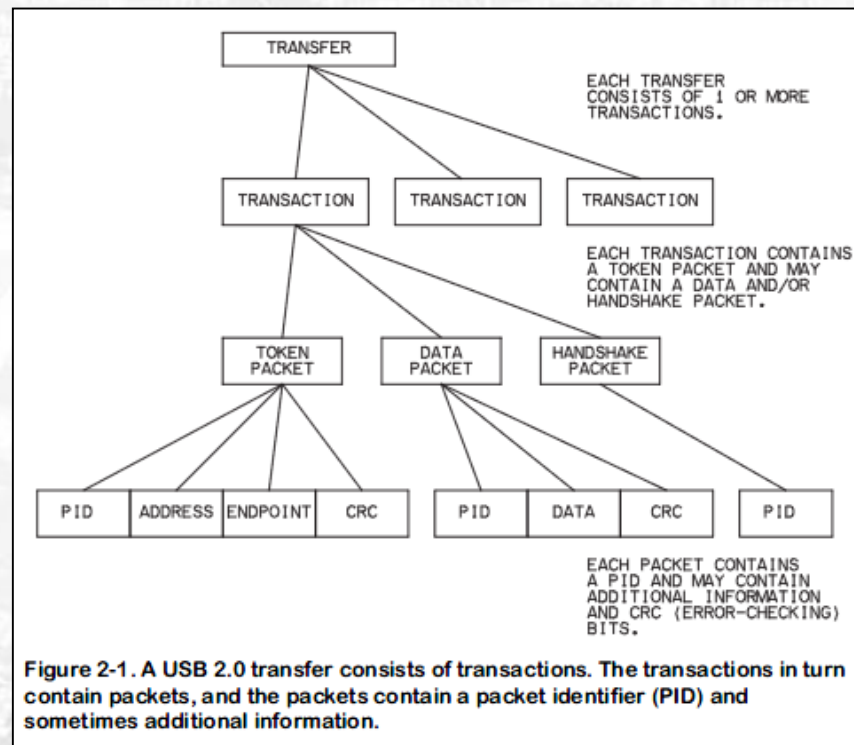
# Estándar USB

- **Endpoints:**
  - en los dispositivos (el host no posee)
  - extremos de cada comunicación
  - buffer en el dispositivo que transmite o recibe datos
  - se definen con un número (0-15) y un sentido
    - IN endpoint: provee datos al host
    - OUT endpoint: recibe datos provenientes del host
  - cada dispositivo debe tener configurado su endpoint 0 para control (IN-OUT).
- **Limitaciones en cantidad de endpoints en función de tipo de dispositivo: low, high o full speed.**



# Estándar USB

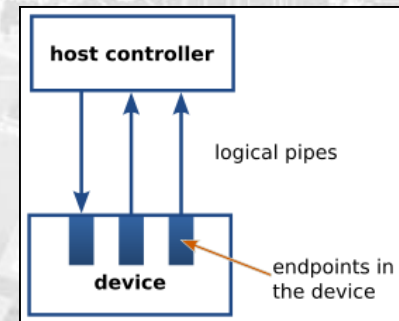
- Flujo de datos en el bus mediante transferencias, transacciones y paquetes:





# Estándar USB

- **Tipos de transferencia:**
  - isocrónica (isochronous): garantiza tasa de transferencia a costa de perder datos (por ej. video en tiempo real)
  - usando interrupciones (interrupt): latencia de respuesta acotada (ej. mouse, teclado)
  - masivas (bulk): largas y esporádicas (consumen todo el ancho de banda disponible – por ej. Hard Drives)
  - de control (control): para enviar comandos cortos
- **Pipes: establecidos por el host:**
  - stream pipes: unidireccional (isocrónica, interrupciones o masivas).
  - message pipes: bidireccional (control).





