

I NUOVI BUS DI COMUNICAZIONE SERIALI E LE RETI DI MISURA WIRELESS

Luigi Ferrigno
ferrigno@unicas.it



*Gruppo Misure Elettriche ed
Elettroniche*

Facoltà di Ingegneria, DAEIMI.

Università degli Studi di Cassino

IL BUS DI
COMUNICAZIONE
SERIALE
USB

Il sistema USB

L'Universal Serial Bus (USB) è un particolare tipo di bus sviluppato nel 1995 (nel 1998 per Personal Computer) da un insieme d'aziende (Compaq, Digital, IBM, Intel, Microsoft, NEC e Northern Telecom) allo scopo di utilizzare un'unica interfaccia per collegare periferiche accessorie, sostituendo le porte parallele e seriali e altri ingressi e uscite esistenti sui computer.

Il sistema USB

Generalità

Le tre lettere USB sono le iniziali d'Universal Serial Bus, che indicano come la comunicazione dei dati avviene in modo seriale o in altre parole un bit alla volta. Questa soluzione porta vantaggi concreti rispetto agli standard paralleli preferiti in passato, soprattutto perché facilita la produzione dei chip necessari alla gestione con velocità elevatissime e consumi minimi, mentre è ancora molto difficile ottenere una schiera di circuiti perfettamente identici tra loro. Ciò rende difficile e costoso il collegamento parallelo, dove un intero gruppo di bit deve viaggiare lungo il cavo alla stessa velocità.

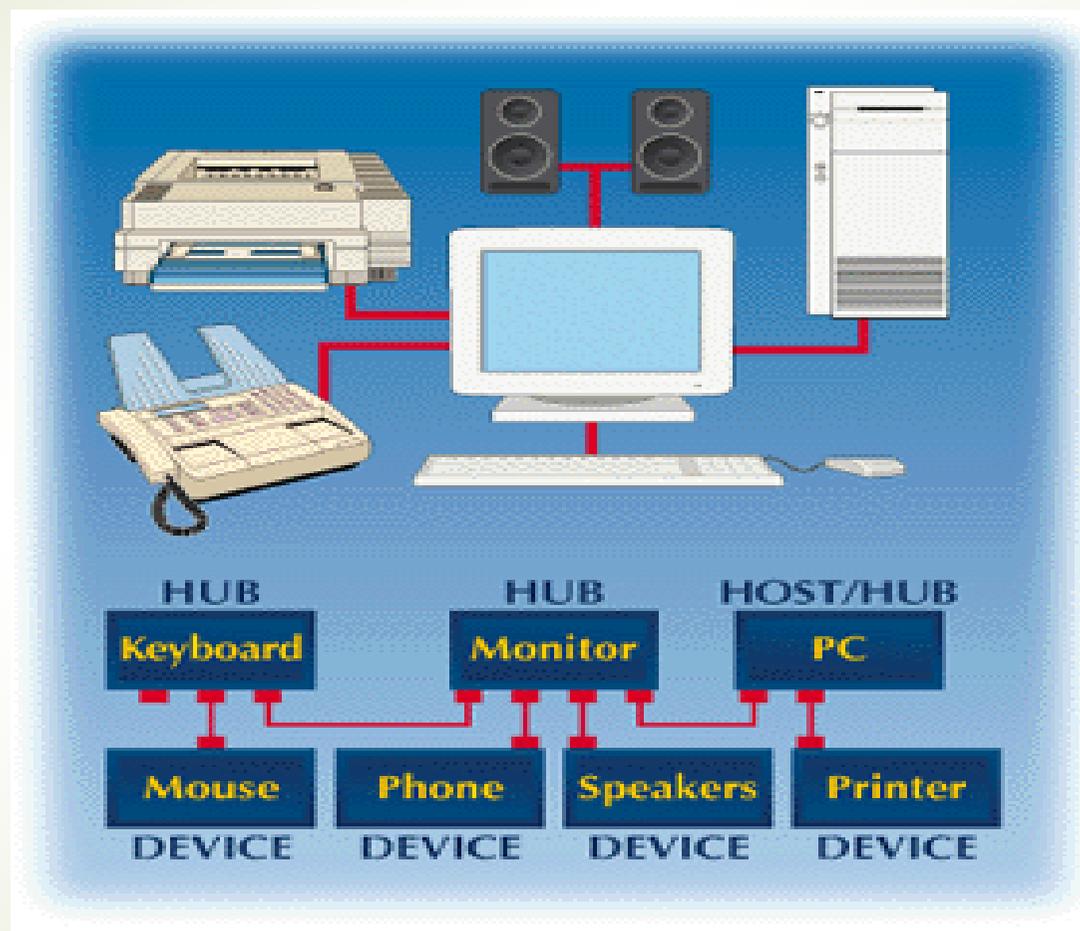
Il sistema USB

La versione 1.0 (96) non aveva previsto adeguate protezioni contro gli inconvenienti accidentali che possono verificarsi nella realtà, come i picchi d'assorbimento causati dall'inserzione del cavo di una periferica nel connettore. Per raggiungere un vero successo, l'interfaccia ha dovuto attendere il rilascio delle specifiche USB 1.1 (98), che ancora oggi rappresentano il riferimento tecnico adottato dalla quasi totalità delle periferiche e delle schede madri.

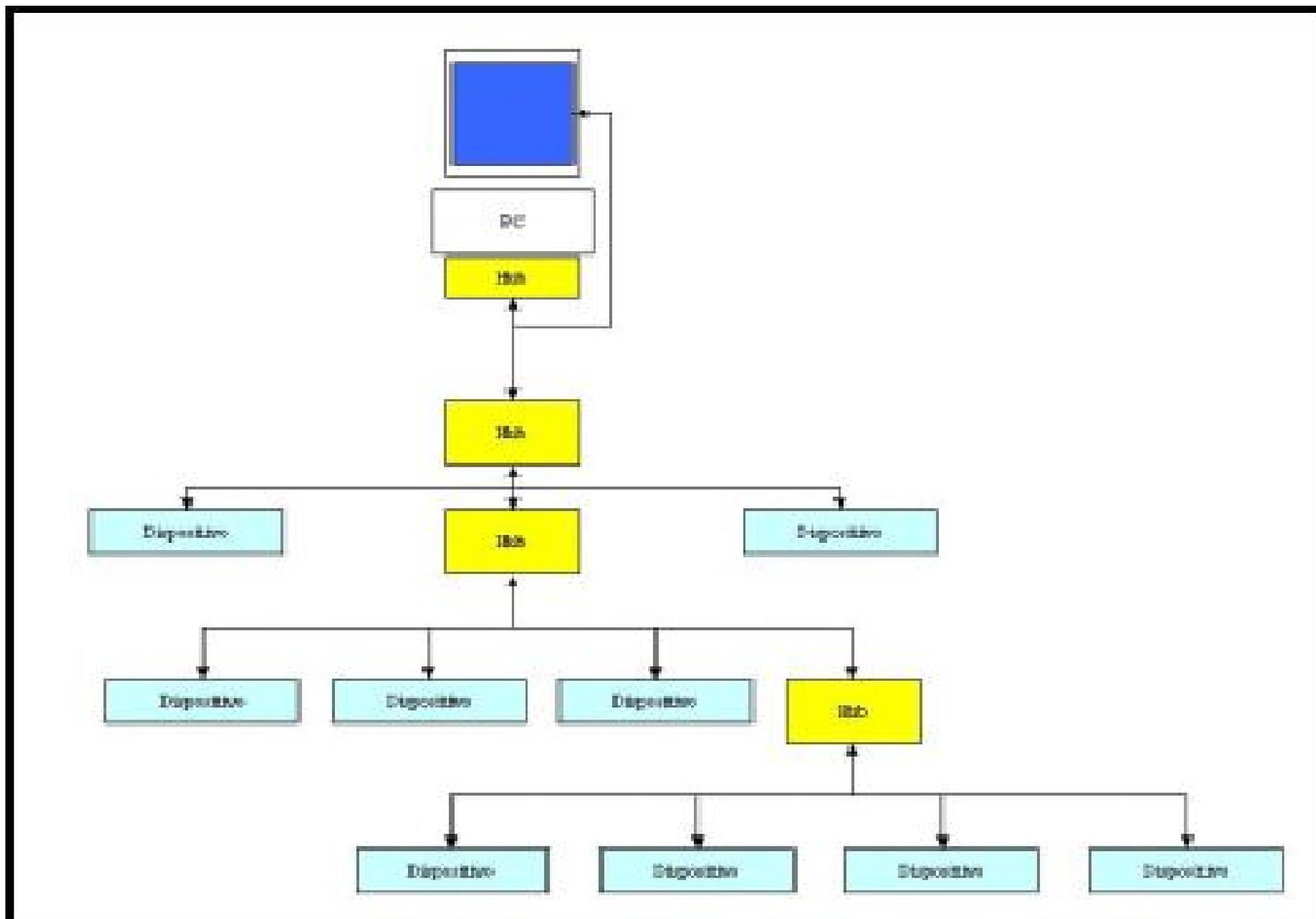
La versione 2.0 dello standard, pubblicata nell'Aprile del 2000 rappresenta un notevole passo in avanti che aumenta di 40 volte la velocità di trasferimento dei dati rispetto alla connessione USB 1.1, arrivando ad una cadenza di 480 Mbit/s senza però cambiare il protocollo di trasferimento dei dati.

Il sistema USB

Allo stato attuale è possibile connettere al PC attraverso la porta USB fino a 127 dispositivi, contro un numero massimo pari a poche unità degli altri bus. I dispositivi sono collegati al PC in una struttura a “radice” (stella) con un massimo di 6 livelli di profondità, in cui il vertice principale è occupato dal PC detto anche host.



Possibile struttura del bus USB



Il sistema USB

La topologia del bus è a stella: i dispositivi periferici non possono comunicare tra loro, ma rispondono unicamente ai comandi e alle direttive dell'Host. Non è possibile servirsi di cavi e commutatori ordinari per collegare simultaneamente lo stesso dispositivo a più computer: il protocollo di comunicazione funziona solo se l'Host è unico.

Confrontato con gli altri bus l'USB risulta molto più flessibile.

Per esempio: *Consente la modalità "Hot Plug & Play"*

È permesso connettere e disconnettere periferiche al PC senza riavviarlo. Con gli altri tipi di interfacce la connessione di una periferica col PC acceso spesso comporta il blocco del sistema operativo o addirittura in casi estremi il danneggiamento dell'hardware.

Il sistema USB

Per esempio: *Permette di alimentare le periferiche.*

Il connettore dell'USB dispone di quattro terminali: due per il trasferimento dei dati vero e proprio, e altri due dedicati all'alimentazione (5V), che possono essere sfruttati dalle periferiche per prelevare l'energia di cui necessitano. Questa possibilità comporta l'assenza di alimentatori esterni, riducendo notevolmente il volume delle connessioni esterne al PC.

In realtà da questo punto di vista esistono due categorie di periferiche USB: *Self Powered e Bus Powered.*

Solo le periferiche Bus Powered vengono alimentate dal bus, rientrano in questa categoria le periferiche che richiedono una potenza inferiore a 2.5 W.

Il sistema USB

Nella pratica il bus USB può quindi alimentare dispositivi come modem, scanner, webcam senza problemi, mentre ancora non è in grado di farlo per periferiche come stampanti o monitor, che allo stato attuale richiedono molta potenza.

L'aumento di flessibilità comporta però una **maggior complicazioni a livello di protocollo di trasmissione**, ossia tutte le regole che periferica e PC devono rispettare affinché il trasferimento dei dati avvenga in modo corretto .

Il sistema USB

Lo Standard USB 1.1 prevede due tipi di dispositivi:

High-Speed Devices: i dispositivi high-speed sfruttano un funzionamento completo (full-speed), cioè mirato a raggiungere la massima velocità di trasferimento, potendo così inviare e ricevere dati ad una velocità massima di 12Mbit/s.

Low-Speed Devices: i dispositivi low-speed sono limitati ad una velocità massima di 1.5Mbit/s; ad essi sono indirizzati pacchetti informativi preceduti da un pacchetto “premessa” (Preamble Packet), per avvisare che la transazione che seguirà è destinata ad un dispositivo low-speed. Le porte low-speed vengono disabilitate durante trasferimenti full-speed, impegnando così i soli cavi full-speed per il flusso di comunicazione.

Il sistema USB

La richiesta sempre maggiore di banda passante per applicazioni dati multimediali ha portato all'aggiornamento dello standard ed al rilascio di un nuovo protocollo USB2.0.

La sua caratteristica innovativa principale è appunto la massima velocità raggiungibile 480 Mbit/s sostenibile solo dalle periferiche e dagli Host che gestiscono il protocollo USB2.0, certificato dalla presenza dell'apposito logo blu e rosso "Usb Hi-Speed" mostrato in figura.



Il sistema USB

Tipologia di periferiche

1. **Host** è il controller del bus e sovrintende a tutte le transizioni indirizzando e interrogando i device.
2. **Device USB** o function aggiunge capacità all'host può essere low speed o full speed è caratterizzato da un indirizzo e dai suoi endpoints
3. **Hub USB** è un replicatore di porte che rigenera i segnali

Il sistema USB

Host

La composizione logica del host include quanto segue:

- Controllore USB Host
- Sistema Software globale dell'USB (driver del USB, driver del controllore host e software host).
- Client

Oltre che la relativa posizione fisica speciale, l'host ha responsabilità specifiche riguardo al USB ed ai relativi dispositivi fissi. L'host controlla tutti gli accessi al bus USB. Un dispositivo del bus USB accede al bus soltanto dopo l'assegnazione da parte del host.

