

## Hardware

9.1	Alimentatori AT e ATX	236
9.1.1	Interruttore e cambio tensione	236
9.1.2	Fusibile	238
9.1.3	Dissipazione del calore	239
9.1.4	Pulizia	240
9.1.5	Alimentazione e affidabilità dell'elaboratore	241
9.1.6	Connettori comuni di alimentazione	241
9.2	Alimentatori ATX	242
9.2.1	Collegamento principale con la scheda madre: 20 poli	242
9.2.2	Collegamento principale con la scheda madre: 24 poli	243
9.2.3	Collegamento ausiliario con la scheda madre, +12 V	244
9.2.4	Collegamento ausiliario con la scheda madre «P6», +3,3 V, +5 V,	244
9.2.5	Controllo del funzionamento	244
9.2.6	Utilizzo di un alimentatore ATX per scopi vari	245
9.3	(P)ATA: AT attachment	245
9.3.1	Caratteristiche generali del sistema PATA	246
9.3.2	Geometria dei dischi	247
9.3.3	Caratteristiche del protocollo PATA e configurazione del firmware	248
9.3.4	Standard PATA e gergo comune	249
9.3.5	Gestione da parte di un kernel Linux delle unità PATA	250
9.3.6	Informazioni da un kernel Linux	251
9.4	SATA: Serial ATA	251
9.4.1	Adattatori	252
9.4.2	Gestione da parte del kernel Linux	252
9.4.3	Informazioni da un kernel Linux	253
9.4.4	Sistemi RAID SATA	253
9.5	Da ISA a PCI	254
9.5.1	ISA	254
9.5.2	MCA	255
9.5.3	EISA	255
9.5.4	VESA local bus	255
9.5.5	PCI	255
9.5.6	Informazioni da un kernel Linux	256
9.5.7	AGP	257
9.5.8	AMR	258
9.5.9	PCI Express	259
9.6	USB	259
9.6.1	Caratteristiche generali	259
9.6.2	Alimentazione elettrica	260
9.6.3	Collegamento di una porta USB	260
9.6.4	Tipi di unità di controllo	261
9.7	USB e il Kernel Linux	261
9.7.1	Avvio di un sistema GNU/Linux da un'unità USB, con kernel monolitico	261
9.7.2	Sequenza di caricamento dei moduli del kernel	263
9.8	Introduzione al BIOS	265
9.8.1	Collocazione fisica del BIOS	265
9.8.2	Blocco di avvio	266
9.8.3	Modifica del BIOS	266

9.8.4	BIOS di «marca»	267
9.8.5	Salvare e riscrivere il BIOS	273
9.8.6	BIOS Award	274
9.8.7	BIOS AMI	274
9.8.8	UniFlash	274
9.8.9	Recupero disperato di un BIOS	276
9.9	Modifica di un BIOS modulare	276
9.9.1	BIOS modulari Award	277
9.9.2	BIOS modulari AMI	278
9.10	Memoria centrale difettosa	280
9.10.1	Esclusione di altri fattori	280
9.10.2	Memtest86+	281
9.10.3	Parametri di avvio del kernel	282
9.11	Restauro	283
9.11.1	Ripulitura	283
9.11.2	Lavaggio di unità a dischetti	284
9.11.3	Pulizia dei monitor	285
9.11.4	Pulizia azzardata dei monitor	286
9.11.5	Pulizia della tastiera	287
9.11.6	Batteria	288
9.11.7	Osservazioni sulla configurazione BIOS di elaboratori molto vecchi	288
9.12	Connettori e cablaggi	290
9.12.1	Video MGA, CGA, EGA, Hercules	290
9.12.2	Tastiere	291
9.12.3	Porte seriali	292
9.12.4	Porte parallele	292
9.12.5	Reti Ethernet	294
9.13	Riferimenti	296

Lo sviluppo attuale della tecnologia è tale da rendere poco utile la conoscenza di dettagli sull'hardware, in quanto gli strumenti informatici sono sempre più integrati e sempre meno espandibili. Pertanto, le informazioni di questo capitolo sono utili soprattutto per chi ha a che fare o vuole riutilizzare hardware del passato.

## 9.1 Alimentatori AT e ATX

Gli alimentatori «AT» sono quelli che sono stati usati inizialmente per gli elaboratori con microprocessore i286; gli alimentatori «ATX» sono stati usati più avanti, per aggiungere delle funzionalità che non era possibile realizzare con quelli della generazione precedente. Tuttavia, tali alimentatori hanno delle caratteristiche comuni, che vengono considerate in questo capitolo.

Quando si apre un alimentatore, si deve ricordare che al suo interno viene elaborata l'energia proveniente dalla rete elettrica comune, con la quale è possibile rimanere folgorati, se non si agisce con la competenza e la prudenza adeguate alla circostanza.

### 9.1.1 Interruttore e cambio tensione

Sul lato posteriore, vicino alla spina tripolare da pannello per l'ingresso dell'alimentazione elettrica, ci può essere un piccolo interruttore, con il quale si può spegnere l'alimentatore.

Figura 9.1. Pannello posteriore di un alimentatore tipico. Alla destra della grata della ventola, si vede: la spina tripolare da pannello; il cambio tensione; l'interruttore.

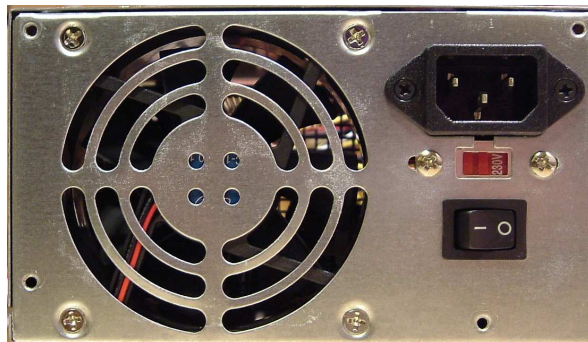
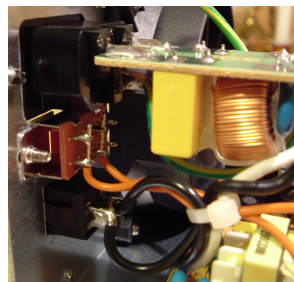


Figura 9.2. Spina tripolare da pannello, cambio tensione e interruttore, visti dall'interno.



Sempre sul lato posteriore, può essere presente un piccolo commutatore a slitta, per selezionare la tensione di alimentazione della rete elettrica. Di solito, il commutatore è collegato soltanto a due fili, che, se vengono messi in cortocircuito, fanno sì che l'alimentatore si aspetti di ricevere dall'esterno una tensione più bassa. In tal caso (lo si può verificare facilmente con un piccolo *tester*), considerato che in Italia viene fornita soltanto la tensione di 220 V a 50 Hz e che il cambio tensione serve, di norma, per selezionare una tensione di alimentazione pari a circa la metà di quella normale, può essere utile rimuovere il collegamento a questo cambio tensione, in modo da evitare errori accidentali nella sua configurazione.

Figura 9.3. Rimozione del commutatore del cambio tensione: distacco del commutatore dal pannello.

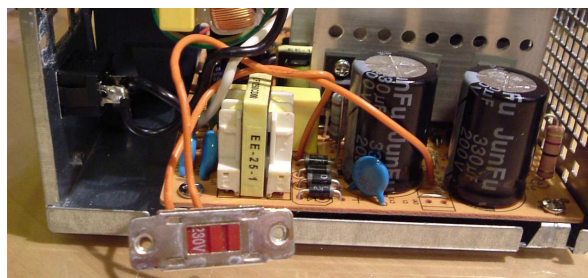
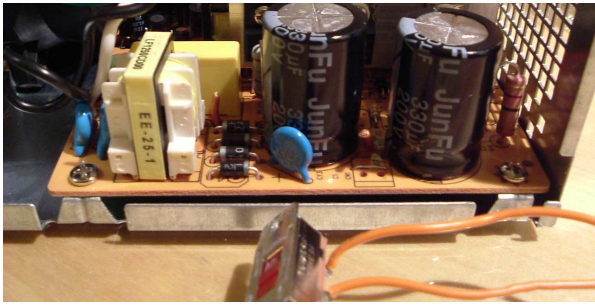


Figura 9.4. Rimozione del commutatore del cambio tensione: taglio dei fili.



### 9.1.2 Fusibile

All'interno dell'alimentatore, si trova normalmente un fusibile di protezione contro gli assorbimenti eccessivi, soprattutto allo scopo di evitare incendi. Generalmente, quando si brucia il fusibile, l'alimentatore ha un guasto; tuttavia, se non si riscontrano bruciature o rotture di componenti, si può tentare la sostituzione di un fusibile interrotto.

Figura 9.5. Integrato di regolazione bruciato; lo si capisce dalla rottura che appare nella parte inferiore. In una situazione come questa non serve a nulla cambiare il fusibile.



Figura 9.6. Fusibile. In questo caso, il fusibile è saldato e non può essere sostituito facilmente.



La sostituzione del fusibile deve avvenire utilizzandone un altro con le stesse caratteristiche. Generalmente si tratta di fusibili rapidi (lettera «F», per *fast*), per la corrente va letta sulla coroncina metallica del fusibile stesso. Eventualmente, si può stimare il valore della corrente di un fusibile, in base alla potenza che deve essere erogata dall'alimentatore, come si vede nel grafico contenuto nella figura successiva.

Figura 9.7. Stima della corrente massima del fusibile in funzione della potenza continua erogabile dall'alimentatore. Si osservi che il valore della corrente cambia anche in funzione della tensione di alimentazione usata in ingresso.

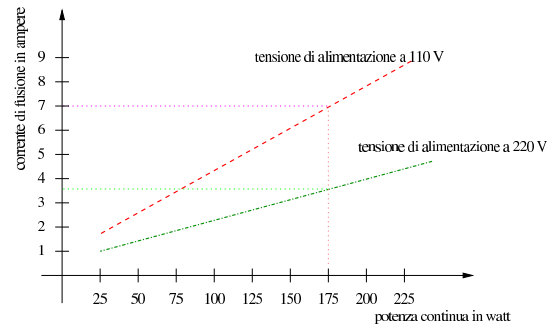
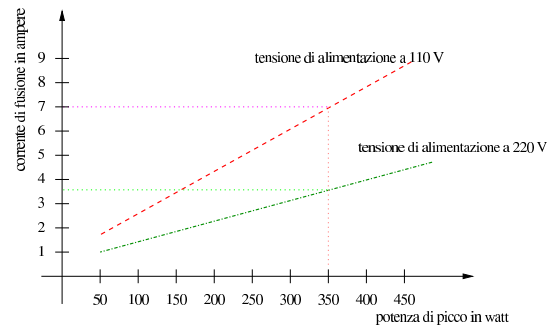


Figura 9.8. Stima della corrente massima del fusibile in funzione della potenza massima (di picco) erogabile dall'alimentatore. Si osservi che di norma, la potenza massima corrisponde al doppio della potenza continua (media) erogabile.



