

Appunti sul BUS USB

A cura di Chiara Orrù

1 INTRODUZIONE

1.1 Nascita di USB

All'inizio degli anni '90, il numero sempre crescente di periferiche ha fatto nascere l'esigenza di un bus più facilmente espandibile rispetto a quelli esistenti; infatti, fino ad allora, per collegare un nuovo dispositivo al proprio PC, c'era bisogno di un particolare cavo (periferiche come stampante, modem, mouse hanno cavi e connettori completamente diversi), di spegnere e riaccendere il computer e di installare il software che riconoscesse e configurasse il dispositivo appena connesso. Tutto questo se si tratta di un dispositivo esterno: altrimenti, bisogna anche collegare la linea IRQ e il canale DMA esatti, sperando che non si creino conflitti. Questo crea problemi soprattutto a chi non è molto esperto; oltretutto, se aumenta il numero di periferiche, aumentano molto anche i costi, soprattutto per quanto riguarda i cavi ed eventuali expansion card necessarie per replicare il collegamento al PC.

1.2 Obiettivi

Per venire incontro a queste esigenze di mercato, nel 1995 quattro giganti del settore (Compaq, Intel, NEC, Microsoft) formano un consorzio dal quale nasce lo standard USB, ormai immancabile su qualsiasi postazione.

Gli obiettivi del consorzio erano principalmente:

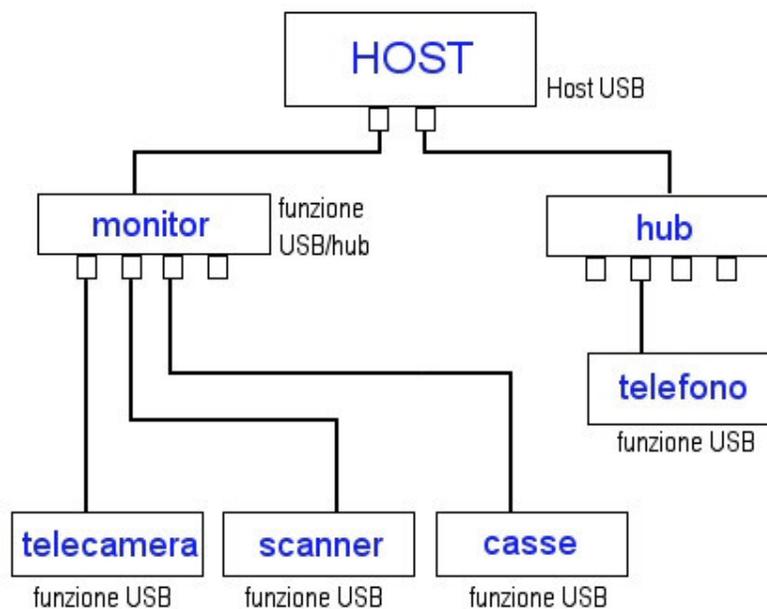
- Facilità d'uso per quanto riguarda la connessione delle periferiche;
- Un solo tipo di cavo e di porta per qualsiasi genere di periferica;
- Possibilità di attaccare molte periferiche alla stessa porta;
- Risoluzione automatica dei conflitti;
- Connessione "a caldo" dei dispositivi;
- Rilevazione e configurazione automatica delle periferiche;
- Supporto per funzioni in tempo reale come voce, audio e video;
- Compatibilità con tutte le configurazioni e tutti i costruttori;
- Basso costo sia dal lato PC che dal lato periferica;
- Bassi consumi;

1.3 Caratteristiche

- Bus molto complesso sia dal lato host (il PC) che dal lato periferica. L' host infatti deve essere in grado di riconoscere la connessione e la disconnessione di una periferica, e capire che tipo di periferica sia; le periferiche, invece, nonostante siano passive (ogni operazione sul bus USB, infatti, è comandata dall'host), devono avere un

microcontrollore che gestisca le comunicazioni e che implementi lo stack di protocolli necessario al funzionamento di questo sistema.