Il sistema USB

Host

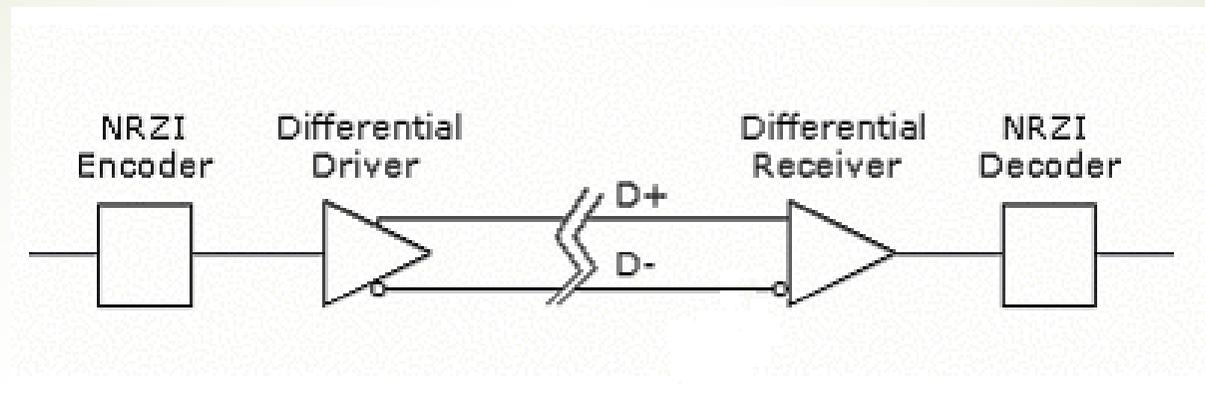
L'host è inoltre responsabile del controllo e monitoraggio della topologia del dispositivo USB collegato al bus.

Tali mansioni richiedono risorse Hw, quali memoria per i descrittori e capacità di calcolo, considerevoli per questo motivo l'host è solitamente un PC.

CARATTERISTICHE ELETTRICHE

Come avviene il trasferimento?

Il trasferimento dei dati avviene fisicamente attraverso le variazioni di tensione differenziale fra due dei quattro cavetti costituenti il cavo USB. I dati da scambiare sono dapprima trasformati in una sequenza continua e poi modulati con la codifica NRZI (Non-Return to Zero, Inverted):



Tale codifica unita alla trasmissione di tipo differenziale, è facilmente gestibile da circuiti elettronici a basso costo garantendo ottima immunità ai disturbi.

RICONOSCIMENTO DEGLI APPARATI

(Detection)

Prima di trasferire informazioni da o a un dispositivo, il software di sistema deve accertarsi della sua presenza e riconoscere se la periferica è ad alta velocità o bassa velocità.

L'osservazione delle variazioni della tensione differenziale consente di riconoscere la connessione/disconnessione degli apparati.

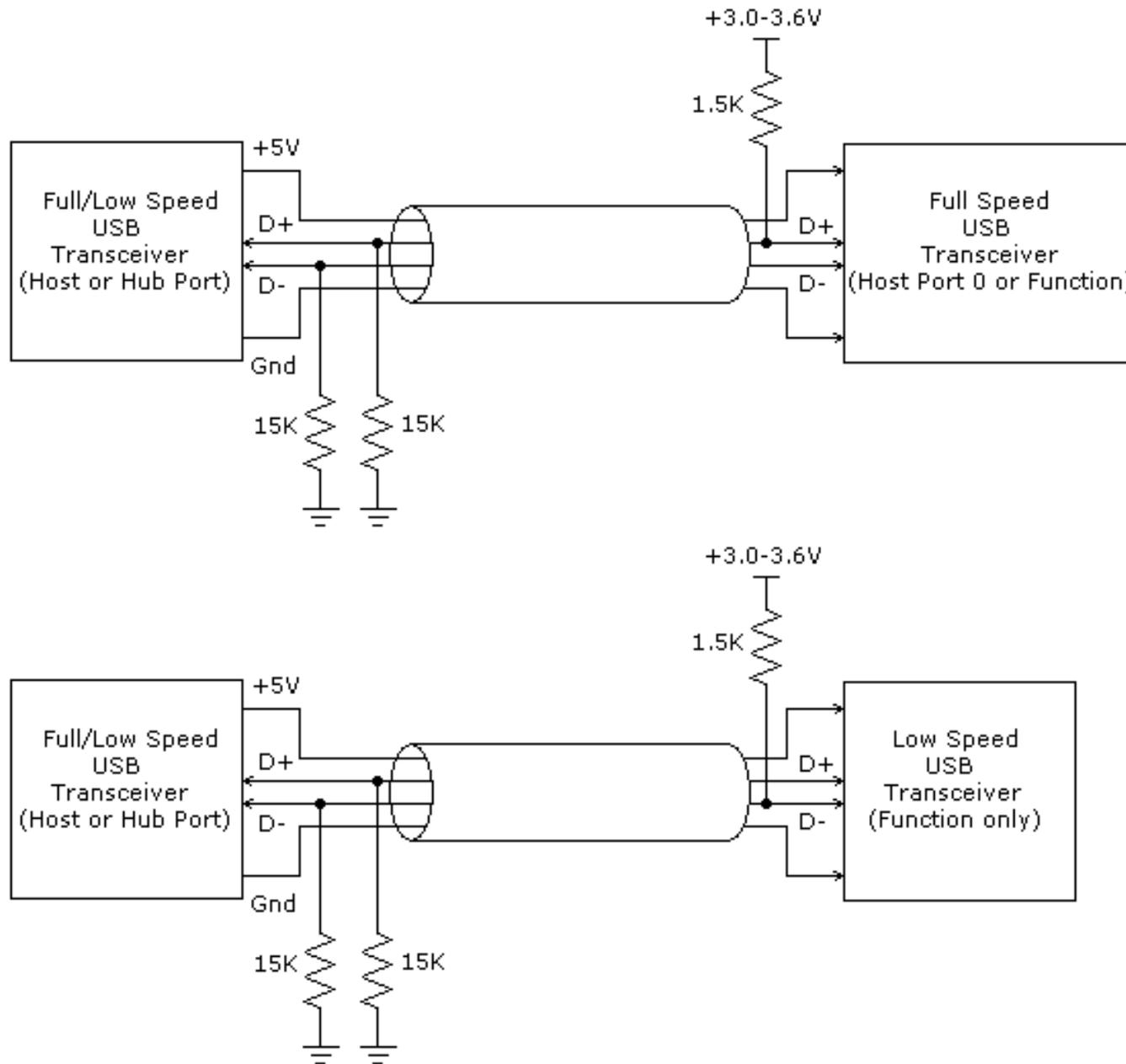
Ciò significa, che l'hub, individua la presenza di un dispositivo a una delle sue porte monitorando le linee dei dati dopo l'inserimento del cavo.

Quando la periferica non è connessa al bus, le resistenze di pull-down ($15k\Omega$) chiudono le linee D+ e D- a massa.

Se invece, la periferica è collegata, si ottiene un partitore di tensione costituito dalla resistenza di pull-down dell'hub e dalla resistenza di pull-up ($1.5K\Omega$) della periferica (posta su D+ o D- a seconda della velocità).

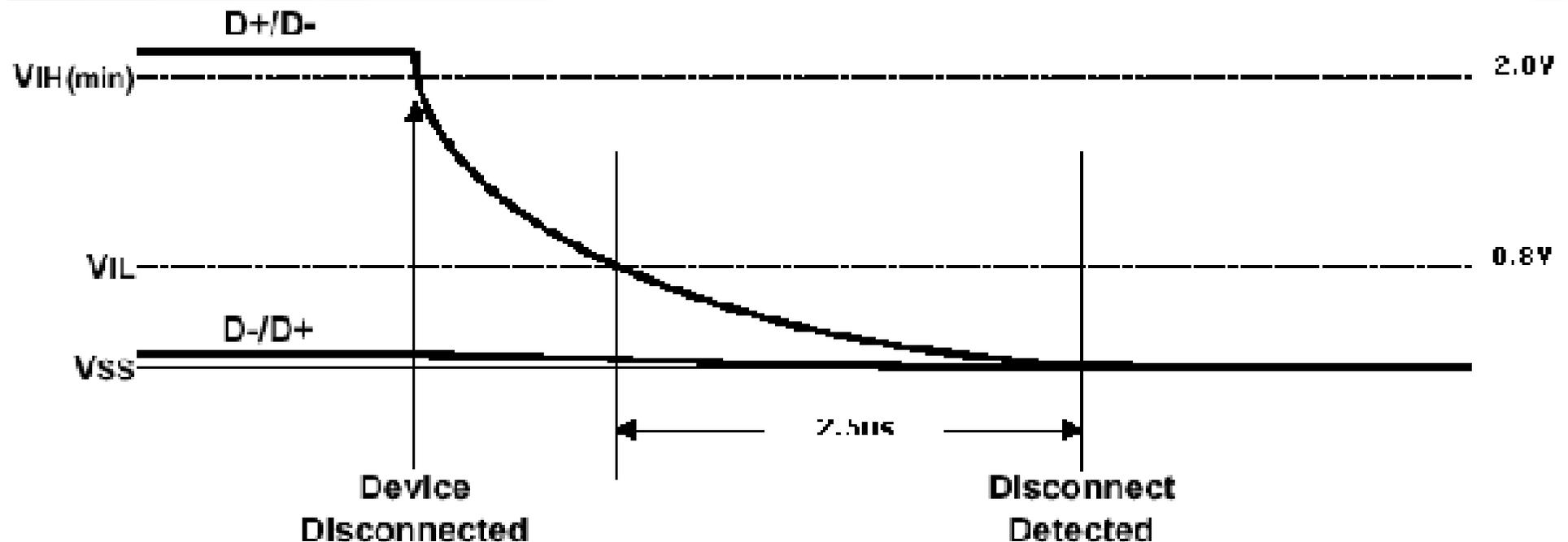
Infatti una periferica ad alta velocità presenta una resistenza di pull-up sulla linea D+, mentre una periferica a bassa velocità presenta una resistenza di pull-up sulla linea D-, permettendo all'hub di identificare la velocità del dispositivo.

Riconoscimento degli apparati



Riconoscimento degli apparati

Volendo analizzare più in dettaglio i livelli di tensione che indicano la connessione/disconnessione di una periferica, si osserva che quando entrambe le linee dati D+, D- scendono al di sotto di VIL (0.8V), per una durata di tempo superiore a 2.5 μ s, l'hub riconosce che la periferica non è connessa figura A.8; se invece, D+ o D- superano il valore VIH (2.0V), per una durata di tempo superiore a 2.5 μ s, l'hub riconosce che la periferica è connessa.



Successivamente alla fase di riconoscimento, l'hub setta a 1 (SE1) un bit di stato e lo resetta (SE0) non appena la periferica è disconnessa. Per evitare confusioni sulla natura dei segnali destinati per periferiche ad alta o a bassa velocità, si fa uso di due stati elettricamente opposti come si può notare in figura seguente:

Riconoscimento degli apparati

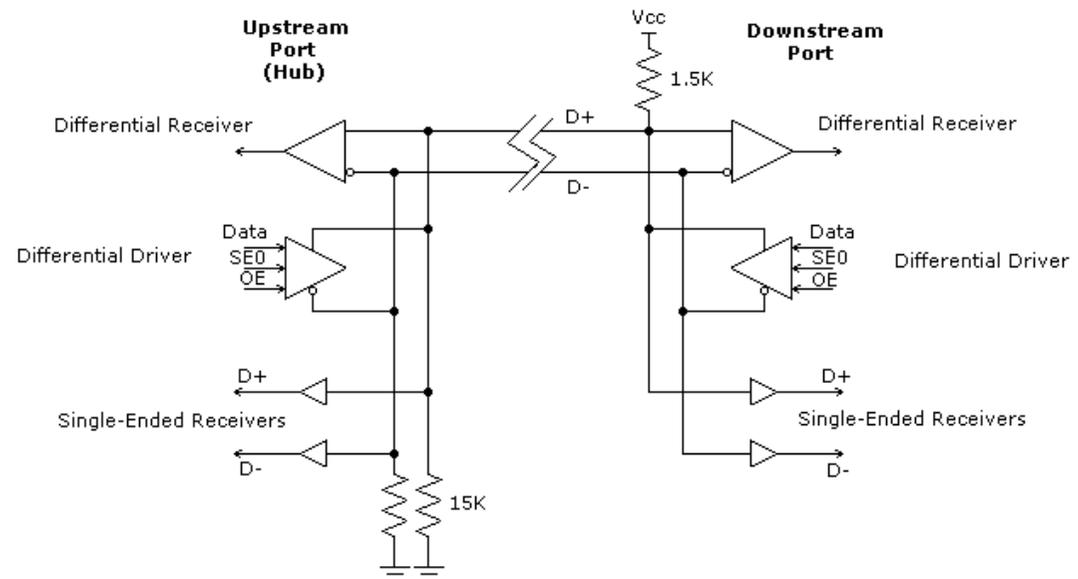
Bus State	Signaling Levels		
	At originating source connector (at end of bit time)	At final target connector	
		Required	Acceptable
Differential "1"	$D+ > V_{OH}(\text{min})$ and $D- < V_{OL}(\text{max})$	$(D+) - (D-) > 200\text{mV}$ and $D+ > V_{IH}(\text{min})$	$(D+) - (D-) > 200\text{mV}$
Differential "0"	$D- > V_{OH}(\text{min})$ and $D+ < V_{OL}(\text{max})$	$(D-) - (D+) > 200\text{mV}$ and $D- > V_{IH}(\text{min})$	$(D-) - (D+) > 200\text{mV}$
Single-ended 0 (SE0)	$D+$ and $D- < V_{OL}(\text{max})$	$D+$ and $D- < V_{L}(\text{max})$	$D+$ and $D- < V_{IH}(\text{min})$
Data J state: Low-speed Full-speed	Differential "0" Differential "1"	Differential "0" Differential "1"	
Data K state: Low-speed Full-speed	Differential "1" Differential "0"	Differential "1" Differential "0"	
Idle state: Low-speed Full-speed	N.A.	$D- > V_{IHZ}(\text{min})$ and $D+ < V_{IL}(\text{max})$ $D+ > V_{IHZ}(\text{min})$ and $D- < V_{IL}(\text{max})$	$D- > V_{IH}(\text{min})$ and $D+ < V_{IH}(\text{min})$ $D+ > V_{IHZ}(\text{min})$ and $D- < V_{IH}(\text{min})$
Resume state	Data K state	Data K state	
Start-of-Packet (SOP)	Data lines switch from Idle to K state		
End-of-Packet (EOP) ^d	SE0 for approximately 2 bit times ¹ followed by a J for 1 bit time ³	SE0 for ≥ 1 bit time ² followed by a J state for 1 bit time	SE0 for ≥ 1 bit time ³ followed by a J state
Disconnect (at downstream port)	N.A.	SE0 for $\geq 2.5\mu\text{s}$	
Connect (at downstream port)	N.A.	Idle for $\geq 2\text{ms}$	Idle for $\geq 2.5\mu\text{s}$
Reset	$D+$ and $D- < V_{OL}(\text{max})$ for $\geq 10\text{ms}$	$D+$ and $D- < V_{IL}(\text{max})$ for $\geq 10\text{ms}$	$D+$ and $D- < V_{IL}(\text{max})$ for $\geq 2.5\mu\text{s}$

CODIFICA E TRASMISSIONE DATI

Codifica NRZI: i dati vengono trasferiti facendo uso della codifica NRZI (Non-Return to Zero, Inverter);

BIT STUFFING: per evitare perdite di sincronizzazione, il trasmettitore inserisce nel flusso dei dati il cosiddetto stuffed bit; ogni volta che vengono inviati sei 1 consecutivi viene inserito questo bit che non porta informazione e viene rimosso dal demodulatore durante la ricezione della stringa dei dati.