# Estándar USB

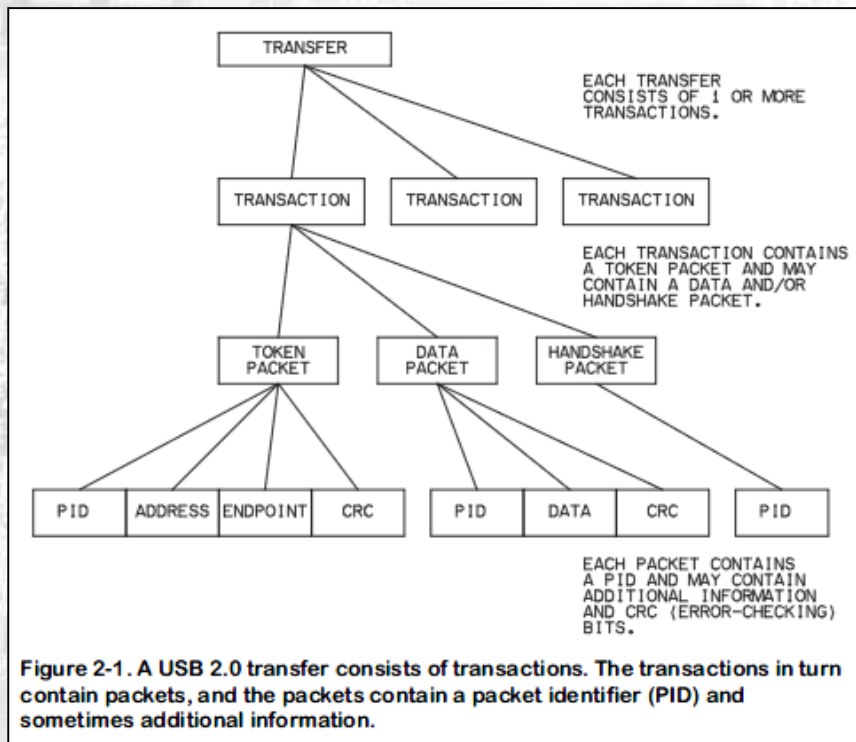
- Tipos de transferencia:

Table 2-1: Each of the USB's four transfer types is suited for different uses.

Transfer Type	Control	Bulk	Interrupt	Isochronous
Typical Use	Identification and configuration	Printer, scanner, drive	Mouse, keyboard	Streaming audio, video
Support required?	yes	no	no	no
Low speed allowed?	yes	no	yes	no
Maximum packet size; maximum guaranteed packets/interval (SuperSpeed).	512; none	1024; none	1024; 3 / 125 $\mu$ s	1024; 48 / 125 $\mu$ s
Maximum packet size; maximum guaranteed packets/interval (high speed).	64; none	512; none	1024; 3 / 125 $\mu$ s	1024; 3 / 125 $\mu$ s
Maximum packet size; maximum guaranteed packets/interval (full speed).	64; none	64; none	64; 1 / ms	1023; 1 / ms
Maximum packet size; maximum guaranteed packets/interval (low speed).	8; none	not allowed	8; 1 / 10 ms	not allowed
Direction of data flow	IN and OUT	IN or OUT	IN or OUT (IN only for USB 1.0)	IN or OUT
Reserved bandwidth for all transfers of the type	10% at low/full speed, 20% at high speed & SuperSpeed	none	90% at low/full speed, 80% at high speed and SuperSpeed (isochronous and interrupt combined, maximum)	
Message or Stream data?	message	stream	stream	stream
Error correction?	yes	yes	yes	no
Guaranteed delivery rate?	no	no	no	yes
Guaranteed latency (maximum time between transfers attempts)?	no	no	yes	yes

# Estándar USB

- Componentes de una transferencia:

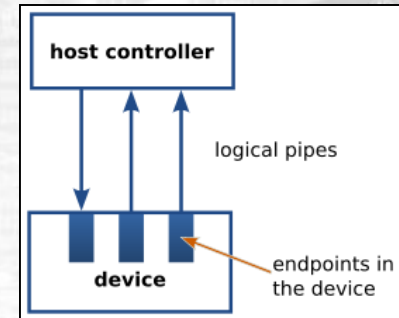


**Table 2-2: Each USB 2.0 transaction has two or three phases. (Not shown are additional transactions required for split transactions, the PING protocol used in some transfers, and the PRE packet that precedes downstream, low-speed packets.)**

Transfer Type	Number and Direction of Transactions		Phases (packets)
Control	Setup Stage	One (SETUP)	Token
			Data
			Handshake
	Data Stage	Zero or more (IN or OUT)	Token
			Data
			Handshake
	Status Stage	One (opposite direction of the transaction(s) in the Data stage or IN if there is no Data stage)	Token
			Data
			Handshake
Bulk	One or more (IN or OUT)		Token
			Data
			Handshake
Interrupt	One or more (IN or OUT)		Token
			Data
			Handshake
Isochronous	One or more (IN or OUT)		Token
			Data

# Estándar USB

- **Transacciones:** cada transacción se inicia con un paquete que indica la dirección del dispositivo, el número de endpoint y su sentido:



Transaction Type	Source of Data	Types of Transfers that Use the Transaction Type	Contents
IN	device	all	data or status information
OUT	host	all	data or status information
Setup	host	control	a request

- Las transacciones de Setup son similares a las OUT pero no pueden ser rechazadas por el device. Inician una transferencia de control.