### 9.1.3 Dissipazione del calore

Gli alimentatori richiedono un sistema di dissipazione del calore, prodotto dai componenti che regolano la tensione di alimentazione, attraverso il controllo del flusso di corrente elettrica. I componenti che producono calore sono attaccati fisicamente a delle alette di raffreddamento di alluminio, per facilitare la dissipazione del calore stesso; tuttavia, queste alette richiedono un flusso continuo di aria per potersi raffreddare.

L'alimentatore tipico contiene una ventola per il raffreddamento dei dissipatori contenuti all'interno. Questa ventola si colloca normalmente sul pannello posteriore dell'alimentatore ed è orientata in modo da espellere l'aria all'esterno. Eccezionalmente, gli elaboratori che puntano sulla silenziosità, utilizzano piuttosto alimentatori con ventole aspiranti, oppure non utilizzano alcuna ventola, ma si collocano sopra la CPU, dove si avvalgono della ventilazione della CPU stessa.

Figura 9.9. Ventola posta sul pannello posteriore dell'alimentatore, vista dall'interno. Questa è la situazione comune di una ventola che spinge l'aria verso l'esterno.

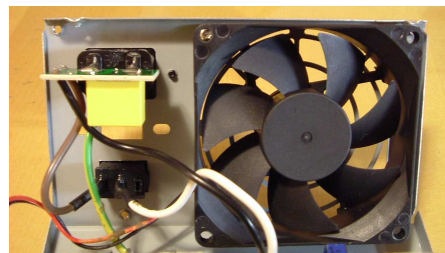
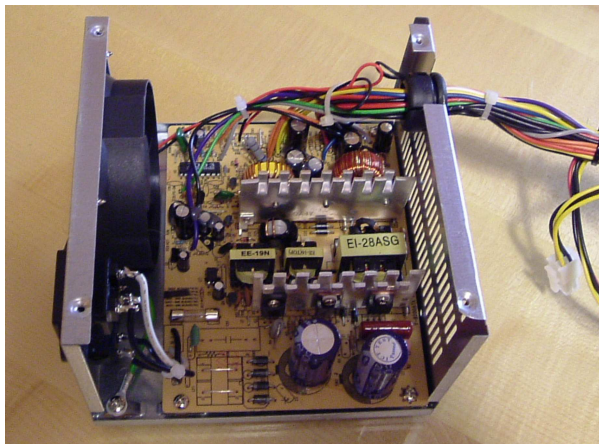
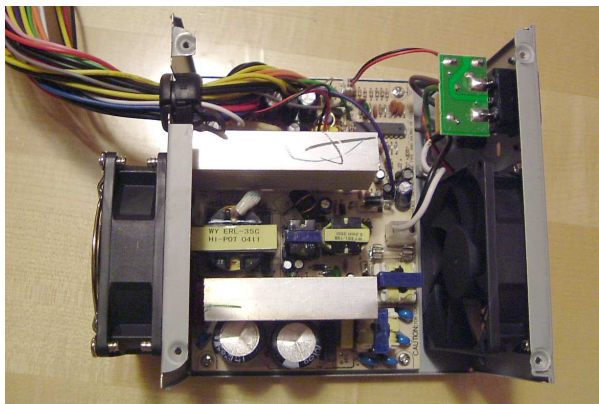


Figura 9.10. Interno di un alimentatore in grado di erogare una potenza molto bassa. Si vedono delle alette di raffreddamento di piccole dimensioni e una sola ventola (sulla sinistra).



Osservando il sistema di dissipazione di un alimentatore, si può intuire la potenza massima erogabile: un circuito con componenti di piccole dimensioni e molto spazio libero, deve generare poco calore, di conseguenza non può erogare una grande potenza; al contrario, un circuito con componenti più grandi e con un sistema di dissipazione più appariscente, dovrebbe sostenere carichi più importanti.

Figura 9.11. Interno di un alimentatore in grado di erogare una potenza relativamente alta. Si vedono delle alette di raffreddamento di dimensioni più grandi e due ventole cooperanti.



#### 9.1.4 Pulizia

L'alimentatore contiene componenti che richiedono la dissipazione del calore, attraverso una ventilazione forzata che ha lo svantaggio di depositare velocemente uno strato di sporco sui componenti. Lo sporco riduce la capacità di dissipazione termica dei componenti, creando anche la possibilità di formare dei cortocircuiti.

Dopo un certo numero di ore di esercizio, un alimentatore dovrebbe essere pulito per garantirne il funzionamento nel tempo successivo, ma la pulizia migliore è quella fatta con l'acqua. Infatti, di norma un alimentatore sopporta bene un lavaggio in lavastoviglie (purché sia stato prima aperto), a una temperatura di 50 gradi centigradi (con tutto quello che serve per la lavastoviglie, come se si trattasse di un lavaggio normale); in alternativa, si può ripiegare per un lavaggio sotto una doccia calda, con sola acqua. Dopo il lavaggio e dopo qualche giorno di attesa per l'asciugatura, l'alimentatore può essere rimesso in funzione. Si veda anche la sezione 9.11.1 a proposito del lavaggio degli alimentatori e di altri componenti di un elaboratore.

#### 9.1.5 Alimentazione e affidabilità dell'elaboratore

Il buon funzionamento dell'alimentatore può condizionare l'affidabilità dell'elaboratore. Infatti, quando un alimentatore non è in grado di erogare tutta la corrente richiesta, può succedere che, di conseguenza, abbassi la tensione di alimentazione. Quando ciò accade, un buon alimentatore deve essere in grado di accorgersene e inviare un segnale di cattiva alimentazione alla scheda madre (in pratica, il segnale noto come *power good*, o *power ok*, deve venire a mancare). Se però l'alimentatore non fa questo in modo corretto, l'elaboratore può anche continuare a funzionare, ma in tal caso non è detto che lo faccia senza conseguenze negative.

Un sintomo di cattiva alimentazione elettrica, può essere dato da un'alterazione casuale dei dati. Per esempio, copiando un file, la copia potrebbe essere differente dall'originale. Sintomi più appariscenti sono dei **malfunzionamenti delle unità a disco**, che spesso si manifestano facendo sentire un **movimento brusco delle testine prima di un blocco del sistema**.

Tutti gli alimentatori sono fatti per sostenere, per periodi brevi, un assorbimento maggiore di quello «normale» per i quali sono stati progettati. Di conseguenza, i sintomi di un'alimentazione insufficiente non si presentano subito, ma durante il funzionamento, ed è questo particolare che rende difficile determinarne la causa. In pratica, durante il funzionamento gli alimentatori si scaldano e se la dissipazione del calore è insufficiente, si presentano i problemi in questione.

Purtroppo, le etichette che appaiono sugli alimentatori non sono sempre veritiere; al contrario, spesso si dichiarano potenze doppie rispetto alla realtà. La situazione più comune in cui i dati riportati dalle etichette non sono attendibili si manifesta con gli alimentatori incorporati in un contenitore da elaboratore (*case*), venduto per gli assemblaggi personalizzati; meglio ancora se il contenitore è di piccole dimensioni, così da rendere improbabile l'inserimento di tanti componenti.

Quando la potenza dell'alimentatore è importante, è altrettanto importante poter capire se l'etichetta riporta valori verosimili o meno. La prima cosa che si può osservare è il peso: un alimentatore leggero non riesce a erogare una potenza elevata. Un'altra cosa da osservare è il sistema di dissipazione: per erogare una potenza elevata i componenti di regolazione devono essere raffreddati da alette di alluminio profilato, con una superficie molto grande; inoltre queste alette devono essere sottoposte a una ventilazione significativa, quindi spesso servono due ventole, una di fronte all'altra. Se si può aprire l'alimentatore, si può osservare la dimensione degli altri componenti: se nell'insieme i componenti sono abbastanza piccoli da lasciare molto spazio libero, l'alimentatore può erogare solo potenze minime; al contrario, un alimentatore che deve erogare potenze elevate, richiede componenti più grandi (diodi, condensatori, trasformatori, ecc.).

#### 9.1.6 Connettori comuni di alimentazione

Per il collegamento di unità separate dalla scheda madre, si usano normalmente dei connettori a quattro poli (femmine volanti), che forniscono le tensioni di +5 V e +12 V, oltre alla massa (0 V), come si vede nelle figure successive.

Figura 9.12. Connettore di alimentazione usato comunemente per le unità PATA.

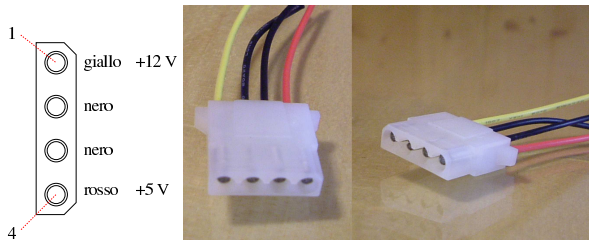
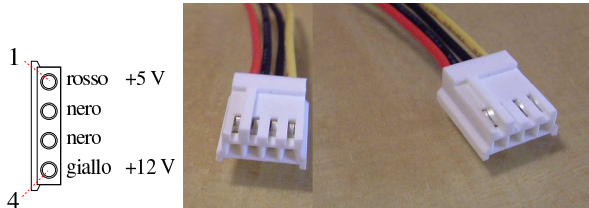


Figura 9.13. Connettore di alimentazione usato comunemente per le vecchie unità a dischetti da 9 cm.



## 9.2 Alimentatori ATX

L'alimentatore «ATX» rappresenta la generazione successiva a quella dell'alimentatore «AT», distinguendosi principalmente da questo, per il fatto di avere una gestione dell'accensione e dello spegnimento tramite un controllo remoto (il filo verde).

Quando si apre un alimentatore, si deve ricordare che al suo interno viene elaborata l'energia proveniente dalla rete elettrica comune, con la quale è possibile rimanere folgorati, se non si agisce con la competenza e la prudenza adeguate alla circostanza.