Trasmissione differenziale: il sistema USB implementa un trasferimento di dati in modalità differenziale per ridurre in maniera significativa eventuali segnali di rumore sovrapposti al segnale utile.



CODIFICA E TRASMISSIONE DATI

Da notare la presenza del ricevitore (single-ended) implementato nel sistema USB, per riconoscere particolari stati del bus, ad esempio quando entrambe le linee dati sono a livello basso.

Dal punto di vista elettrico lo scambio dei dati tra l'Host o hub e la periferica è di tipo seriale half-duplex, e cioè la coppia di conduttori che trasporta il segnale viene utilizzata per trasmettere e per ricevere i dati in momenti diversi, perché la ricezione e la trasmissione simultanea non sono possibili.

L'implementazione half-duplex impone che i drivers siano posti in uno stato di alta impedenza quando non trasmettono dati.

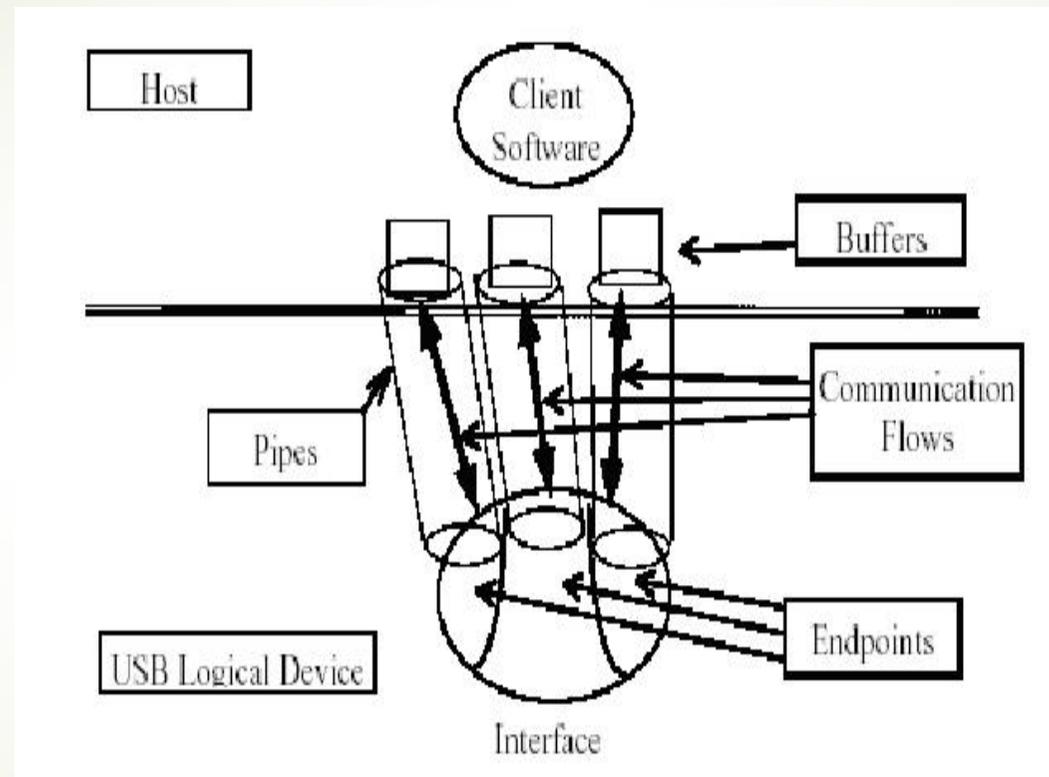
La connessione viene inoltre classificata come "punto-punto": su ogni cavetto USB i segnali sono gestiti in modo indipendente dagli altri.

Se la periferica è connessa a un hub a sua volta collegato alla porta posteriore del PC, lo scambio dei dati tra periferica e hub è gestito in modo indipendente da quello tra hub e computer.

Il sistema USB

I canali di comunicazione

1. Control
2. Isochronous
3. Interrupt
4. Bulk



- Pipes: canali attraverso i quali fluiscono dati tra Host e device
- Endpoint: terminatori lato device del canale

ENDPOINTS

Un endpoint è una parte unicamente identificabile del dispositivo USB e costituisce un estremo di un flusso di comunicazione tra Host e Device.

Ad ogni Device viene assegnato, in fase di collegamento e configurazione, un indirizzo.

Inoltre ad ogni Endpoint, in fase di progetto e realizzazione del Device, viene associato un numero identificativo e una direzione del flusso dati con l'Host.

Per cui la combinazione dell'indirizzo del Device, del numero dell'Endpoint e della sua direzione costituiscono un'identificazione univoca del destinatario/sorgente dei dati.

Questa tecnica permette di veicolare dati, destinati a funzioni differenti, attraverso un bus a singola linea.

ENDPOINTS

Un Endpoint ha caratteristiche che determinano il tipo di trasferimento tra l'Endpoint e il Client Software e sono:

- La frequenza d'accesso al bus e la massima latenza consentita.
- La larghezza di banda richiesta.
- Il numero assegnato all'Endpoint e la direzione del flusso.
- La massima dimensione del pacchetto che l'Endpoint è capace di ricevere o trasmettere.
- Il tipo di trasferimento che può sostenere l'Endpoint (tale argomento sarà approfondito in seguito).

Una considerazione particolare merita l'Endpoint numero zero, chiamato anche Default Control Pipe, l'unico ad essere sempre accessibile anche quando il dispositivo non è stato ancora configurato.

ENDPOINTS

Endpoint zero caratteristiche:

Tutti i dispositivi USB devono implementare un canale di controllo di default che è costituito dall'Endpoint zero.

L'USB System Software utilizza tale canale per inizializzare e configurare un Device. Il Default Control Pipe provvede all'accesso al dispositivo per recuperare le informazioni necessarie alla configurazione e di monitorare lo stato del Device. L'endpoint numero zero è sempre accessibile non appena il dispositivo è collegato, alimentato e ha ricevuto un comando di Reset.

Endpoints non-zero caratteristiche:

Le Funzioni possono avere Endpoint addizionali come richiesto dalla specifica implementazione.

I Device Low Speed sono limitati ad avere al massimo due Endpoints opzionali oltre i due necessari per l'implementazione del Default Control Pipe. I Device Full Speed possono avere un numero di Endpoint opzionali limitati solo dalle specifiche del protocollo, cioè un massimo di 15 endpoint di input e 15 di output.

ENDPOINTS

Pipes:

Un USB Pipe (tubo) è il collegamento che si instaura tra una Endpoint sul Device e un Client Software sull'Host tramite il buffer di memoria associato.

Vi sono due differenti, mutuamente esclusivi, modi di comunicazione:

- Stream: i dati vengono trasferiti attraverso un pipe senza una struttura imposta dalle norme.
- Message: i dati vengono trasferiti attraverso un pipe con una struttura definita dalle norme. *(Ad esempio i dispositivi instrument di NI)*

L'USB non interpreta il contenuto dell'area dati che viene trasferita attraverso un pipe ma si limita a verificare che la struttura che li accoglie sia in accordo con le specifiche USB.

ENDPOINTS

Inoltre le pipe hanno le seguenti caratteristiche associate:

- La richiesta di accesso al bus e la larghezza di banda usata.
- Il tipo di trasferimento.
- Le caratteristiche associate, come la direzione e la massima dimensione del payload.

La pipe che ha come estremi i due endpoint numero zero è chiamato Default Control Pipe.

Stream Pipes

I dati in uno Stream Pipe fluiscono con una logica FIFO e tali canali sono sempre unidirezionali.

Il flusso dati è sempre riferito ad un unico client l'USB System Software non fornisce la sincronizzazione tra client multiple che necessitano degli stessi dati (non supporta il multicasting).

Un numero di endpoint di un device nella direzione opposta può essere usato per un altro stream pipe associato al device.

Uno stream pipe supporta trasferimenti di tipo bulk, isocroni e ad interrupt che verranno descritti in seguito.

MESSAGE PIPES

I message pipes interagiscono con un endpoint in maniera differente da uno stream pipe. Ogni transazione comincia con una richiesta da parte dell'Host verso il Device. Questa richiesta è seguita da uno o più trasferimenti di dati nella direzione appropriata. Infine segue una fase di resoconto dello Status.

Tale ordine nella comunicazione in un Message Pipe permette che gli ordini siano identificati e comunicati attendibilmente. I canali del messaggio permettono il flusso di comunicazione in entrambi i sensi, anche se il flusso di comunicazione può essere principalmente one-way.

Il Default Control Pipe è sempre un Message Pipe.

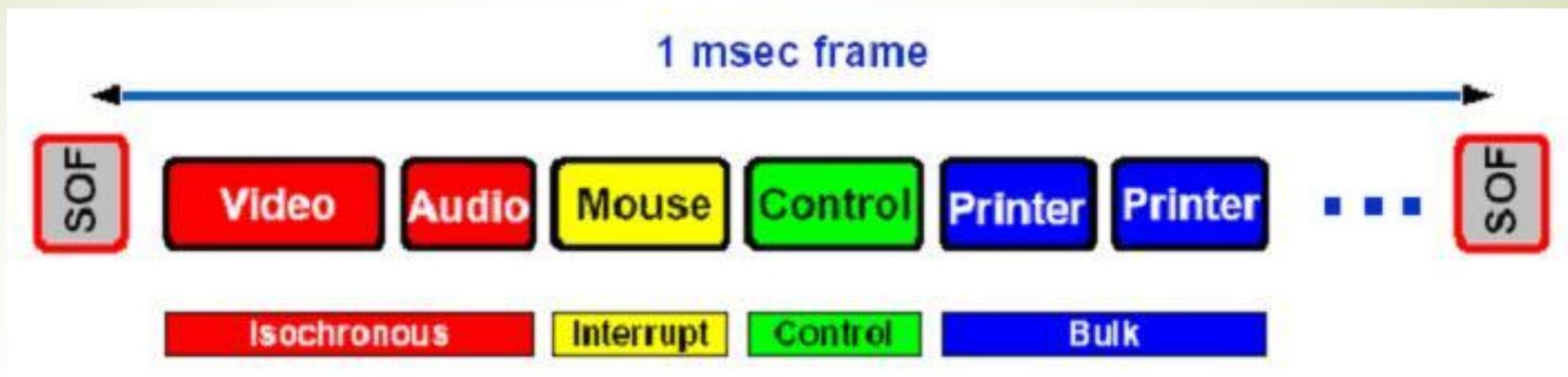
L'USB System Software assicura che non vengano inviati richieste multiple con un Message Pipe. Eventuali richieste provenienti da più client e destinate al Default Control Pipe vengono smaltite con strategia FIFO.

MESSAGE PIPES

Normalmente un Host non spedisce un nuovo message fin quando il message corrente non è stato completamente ricevuto dal device. Tuttavia, ci sono condizioni di errore per cui un trasferimento del messaggio può essere abbandonato dall'host ed un nuovo message può essere trasmesso prematuramente. Dal punto di vista del software che gestisce un message pipe, un errore su una certa parte di un IRP comporta il ritiro dell' IRP corrente e di tutti gli IRPs in coda. Al cliente del software che ha chiesto l'IRP è comunicato il mancato completamento dell' IRP con un'indicazione adatta di errore. Un canale del messaggio ad un dispositivo richiede un singolo numero di Endpoint del dispositivo in entrambi i sensi (IN e OUT token).