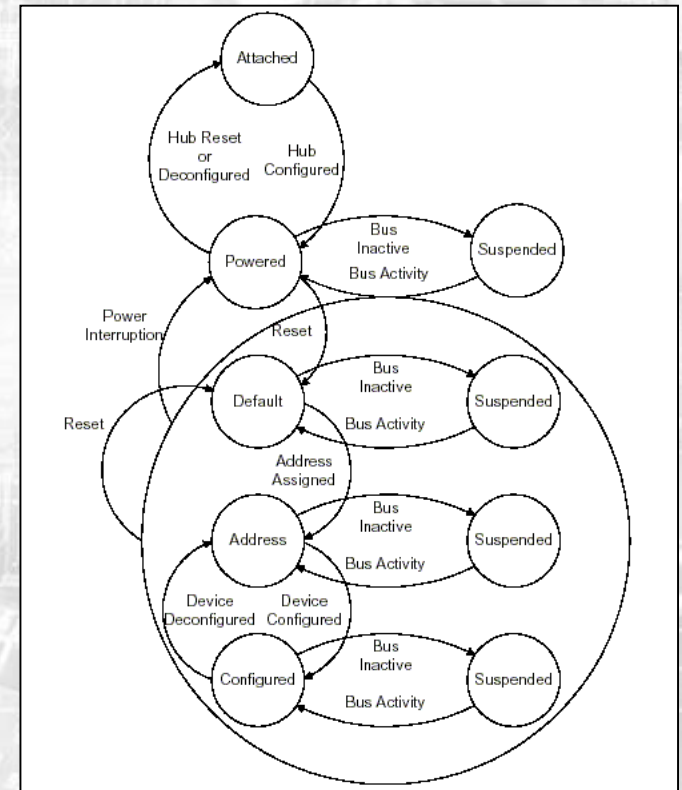
# Estándar USB

- Paquetes USB:

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

# Estándar USB

- El dispositivo debe respetar el protocolo USB y sus restricciones temporales,
  - Responder a los paquetes del host (cada cierto tiempo).
  - En caso contrario, el host puede suponer una desconexión del dispositivo y la consecuente pérdida de funcionalidad digital.



- Ej: Invocación a función USBTasks cada cierto tiempo (USB stack/API de Microchip).

# Estándar USB

- **Proceso de enumeración de dispositivos: al conectar por primera vez un dispositivo, el host:**
  - negocia tasa de transferencia,
  - asigna dirección única de 7 bits al dispositivo,
  - lee descriptores del dispositivo,
  - asigna y carga un driver para el dispositivo,
  - selecciona una configuración de dispositivo (requerimientos de alimentación, interfaces, etc.).
- **Descriptores: permiten al Host descubrir las características del dispositivo que se conecta (mediante transferencias de control).**



# Estándar USB - Descriptores

- Todos los dispositivos USB deben responder a las solicitudes del host (descriptores estándar USB).
- Veremos descriptores para:
  - Device
  - Configuration
  - Interface
  - Endpoint
- Existen descriptores adicionales (Interface Association, SuperSpeed endpoints, String, Binary Object Store, etc.), y también particulares para ciertas clases de dispositivos (por ej. HID).

# Estándar USB - Descriptores

- **Descriptor USB:** punteros a los descriptores e información del soporte USB.

**Table 4-1: The bDescriptorType field in a descriptor contains a value that identifies the descriptor type.**

bDescriptorType	Descriptor Type	Required?
01h	device	Yes.
02h	configuration	Yes.
03h	string	No, unless a driver requires it. Optional descriptive text.
04h	interface	Yes.
05h	endpoint	Yes, to use other than endpoint zero.
06h	device_qualifier	Yes for devices that support both full and high speeds. Not allowed for other devices.
07h	other_speed_configuration	Yes for devices that support both full and high speeds. Not allowed for other devices.
08h	interface_power	No (proposed but never approved or implemented).
09h	OTG	Yes for On-The-Go devices.
0Ah	debug	No.
0Bh	interface_association	Yes for some composite devices.
0Ch	security	For wireless devices.
0Dh	key	
0Eh	encryption type	
0Fh	binary device object store (BOS)	Yes for SuperSpeed devices, wireless devices, and devices that support link power management.
10h	device capability	
11h	wireless endpoint companion	For wireless devices.
30h	SuperSpeed_endpoint_companion	Yes for SuperSpeed devices. Not supported for other speeds.



# Estándar USB - Descriptores

- Descriptores de dispositivo: product ID, fabricante, clase, etc.

Table 4-2: The device descriptor identifies the product and its manufacturer, sets the maximum packet size for endpoint zero, and can specify a device class.