### 9.2.1 Collegamento principale con la scheda madre: 20 poli

La figura successiva mostra schematicamente il connettore principale tipico, che serve ad alimentare la scheda madre. Il terminale numero otto, collegato con un filo di colore grigio, fornisce una tensione di +5 V, quando l'alimentatore è acceso e sta funzionando regolarmente. Il terminale numero nove, collegato con un filo di colore viola, fornisce una tensione di +5 V, sia durante il funzionamento normale dell'alimentatore, sia quando questo figura «spento», ma ancora alimentato dalla rete elettrica esterna. Il terminale numero 14, collegato con un filo di colore verde, consente di attivare o disattivare l'alimentatore; per l'attivazione, deve essere collegato alla massa, mentre è sufficiente tenerlo isolato per disattivare l'alimentatore. Il filo marrone, si inserisce, assieme a un filo di colore arancio, nel terminale numero 11; il suo scopo è quello di consentire all'alimentatore di sondare la presenza dell'alimentazione a 3,3 V, ma alcuni alimentatori non hanno proprio tale filo marrone.

Figura 9.14. Schema del connettore principale per l'alimentazione della scheda madre.

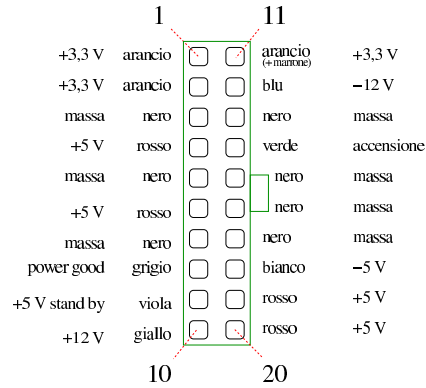
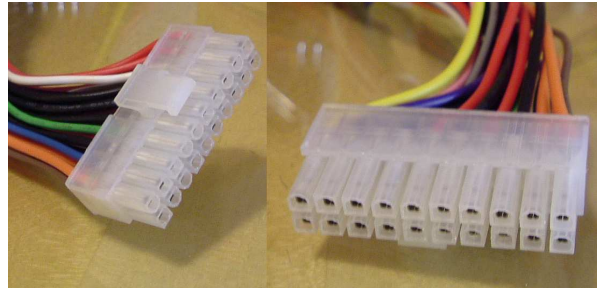


Figura 9.15. Foto del connettore principale per l'alimentazione della scheda madre.



### 9.2.2 Collegamento principale con la scheda madre: 24 poli

Esistono alimentatori ATX con un connettore per il collegamento della scheda madre, a 24 poli. In questo caso, oltre all'aggiunta di quattro poli addizionali, può venire a mancare la linea da -5 V (filo bianco).

Figura 9.16. Schema del connettore principale per l'alimentazione della scheda madre, a 24 poli.

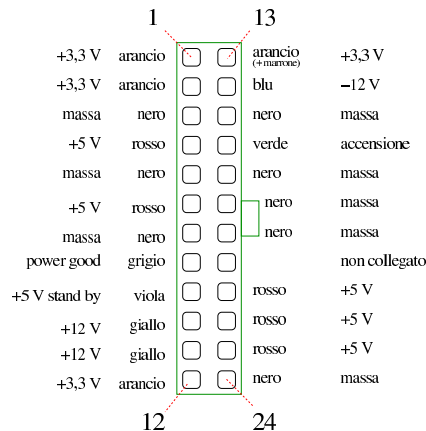
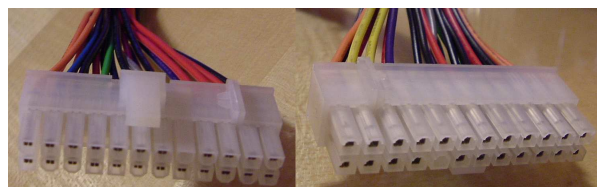


Figura 9.17. Foto del connettore principale per l'alimentazione della scheda madre, a 24 poli.

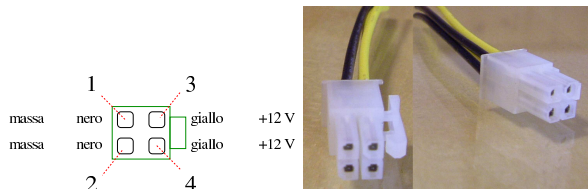


Osservando la foto, si vede che il connettore può essere formato dall'unione di due connettori, che eventualmente possono anche essere separati. Infatti, la porzione più grande del connettore (quella con 20 poli), è conforme allo standard più vecchio, a parte il fatto che manca il filo bianco. Tuttavia, nella maggior parte dei casi, le schede madri che usano un connettore a soli 20 poli, non richiedono la tensione di -5 V, che dovrebbe essere fornita da quel filo mancante.

### 9.2.3 Collegamento ausiliario con la scheda madre, +12 V

« Alcune schede madri richiedono una sorgente di alimentazione addizionale, attraverso un piccolo connettore a quattro poli, come si vede nella figura successiva, che fornisce +12 V.

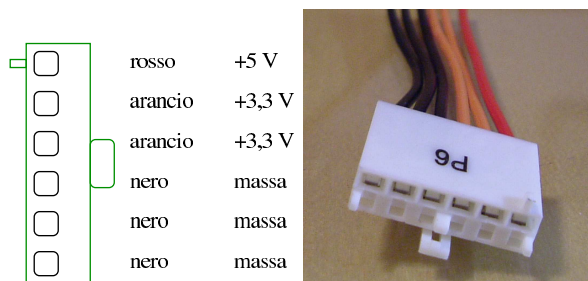
Figura 9.18. Connettore ausiliario per l'alimentazione della scheda madre con +12 V.



### 9.2.4 Collegamento ausiliario con la scheda madre «P6», +3,3 V, +5 V,

« Alcune schede madri ATX richiedono una fonte di alimentazione ausiliaria per le tensioni più basse. Per questo si utilizza solitamente un connettore noto con la sigla «P6», molto simile a quelli usati negli alimentatori AT (ma nel caso degli alimentatori AT, si tratta dei connettori «P8» e «P9»).

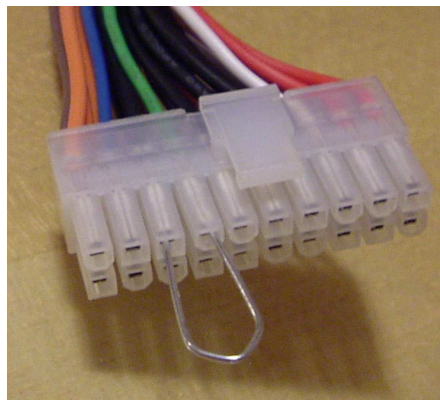
Figura 9.19. Connettore ausiliario per l'alimentazione della scheda madre con +3,3 V e +5 V.



### 9.2.5 Controllo del funzionamento

« Quando si accende un alimentatore ATX senza che sia stato prima collegato alla scheda madre, questo rimane in attesa, senza erogare energia dalle linee principali (lasciando ferma la stessa ventola). Per mettere in funzione l'alimentatore, anche senza collegarlo alla scheda madre, occorre cortocircuitare il terminale del filo verde (a seconda della dimensione del connettore, può trattarsi del terminale numero 14, oppure del numero 16) con la massa. Per esempio, nella figura successiva vengono collegati i terminali 13 e 14 di un connettore a 20 poli, con un pezzo di fermaglio per lettera, proprio per mettere subito in funzione l'alimentatore quando questo riceve energia dalla rete esterna.

Figura 9.20. Accensione di fortuna di un alimentatore ATX.



In questo modo, è possibile fare un controllo di massima del funzionamento di un alimentatore, semplicemente osservando se si avvia la ventola di raffreddamento.

### 9.2.6 Utilizzo di un alimentatore ATX per scopi vari

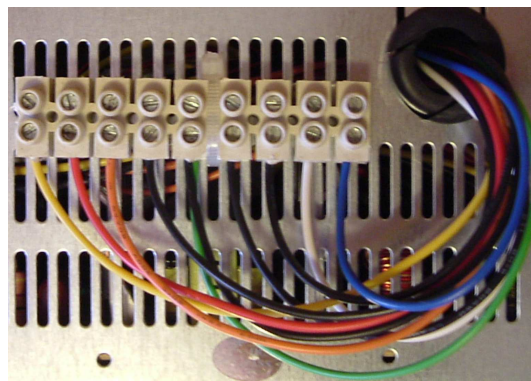
« Un alimentatore ATX che non serve più, potrebbe essere riutilizzato per compiti diversi, fuori dal contenitore di un elaboratore. Per trasformare un alimentatore in questo modo, occorre tenere presente almeno due cose:

- se è presente il filo marrone, questo deve essere collegato a un filo arancione, altrimenti l'alimentatore non si metterebbe in funzione;
- è necessario utilizzare un piccolo interruttore per collegare il filo verde con la massa, in modo da poter mettere in funzione l'alimentatore o rimetterlo a riposo.

Dell'alimentatore che si ottiene, oltre alla massa, si possono utilizzare le linee normali da +3,3 V, +5 V, +12 V, -12 V e -5 V. Pertanto, il filo grigio e il filo viola, non servono.

Eventualmente, il filo grigio può essere usato per applicarvi un diodo LED, con una resistenza appropriata in serie (500 ohm, 0,25 W), in modo da sapere che l'alimentatore è acceso, anche se eventualmente non risulta ancora attivo.

Figura 9.21. Trasformazione molto semplice di un alimentatore per usi generali. Si può osservare che il filo verde è stato collegato in modo stabile a un filo nero, per attivare l'alimentatore appena gli viene fornita energia dalla rete elettrica esterna.



## 9.3 (P)ATA: AT attachment

« ATA, ovvero *AT attachment*, è stato originariamente un bus a 16 bit, nato per il collegamento di dischi fissi. Questo tipo di bus è conosciuto anche con altri nomi, che però non sono standard; in particolare «IDE», ovvero *Integrated drive electronics*. Il bus ATA (parallelo) è stato definito quando la tecnologia ha iniziato a integrare l'elettronica dell'interfaccia nel disco fisso stesso, da cui viene l'acronimo

IDE. Il primo elaboratore che abbia incorporato un bus ATA è stato proprio un IBM PC/AT, da cui il nome che giustifica l'acronimo ATA. Successivamente, nel 2001, con l'introduzione di un bus seriale, derivato dallo standard ATA, per distinguere i due tipi si sono introdotte le sigle «PATA», per «paralel ATA», e «SATA» per «serial ATA». Questo capitolo tratta del bus ATA parallelo.

Il bus PATA ha avuto subito un grande successo ed è diventato uno standard molto importante. L'evoluzione dello standard ha portato a espandere le possibilità del bus, che originariamente permetteva solo il collegamento di dischi fissi, consentendo l'uso di altri tipi di dischi e non solo questo, attraverso le estensioni ATAPI (*ATA packet interface*).