Il sistema USB

Tutte le trasmissioni devono essere incapsulate in una frame di durata 1ms



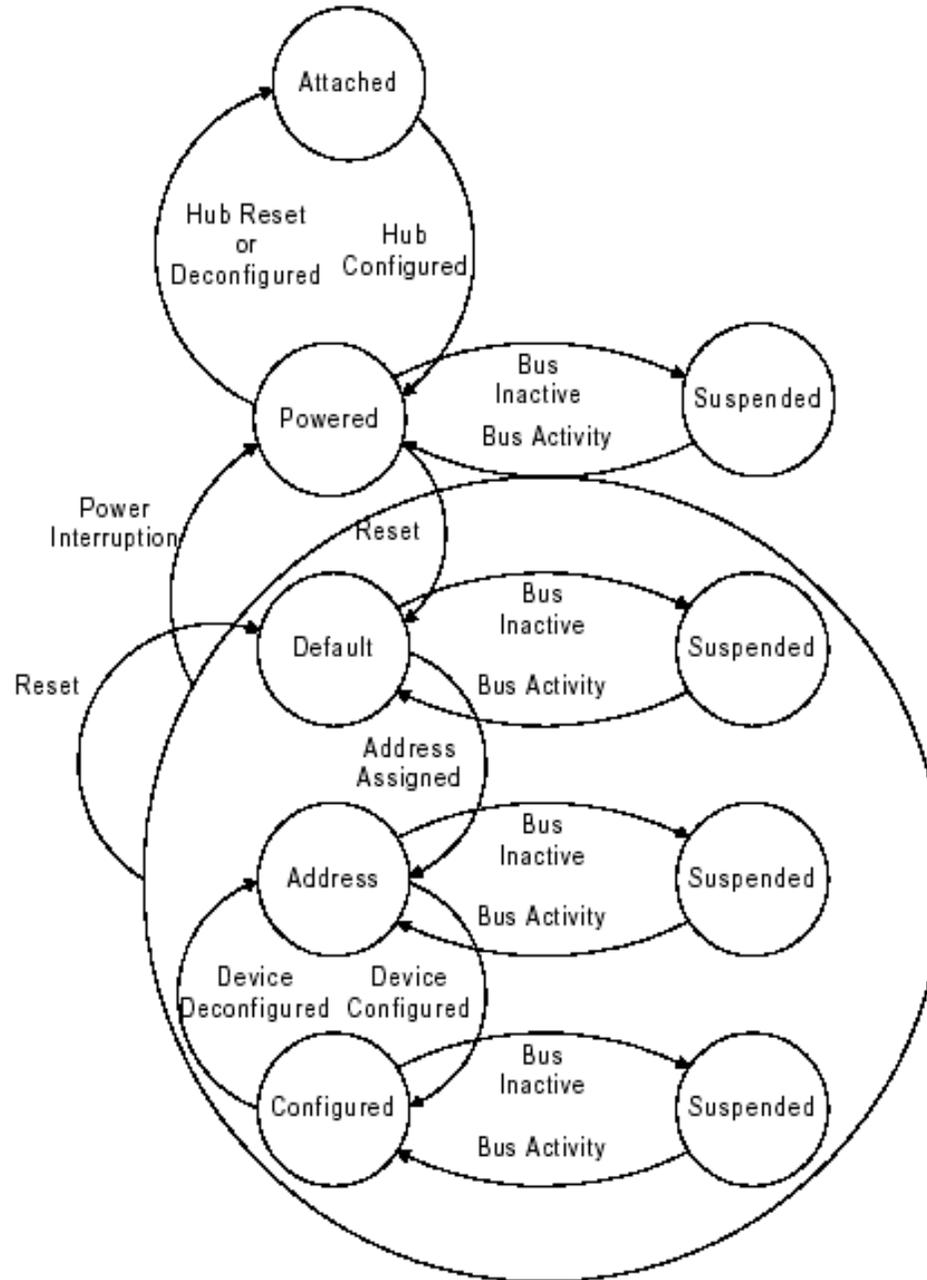
La configurazione delle periferiche

Quando si collega una periferica USB al PC, prima che inizi a comunicare esiste una fase di configurazione o “enumeration”. La configurazione avviene attraverso trasferimenti di tipo control utilizzando il Default Control Pipe. Per questo motivo, tutte le periferiche devono essere in grado di gestire il trasferimento control. In questa fase periferica e Host si scambiano alcune informazioni memorizzate nel dispositivo e organizzate in descrittori. Per esempio la periferica, su richiesta del PC, comunica quale tipo di trasferimento supporta, il numero degli eventuali endpoint che possiede etc. Mentre l’host comunica al device un indirizzo, tramite il quale il dispositivo sarà indirizzato, e sceglie una tra le possibili configurazioni.

Come avviene la configurazione sull'host

Una delle caratteristiche del sistema USB è quella di rilevare la connessione o la rimozione di una periferica in modo autonomo. Dalla prospettiva dell'utente il processo di configurazione è quasi invisibile, eccetto la prima volta, quando il sistema operativo può avere necessità di un file con estensione “.inf” nel quale sono scritte informazioni sulla periferica ed eventualmente di un driver. In questo caso l'utente dovrà fornire un dischetto di installazione.

I passi della configurazione del device



Per capire i differenti stati si può seguire cosa avviene quando una periferica viene connessa al PC:

1 L'utente collega la periferica al PC: la variazione di tensione sulle linee D+/D- segnala all'host l'avvenuta connessione. Il dispositivo inizia a prelevare dal connettore l'alimentazione e passa nello stato di Powered.

2 L'host riconosce il dispositivo: l'hub controlla le tensioni di ciascuna linea del bus e determina se il dispositivo è di tipo Low Speed o Full Speed.

3 L'host esegue il reset del device: accertata la presenza di una nuova periferica, il PC host emette un comando di reset (passiamo nello stato di Default). La periferica viene inizializzata ed attiva la comunicazione di tipo control sull'endpoint zero. Finora periferica e host non hanno scambiato nessuna informazione, ma semplicemente eseguito operazioni preliminari.

Per capire i differenti stati si può seguire cosa avviene quando una periferica viene connessa al PC:

4 L'host assegna un indirizzo alla periferica: il Pc assegna alla periferica un indirizzo (da 0 a 127), tramite il quale indirizzare il device una volta configurato. La periferica, ricevuto l'indirizzo, lo memorizza ed emette un handshake di conferma.

A questo punto la periferica è nello stato Address. L'indirizzo assegnato non è univoco, la stessa periferica infatti può ricevere un indirizzo differente ogni volta che viene disconnessa e poi ricollegata al PC.

5 Il PC host richiede alla periferica i descrittori: le richieste da parte dell'host avvengono tramite una transazione control. A tali richieste, in fase di configurazioni sono di tipo standard, il device risponde con un descrittore.

Con tali descrittori la periferica fornisce al PC host una sorta di scheda personale nella quale sono scritte tutte le informazioni necessarie per la corretta configurazione.

Ad esempio:

- Costruttore
- Tipo di periferica
- Numero degli eventuali endpoint
- Corrente assorbita
- Altri eventuali parametri

La corrente richiesta dalla periferica potrebbe impedirne la configurazione. Infatti un hub può erogare fino ad un massimo di 0.5 A (2.5 W), da distribuirsi fra tutte le periferiche ad esso collegate. Quindi, se la periferica richiede più corrente di quella disponibile, viene portata in uno stato denominato Attached e la procedura di configurazione viene interrotta.

6 Il PC host assegna un driver alla periferica: in base ai descrittori ed eventualmente in base al file con estensione “.inf” che il sistema operativo può richiedere, viene associato alla periferica il driver che si adatta meglio alle sue caratteristiche. Nel caso il sistema operativo non disponga di un driver adatto, questo verrà richiesto all’utente, da una finestra di dialogo.

7 Il driver sceglie la configurazione: dopo aver acquisito tutte le informazioni sulla periferica, il driver le assegna la configurazione. Alcune periferiche possono avere differenti configurazioni in tal caso il driver può decidere quale assegnarle, o eventualmente può chiederla all’utente tramite una finestra di dialogo. A questo punto il dispositivo è

Vantaggi e limiti del bus USB

L'utilizzo dell'interfaccia USB presenta numerosi vantaggi rispetto a bus concorrenti a basso costo quali RS232, LPT e 488.

Rispetto al bus seriale RS232 e LPT il bus USB offre:

1. maggiore larghezza di banda e diversi tipi di trasferimento con latenza o larghezza di banda assicurata.
2. possibilità di connettere un numero maggiore di periferiche.
3. connettività a “caldo” delle periferiche (Hot Plug and Play)

Rispetto alla più preformante 488 l'USB presenta :

1. maggiore diffusione del bus sui computer
2. minor costo e complessità dei controller

I limiti di tale bus sono rappresentati invece dalla scarsa diffusione di periferiche di misura dotate di tale interfaccia ma soprattutto della scarsa trasparenza della gestione del bus da parte del sistema operativo e dalla difficoltà nello scrivere driver generici.

IL BUS DI
COMUNICAZIONE
SERIALE VELOCE
FireWire o Ilink
(IEEE 1394)

GENERALITA'

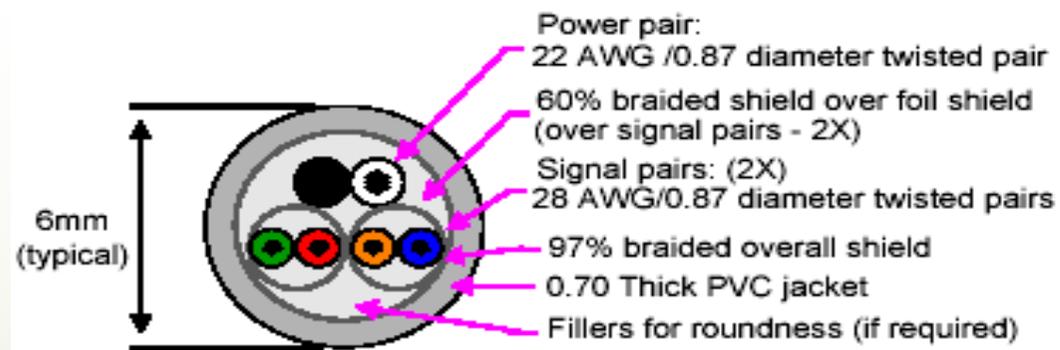
In principio introdotto come “Firewire” dalla Apple Computer verso la fine del 1980, successivamente approvato come standard IEEE 1394 – 1995, fu sviluppato per supportare i requisiti di elevata larghezza di banda di dispositivi come apparati digitali video e memorie di massa ad alte prestazioni. Dopo il primo standard, un successivo ne viene approvato nel Marzo 2000, è lo standard IEEE 1394a – 2000. Quest’ultimo descrive un bus seriale che supporta velocità di 100 Mbytes per secondo, 200 Mbps fino ai 400 Mbps con notevoli miglioramenti nel controllo del traffico e nella configurazione della gestione della potenza .

GENERALITA'

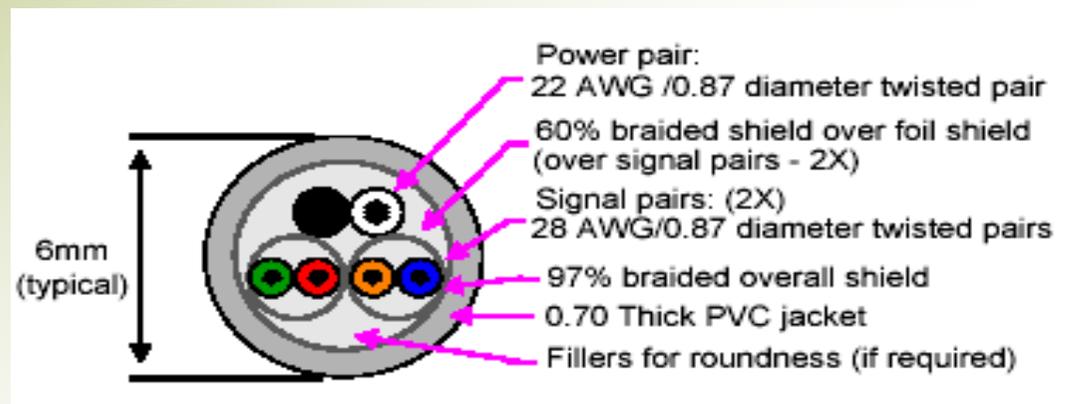
Tale bus supporta l'hot swapping ed il plug and play per rendere semplice l'uso di prodotti che supportano la IEEE 1394. Inoltre si è in attesa dello standard 1394b che dovrebbe portarci a lavorare con velocità di 1600 Mbps fino a 3200 Mbps. Uno dei vantaggi della scelta di una interfaccia seriale su una parallela è la taglia ridotta dei connettori richiesti e dei cavi e questo è un fattore importante nei prodotti portatili dove l'IEEE 1394 si è fatto per prima strada.

GENERALITA'

La specifica 1394 definisce due tipi di connettori: a quattro e a sei pin; il primo ha due coppie differenziali per il segnale ed ha dimensioni vantaggiose: 5 mm x 3 mm. Il secondo oltre alla doppia coppia differenziale ha anche due cavetti interni per l'alimentazione (potenza e massa) .



GENERALITA'



Il connettore a 4 pin è più piccolo dell'USB. Quello a 6 pin avendo in sé l'alimentazione è stato facilmente accettato dai produttori di dispositivi portatili ed inoltre molte periferiche di PC come camere per conferenze o drive per dischi portatili hanno tratto vantaggio da un bus autoalimentato; così infatti i costruttori possono produrre periferiche a costi considerevolmente ridotti rispetto ai modelli portatili alimentati a batteria delle stesse periferiche.

GENERALITA'

Quindi quando le dimensioni ridotte non sono il principale requisito, il connettore a 6 pin è ambito per le sue qualità.

In aggiunta esse hanno un basso costo alta velocità e una grande quantità di spazio indirizzi memory-mapped.

Per le applicazioni multimediali la IEEE 1394 supporta sia la comunicazione asincrona che quella isocrona.

Il trasporto asincrono ci assicura consegna di dati libera da errori, ma il suo protocollo di controllo degli errori introduce un tempo di ritardo che è non idoneo per le applicazioni multimediali; per contro esso è un passaggio fondamentale in altri compiti come la scrittura su disco.

Il trasferimento di dati isocrono garantisce il throughput ad un predeterminato tasso cosa che è necessaria per la trasmissione di dati multimediali in tempo critico dove alcuni errori sono tollerati.

Nodi

L'architettura del bus seriale è definita in termini di nodi.