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



# Estándar USB - Descriptores

- Descriptores de configuración: requerimientos de consumo (requisitos de corriente en el bus)

Table 4-5: The configuration descriptor specifies the maximum amount of bus current the device will require and gives the total length of the subordinate descriptors.

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).

# Estándar USB - Descriptores

- Descriptores de interface: número de endpoints para la interface y clase USB (p/dis. con clase definida por la interface).

Table 4-8: The interface descriptor specifies the number of subordinate endpoints and may specify a USB class.

Offset (decimal)	Field	Size (bytes)	Description
0	bLength	1	Descriptor size in bytes (09h)
1	bDescriptorType	1	The constant Interface (04h)
2	bInterfaceNumber	1	Number identifying this interface
3	bAlternateSetting	1	A number that identifies a descriptor with alternate settings for this bInterfaceNumber.
4	bNumEndpoints	1	Number of endpoints supported not counting endpoint zero
5	bInterfaceClass	1	Class code
6	bInterfaceSubclass	1	Subclass code
7	bInterfaceProtocol	1	Protocol code
8	iInterface	1	Index of string descriptor for the interface



# Estándar USB - Descriptores

- Descriptores de endpoint:

**Table 4-10: The endpoint descriptor provides information about an endpoint address.**

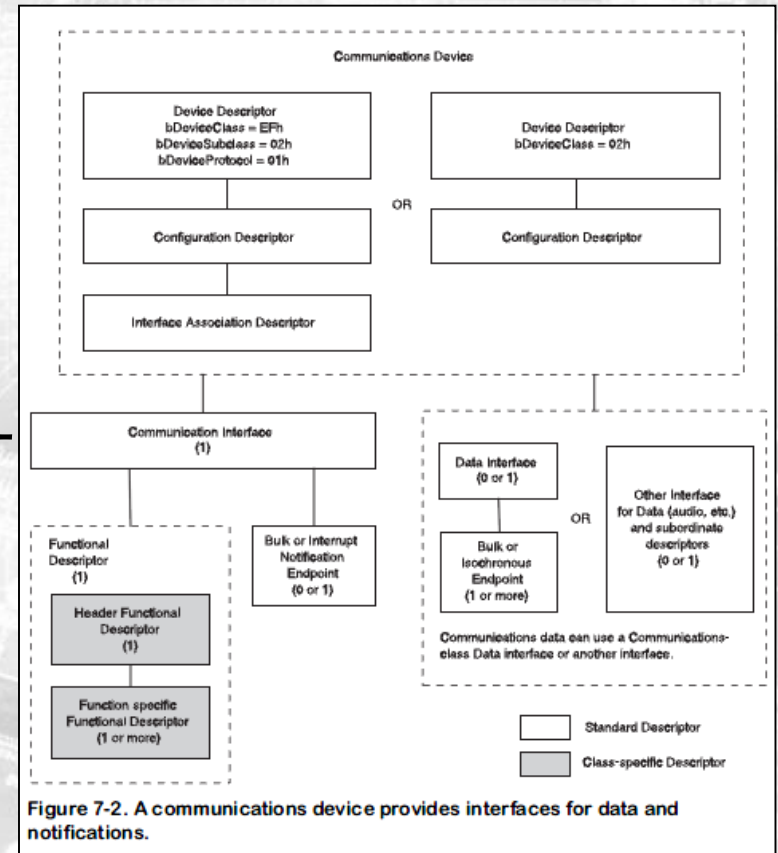
Offset (decimal)	Field	Size (bytes)	Description
0	bLength	1	Descriptor size in bytes (07h)
1	bDescriptorType	1	The constant Endpoint (05h)
2	bEndpointAddress	1	Endpoint number and direction
3	bmAttributes	1	Transfer type and supplementary information
4	wMaxPacketSize	2	Maximum packet size supported
6	bInterval	1	Service interval or NAK rate