- Stack di protocolli: per implementare una struttura così complessa c'è bisogno di un software adeguato, organizzato a livelli, che nasconda la struttura fisica di ogni periferica e presenti un'interfaccia uniforme ai programmi utente. Per fare un esempio, si pensi all'impossibilità di accedere a una rete senza aver installato la pila di protocolli rappresentata da TCP/IP; allo stesso modo, non è possibile far funzionare il bus USB senza il suo software. Sull'host deve quindi essere presente il software USB di sistema (SWU), il quale ha come scopo principale di rendere le periferiche di semplice utilizzo per qualsiasi utente, che non si deve così preoccupare di fasi complesse come configurazione o installazione.
- Le comunicazioni sul bus sono seriali e asincrone: l'iniziativa viene presa sempre dall'host che è l'unico master del sistema, e le periferiche devono solo rispondere alle sue richieste. Sono previsti anche trasferimenti isocroni, necessari alla trasmissione di traffico voce o video.
- Gli Hub sono particolari periferiche con la sola funzione di replicare il collegamento al bus: in questo modo si realizza la struttura ad albero del bus usb: infatti, sul PC ci sono solo due porte (in realtà è solo una, ma di solito è presente un hub integrato che sdoppia questo collegamento) ma se, per esempio, su una di esse colleghiamo un hub a 4 porte, le periferiche collegabili diventeranno 5. L'hub deve avere, come tutte le periferiche, un controllore per poter riconoscere su quali porte sono collegati dei dispositivi, in modo da poterlo comunicare all'host quando quest'ultimo lo interroga. Esistono anche periferiche miste, che hanno sia funzione di hub che di periferica.



- Il bus è in grado di alimentare i dispositivi a basso consumo tramite le stesse linee su cui passano i dati; dato che si tratta di un sistema ad albero, più dispositivi sono connessi, e meno potenza sarà a disposizione di ognuno.

- Si possono collegare fino a 127 periferiche sul bus, grazie a questa struttura ad albero; più aumenta il numero di periferiche, però, peggiori saranno le prestazioni, dato che le risorse disponibili (in particolare la banda) devono essere distribuite fra le varie periferiche. Inoltre, poiché ogni periferica fisica può avere più funzioni, le periferiche "logiche" totali potranno essere anche più di 127.
- Gran parte del risparmio che USB introduce deriva dal fatto che non è più necessario, per connettere un maggior numero di periferiche, installare una nuova scheda, dato che le periferiche si possono attaccare direttamente alla porta USB. Oltretutto, cavi e i connettori sono ormai molto economici, dato che il loro mercato è molto ampio e dunque ci sono molti costruttori che li producono, abbattendone il prezzo. Poi, i costi si riducono ulteriormente quando si pensa a quanti piedini sono necessari su un qualsiasi bus parallelo, che sono molti di più rispetto a quelli su USB.
- La priorità è statica; viene decisa in fase di configurazione dal SWU, e dipende dalla natura della periferica e dalla sua posizione nell'albero.