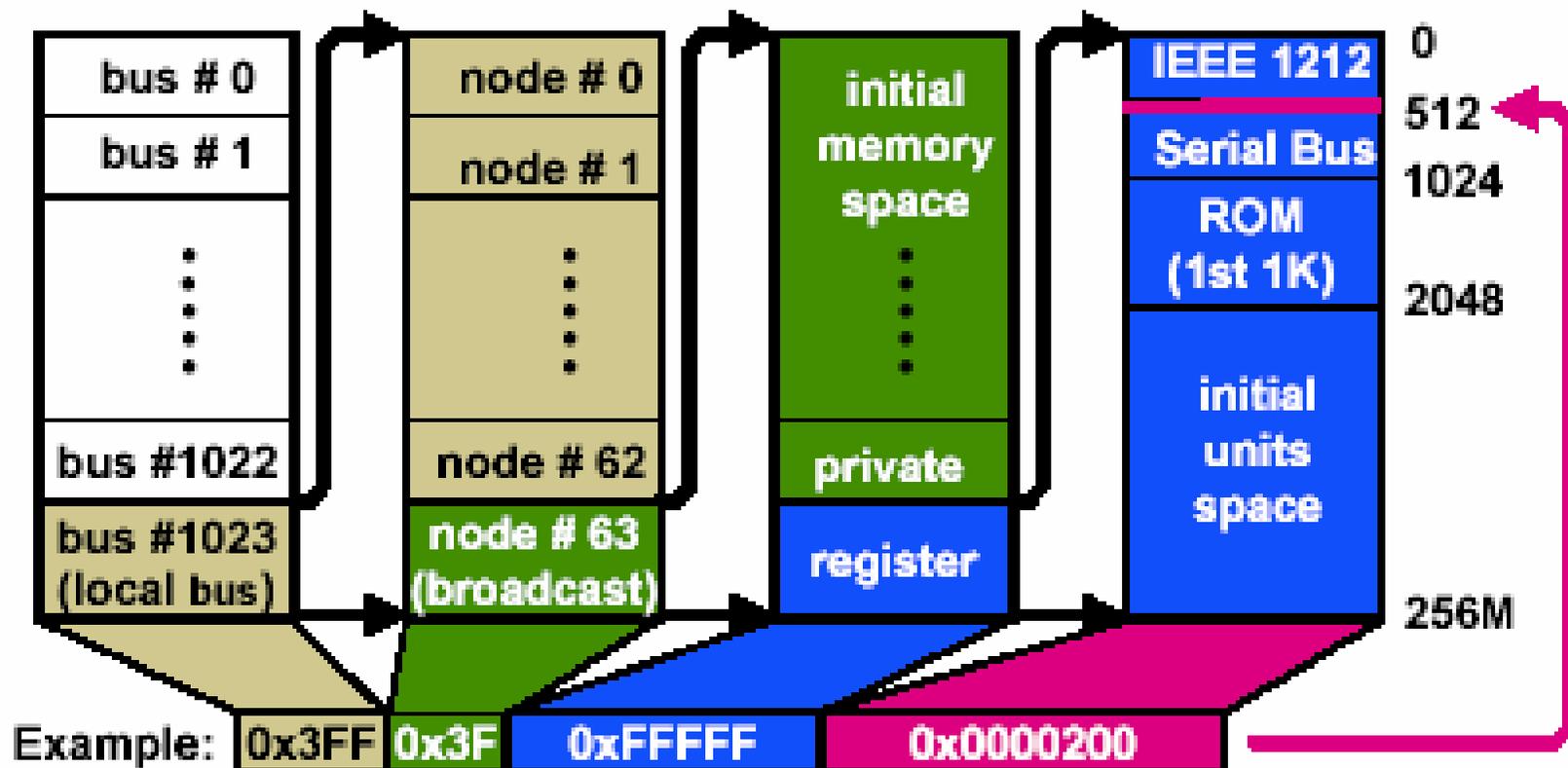
Un nodo è un'entità indirizzabile. Un modulo, che è definito come un dispositivo sostituibile, contiene uno o più nodi e ogni nodo contiene una o più unità come: un processore, un dispositivo di I/O, o una unità di memoria. Ogni nodo ha una o più porte, che sono connesse o ai cavi o ai backplane e provvedono ad una connessione fisica con un altro nodo.

La topologia dei cavi permette due tipi di configurazioni: o quella ad albero o la daisy chain.

Gestione ad albero: Il bus seriale limita il numero di nodi a 63, usando la gestione ad albero possiamo produrre un sistema di bus multipli con la possibilità di indirizzare fino a circa 216 nodi. Un ponte (bridge) del bus è una unità che provvede ad una connessione fisica tra due bus nella gerarchia dei bus. Cioè l'interfaccia del bus seriale 1394 può essere usata per collegare insieme altri bus 1394, o bus con architetture differenti.

INDIRIZZAMENTO

IEEE 1212 addressing



= all cycle timer registers on local bus

INDIRIZZAMENTO

Come si evince dalla figura sopra il campo dei primi 10 bit identifica il bus (bus ID) da cui provengono i dati per cui abbiamo l'identificazione di 1022 bus più il bus locale che è identificato al posto 1023 (La sequenza con tutti 1 è appunto utilizzata per scopi speciali).

I successivi 6 bit identificano invece il nodo sul bus (physical ID) per cui abbiamo da 0 a 62 nodi possibili mentre il nodo 63 è usato per l'audio/video(anche in questo caso è riservata la sequenza con tutti 1 per scopi speciali); quindi in totale abbiamo la possibilità di indirizzare 64 nodi (dove per nodi si intende un dispositivo base indirizzabile). Tutti i 16 bit sono comunque considerati come identificatori di nodo, per cui riusciamo ad identificare 1023 bus con 63 nodi per bus ovvero 64K nodi indirizzabili.

Infine i 48 bit restanti specificano l'area di memoria (virtuale) locale.

INDIRIZZAMENTO

Essi sono così suddivisi:

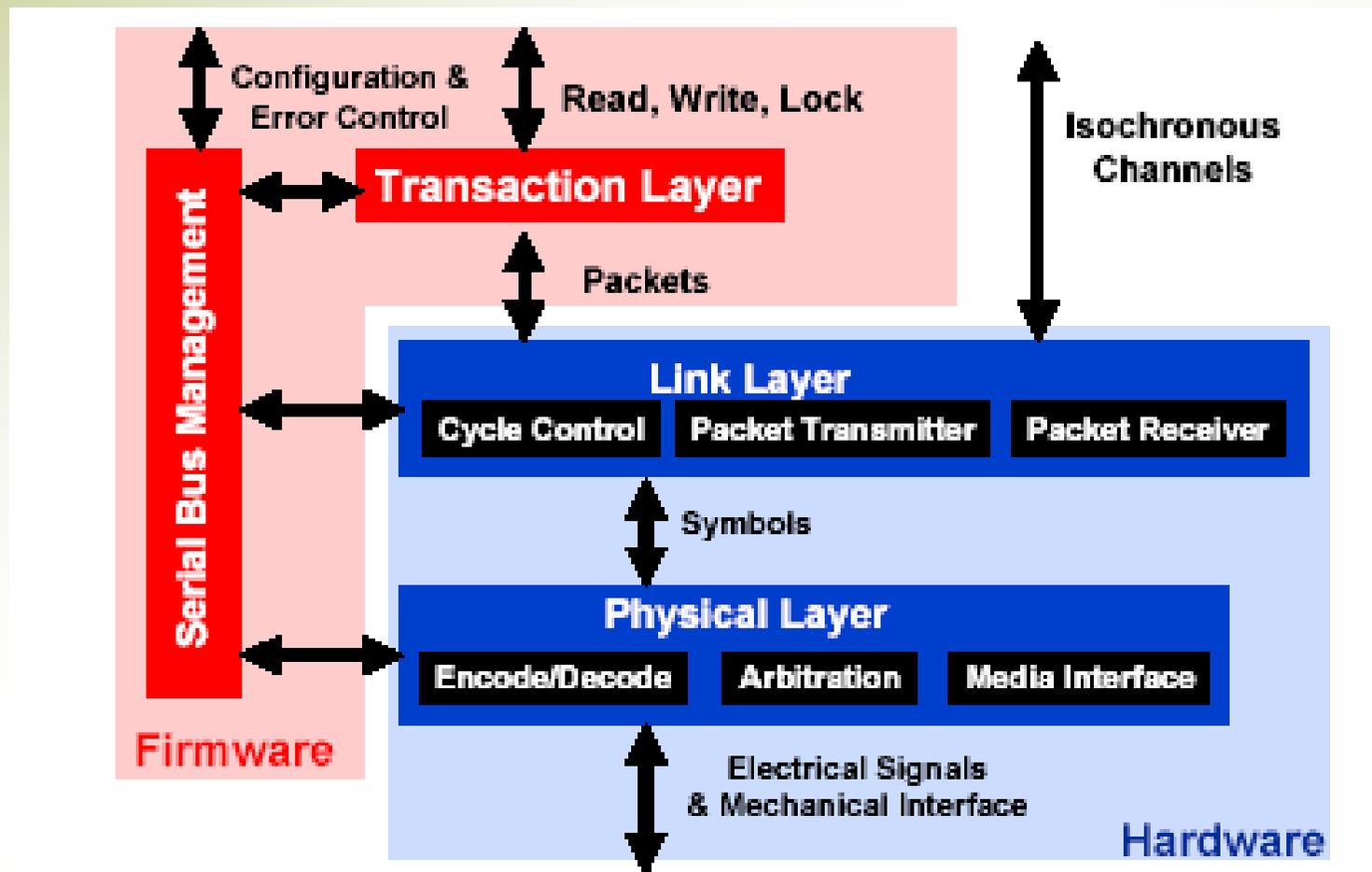
Spazio iniziale di memoria;

Spazio di memoria privata: riservato per usi del nodo locale;

Spazio iniziale di registro, 2048 bytes allocati come segue:

Gli ultimi 256 MB dell'area locale d'indirizzo sono riservati per i registri (0 x FFFFFFF0000000). Di questi, i primi 512 byte sono detti core control e status register, i 512 byte successivi costituiscono il register del bus seriale (serial bus register), segue la Rom di configurazione (configuration ROM) a cui è riservato 1Kbyte.

Blocchi costruttivi del bus seriale veloce IEEE1394



Blocchi costruttivi del bus seriale veloce IEEE1394

Lo standard definisce 4 livelli:

fisico (physical),

linea (link),

transazioni (transaction)

gestione seriale del bus (Serial Bus Management).

I livelli fisico e di linea sono i principali blocchi di un'interfaccia 1394. Il livello linea comunica con il livello transazioni usando un datagram noto, un trasferimento di dati a una via con richiesta di conferma.

Il livello linea tratta trasmissione e ricezione di pacchetti di dati.

Blocchi costruttivi del bus seriale veloce IEEE1394

Un transceiver (interfaccia fisica) esegue servizi di inizializzazione e arbitraggio e gestisce il livello fisico assicurando che solo un nodo (dispositivo base indirizzabile) invia i dati in un certo istante. Il transceiver traduce il flusso dei dati seriali e i livelli di segnale prima di mandarli al livello connessioni. I dispositivi di interfaccia fisica possono anche permetterci di far passare i dati da un dispositivo ad un altro senza passare attraverso il livello connessioni, per esempio se il PC ha 2 porte 1394 una connessa ad una stampante e l'altra ad una videocamera, ed il PC deve mantenere l'alimentazione dei dispositivi di interfaccia fisica, il flusso di dati tra le due periferiche può allora avvenire senza interessare il PC ed anche se questo è spento.

Blocchi costruttivi del bus seriale veloce IEEE1394

Il livello di transazione implementa i protocolli di richiesta –risposta. Il livello di transazione può essere implementato in firmware o in hardware. Il link in tecnologia digitale ed il fisico in hardware. La gestione seriale del bus ci permette di ottimizzare il tempo di arbitraggio, garantisce un'adeguata alimentazione elettrica per tutti i dispositivi sul bus che lo richiedono, assegna i canali isocroni e notifica gli errori.

Confronto FireWire vs. USB

CARATTERISTICHE	FIREWIRE	USB
Velocità di trasferimento dati	400 Mbps	12 Mbps
Numero di dispositivi	63	127
Plug and play	Si	Si
Hot pluggable	Si	Si
Dispositivi isocroni	Si	Si
Bus power	Si	Si
Tipo di bus	Seriale	Seriale
Bus termination required	No	No
Tipo di cavi	Doppino intrecciato schermato a 6 fili (1 coppia per l'alimentazione e 2 coppie per lo scambio dati)	Doppino schermato intrecciato a 4 fili (1 coppia per l'alimentazione, 1 coppia per lo scambio dati)
Networkable	Si	Si
Topologia della rete	Daisy chain	Hub

Confronto FireWire vs. USB

Eccetto che per le velocità i due bus sono alquanto simili e proprio questo ha fatto la differenza:

Firewire oggi ricopre il campo audio/video dove era attesa l'alta velocità per consentire trasmissioni real-time

USB ricopre tutte quelle applicazioni dove l'alta velocità non è fondamentale e si preferisce avere un costo più basso

Comunque sia la Firewire ha notevoli risorse in quanto lo standard successivo ovvero l'IEEE 1394b ha portato la velocità a 1600 Mbps e si pensa di poter arrivare a 3200 Mbps

USB è host based, significa che i dispositivi devono connettersi ad un computer per comunicare, mentre la Firewire è peer-to-peer, cioè i dispositivi Firewire possono parlare tra loro senza dover passare attraverso il computer, forse è proprio questo il motivo che ha spinto molti costruttori a realizzare stampanti, scanner e hard disk con bus seriali Firewire.

SISTEMI PER LA
TRASMISSIONE
WIRELESS A
BASSO COSTO

INTRODUZIONE

Sempre maggiore è l'interesse che gli ambienti industriali rivolgono verso le numerose opportunità offerte per connessioni wireless tra strumenti di misura.

VANTAGGIO DI UNA CONNESSIONE WIRELESS IN UNA STAZIONE AUTOMATICA DI MISURA

ॐ Assenza di cavi: ambiente più “vivibile” e “ordinato”

ॐ *scalabilità* è possibile in ogni momento aggiungere un dispositivo di comunicazione alla rete senza aggiunta di prese o altre predisposizioni

ॐ **Mobilità: è possibile ricevere o trasmettere informazioni anche quando si è in movimento senza perdere operatività**

APPLICAZIONI

Ambientali: Monitoraggio aree a rischio di calamità
Analisi di un territorio per scopi scientifici

Militari: *Disseminare la zona delle operazioni militari con dei nodi in modo che si riescono ad ottenere informazioni sul territorio “nemico” senza l'utilizzo d'uomini sul campo*

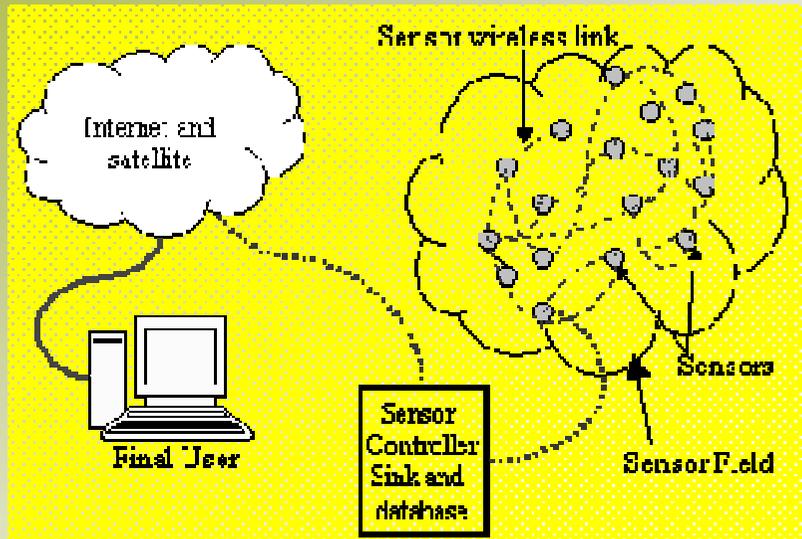
Industriali: Possibilità di automatizzare misure e processi o creare dispositivi remoti intelligenti