# Estándar USB - Clases

- **CDC Class: Communications Device Class**

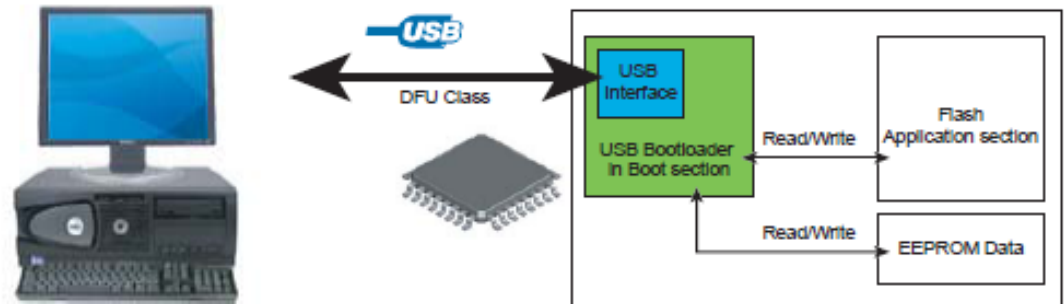
- Para dispositivos de comunicaciones: teléfonos, modems, terminales y adaptadores ISDN, dispositivos con puertos COM virtuales (ej Arduino Uno)...
- Para dispositivos con funciones de red: ADSL modems, cablemodems, adaptadores y hubs ethernet...
- Administrar dispositivos, llamadas, transmitir datos y notificaciones.



# Estándar USB - Clases

- **DFU Class – Ej. ATmega16U2:**
  - Permite colocar un dispositivo en modo de actualización del firmware.
  - Solicitudes específicas en el protocolo.

Figure 2-1. Physical Environment



Request	Description
DFU_Detach	Detach and re-attach to the bus or wait for bus reset within the time period specified in the DFU Functional descriptor. On reattach or a reset within the specified time, enumerate using the DFU-mode descriptors.
DFU_Dnload	Accept new firmware in the request's Data stage. A request with wLength = 0000h means all of the firmware has been transferred.
DFU_Upload	Send firmware to the host in the request's Data stage.
DFU_GetStatus	Return status and error information. On error, enter the dfuError state.
DFU_ClrStatus	Clear the dfuError state reported in response to a DFU_GetStatus request and enter the dfuIdle state.
DFU_GetState	Same as DFU_GetStatus but with no change in state on error.
DFU_Abort	Return to the dfuIdle state.



# Estándar USB - Clases

- **HID Class: Human Interface Device**
  - Incluye mouses, teclados, joysticks, etc.
  - Los SOs en los hosts, suelen tener drivers para HIDs.
  - Limitados a transferencias control e interrupt.
  - Los datos HID viajan en reports (estructuras bien definidas).
    - Input item: lleva info hacia el host.
    - Output item: lleva info hacia el dispositivo.
    - Feature item: es bidireccional.
  - Solicitudes específicas en el protocolo para obtener reports.



# Estándar USB - Clases

- **Mass Storage (MSC):**
  - Para dispositivos de almacenamiento masivo.
  - Discos rígidos, unidades de CD/DVD, cámaras que presentan su contenido mediante un sistema de archivos, etc.
  - Usan transferencias bulk para intercambiar datos.
- **Media Transfer Protocol (MTP)**
  - Extensión al Picture Transfer Protocol (PTP - USB Image Class) usado para transferir imágenes (cámaras digitales)
  - Para transferir archivos transaccionalmente sin requerir acceso exclusivo al medio (ej. acceso a memoria externa en smartphones sin bloquear Apps).
  - Utilizan transferencias de tipo bulk e interrupt.

# Estándar USB - Clases

- **Otras clases:**
  - Audio (streaming de audio / MIDI)
  - IrDA Bridge: intercambio de datos por enlaces infrarrojos.
  - Personal Health Care
  - Printer: impresoras, CNCs
  - Smart Cards
  - Still Image Capture: cámaras, scanners, PTP, MTP, etc.
  - Video
  - Etc...
- **Dispositivos genéricos (HID, CDC, MSC, etc)**
- **Dispositivos Vendor-Specific (drivers específicos).**



# Estándar USB

- El estándar también define la forma y las señales de los conectores.