1.4 Velocità

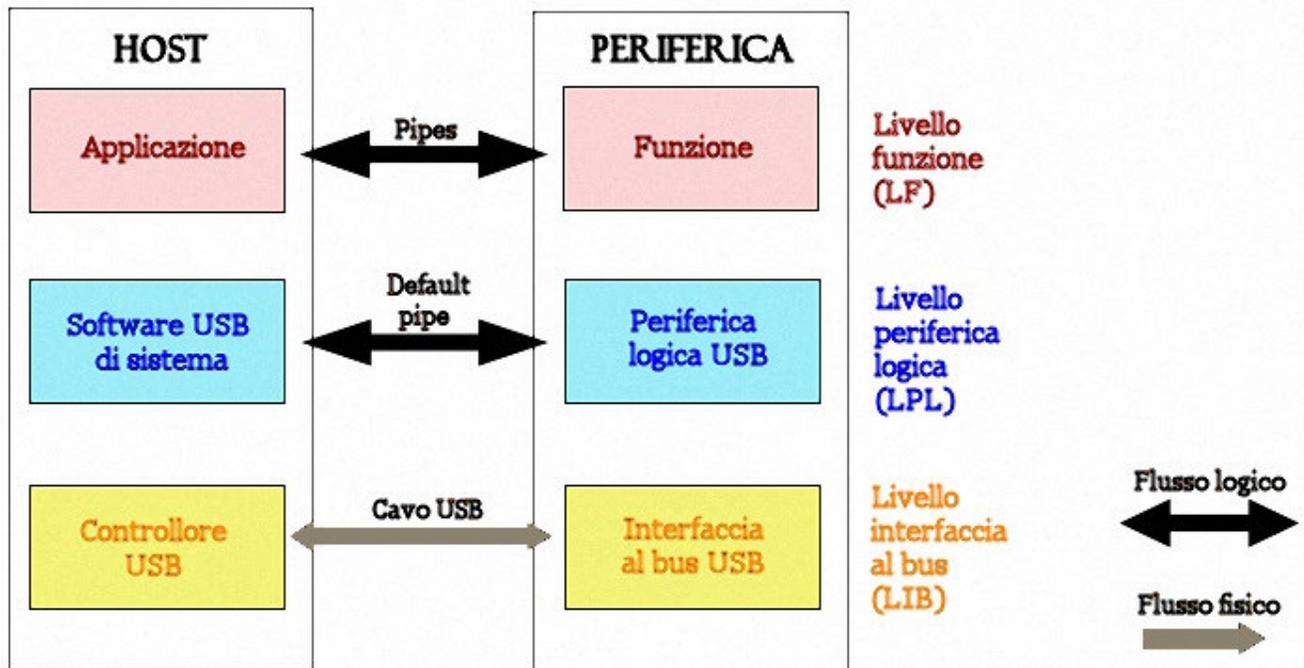
Versione	Banda (Mbit/s)
1.x	1.5
	12
2.0	Fino a 480 (supporta anche le altre due velocità)

La creazione dello standard USB 2.0 risponde al fatto che i PC ormai sono in grado di processare una grande quantità di dati alla volta e, allo stesso tempo, le periferiche hanno sempre più funzionalità. Questa versione consente poi a un gran numero di periferiche a bassa e media velocità di operare sullo stesso bus.

2 FUNZIONAMENTO

2.1 Stack di software

Dal punto di vista logico, a parte la priorità, tutte le periferiche sono allo stesso livello; attraverso il Software Usb di sistema (SWU), l'utente vede solo le loro funzioni: i programmi applicativi non devono preoccuparsi delle comunicazioni sul bus che vengono gestite dal SWU, il quale presenta un'interfaccia molto semplice verso il livello più alto. La pila, in generale, si può schematizzare in questo modo:



Livello di interfaccia al bus (LIB): contiene l'hardware necessario a mandare i pacchetti sul cavo; ha una struttura diversa su host o periferica, ma in ogni caso contiene della logica che serializza la trasmissione.

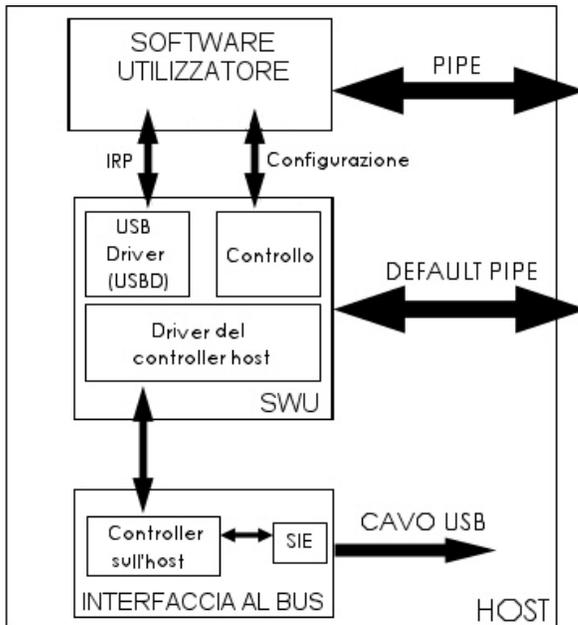
Livello di periferica logica (LPL): contiene il software che maschera la struttura fisica della periferica; si tratta allora del SWU per quanto riguarda l'host, e, invece, per la periferica, della logica che elabora i flussi di controllo.