Medicali: Impiantazione di sensori nel corpo umano

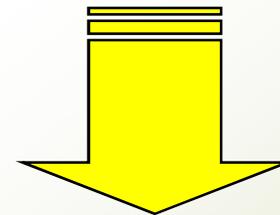


Architettura di una wireless sensor network



I nodi sfruttano un canale radio come mezzo di comunicazione, sono cioè *wireless*

Non occuperanno posizioni “regolari”



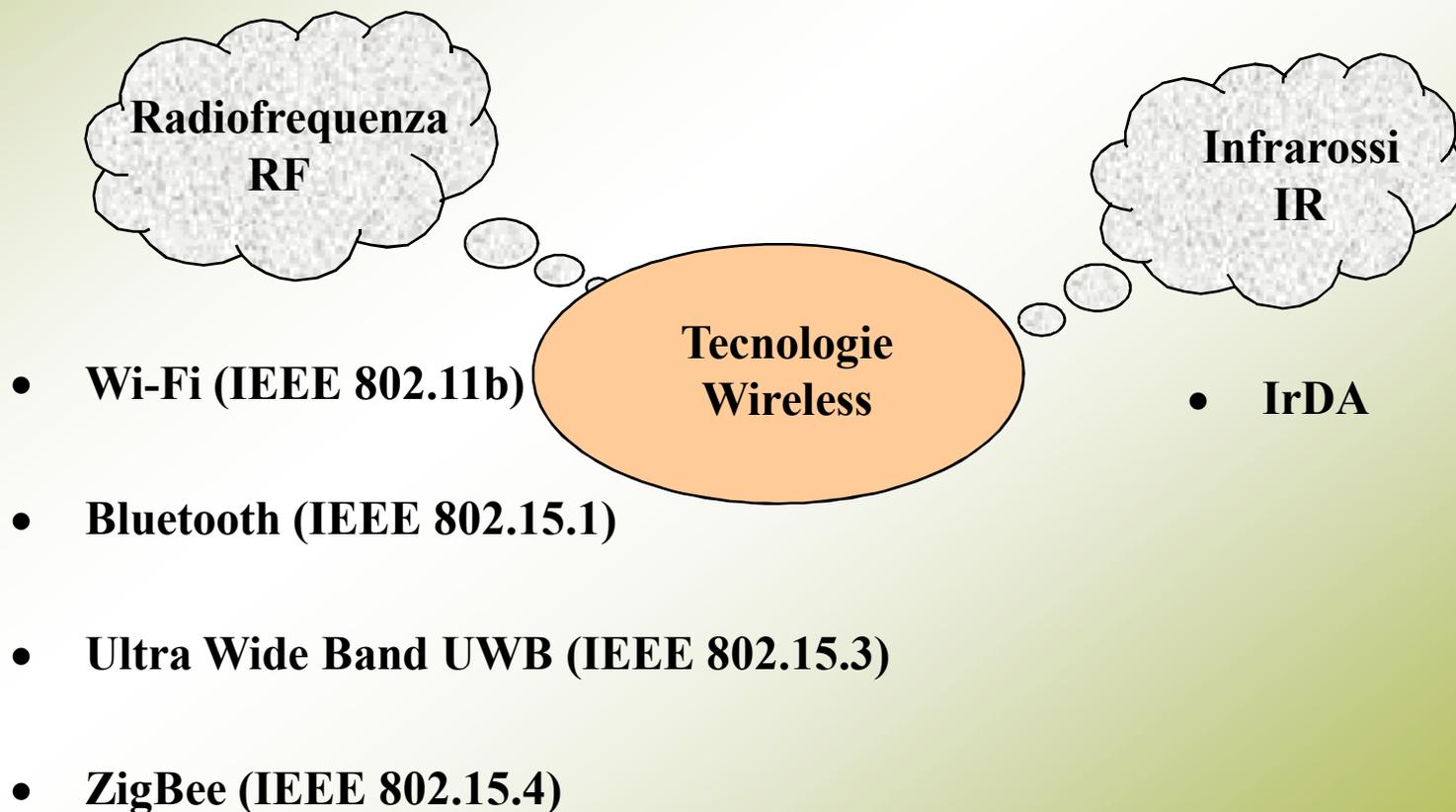
- E' impraticabile connettere i nodi ad una rete d'alimentazione fissa, ma si deve adottare un'alimentazione basata su batterie.
- Si devono ottimizzare, sia i dispositivi fisici di elaborazione, sia il protocollo per lo scambio dei dati.

Molte sono le tecnologie wireless presentate sul mercato

- GSM
 - GPRS
 - DECT
- alto range di copertura,
elevata banda disponibile,
alti costi.
- BlueTooth
 - WiFi
 - IrDA
- range minori,
minore banda,
costi di gestione
notevolmente più piccoli.

TECNOLOGIE WIRELESS PER APPLICAZIONI DI MISURA

Contesto: Applicazioni a medio e corto raggio con basso data rate e bassi consumi



TECNOLOGIE WIRELESS PER APPLICAZIONI DI MISURA

Standard	IrDA	Wi-Fi™ 802.11b	Bluetooth™ 802.15.1	UWB™ 802.15.3	ZigBee™ 802.15.4
Applicazioni	Comunicaz. Seriale a basso Data Rate	Networking Video	Audio e dati	Video, dati, tracking	Sensori, Monitoraggio & Controllo
Dimensioni Stack	-	1MB+	250KB+	-	32KB
Nodi	1	32	7	-	255
Data Rate (kb/s)	4.000	5.500+	723.2 e 57.6	100.000+	20 - 250
Raggio di copertura (Metri)	1 - 2	1 - 100	100+ (Cl. 1) 10 (Cl. 2) 1 (Cl. 3)	-	1 - 100+
Consumo	Medio	Medio/Alto	Medio	Basso	Basso

LE OFFERTE DEL MERCATO

Numerosi siti web tecnici e commerciali propongono ogni giorno nuove soluzioni basate su sistemi wireless.

(HIPERLAN, WIRELESS LAN, IRDA, ecc.)

Tra le tante molto crescente è l'interesse verso soluzioni basate su

dispositivi in tecnologia  **Bluetooth™** (d'ora innanzi BT)

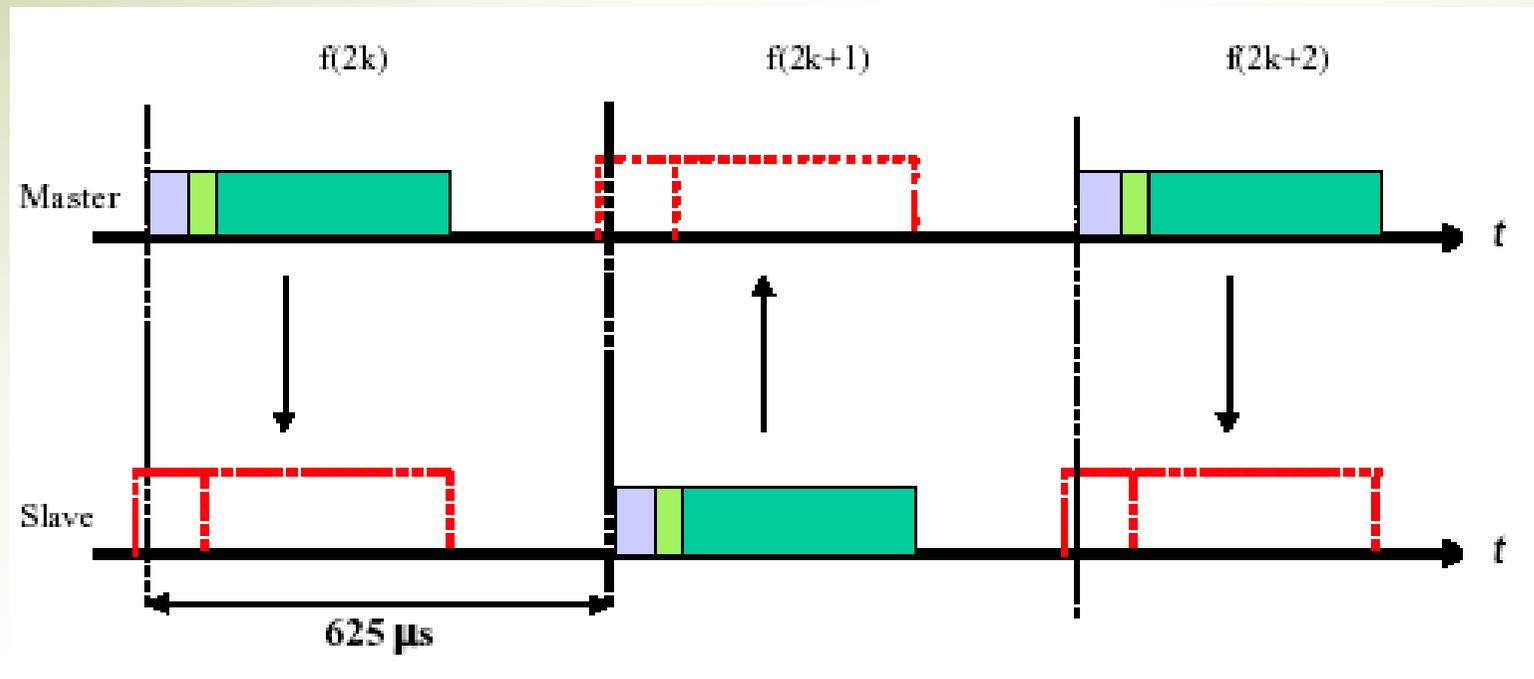
BT è un sistema di comunicazione wireless che nasce per sostituire le connessioni via cavo tra dispositivi elettronici fissi e/o portatili.

Esso è capace di realizzare connessioni radio a corto raggio (10-100 m) tra dispositivi elettronici

Caratteristiche dichiarate: robustezza, bassa potenza di trasmissione e basso costo.

CARATTERISTICHE GENERALI

- BT opera nella banda ISM1 (2.4 GHz), con una baud rate di 1 Ms/s.
- Architettura Master Slave
- trasmissione basata su canali a divisione di tempo ($625 \mu\text{s}$ x intervallo).



- Le informazioni sul canale sono scambiate attraverso pacchetti.
- La trasmissione può essere half duplex o full duplex.

CARATTERISTICHE GENERALI

La tecnica di modulazione usata

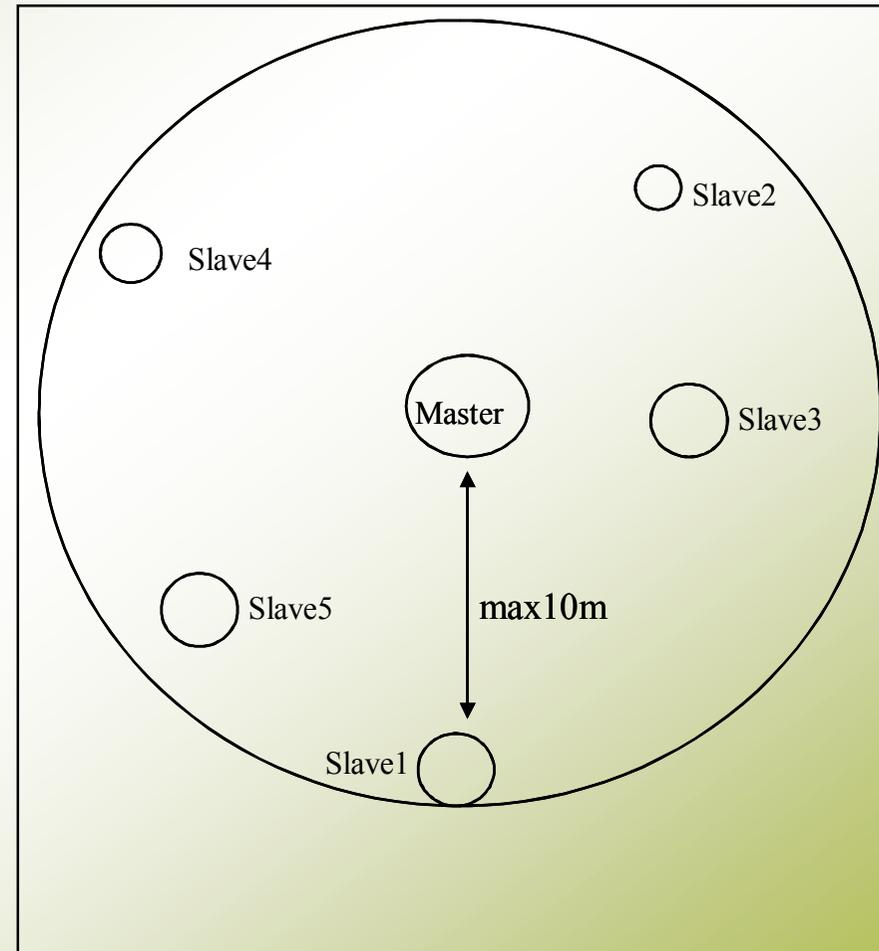
La modulazione utilizzata è la GFSK (Gaussian Frequency Shift Keying) con un coefficiente di modulazione BT=0.5 ad 1Mbps.

With modulating waveform $m(t)=A_m\cos(2\pi f_m t)$ the instantaneous carrier frequency is $f_i(t)=f_c+\Delta f\cos(2\pi f_m t)$ and Δf is peak frequency deviation. The modulation index, $\beta=\Delta f/f_m$, should be kept between 0.28 and 0.35, which defines it as a narrowband FM system.

A binary one is represented by a positive frequency deviation, and a binary zero is represented by a negative frequency deviation, and must be kept with a timing better than +/-20ppm. With GFSK the information data has to pass a Gaussian shaped lowpass filter in order to maintain the lowest possible frequency occupancy. GFSK is also used in the DECT system and is also close to GMSK used in GSM systems.

CARATTERISTICHE GENERALI

- BT supporta connessioni “*punto a punto*” nelle quali solo due moduli sono coinvolti, o connessioni “*multipunto*”, laddove in quest’ultimo caso diversi moduli BT condividono lo stesso canale fisico (**Piconet**)
- In ogni piconet vi è un solo modulo *master*, e diversi moduli *slave*: vi possono essere fino a sette slave attivi per volta.