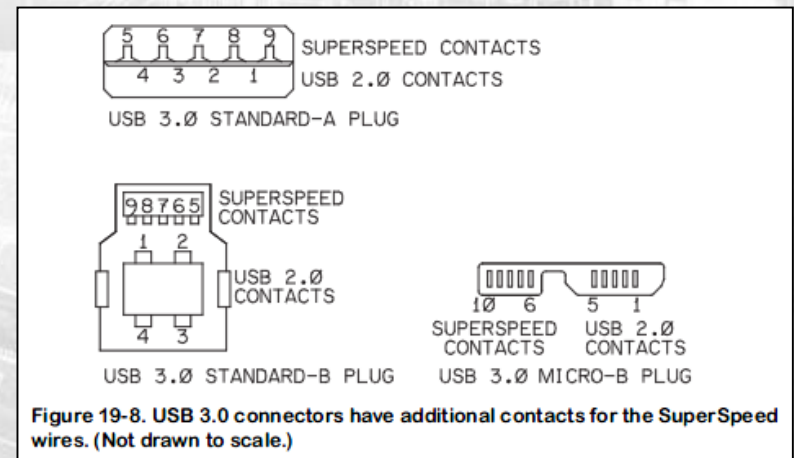
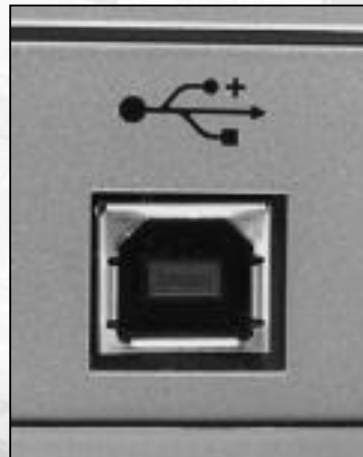
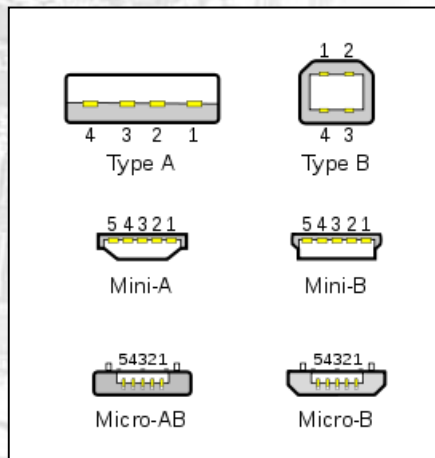
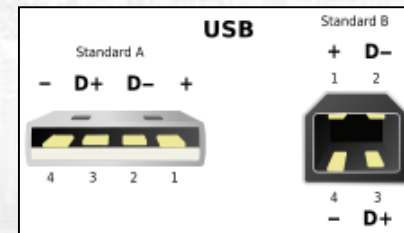
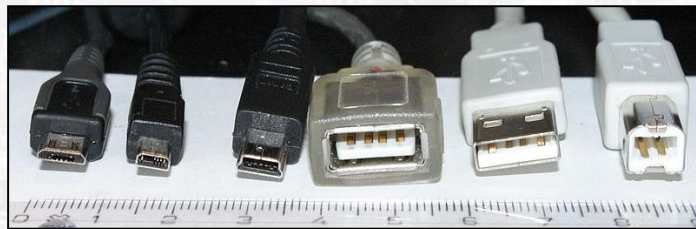


Figure 19-8. USB 3.0 connectors have additional contacts for the SuperSpeed wires. (Not drawn to scale.)

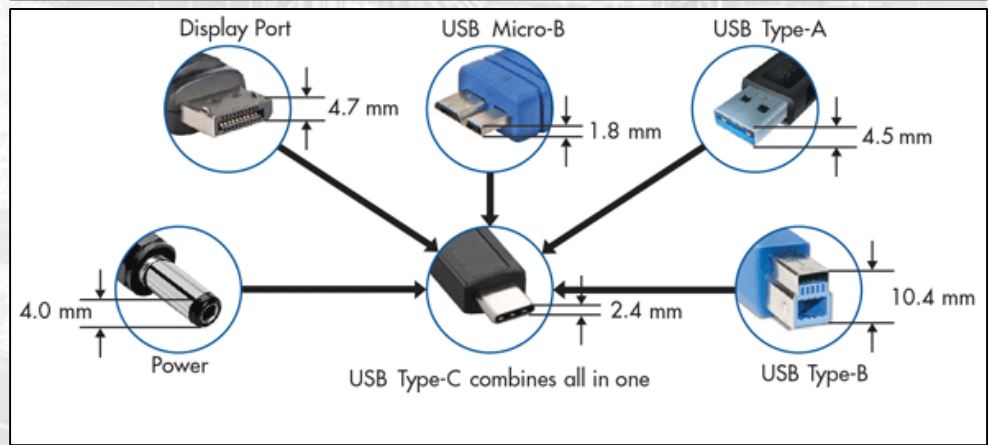
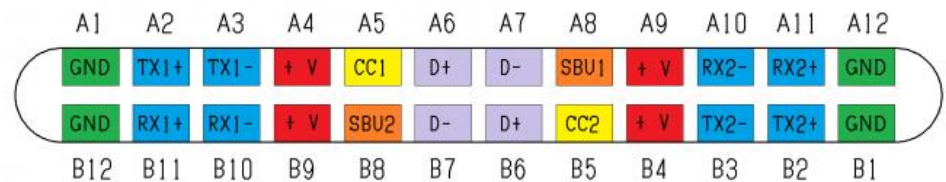


# Estándar USB

- El estándar también define la forma y las señales de los conectores.