Livello funzionale (LF): per l'host, è il software utilizzatore, quindi i programmi applicativi, i quali vedono le funzioni implementate dalla periferica come se essa fosse direttamente connessa al computer; per la periferica, sono le funzioni vere e proprie.

2.2 Lato Host

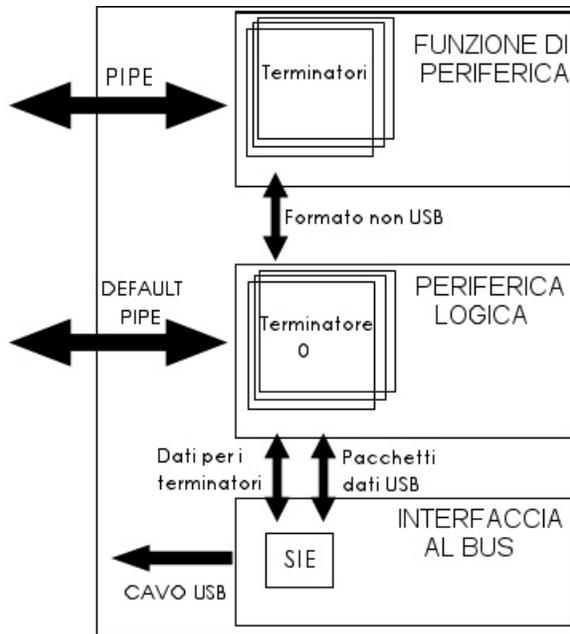
- **LF:** composto da programmi applicativi, i quali, per comunicare con la periferica, devono instaurare un canale (vedi Modello dei flussi informativi), operazione che viene fatta nella fase di configurazione. Ovviamente, per trasmettere dati alla periferica, si deve trasmettere al livello inferiore, e i dati inviati o ricevuti sono sempre organizzati in pacchetti.
- **LPL:** a questo livello troviamo il Software USB di Sistema, che gestisce il bus dal punto di vista operativo, come per esempio l'amministrazione della banda e dell'alimentazione. Nel software sono inclusi due driver, uno per comunicare con il livello superiore (USB Driver, USBD), e uno con il livello inferiore. Il primo ha sostanzialmente la funzione di nascondere al livello più alto la struttura fisica della periferica, fornendone una rappresentazione.
- **LIB:** l'interfaccia al bus è composta dal controllore e da un hub integrato (all'esterno infatti di solito si vedono due o più porte, ma non si tratta di due bus diversi); i dati in

ingresso e in uscita sono confezionati dal controllore e spediti sul cavo sotto forma di impulsi elettrici generati dal SIE (Serial Interface Engine).



2.3 Lato Periferica

- **LF:**
le funzioni della periferica vengono viste ognuna come un terminatore (vedi Modello dei flussi informativi), collegati al software USB di sistema tramite altrettanti canali.
- **LPL:**
a questo livello si instaura il canale di base che, in fase di configurazione, serve all'host per portare a termine il riconoscimento della periferica.