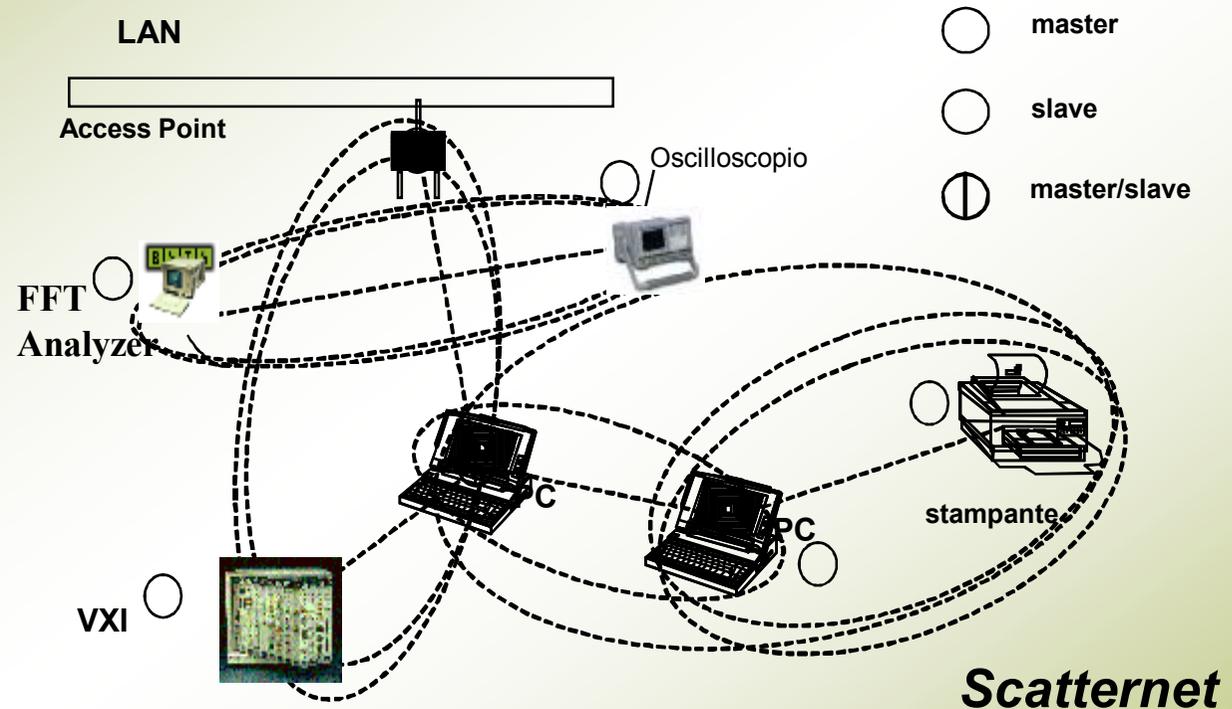


Esempio di Piconet

IL MONDO DI BT

La Scatternet:

se più *piconet*, ognuna con il proprio master ed il proprio ciclo di hop, ricoprono la stessa area, lo stesso modulo BT può operare come master in una *piconet* e come slave nelle altre che si vengono a formare.



L'insieme dei comandi necessari per gestire una connessione tra dispositivi BT è racchiuso in un protocollo denominato *Host Controller Interface* (HCI).

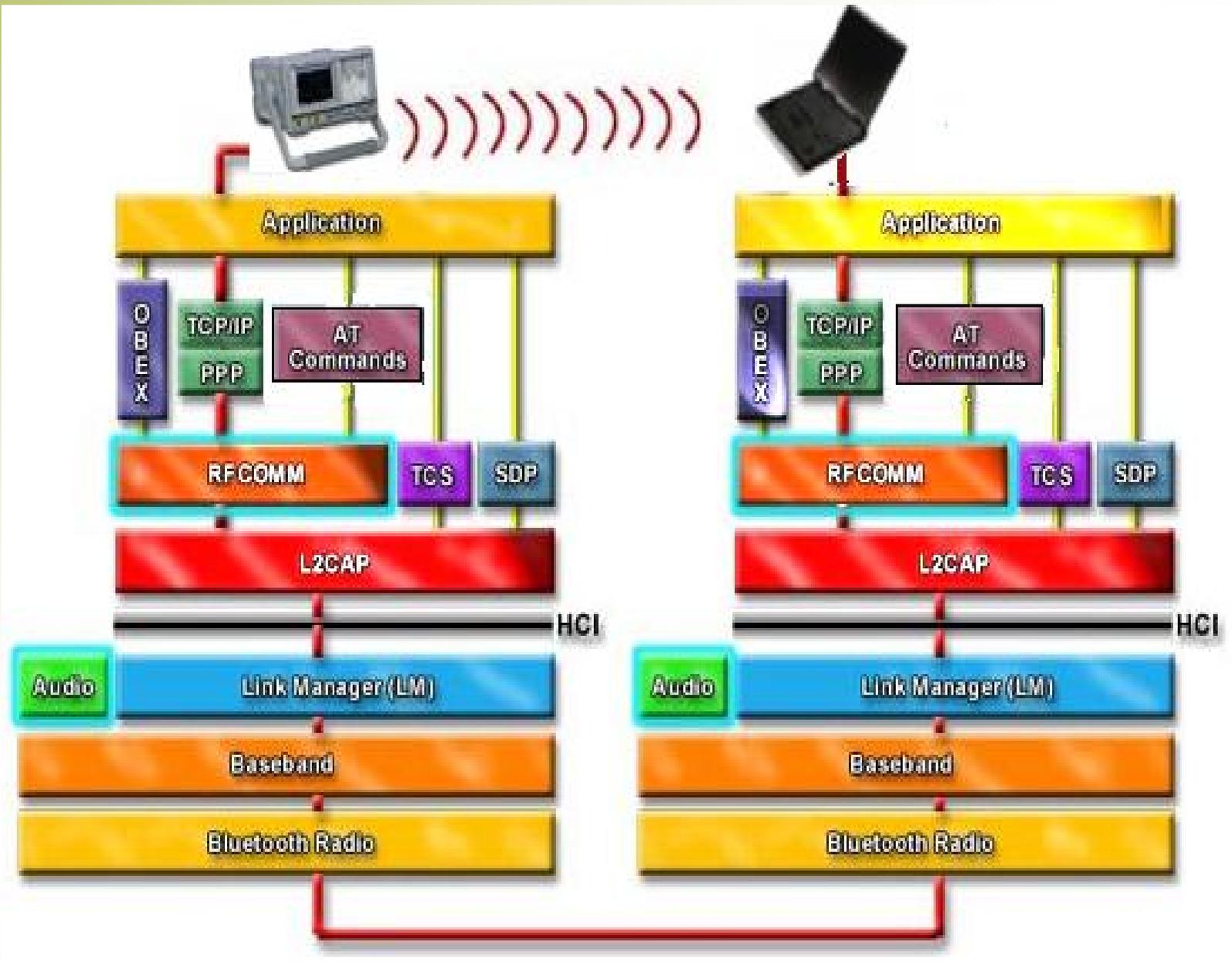
LA TECNOLOGIA BLUETOOTH

Caratteristiche principali

Ogni dispositivo è classificato in una delle tre classi di potenza:

- *Classe 1: dispositivi con ampia portata (fino a 100 metri) con una potenza massima di trasmissione di 20 dBm (100 mW)*
- *Classe 2: dispositivi per distanza ordinarie (10 metri) con potenza massima di trasmissione di 4 dBm (2,5 mW)*
- *Classe 3: dispositivi a corto raggio (10 cm) con potenza di trasmissione di 0 dBm (1 mW)*

LO STACK BT



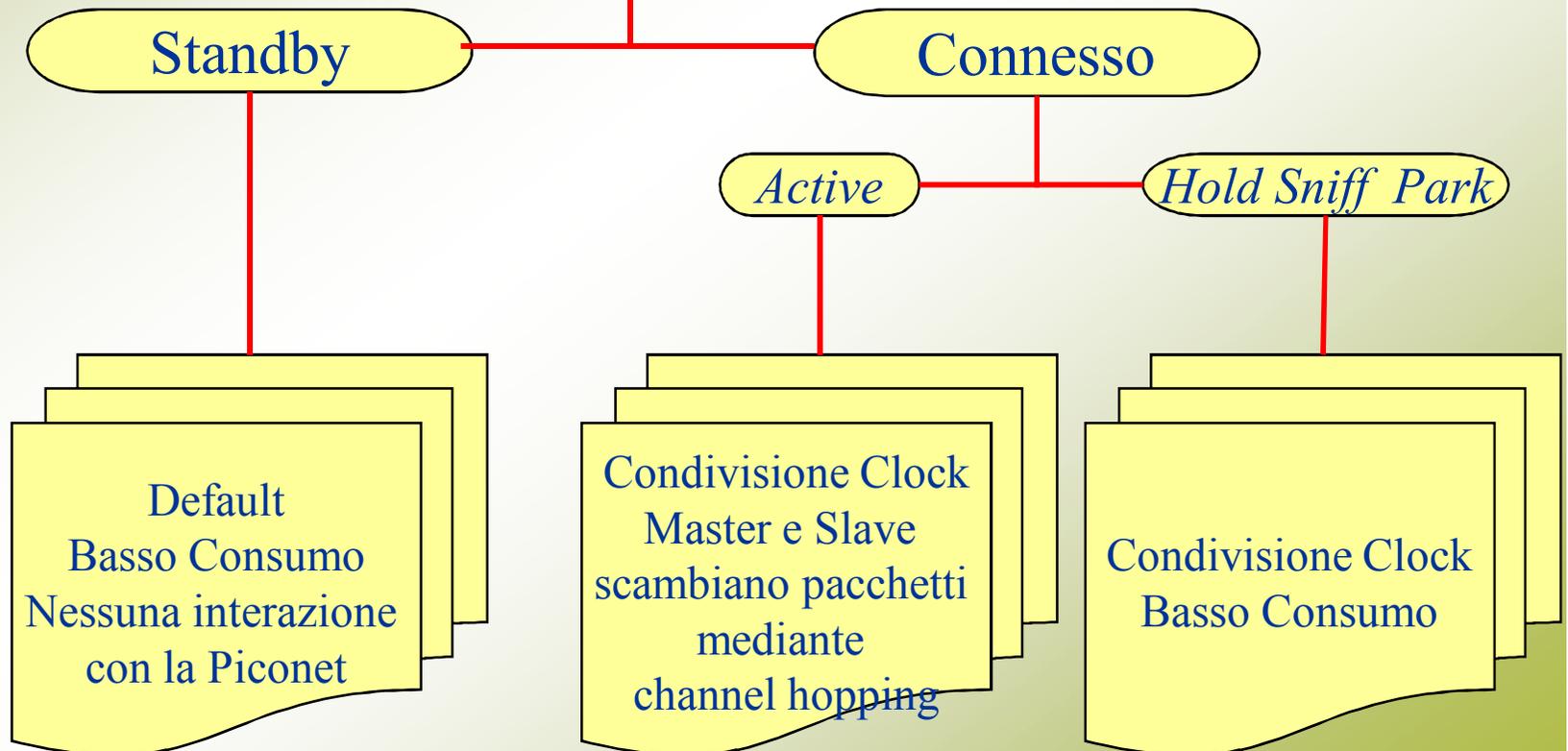
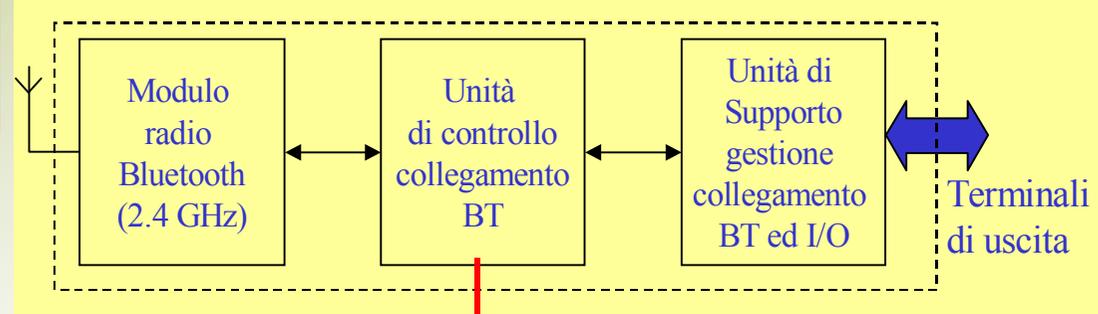
Profili

- **Generic**
Generic Access, Service Discovery
- **Serial**
Serial Port, Generic Object Exchange, File Transfer, Object Push, Synchronization
- **Telephony**
Cordless Telephony, Headset, Intercom
- **Networking**
Dial Up Networking, Fax, LAN access

Composizione di un dispositivo BT



IL MONDO DI BT



Tipi di pacchetti e Tassi

Packet Types			
SEGMENT	TYPE	SCO link	ACL link
1	0000	NULL	NULL
	0001	POLL	POLL
	0010	FHS	FHS
	0011	DM1	DM1
2	0100		DH1
	0101	HV1	
	0110	HV2	
	0111	HV3	
	1000	DV	
	1001		AUX1
3	1010		DM3
	1011		DH3
	1100		
	1101		
4	1110		DM5
	1111		DH5

Data Rates (Kbps)			
TYPE	symmetric	asymmetric	
DM1	108.8	108.8	108.8
DH1	172.8	172.8	172.8
DM3	256.0	384.0	54.4
DH3	384.0	576.0	86.4
DM5	286.7	477.8	36.3
DH5	432.6	721.0	57.6



IL BUS PER STRUMENTAZIONE PROPOSTO



Principali Caratteristiche

- **BUS a STELLA:** ogni comando di *periferica* inviato o ricevuto dagli strumenti è sempre letto o scritto dal controller.
- Fino a 64 periferiche presenti sul bus (moduli BT di tipo slave) utilizzando le modalità *hold*, *sniff* e *park*, e fino a 7 slave contemporaneamente attivi.
- Possibilità di creare più bus di misura (scatternet):
 - (a) Possibilità di interazione tra diversi bus,
 - (b) aumento del raggio coperto dal bus wireless.