## USB Type-C Connector Pin Assign



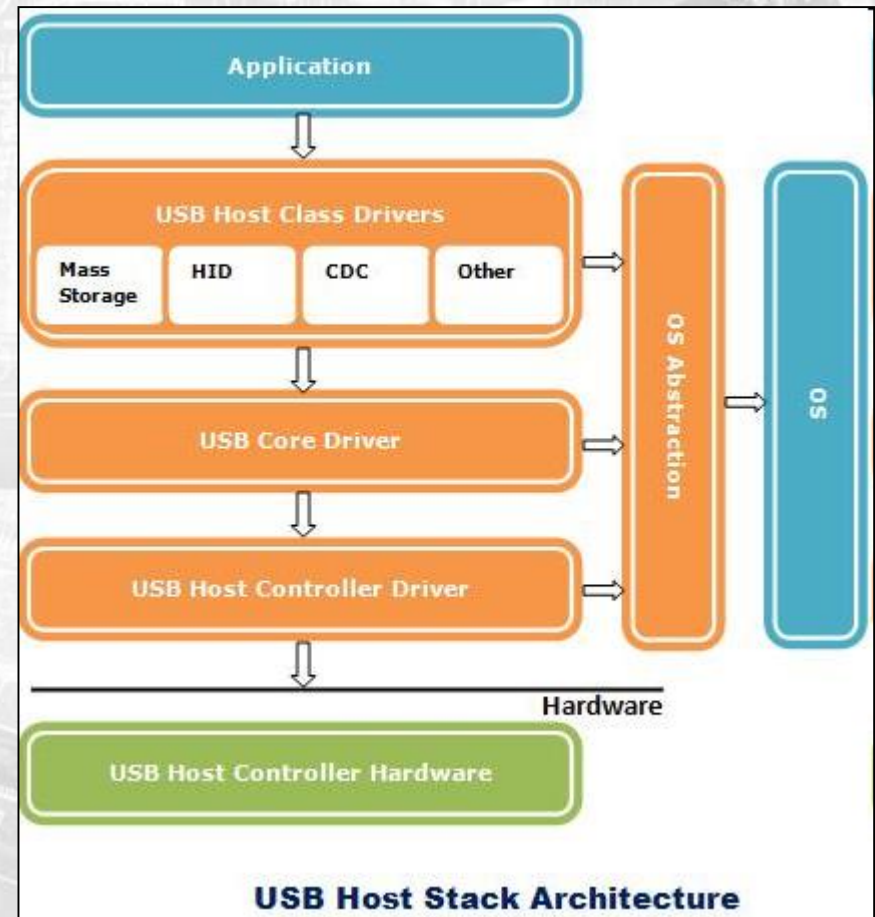
# Estándar USB

- También se definen consideraciones en cuanto a consumo y alimentación de los puertos y dispositivos.
  - Devices bus-powered (5V, 3.3V).
  - Devices self-powered.
- Numerosas variantes del estándar:
  - Distintas versiones: USB 1.0, 1.1, 2.0 y 3.0 (SuperSpeed)
  - Sleep & charge: Para carga con hosts suspendidos.
  - Powered USB: USB para alimentación.
  - USB On the Go (OTG): Para conexión de 2 dispositivos sin hosts (par a par): Uso de embedded hosts.
  - Wireless USB.



# Estándar USB

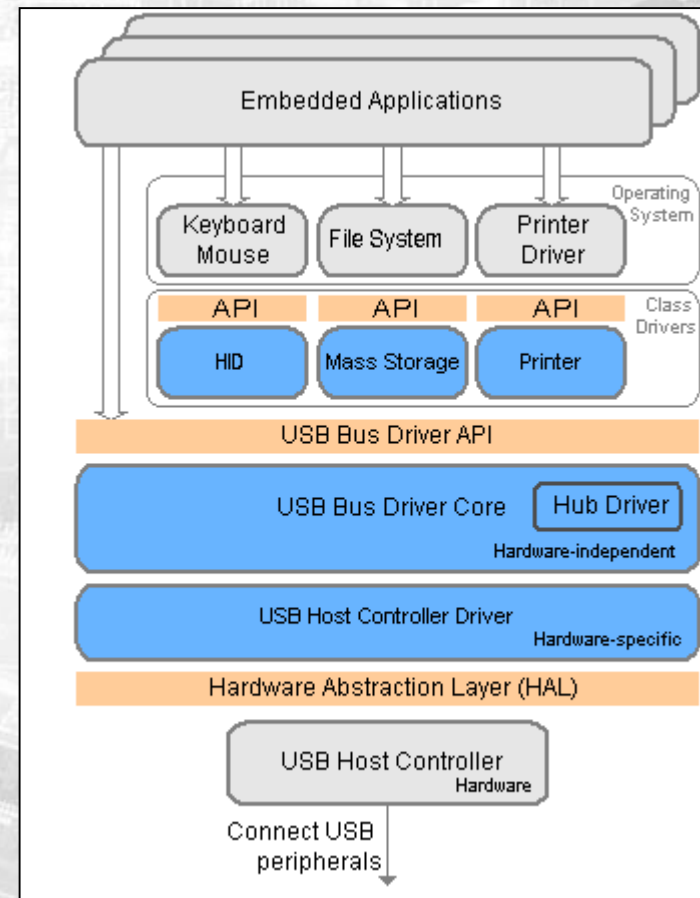
- Dada la complejidad del protocolo, se suele manejar con dispositivos específicos en los SoC y mediante librerías que gestionan los diversos niveles en el stack.