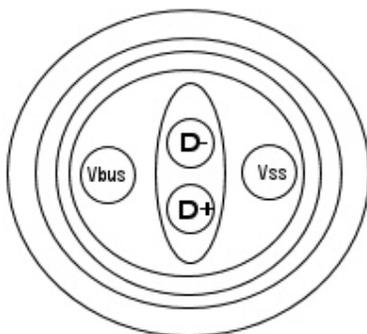


2.4 Modello dei flussi informativi

Le comunicazioni sul bus USB avvengono tramite *terminatori* e *canali*. Un terminatore è un registro o un'area di memoria presente sulla periferica e indirizzabile dall'host. I flussi logici di informazioni fra host e un terminatore sono detti invece canali. Su ogni periferica deve essere presente il terminatore 0, il quale contiene i dati caratteristici della periferica stessa; deve quindi poter essere letto dall'host in fase di configurazione, al fine di assegnare al dispositivo le quantità di banda e alimentazione più adatte al suo funzionamento, e un indirizzo. Fra i livelli LPL di host e periferica (quindi fra Software USB di sistema e terminatore 0) si instaura un canale, detto canale di base o Default Control Pipe. A livello più alto, le funzioni della periferica vengono ancora viste come terminatori, e il SWU vi accede dedicando un canale a ognuna di esse.

2.5 Il cavo

Il cavo USB è formato da quattro fili: due, intrecciati fra loro (D+ e D-) servono al trasferimento delle informazioni, gli altri due (Vbus e Vss) per l'alimentazione. D- è di solito tenuto a massa, mentre D+ ha una tensione positiva variabile. La trasmissione dei dati avviene tramite variazioni della tensione $\Delta V = (D+) - (D-)$: questa differenza può assumere diversi valori, a ogni intervallo dei quali corrisponde un diverso modo di interpretare i dati sul bus. Il SIE riconosce, al variare di D+ e di ΔV , quattro stati fisici, non solo due come nei normali sistemi di trasmissione.

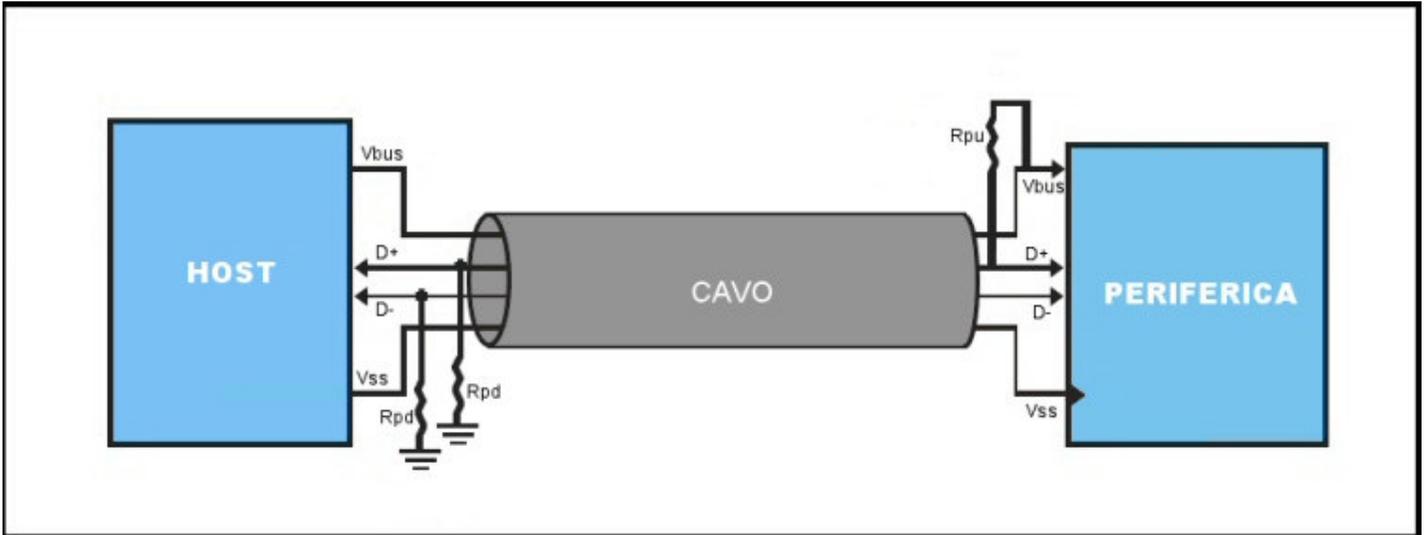


Per quanto riguarda le altre due linee, Vss è solitamente collegato alla massa, e Vbus all'alimentazione; forniscono una piccola quantità di potenza ai dispositivi a basso consumo (circa 0.5 W per periferica) e, come vedremo, permettono di riconoscere la connessione e la disconnessione delle periferiche.