### 9.3.1 Caratteristiche generali del sistema PATA

« Un bus PATA tradizionale (ma in questo contesto si potrebbe usare il termine «canale») consente di collegare uno o due unità, attraverso una piattina con tre connettori da 40 poli (un connettore si collega all'unità di controllo PATA, mentre gli altri due servono per i componenti), come si vede nella figura 9.22. Un'estremità della piattina è colorata in modo diverso a indicare il lato in cui si trova il terminale numero uno.

Figura 9.22. Connettore a 40 poli e pezzo di piattina a 40 fili per una connessione PATA normale. In questo caso, il riferimento al terminale numero uno si trova sul lato destro.

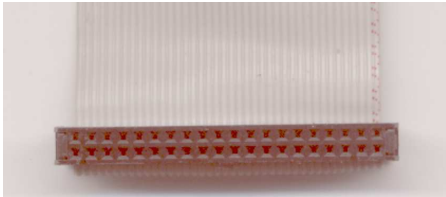
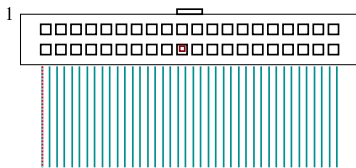


Figura 9.23. Disegno di un connettore a 40 poli con piattina completo. La tacca superiore e il foro otturato non sono sempre presenti.



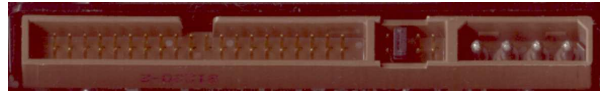
I due dischi, o comunque i due componenti che si collegano all'unità di controllo devono essere distinti, essendo uno *master* e l'altro *slave*. Questa distinzione è indispensabile e in condizioni normali è necessario dichiarare questo ruolo dei componenti utilizzando dei ponticelli o dei micro interruttori. In questa situazione, utilizzando piattine normali diventa indifferente il collegamento dei componenti su questo o su quel connettore. In particolare, a differenza del bus SCSI non è richiesta alcuna terminazione.

Figura 9.24. Un'etichetta che appare su un disco fisso per guidare la configurazione in qualità di *master* o di *slave*, oppure lasciando che sia il cavo a definire la posizione.



Eventualmente esistono dei cavi PATA speciali che consentono di distinguere l'unità *master* da quella *slave* in base alla scelta del connettore utilizzato. Ma questo fatto richiede comunque la selezione nelle unità stesse della modalità *cable select*, sempre attraverso ponticelli o micro interruttori.

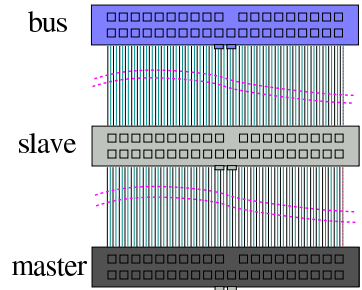
Figura 9.25. Lato posteriore di un disco fisso in cui si vede il connettore da 40 poli (il terminale numero uno si trova nel lato destro); una fila di ponticelli per la selezione di *master*, *slave* o *cable select*; il connettore di alimentazione. Si osservi l'assenza di un piedino nella parte centrale del connettore.



Lo standard PATA, nella sua evoluzione, ha cercato di mantenere la compatibilità con il passato, per garantire che un'unità di controllo aggiornata sia in grado di funzionare anche con componenti PATA più vecchi e viceversa. In particolare, questo fatto serve a garantire che l'accoppiamento di due componenti eterogenei, collegati allo stesso canale, non comporti complicazioni. In pratica, nonostante le buone intenzioni, possono anche sorgere dei problemi di compatibilità; inoltre, l'accoppiamento di due componenti eterogenei implica solitamente l'utilizzo del protocollo di comunicazione migliore che entrambi sono in grado di gestire, costringendo uno dei due a lavorare con prestazioni inferiori rispetto alle proprie possibilità.

Il protocollo PATA si è sviluppato nel tempo utilizzando tecniche diverse per la comunicazione dei dati. In particolare, le versioni più recenti che consentono velocità di trasferimento dati più elevate richiedono una piattina speciale, in cui ogni collegamento dei 40 poli esistenti viene separato da un filo di massa, allo scopo di ridurre le interferenze reciproche. Questo tipo di piattina a 80 fili ha anche la caratteristica di selezionare automaticamente il componente *master* e *slave*, pertanto questi devono essere configurati in modalità *cable select*.

Figura 9.26. Disegno di una piattina completa a 80 fili.



In particolare, il connettore blu, o azzurro, va collegato all'unità di controllo; il connettore nero, all'estremità opposta, va collegato al disco *master*; il connettore grigio va collegato al disco *slave*.

Figura 9.27. Connettori per due canali PATA su una scheda madre. Il terminale numero uno si trova sul lato destro. Si osservi l'assenza di un piedino nella parte centrale del connettore.



### 9.3.2 Geometria dei dischi

« I dischi (fissi o rimovibili che sia) hanno una geometria, definita in base a tre coordinate: cilindro, testina e settore. Nella prima edizione dello standard PATA è necessario indicare i valori massimi di queste coordinate nella configurazione del firmware. A partire da ATA-2 si è introdotta la capacità di identificare automaticamente la geometria dei dischi e di poterla ridefinire in base a ciò che è stato chiamato LBA, ovvero *Logical block addressing*.

Attraverso questo meccanismo, se il firmware è predisposto per farlo, è possibile superare i limiti alla quantità di cilindri e di testine imposti dalla struttura tradizionale del firmware stesso. Tuttavia, perché

sia possibile l'utilizzo di questa possibilità, è necessario anche che il sistema operativo e il sistema di avvio siano in grado di gestirla.

### 9.3.3 Caratteristiche del protocollo PATA e configurazione del firmware

Attualmente, nell'ambito del protocollo PATA si distinguono tre modalità di comunicazione: PIO (*Programmed I/O*), DMA (che comunque si articola in due sotto categorie) e Ultra DMA. Ognuna di queste modalità ha poi dei livelli differenti, costituiti da un numero intero, dove lo zero rappresenta quello minimo con le prestazioni peggiori, mentre i valori successivi indicano una progressione.

- Il protocollo PIO è il primo a essere stato realizzato e ha la caratteristica di impegnare direttamente l'unità centrale (la CPU) per il suo funzionamento. La tabella 9.28 elenca le modalità PIO, ovvero i livelli di questo protocollo, in cui si può vedere che il numero cinque non appartiene allo standard codificato, ma probabilmente viene indicato da qualche costruttore, facendo riferimento a una velocità di trasferimento di 22 Mbyte/s.

Tabella 9.28. Modalità PIO.

Modalità PIO	Velocità massima in Mbyte/s (10 <sup>6</sup> byte/s)	Note
0	3,3	
1	5,2	
2	8,3	
3	11,1	
4	16,7	
5	22,2	Non standard.

Di solito, la gestione del protocollo PIO può essere controllata dal firmware (il BIOS), con cui si può imporre una modalità particolare, oppure si può escludere del tutto, in favore di altri protocolli. Tuttavia, questo tipo di intervento si può richiedere solo se si incontrano dei problemi, in cui i componenti installati non sono in grado di determinare automaticamente il metodo migliore di funzionamento. Il programma di accesso alla configurazione del firmware potrebbe presentare le voci seguenti, dove in questo caso si lascia una gestione automatica del protocollo:

Primary Master	PIO: Auto
Primary Slave	PIO: Auto
Secondary Master	PIO: Auto
Secondary Slave	PIO: Auto

- Il protocollo DMA utilizza in pratica un accesso diretto alla memoria (*Direct memory access*). Inizialmente, il trasferimento dei dati avveniva a pacchetti di 16 bit (*word*), ovvero quanto l'ampiezza del canale PATA. Successivamente si è introdotta la possibilità di trasmettere sequenze più lunghe. Nel primo caso si parla di *single word*, mentre nel secondo si fa riferimento al termine *multi word*. Queste due varianti del protocollo DMA hanno livelli di identificazione distinti, come se fossero due protocolli diversi; tuttavia, attualmente il trasferimento a blocchi singoli di 16 bit non si usa più, per cui il protocollo DMA è inteso essere implicitamente di tipo *multi word*.

Tabella 9.30. Modalità DMA.

Modalità <i>single word</i>	Modalità <i>multi word</i>	Velocità massima in Mbyte/s (10 <sup>6</sup> byte/s)
0		2,1
1		4,2
2		8,3
	0	4,2
	1	13,3
	2	16,7

La comunicazione diretta con la memoria può avvenire attraverso un sistema di controllo integrato nella scheda madre, oppure direttamente attraverso l'unità di controllo PATA. In pratica, il secondo caso richiede che il bus in cui si inserisce a sua volta l'unità di controllo PATA consenta la tecnica conosciuta come *bus mastering*, attraverso cui un componente del bus può prendere il controllo.

Il primo tipo di sistema, in cui si passa per l'intermediazione della scheda madre, viene indicato come *third party DMA*, mentre l'accesso diretto al DMA si indica come *first party DMA*. L'utilizzo dell'accesso DMA senza intermediazioni consente di gestire velocità più elevate e diventa necessario per mettere in pratica gli ultimi livelli di questo protocollo.

- Il protocollo Ultra DMA, o solo UDMA, è un'evoluzione del DMA, in cui vengono usate delle tecniche diverse per aumentare la frequenza di trasmissione effettiva dei dati, introducendo anche un sistema di controllo CRC per la verifica che questi siano stati ricevuti correttamente, in modo da poter regolare la velocità effettiva in base alle caratteristiche reali del mezzo.

Tabella 9.31. Modalità UDMA.

Modalità UDMA	Velocità massima in Mbyte/s (10 <sup>6</sup> byte/s)
0	16,7
1	25,0
2	33,3
3	44,4
4	66,7
5	100,0
6	133,3

Le tecniche di trasmissione introdotte dal protocollo UDMA richiedono l'uso di una piattina speciale, a 80 fili, quando si supera la velocità di 33,3 Mbyte/s. Come già accennato, ognuno dei fili corrispondenti ai 40 poli viene alternato da un filo di massa che serve a ridurre le interferenze; inoltre, questo tipo di cavo implica la selezione del tipo *cable select* nei componenti che si installano. Pertanto, i vari connettori sono colorati in modo differente per evitare confusione.