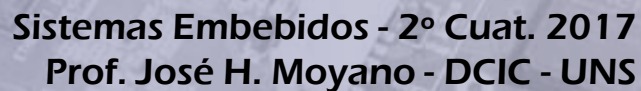


# Estándar USB


- Dada la complejidad del protocolo, se suele manejar con dispositivos específicos en los SoC y mediante librerías que gestionan los diversos niveles en el stack.



- Ej: USB DFU/CDC – Arduino Uno



# GrIDSE

The background of the slide is a detailed, high-resolution image of a printed circuit board (PCB). It shows various electronic components such as integrated circuits, capacitors, and resistors, along with a complex network of copper traces. The image is slightly faded and serves as a technical backdrop for the text.

# **Otros estándares de comunicación serie**



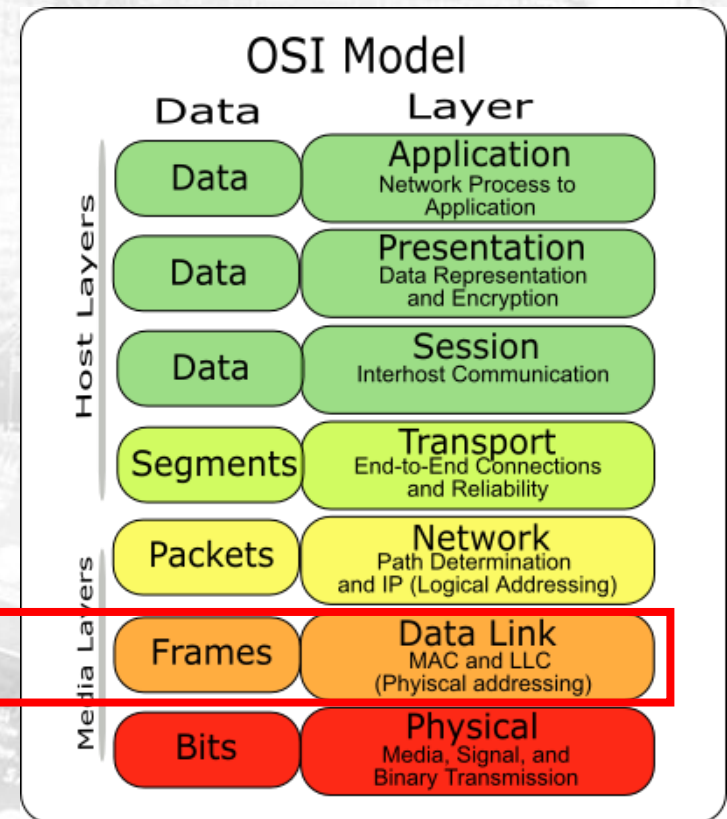
# Otros estándares de comunicación

- Otros estándares de comunicación serie adicionales a los vistos son:

- Ethernet
- Point to Point Protocol (PPP)
- Serial Line Internet Protocol (SLIP)
- Wireless LAN

- No los veremos en detalle.

**Eth – PPP – SLIP –  
WLAN**



# Referencias

- Atmel AVR ATmega328P Datasheet.
- Atmel AVR ATmega16U2 Datasheet.
- Atmel USB DFU Bootloader Datasheet.
- Axelson, J. *USB Complete: The Developer's Guide (4th Ed.)*. Lakeview Research. 2009. ISBN: 978-1931448086. Capítulos 1 a 7, 11 y 12.
- Noergaard, T. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*. Newnes. 2005. ISBN: 978-0750677929. Capítulo 6.
- PIC18F2455/2550/4455/4550 Data Sheet – Sec. 17 – USB
- <http://en.wikipedia.org>