2.6 Connessione di una periferica

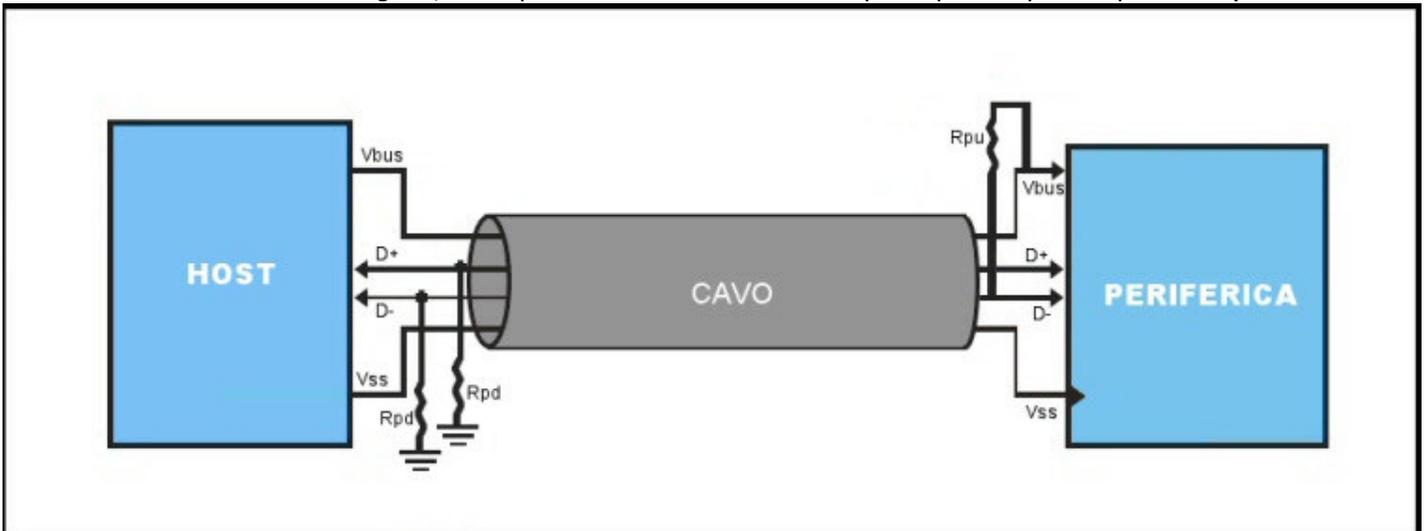
Periferica "veloce":

$V_{ss} = 0\text{ V}$
 $V_{bus} = 5\text{ V}$
 $R_{pd} = 15\text{ K}\Omega$
 $R_{pu} = 1,5\text{ K}\Omega$



Periferica "lenta":

Come si vede dalla figura, sono presenti delle resistenze di pull-up sulle porte upstream (cioè



dal lato periferica, che collegano D+ o D- all'alimentazione, a seconda che la periferica sia "veloce" o "lenta") e di pull-down sulle porte downstream (lato host, che portano sia D+ che D- a massa). Questi collegamenti fanno sì che il riconoscimento di una connessione sia automatico e che si riconosca anche il tipo di periferica: vediamo come.

Nel caso in cui il cavo sia collegato dal lato host ma non da quello periferica, sia D+ che D- sono a massa e non succede niente. Quando collego il cavo anche alla periferica, si forma un partitore di tensione fra Vbus e Vss, su D+ se la periferica è lenta, o su D- se è veloce. Si avrà

quindi una caduta di tensione su D+ o D-, circa del 10%, che supera la soglia V_{ih} oltre la quale, se questa condizione permane per 2.5 μs , l'hub riconosce l'avvenuta connessione. Tutte le periferiche USB (Hub compresi), però, sono passive: l'hub non può dare all'host notizia della connessione di sua iniziativa: allora è l'host che, periodicamente, interroga l'hub; quest'ultimo notificherà così la presenza della nuova periferica, e l'host risponde abilitando la porta su cui è connessa e dando inizio alla fase di configurazione.