Anche se non dovrebbe essere necessario, è probabile che il firmware consenta di stabilire espressamente la modalità UDMA che deve essere utilizzata. Nell'esempio seguente si vede la richiesta di lasciare che il protocollo venga definito in modo automatico, in base a quanto riportato dai componenti e dalla verifica delle comunicazioni:

Primary Master	UDMA: Auto
Primary Slave	UDMA: Auto
Secondary Master	UDMA: Auto
Secondary Slave	UDMA: Auto

Alcune unità di controllo, in base ai protocolli PATA, consentono di inviare dei blocchi di dati più consistenti attraverso il bus, per ogni singolo ciclo di interruzione, secondo una tecnica definita *block mode*. In generale questo fatto non crea complicazioni; tuttavia viene riportato solo per chiarire il senso di ciò quando una voce del genere appare nella configurazione del firmware:

IDE HDD Block Mode:	Enabled
---------------------	---------

Di solito sono disponibili due bus PATA, ovvero due canali per ogni unità di controllo, specialmente se questa è integrata nella scheda madre. Ogni canale, o bus, utilizza delle risorse, in particolare un livello IRQ; pertanto, se uno dei due canali non viene utilizzato, conviene disabilitarne il funzionamento attraverso il firmware:

OnChip IDE Channel0:	Enabled
OnChip IDE Channel1:	Disabled

Tabella 9.35. Risorse utilizzate da un'unità di controllo PATA comune. Si tenga presente che un'unità di controllo collegata a sua volta su un bus PCI, utilizza anche altre aree di indirizzi di I/O.

Canale	IRQ	I/O
0	14	da 01F0 <sub>16</sub> a 01F7 <sub>16</sub> e da 03F6 <sub>16</sub> a 03F6 <sub>16</sub>
1	15	da 0170 <sub>16</sub> a 0177 <sub>16</sub> e da 0376 <sub>16</sub> a 0376 <sub>16</sub>

### 9.3.4 Standard PATA e gergo comune

Lo standard ufficiale per quanto riguarda il bus PATA è definito da ANSI e la tabella 9.36 ne riassume le caratteristiche salienti.



Tabella 9.36. Standard PATA secondo ANSI.

Sigla	Denominazione	PIO	DMA single word	DMA multi word	UDMA
ATA-1	AT Attachment Interface for Disk Drives	0, 1, 2	0, 1, 2	0	--
ATA-2	AT Attachment Interface with Extensions	0, 1, 2, 3, 4	0, 1, 2	0, 1, 2	--
ATA-3	AT Attachment 3 Interface	0, 1, 2, 3, 4	--	0, 1, 2	--
ATA/ATAPI-4	AT Attachment with Packet Interface Extensions	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2
ATA/ATAPI-5	AT Attachment with Packet Interface - 5	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4
ATA/ATAPI-6	AT Attachment with Packet Interface - 6	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4, 5
ATA/ATAPI-7	AT Attachment with Packet Interface - 7	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4, 5, 6

Purtroppo si sono diffuse una grande quantità di definizioni non standard, il cui significato diventa a volte ambiguo. Per essere certi delle possibilità di un bus PATA o di un componente PATA, occorre verificare le caratteristiche che compongono lo standard, ovvero le modalità dei protocolli. Il problema legato alla terminologia dipende molto dal fatto che i prodotti che si trovano in commercio sono spesso più avanzati degli standard già definiti, anche se è molto probabile che vengano poi incorporati negli standard ufficiali successivi. Pertanto, questo favorisce la diffusione di appellativi provvisori e altisonanti. Quello che segue è un elenco molto approssimativo dei nomi attribuiti nel gergo a vari aspetti dello standard PATA.

- IDE, EIDE

La sigla IDE sta per *Integrated drive electronics*, ovvero quel tipo di disco che integra l'elettronica di controllo. Pur non essendo codificata negli standard, questa sigla viene usata comunemente, al posto di PATA.

La sigla EIDE è stata conosciuta da un'azienda particolare, con il significato di *Enhanced IDE*, a sottolineare l'utilizzo di tecniche nuove, non ancora codificate fino a quel momento, ma senza fare riferimento a un'epoca o a delle caratteristiche particolari, rendendo così la sigla altrettanto vaga quanto IDE.

- Fast ATA

La definizione Fast ATA è stata usata in vari contesti per identificare qualcosa di più veloce del normale, ma senza un legame particolare con lo standard ufficiale. In particolare, quando si legge qualcosa come «Fast ATA-*n*», il numero non corrisponde a quello usato dallo standard ANSI. Per esempio, Fast ATA-4 potrebbe voler solo dire che offre il protocollo UDMA 4, mentre lo standard ufficiale corrispondente è ATA/ATAPI-5.

- Ultra ATA, Ultra ATA/33, ATA/33, Ultra ATA/66, ATA/66,...

Questo genere di definizioni sono solitamente ottenute sommando assieme porzioni delle caratteristiche dei protocolli gestibili. Il termine «ultra» si riferisce normalmente al protocollo UDMA, mentre il numero indica presumibilmente la velocità di trasferimento dei dati. Pertanto, «Ultra ATA/33» o solo «ATA/33» si riferisce presumibilmente a un protocollo UDMA che consente un trasferimento a 33,3 Mbyte/s.

### 9.3.5 Gestione da parte di un kernel Linux delle unità PATA

Nei sistemi GNU/Linux i file di dispositivo usati per accedere alle unità ATA hanno un nome che rispetta il modello:

```
/dev/hdx[n]
```

In pratica, *x* può essere una lettera minuscola, da 'a' a 't', che rappresenta un disco PATA completo; per esempio, '/dev/hdd' indica il disco *slave* del secondo canale PATA disponibile. Se al nome segue un numero, questo indica l'*n*-esima partizione, a partire da uno. Pertanto, '/dev/hdd1' è la prima partizione della quarta unità PATA.

La gestione dei componenti PATA da parte del kernel Linux viene definita nella sezione *Device drivers, serial ATA and parallel ATA drivers* (8.3.8.5). La gestione accurata delle opzioni di configurazione di quella sezione consente di ottenere le prestazioni massime disponibili dal bus PATA di cui si dispone.

### 9.3.6 Informazioni da un kernel Linux

Le informazioni che si possono ottenere da un kernel Linux consentono di verificare la configurazione effettiva del firmware e il riconoscimento dell'unità di controllo PATA esistente. La prima verifica va fatta sui messaggi iniziali di avvio del kernel, i quali si possono rileggere con il comando '**dmesg**':

```
$ dmesg | less [Invio]
```

Nella prima parte si può osservare la gestione generale del bus PATA:

```
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 33MHz system bus speed for PIO modes;
      override with idebus=xx
```

Nella seconda parte si può osservare l'identificazione delle particolarità dell'unità di controllo PATA, ammesso che sia in grado di farlo:

```
VP_IDE: IDE controller on PCI bus 00 dev 39
VP_IDE: chipset revision 16
VP_IDE: not 100% native mode: will probe irqs later
ide: Assuming 33MHz system bus speed for PIO modes; override
      with idebus=xx
VP_IDE: VIA vt82c596b (rev 23) IDE UDMA66 controller on
      pci00:07.1
      ide0: BM-DMA at 0xd000-0xd007, BIOS settings:
            hda:DMA, hdb:DMA
      ide1: BM-DMA at 0xd008-0xd00f, BIOS settings:
            hdc:pio, hdd:pio
```

Quindi si mostrano le caratteristiche dei componenti installati, che in questo caso sono due dischi fissi:

```
hda: IBM-DTLA-307030, ATA DISK drive
hdb: QUANTUM FIREBALLct10 20, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hda: 60036480 sectors (30739 MB) w/1916KiB Cache,
      CHS=3737/255/63, UDMA(66)
hdb: 39876480 sectors (20417 MB) w/418KiB Cache,
      CHS=2482/255/63, (U)DMA
```

Infine si vedono anche le partizioni di questi:

```
Partition check:
hda: hda1 hda2 hda4 < hda5 hda6 hda7 >
hdb: hdb1 hdb2
```

Se l'unità di controllo PATA è collegata a sua volta su un bus PCI, cosa che ormai rappresenta la situazione normale, se ne possono trovare le tracce nella directory virtuale '/proc/', attraverso '**lspci**' (l'esempio si riferisce a un elaboratore differente, ma sulle stesse coordinate PCI):

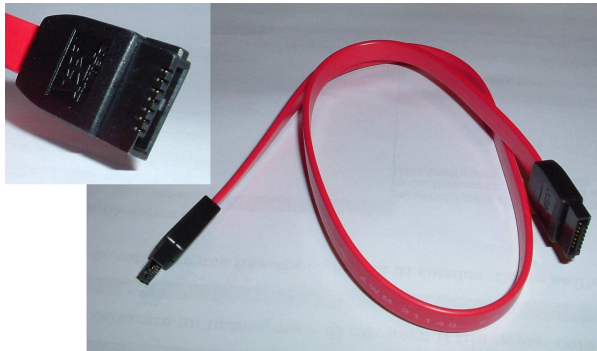
```
$ lspci -v | less [Invio]
```

```
00:07.1 IDE interface: VIA Technologies, Inc.
↳VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus
↳Master IDE (rev 06) (prog-if 8a [Master SecP PriP])
      Flags: bus master, stepping, medium devsel, latency 32
      I/O ports at fff0 [size=16]
```

### 9.4 SATA: Serial ATA

Terminato lo sviluppo del bus ATA tradizionale, si introduce un bus diverso meccanicamente, ma che conserva qualche compatibilità a livello di software, denominato SATA, ovvero *Serial ATA*. Da un punto di vista fisico, questo nuovo bus si presenta con un collegamento differente, attraverso un cavo che non è più realizzato in forma di piatina.

Figura 9.42. Un cavo SATA, con il dettaglio di un connettore.



#### 9.4.1 Adattatori

Durante la fase di transizione dal bus ATA tradizionale (ovvero PATA) a quello SATA, sono stati disponibili dei piccoli convertitori in grado di adattare dischi fissi SATA su bus PATA e dischi PATA su bus SATA. Si tratta però di rimedi di emergenza, il cui utilizzo è decisamente sconsigliabile.

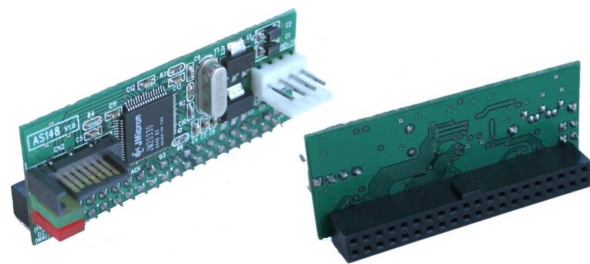
Figura 9.43. Adattatore da bus PATA a disco SATA. Si può osservare che l'adattatore richiede un'alimentazione supplementare, con un cavetto di quelli usati per le unità a dischetti tradizionali. Questa immagine è stata tratta da <http://www.extrememhz.com/idetosata-p1.shtml>.