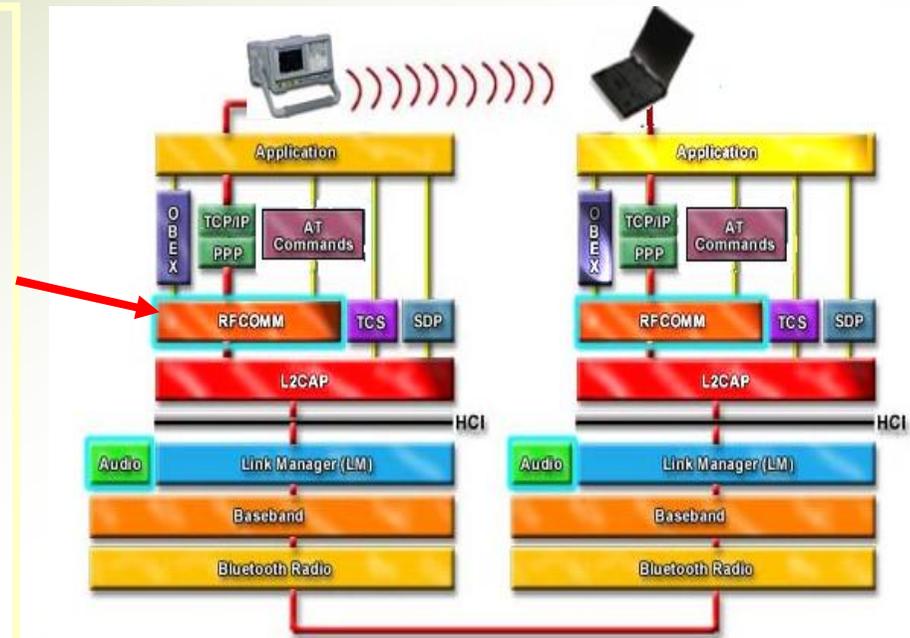
UN GROSSO PROBLEMA!!!

PREMESSA

BT realizza l'emulazione del collegamento seriale attraverso un opportuno livello di protocollo chiamato RFCOMM.

CONSEGUENZA

- (i) Ogni periferica deve possedere una porta seriale RS232
- (ii) GESTIRE lo stack HCI
- (iii) GESTIRE RFCOMM per l'emulazione della porta seriale.



PC controller
(BT Master)



Non vi sono grossi problemi, bisogna realizzare un driver software capace di allocare sullo stack HCI le necessarie funzioni RFCOMM.

Periferica
(BT Slave)



Ogni strumento di misura utilizza la propria porta seriale solo per le istruzioni di misura e la comunicazione dei dati e non possiede in alcun modo le funzioni RFCOMM indispensabili per gestire un modulo BT in emulazione seriale.

Lo Standard ZigBee

LA TECNOLOGIA ZIGBEE

- ***La tecnologia ZigBee si basa sullo standard IEEE 802.15.4 che ne definisce le direttive a livello fisico e del controllo degli accessi al canale***
- ***ZigBee identifica l'alleanza industriale che mira a promuoverne lo sviluppo e la diffusione***



LA TECNOLOGIA ZIGBEE

Tipologie di dispositivi

Esistono due tipologie di dispositivi:

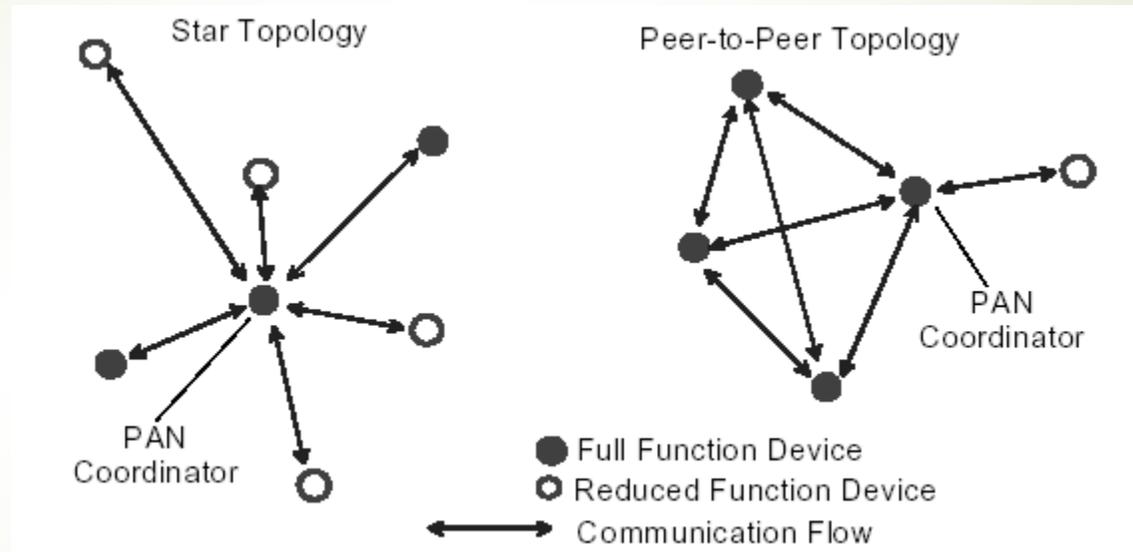
- ***“FFD”***: *Dispositivi sempre connessi alla rete che possono funzionare sia come coordinatore della rete (PAN Coordinator) che come un dispositivo qualunque. Possono dialogare con qualsiasi altro dispositivo (FFD/RFD)*
- ***“RFD”***: *Dispositivi non sempre connessi alla rete che possono funzionare solo come un dispositivo qualunque. Possono dialogare solo con dispositivi “FFD”*

LA TECNOLOGIA ZIGBEE

Topologie di reti

Due topologie di reti: “Star” e “Peer-to-Peer”

- *“Star”*: Ogni dispositivo può comunicare solo col coordinatore della rete (PAN Coordinator)
- *“Peer-to-Peer”*: Tutti i dispositivi possono comunicare tra di loro



LA TECNOLOGIA ZIGBEE

Topologie di reti

MIN un dispositivo FFD (Master)

**Numero MAX di elementi presenti in
una rete:**

- Fino a 65,536 nodi - Indirizzo a 16 bit



- **Network Coordinator**
- **Full Function node**
- **Reduced Function node**

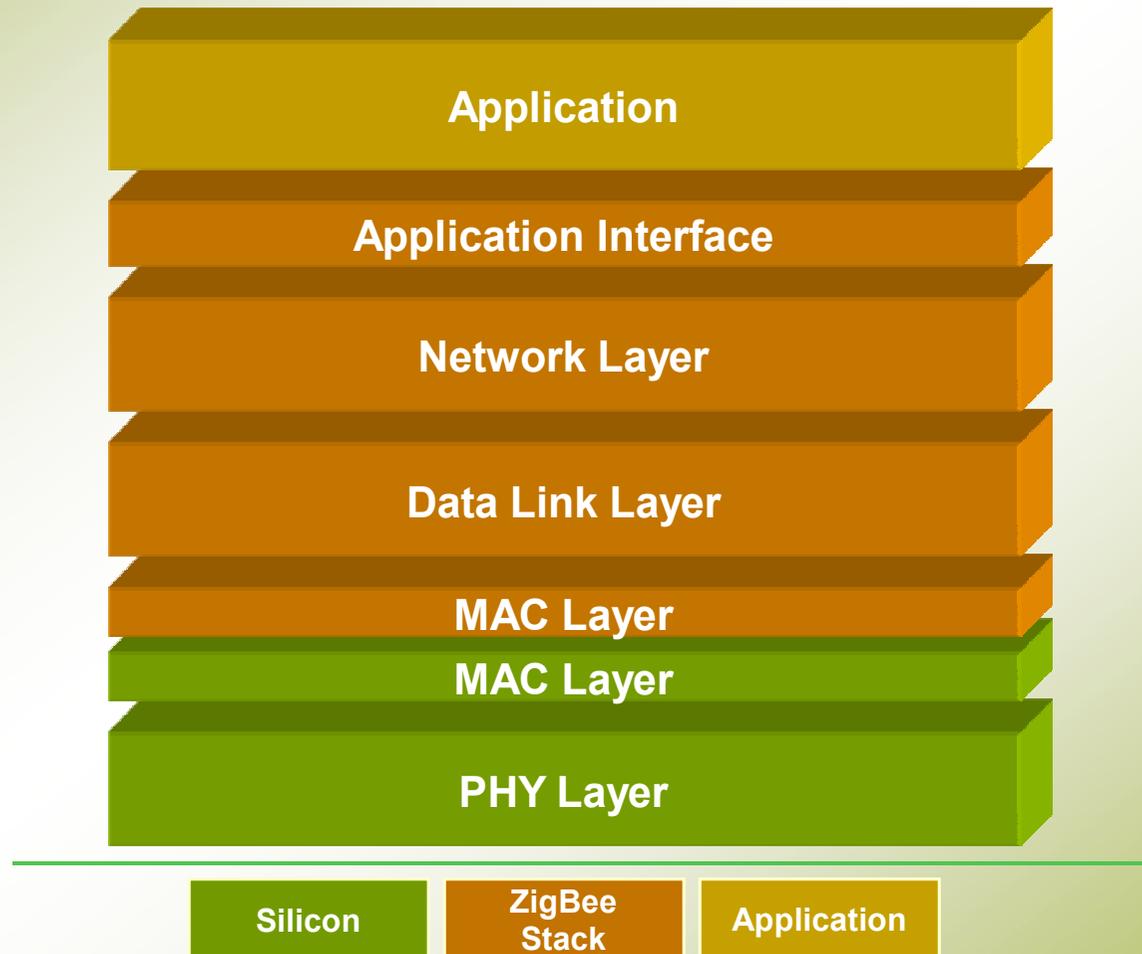
LA TECNOLOGIA ZIGBEE

Livello Fisico (PHY)

*Lo standard prevede tre bande di frequenza per la comunicazione, ognuna delle quali offre un diverso numero di canali e una differente velocità per i dati
Adotta un sistema di modulazione del tipo Direct Sequence Spread Spectrum (DSSS)*

	<u>MODULAZIONE</u>	<u>UTILIZZO</u>	<u>DATA RATE</u>	
<u>CANALI</u>				
2.4 GHz	O-QPSK	Ovunque	250 kbps	16
915 MHz	BPSK	Americhe	40 kbps	10
868 MHz	BPSK	Europa	20 kbps	1

Stack del Protocollo ZigBee



LA TECNOLOGIA ZIGBEE

Sottolivello Medium Access Control (MAC)

Lo standard prevede due modalità di accesso al canale:

- **“Senza Beacon”**: *La trasmissione può avvenire in qualsiasi momento, non vi è sincronizzazione tra i dispositivi*
- **“Con Beacon”**: *La trasmissione è originata dal coordinatore attraverso un pacchetto detto “Beacon Frame”, vi è quindi sincronizzazione tra i dispositivi, una migliore organizzazione della rete ed una notevole riduzione dei consumi*

LA TECNOLOGIA ZIGBEE

Tipi di pacchetti

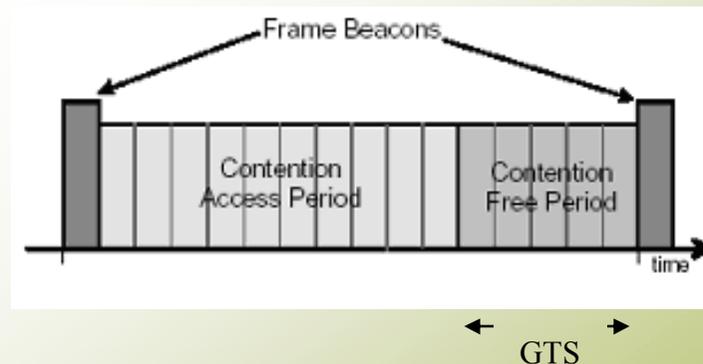
Lo standard definisce 4 strutture diverse di pacchetti:

- *“Beacon Frame”*: usati per la sincronizzazione tra i dispositivi
- *“Data Frame”*: usati per i dati da trasferire
- *“Acknowledgement Frame”*: usati per confermare l'avvenuta ricezione di un pacchetto
- *“MAC Command Frame”*: usati per il controllo dei trasferimenti

LA TECNOLOGIA ZIGBEE

Sottolivello Medium Access Control (MAC)

La modalità di accesso al canale “Con Beacon” prevede una “Struttura Superframe”: ogni coordinatore trasmette “Beacon Frame” ad intervalli. Tra ogni coppia di “Beacon Frame” vi sono 16 minislot temporali uguali per l’accesso al canale, di cui alcuni (GTS, fino a 7) possono essere riservati per applicazioni a bassa latenza o che richiedono una larghezza di banda specifica.



PERCHÉ ZIGBEE?

- *Perché consente bassissimi consumi in virtù di un duty-cycle anche $<1\%$*
- *Perché ha un basso costo*
- *Perché consente di ottenere un'alta densità di nodi per rete*



.... potrebbe essere una buona soluzione per applicazioni di misura

CONFRONTO ZIGBEE-BLUETOOTH

Ecco le principali differenze:

- *Bluetooth supporta la sola topologia “Star” mentre ZigBee supporta sia la topologia “Star” che quella “Peer-to-Peer”*
- *Ogni rete ZigBee può comprendere fino a 255 dispositivi (Bluetooth fino a 7)*
- *Lo stack protocollare ZigBee può arrivare ad occupare soli 32 KB (in Bluetooth fino a 250KB)*

CONFRONTO ZIGBEE-BLUETOOTH

- *ZigBee può raggiungere un Data Rate di 250 kbps, assai più basso dei 723 kbps di Bluetooth*
- *Negli stati a basso consumo ZigBee consente di ottenere consumi molto bassi (Sleep Mode ~ 1uA) rispetto a Bluetooth (Park, Hold e Sniff Mode ~ 15mA)*
- *Confronto su “tempi” caratteristici*

Considerazioni sui tempi

ZigBee:

- Tempo di unione alla rete = 30ms
- Tempo di passaggio allo stato “attivo” = 15ms
- Tempo di accesso al nodo “attivo” = 15ms

Bluetooth :

- Tempo di unione alla rete = >3s
- Tempo di passaggio allo stato “attivo” = 3s
- Tempo di accesso al nodo “attivo” = 2ms

ZIGBEE RISPETTO ALLE ALTRE TECNOLOGIE WIRELESS