2.7 Disconnessione

Durante il funzionamento, una delle due linee D (D+ per le periferiche veloci, D- per le lente) è sopra alla soglia V_{ih} , mentre l'altra è vicina alla massa. Quando si ha la disconnessione, sulla linea a livello alto la tensione si abbassa fino a scendere sotto una soglia V_{il} ; se questa condizione permane per 2.5 μs l'hub assume che c'è stata una disconnessione. Come prima, alla seguente interrogazione l'host disabilita la porta e rende nuovamente disponibili le risorse che appartenevano alla periferica.

2.8 Configurazione

La configurazione di una periferica USB avviene in vari passi: prima comunicano host e hub, dato che la porta a cui è connessa la periferica non è ancora abilitata, e poi l'host può comunicare direttamente con il dispositivo.

- I. L'hub riconosce una nuova connessione ma non la specifica periferica, tramite i meccanismi visti;
- II. L'host, tipicamente ogni ms, lancia un messaggio detto SOF per interrogare gli hub;
- III. L'hub comunica che c'è stata una variazione di stato,
- IV. Il Software USB inizia uno scambio di messaggi con l'hub per sapere che tipo di variazione di stato sia avvenuta e riceve in risposta l'identificatore della porta a cui è connessa la periferica;
- V. Si aspettano 100 ms per consentire la stabilizzazione elettrica del bus, e poi l'host manda un comando all'hub per abilitare la porta in questione e attivare il terminatore 0 della periferica: a questo punto la periferica si trova nello *stato di default*;
- VI. L'host assegna un indirizzo a 7 bit alla periferica, che ora si trova nello *stato addressed*;
- VII. Attraverso il canale di base, l'host può leggere il terminatore 0 e ricavare informazioni sulla particolare periferica, e infine assegnare le risorse necessarie alla periferica stessa, che si trova nello *stato configured*.

2.9 Trasferimento dati

Esistono due metodi, mutuamente esclusivi, di comunicazione sui canali:

1. *Sequenziale*: i dati che passano sul canale non hanno una struttura definita dallo standard USB, ma sono nel formato specifico della particolare periferica; viaggiano nello stesso ordine con cui sono stati prodotti. Di solito un canale di questo tipo è destinato a un unico utilizzatore, dato che SWU non è in grado di sincronizzare richieste concorrenti.

SOTTOMODALITA'

- *BULK*: è più o meno un trasferimento a burst; bisogna minimizzare il tempo di trasmissione di un blocco, quindi viene concessa tutta la banda disponibile a questo trasferimento: va bene per blocchi che non hanno vincoli temporali fra di loro.
- *ISOCRONO*: i dati devono arrivare al destinatario con la stessa cadenza temporale con cui sono stati prodotti, e il bus deve fare in modo che questi

vincoli temporali vengano rispettati, allocando la maggior parte di banda disponibile a questa trasmissione.

- *INTERRUPT*: ogni ms, se possibile, l'host fa un controllo di programma per vedere se qualche periferica ha bisogno di interruzioni; questo avviene perchè le periferiche USB non possono prendere l'iniziativa e devono aspettare i comandi dell'host.
2. *A messaggi*: i dati hanno una struttura USB, e la loro struttura non dipende quindi dal tipo di periferica. La comunicazione avviene con una sequenza che comprende una richiesta dell'host, un'operazione completata dalla periferica, e un messaggio di stato inviato dall'host in merito alla conclusione del comando. Il Software USB di sistema garantisce che non ci siano più richieste contemporanee, e se c'è più di una richiesta vengono gestite solitamente secondo la logica FIFO.

RIFERIMENTI

G. Bucci *Architetture dei calcolatori elettronici*, McGraw-Hill