Figura 9.44. Adattatore da bus PATA a disco SATA. A differenza della figura precedente, questo si applica al disco fisso SATA.



Figura 9.45. Adattatore da bus SATA a disco PATA. Questo adattatore si applica al disco PATA.



#### 9.4.2 Gestione da parte del kernel Linux

Il kernel Linux tratta le unità SATA come se fossero unità SCSI, pertanto i file di dispositivo a cui si fa riferimento sono `'/dev/sdx'`, dove `x` è una lettera alfabetica minuscola.

#### 9.4.3 Informazioni da un kernel Linux

Le informazioni che si possono ottenere da un kernel Linux consentono di verificare la configurazione effettiva del firmware e il riconoscimento dell'unità di controllo SATA esistente. La prima verifica va fatta sui messaggi iniziali di avvio del kernel che si possono rileggere con il comando `'dmesg'`:

```
$ dmesg | less [Invio]
```

Nella prima parte si può osservare la gestione generale del bus SATA e l'identificazione delle particolarità dell'unità di controllo SATA:

```
libata version 1.20 loaded.
sata_via 0000:00:0f.0: version 1.1
sata_via 0000:00:0f.0: routed to hard irq line 0
ata1: SATA max UDMA/133 cmd 0xEFF0 ctl 0xEFE6 bmdma 0xEF90
      irq 16
ata2: SATA max UDMA/133 cmd 0xEFA8 ctl 0xEFE2 bmdma 0xEF98
      irq 16
ata1: dev 0 cfg 49:2f00 82:7c6b 83:7f09 84:4673 85:7c69
      86:3e21 87:4663 88:407f
ata1: dev 0 ATA-7, max UDMA/133, 398297088 sectors: LBA48
ata1: dev 0 configured for UDMA/133
ata2: dev 0 cfg 49:2f00 82:7c6b 83:7f09 84:4673 85:7c69
      86:3e21 87:4663 88:407f
ata2: dev 0 ATA-7, max UDMA/133, 398297088 sectors: LBA48
ata2: dev 0 configured for UDMA/133
```

Nella seconda parte si può osservare l'associazione dei dischi a unità SCSI:

```
scsi0 : sata_via
scsil : sata_via
Vendor: ATA Model: Maxtor 6L200S0 Rev: BACE
Type: Direct-Access ANSI SCSI revision: 05
Vendor: ATA Model: Maxtor 6L200S0 Rev: BACE
Type: Direct-Access ANSI SCSI revision: 05
SCSI device sda: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sda: drive cache: write back
SCSI device sda: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sda: drive cache: write back
      sda: sda1 sda2 sda4
sd 0:0:0:0: Attached scsi disk sda
SCSI device sdb: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sdb: drive cache: write back
SCSI device sdb: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sdb: drive cache: write back
      sdb: sdb1 sdb2 sdb4
sd 1:0:0:0: Attached scsi disk sdb
sd 0:0:0:0: Attached scsi generic sg0 type 0
sd 1:0:0:0: Attached scsi generic sg1 type 0
```

Se l'unità di controllo SATA è collegata a sua volta su un bus PCI, se ne possono trovare le tracce nella directory virtuale `'/proc/'`, attraverso `'lspci'` (l'esempio si riferisce a un elaboratore differente, ma sulle stesse coordinate PCI):

```
$ lspci -v | less [Invio]
```

```
0000:00:0f.0 RAID bus controller: VIA Technologies, Inc. ↵
↳VIA VT6420 SATA RAID Controller (rev 80)
  Subsystem: ASUSTeK Computer Inc. A7V600/K8V ↵
↳Deluxe/K8V-X motherboard
  Flags: bus master, medium devsel, latency 64, IRQ 16
  I/O ports at eff0 [size=8]
  I/O ports at efe4 [size=4]
  I/O ports at efa8 [size=8]
  I/O ports at efe0 [size=4]
  I/O ports at ef90 [size=16]
  I/O ports at e800 [size=256]
  Capabilities: [c0] Power Management version 2
```

#### 9.4.4 Sistemi RAID SATA

Generalmente, le unità di controllo SATA sono in grado di gestire alcuni livelli RAID con l'unione di più dischi SATA, in modo indipendente dal sistema operativo. Tuttavia, con i sistemi GNU/Linux risulta più conveniente una gestione RAID software, che può essere controllata completamente durante il funzionamento del sistema operativo. Si veda eventualmente la sezione 19.10 al riguardo.

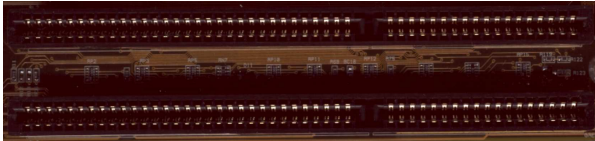
## 9.5 Da ISA a PCI

Da quando esistono gli elaboratori personali, il bus «locale», ovvero quello incorporato nella scheda madre, ha subito una continua evoluzione. In questo capitolo si riassumono le caratteristiche dei bus locali più importanti nell'architettura x86, anche se in molti casi può trattarsi di componenti superati.

### 9.5.1 ISA

La sigla ISA sta per *Industry standard architecture* e rappresenta il primo tipo di bus negli elaboratori personali x86-16. Inizialmente si è trattato di un bus a 8 bit, che successivamente è stato esteso a 16 bit.

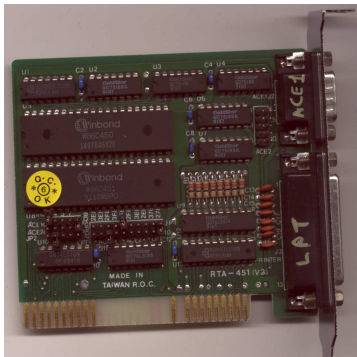
Figura 9.49. Due alloggiamenti ISA a 16 bit. La prima parte dell'alloggiamento, a sinistra, equivale alla vecchia versione a 8 bit.



Il connettore di un alloggiamento a 16 bit è praticamente un prolungamento di quello a 8 bit, come si può vedere nella figura 9.49, in cui si possono ancora inserire le schede più vecchie. Tuttavia, nei primi 8 bit di un bus ISA si può accedere solo ai livelli di IRQ inferiori a nove.

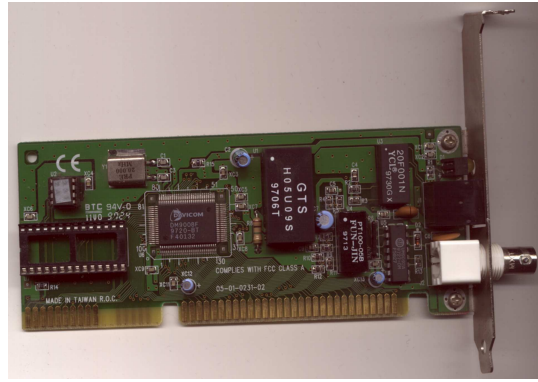
Una limitazione importante del bus ISA, a parte la velocità di trasferimento dei dati molto ridotta, è l'impossibilità di condividere uno stesso livello di IRQ tra diverse schede.

Figura 9.50. L'esempio tipico di una scheda ISA a 8 bit, che non richiede alte prestazioni e nemmeno di accedere a livelli IRQ superiori al numero otto.



Le figure 9.50 e 9.51 mostrano l'esempio di due schede da inserire in alloggiamenti ISA, rispettivamente a 8 bit e 16 bit. Nel primo caso, che rappresenta l'insieme di due interfacce seriali e una parallela, sono visibili i ponticelli che consentono di configurare i livelli di IRQ utilizzati e gli indirizzi di I/O relativi. Nel secondo caso, che riguarda una scheda di rete su cavo coassiale, questi ponticelli mancano, perché sostituiti da una configurazione effettuata attraverso del software speciale, realizzato appositamente per quella scheda.

Figura 9.51. Una scheda di rete che utilizza un alloggiamento ISA a 16 bit.



### 9.5.2 MCA

La sigla MCA sta per *Micro channel architecture* e rappresenta un bus a 32 bit introdotto da IBM nel 1987 nei primi elaboratori con microprocessori x86 a 32 bit.

Il bus MCA è in grado di gestire anche il *bus mastering*, con cui un componente del bus può prendere il controllo. Un'altra caratteristica innovativa per l'epoca è la capacità di configurare automaticamente l'utilizzo delle risorse, senza bisogno di interventi, come invece richiede il bus ISA.

Il bus MCA non è stato accettato bene dal mercato, perché incompatibile con ISA.

### 9.5.3 EISA

La sigla EISA sta per *Extended industry standard architecture* e rappresenta un bus a 32 bit introdotto da Compaq a seguito della comparsa del bus MCA.

La caratteristica più importante di questo bus è la compatibilità con le schede ISA, che possono essere inserite nei suoi alloggiamenti, funzionando regolarmente, consentendo alle schede EISA di comunicare a 32 bit e di gestire il *bus mastering*.

Lo standard EISA, nonostante la compatibilità ISA è stato superato con l'introduzione del bus VESA.

### 9.5.4 VESA local bus

VESA sta per *Video electronics standards association* e rappresenta un'associazione con lo scopo di definire degli standard sulla gestione del video negli elaboratori con architettura i486. Lo standard VESA ha dato vita nel 1992 al bus VLB, ovvero *VESA local bus*, a 32 bit, su un alloggiamento compatibile con le schede ISA comuni, dove si aggiunge un'estensione per i collegamenti mancanti.

A differenza di altri bus a 32 bit, in questo caso non si possono avere molti alloggiamenti (di solito un massimo di due); inoltre, il *bus mastering* non è efficace e non è disponibile un sistema di configurazione automatica delle risorse utilizzate. Infine, il bus dipende strettamente dalle caratteristiche dei microprocessori i486.

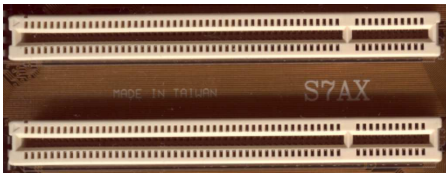
### 9.5.5 PCI

PCI sta per *Peripheral component interconnect* e rappresenta un bus, inizialmente a 32 bit, in grado di gestire *bus mastering* e la configurazione automatica delle risorse, con la caratteristica di non dipendere strettamente dal microprocessore. In particolare, è ammissibile la condivisione dei livelli di IRQ per più componenti simultaneamente.

Il bus PCI è stato introdotto nel 1993 a sostegno delle nuove generazioni di microprocessori i586, ma è stato usato anche in altre architetture, con un'estensione a 64 bit.

Con il bus PCI, si abbandona progressivamente il vecchio bus ISA.

Figura 9.52. Un alloggiamento PCI a 32 bit su una scheda madre comune.



Generalmente, le schede madri che mettono a disposizione un bus PCI e incorporano anche altre funzionalità, lo fanno attraverso lo stesso bus PCI. In pratica, si trovano bus ATA, bus USB e altro, già inseriti nel bus PCI.

Il bus PCI ha un proprio sistema di interruzione, dove i livelli relativi vengono identificati da sigle del tipo '#A', '#B', '#C' e '#D', oppure '#1', '#2', '#3' e '#4'. Di solito non è necessario essere consapevoli di questo, salvo il caso in cui tali indicazioni possano apparire nella configurazione del firmware.

Nell'ambito di un bus PCI, i componenti che vi si collegano possono essere individuati in base all'alloggiamento in cui si trovano, che può essere costituito da un supporto in cui inserire una scheda, oppure può essere qualcosa di integrato nella stessa scheda madre. Inoltre, ogni componente può distinguere al suo interno delle *funzioni*. In pratica, alloggiamenti e funzioni hanno un numero di identificazione, con cui fare riferimento esattamente a un componente e a una funzione precisa.

### 9.5.6 Informazioni da un kernel Linux

Il kernel Linux consente di ottenere informazioni sul bus PCI attraverso la directory virtuale `/proc/`, con l'aiuto del programma `lspci`.<sup>1</sup> In particolare, si può osservare che le coordinate di questi partono dall'indicazione del bus, che solitamente è uno solo, per cui corrisponde all'indirizzo zero. Nell'estratto di esempio vengono mostrati solo alcuni componenti relativi a quanto già integrato in una scheda madre comune:

```
$ lspci -v [lvio]

00:00.0 Host bridge: VIA Technologies, Inc. ↵
↳VT82C693A/694x [Apollo PRO133x] (rev c4)
  Flags: bus master, medium devsel, latency 8
  Memory at ec000000 (32-bit, prefetchable) [size=64M]
  Capabilities: <available only to root>

00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x ↵
↳[Apollo MVP3/Pro133x AGP] (prog-if 00 [Normal decode])
  Flags: bus master, 66Mhz, medium devsel, latency 0
  Bus: primary=00, secondary=01, subordinate=01, ↵
↳sec-latency=0
  Memory behind bridge: feb00000-febfffff
  Prefetchable memory behind bridge: f0000000-fe9fffff
  Capabilities: <available only to root>

00:02.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA ↵
↳[Mobile South] (rev 12)
  Flags: bus master, stepping, medium devsel, ↵
↳latency 0

00:02.1 IDE interface: VIA Technologies, Inc. ↵
↳VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC ↵
↳Bus Master IDE (rev 06) (prog-if 8a [Master SecP PriP])
  Flags: bus master, stepping, medium devsel, latency 32
  I/O ports at fff0 [size=16]

00:02.2 USB Controller: VIA Technologies, Inc. USB ↵
↳(rev 08) (prog-if 00 [UHCI])
  Subsystem: VIA Technologies, Inc. (Wrong ID) ↵
↳USB Controller
  Flags: bus master, medium devsel, latency 22, IRQ 10
  I/O ports at fce0 [size=32]

00:02.3 Host bridge: VIA Technologies, Inc. VT82C596 ↵
↳Power Management (rev 20)
  Flags: medium devsel

00:0f.0 Ethernet controller: 3Com Corporation ↵
↳3c905C-TX/TX-M [Tornado] (rev 74)
```

```
Subsystem: 3Com Corporation 3C905C-TX Fast ↵
↳Ethernetlink for PC Management NIC
  Flags: bus master, medium devsel, latency 80, IRQ 9
  I/O ports at 7480 [size=128]
  Memory at feafff80 (32-bit, non-prefetchable) ↵
↳[size=128]
  Expansion ROM at <unassigned> [disabled] ↵
↳[size=128K]
  Capabilities: <available only to root>

00:12.0 Multimedia audio controller: ESS Technology ↵
↳ES1988 Allegro-1 (rev 10)
  Subsystem: IBM: Unknown device 017b
  Flags: bus master, medium devsel, latency 16, IRQ 5
  I/O ports at 7800 [size=256]
  Capabilities: <available only to root>

01:00.0 VGA compatible controller: S3 Inc. Savage 4 ↵
↳(rev 04) (prog-if 00 [VGA])
  Subsystem: Diamond Multimedia Systems: ↵
↳Unknown device 4906
  Flags: bus master, 66Mhz, medium devsel, ↵
↳latency 248, IRQ 11
  Memory at feb80000 (32-bit, non-prefetchable) ↵
↳[size=512K]
  Memory at f0000000 (32-bit, prefetchable) ↵
↳[size=128M]
  Expansion ROM at <unassigned> ↵
↳[disabled] [size=64K]
  Capabilities: <available only to root>
```

A fianco di `lspci`, si colloca `setpci`, dello stesso pacchetto di programmi di servizio. Con questo si può accedere alla configurazione dettagliata delle caratteristiche dei componenti PCI, cosa che di solito non serve fare. Si vedano eventualmente le pagine di manuale `lspci(8)` e `setpci(8)`.

### 9.5.7 AGP

AGP, ovvero *Accelerated graphics port*, non è un bus vero e proprio, ma soltanto una «porta», rappresentata da un solo alloggiamento speciale nelle schede madri, per l'utilizzo di adattatori video particolarmente potenti, che richiedono un accesso privilegiato alle risorse del sistema.

Figura 9.54. Un alloggiamento AGP X1.



Lo sviluppo di AGP ha portato a distinguere dei livelli con prestazioni differenti; AGP 1X, 2X, 4X, 8X e PRO, dove la sigla *nX* serve a rappresentare le prestazioni in termini di velocità. Dal momento che le caratteristiche delle schede video cambiano da una versione all'altra, l'alloggiamento AGP subisce delle piccole modifiche in modo da ridurre la possibilità di inserire una scheda non compatibile. Rispetto alla figura 9.54 ci può essere uno spostamento della «chiave» di inserzione, come la sua mancanza nel caso di un alloggiamento «universale»; infine, nel caso di AGP PRO, si ha un alloggiamento con un'estensione maggiore.

Figura 9.55. Un alloggiamento AGP «universale», che dovrebbe poter accogliere tutte le schede, escluse quelle «PRO». Come si può vedere manca di proposito la chiave di inserimento.



Figura 9.56. Un alloggiamento AGP PRO.



Figura 9.57. Un tipo di scheda AGP comune.

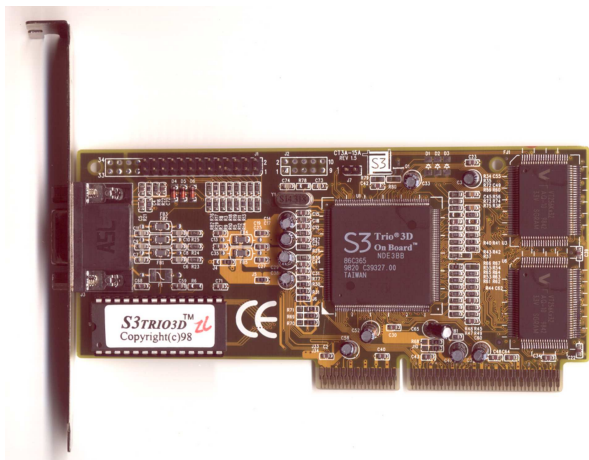
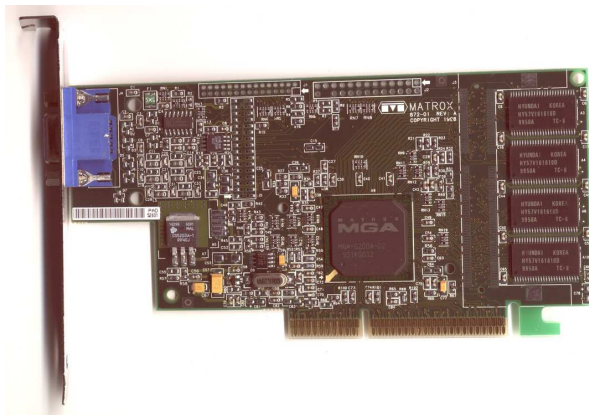


Figura 9.58. Un tipo di scheda AGP PRO; lo si distingue per la presenza del piedino finale, dopo la fila dei contatti standard.



### 9.5.8 AMR

AMR, ovvero *Audio modem riser*, non è un bus, ma soltanto un alloggiamento, normalmente singolo, da usare per funzionalità audio o come modem.

Lo standard AMR si associa a AC97 (*Audio codec 97*) e generalmente questo tipo di alloggiamento è già scomparso dalle schede madri che normalmente tendono a integrare tutte le funzionalità al proprio interno, senza bisogno di schede da alloggiare in un connettore AMR.

Figura 9.59. Un modem per alloggiamento AMR.



La figura mostra un modem AMR e come si può vedere non ci sono

componenti importanti, salvo degli integrati microscopici, pertanto si intende che l'elaborazione elettronica più importante avviene già nella scheda madre (o meglio nella CPU).

Figura 9.60. Un alloggiamento AMR.



### 9.5.9 PCI Express

Il successore del bus PCI potrebbe essere PCI Express, che introduce una comunicazione seriale e un insieme di alloggiamenti differenti, in base al numero di «corsie» (*lane*) previste.

In pratica, l'alloggiamento di un bus PCI Express può essere relativamente corto, quasi come quello di un alloggiamento AMR, per il livello «x1», allungandosi per i livelli successivi, fino al massimo di «x16». Naturalmente, i vari livelli corrispondono a prestazioni maggiori, proporzionalmente, come suggerisce la sigla stessa.

Teoricamente, pur cambiando completamente la sua forma e il modo di comunicare, i componenti PCI Express dovrebbero mantenere almeno una compatibilità a livello di software.

Inizialmente, il bus PCI Express si introduce nelle schede madri per consentire l'uso di schede video grafiche con prestazioni eccezionali, pertanto, proprio per questa ragione, si tratta solitamente di un solo alloggiamento x16.

### 9.6 USB

La sigla USB sta per *Universal serial bus* e rappresenta un tipo di bus in cui i vari componenti si possono collegare in una struttura che ricorda quella dell'albero. In questo tipo di bus, le comunicazioni sono controllate da un protocollo che non consente alle unità periferiche di comunicare direttamente tra di loro.

#### 9.6.1 Caratteristiche generali

Il bus USB è fatto per connettere unità periferiche esterne a un elaboratore, al contrario di SCSI, che prevede la possibilità di un collegamento interno e di uno esterno. Il bus USB ha un collegamento speciale, attraverso connettori di tipo differente, a seconda che si tratti del lato rivolto verso l'unità periferica (il tipo «B»), oppure verso l'unità di controllo (il tipo «A»). La continuazione del bus avviene attraverso dei concentratori (*hub*), che da un lato si collegano come unità periferiche e dall'altro offrono più connettori uguali a quelli dell'unità di controllo. In particolare, l'unità di controllo viene definita anche *root hub*, come dire che si tratta del primo concentratore della struttura.

Figura 9.61. Porta USB di tipo «B» di un'unità periferica.

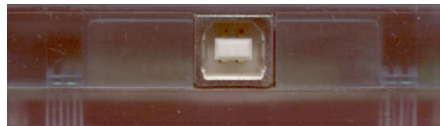


Figura 9.62. Connettori USB: quello in alto va collegato all'unità di controllo (tipo «A»), oppure al concentratore, mentre quello in basso va collegato all'unità periferica (tipo «B»).



Il bus USB consente di collegare un massimo di 127 dispositivi, oltre all'unità di controllo. In questo numero vanno contati anche i concentratori. A differenza del bus SCSI, qui i dispositivi non

devono essere numerati manualmente, perché a questo provvede automaticamente il sistema.

Figura 9.63. Connettore USB «B», miniaturizzato, usato di solito per le macchine fotografiche.



### 9.6.2 Alimentazione elettrica

«

Una caratteristica molto importante del bus USB è la possibilità di alimentare i dispositivi che vi si collegano, attraverso il collegamento del bus stesso. La corrente elettrica che può essere fornita in questo modo arriva normalmente a un massimo di 500 mA.

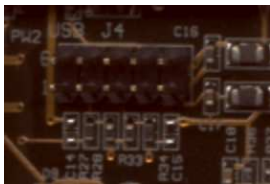
Un dispositivo USB comune può essere alimentato anche in modo indipendente, se il suo consumo è elevato; tuttavia, il fatto che l'alimentazione esterna non sia sempre necessaria, consente di ridurre il cablaggio per la connessione di componenti esterni, soprattutto quando questi sono di piccole dimensioni.

### 9.6.3 Collegamento di una porta USB

«

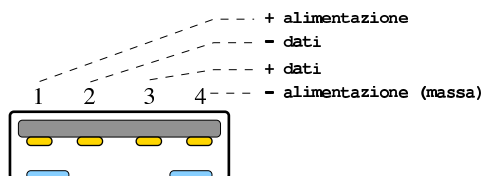
In condizioni normali, non è necessario essere a conoscenza del modo in cui i quattro terminali di una porta USB devono essere collegati da un punto di vista elettrico. Tuttavia, alcune vecchie schede madri hanno un bus USB integrato a cui si accede attraverso una o due file di cinque o di quattro piedini ciascuna, senza che esista uno standard preciso per il collegamento di questi alla porta USB di tipo «A». Si osservi la figura 9.64 a questo proposito.

Figura 9.64. Piedini per collegare due porte USB di tipo «A» su una scheda madre molto vecchia, in cui non è chiaro come siano disposti i vari terminali.



Per prima cosa occorre comprendere un po' la logica delle connessioni USB, dal punto di vista fisico. La figura 9.65 schematizza una porta USB di tipo «A» su un pannello di un elaboratore.

Figura 9.65. Collegamento elettrico di una porta USB di tipo «A».



Si può osservare che i terminali utilizzati per l'alimentazione sono quelli più esterni; inoltre, i terminali interni per i dati, hanno una polarità invertita rispetto all'alimentazione. Con un po' di fortuna, anche i terminali che sporgono da una scheda madre dovrebbero rispettare questa logica, tenendo conto che, se esistono cinque terminali, uno va collegato alla massa esterna, in modo distinto dall'alimentazione negativa. Una disposizione tipica dei piedini di una scheda madre è questa:

1. alimentazione +5 V;
2. dati, terminale negativo;

3. dati, terminale positivo;
4. massa dell'alimentazione (0 V);
5. massa esterna.

Un'altra possibilità comune è la seguente:

1. alimentazione +5 V;
2. dati, terminale negativo;
3. non collegato (piedino assente);
4. dati, terminale positivo;
5. massa dell'alimentazione (0 V);

In queste condizioni, è necessario verificare almeno i terminali che sono adibiti al trasporto dell'alimentazione, attraverso uno strumento di misura. A elaboratore spento, si può verificare quali piedini risultano essere collegati direttamente alla massa; se sono due per ogni fila, quello più interno dovrebbe corrispondere alla massa di alimentazione (0 V), mentre quello più esterno dovrebbe essere collegato alla massa vera e propria. Il terminale esterno opposto di ogni fila, dovrebbe essere il positivo di alimentazione (+5 V), mentre i due terminali rimanenti dovrebbero essere quelli dei dati, con polarità alternate (vicino al positivo di alimentazione dovrebbe trovarsi il negativo dei dati; vicino al negativo di alimentazione, dovrebbe trovarsi il positivo dei dati). Se ci sono due file parallele di terminali, si può verificare che i piedini esterni riguardano l'alimentazione, perché risultano collegati assieme (i due terminali del positivo di alimentazione sono collegati tra loro, così come i due o i quattro terminali di massa). Accendendo l'elaboratore si può verificare che il terminale positivo dell'alimentazione ha una tensione di (+5 V).

### 9.6.4 Tipi di unità di controllo

«

Le unità di controllo USB sono fondamentalmente di tre tipi, in base alla loro compatibilità con lo standard OHCI (*Open host controller interface*), con lo standard UHCI (*Universal host controller interface*) o EHCI.

### 9.7 USB e il Kernel Linux

«

La gestione di un bus USB in un sistema GNU/Linux parte dal kernel che deve essere stato predisposto per questo (sezione 8.3.8.12). In particolare, deve essere stato selezionato il tipo di unità di controllo; in pratica si deve attivare la gestione UHCI, OHCI, oppure EHCI. Eventualmente, per sapere a quale tipo appartiene la propria unità di controllo, dovrebbe essere sufficiente leggere il rapporto relativo al bus PCI (infatti, il bus USB si innesta normalmente in un bus PCI).

```
# lspci -v [Invio]
```

```
00:07.2 USB Controller: VIA Technologies, Inc. VT82C586B USB
(rev 11)
(prog-if 00 [UHCI])
Subsystem: Unknown device 0925:1234
Flags: bus master, medium devsel, latency 32, IRQ 11
I/O ports at d400 [size=32]
Capabilities: [80] Power Management version 2
```

Nell'esempio si vede la sigla UHCI, che chiarisce di che tipo sia il bus USB.

#### 9.7.1 Avvio di un sistema GNU/Linux da un'unità USB, con kernel monolitico

«

L'avvio di un sistema GNU/Linux che risiede fisicamente in un'unità di memoria USB esterna, richiede la preparazione di alcuni accorgimenti. Per cominciare, il kernel deve essere stato predisposto per la gestione del bus SCSI, del bus USB e in particolare per la gestione della memoria di massa su un bus USB (sezione 8.3.8.12). Ovviamente, se le funzionalità che consentono di accedere all'unità USB sono spostate in moduli esterni al kernel, occorre predisporre

un disco RAM iniziale per il caricamento di questi, ma qui si preferisce fare riferimento a un kernel che non abbia bisogno di tale accorgimento.

Durante l'avvio del kernel, è molto probabile che il rilevamento dell'unità di memoria USB sia in ritardo rispetto al momento in cui viene fatto il tentativo di innestare il file system principale, cosa che si traduce normalmente in un blocco dovuto a un *kernel panic*. Per risolvere il problema in modo «standard», sembra non esserci altra strada che predisporre un disco RAM iniziale (sezione 8.10), ma per chi è disposto a ricompilarsi il kernel dovrebbe essere più semplice introdurre nel codice del kernel stesso qualche istruzione per ritardare il procedimento, così da far trovare il disco USB quando è il momento.

Si può intervenire nel file `'sorgenti_linux/init/main.c'` per introdurre una pausa che viene fissata attraverso un parametro di avvio, creato appositamente: `'setupdelay'`. Questo tipo di modifica deriva da un lavoro di Willi Tarrey, pubblicato attraverso un messaggio del 2 maggio 2004 nella lista *Linux-Kernel*, che può essere letto presso <http://kml.indiana.edu/hypermil/linux/kernel/0405.0/0252.html>.

La modifica che viene proposta richiede di aggiungere delle istruzioni al contenuto di `'sorgenti_linux/init/main.c'`. Lo schema seguente dovrebbe permettere di capire dove si deve intervenire; le righe aggiunte sono evidenziate con un carattere nero.

Listato 9.67. File `'sorgenti_linux/init/main.c'`; modifiche utili in un kernel 2.6.

```

...
static char * argv_init[MAX_INIT_ARGS+2] = { "init", NULL, };
char * envp_init[MAX_INIT_ENVS+2] = { "HOME=/", "TERM=linux", NULL, };
static const char *panic_later, *panic_param;
static int setupdelay; /* delay some seconds to let
                        devices set up before root mount */

__setup("profile=", profile_setup);

static int __init setupdelay_setup(char *str)
{
    int par;
    if (get_option(&str, &par)) {
        setupdelay = par;
    }
    return 1;
}

__setup("setupdelay=", setupdelay_setup);

...
...
static int wait_setup_delay (void)
{
    int time_left;
    if (setupdelay) {
        printk("Waiting %d s to let devices set up properly ",
              setupdelay);
        time_left = setupdelay;
        while (time_left) {
            set_current_state (TASK_INTERRUPTIBLE);
            schedule_timeout (HZ);
            time_left--;
            printk(". ");
        }
        printk("\n");
    }
    return 1;
}

static int init(void * unused)
{
    ...
    /*
     * Do this before initcalls, because some drivers want to
     * access firmware files.
     */
    populate_rootfs();

    do_basic_setup();

    wait_setup_delay ();
}

```

```

/*
 * check if there is an early userspace init. If yes, let
 * it do all the work
 */
if (sys_access((const char __user *) "/init", 0) == 0)
    execute_command = "/init";
else
    prepare_namespace();
...

```

Naturalmente, per fare in modo che ci sia effettivamente un ritardo di *n* secondi, occorre usare l'opzione di avvio `'setupdelay=n'`.

Quanto mostrato dovrebbe risolvere il problema dell'avvio del sistema, ma rimane il fatto che alcune unità di memorizzazione connesse sul bus USB rispondono in ritardo ai comandi che gli vengono impartiti. In pratica, ciò significa che quando si esegue il distacco di un file system contenuto in un'unità USB, conviene aspettare un po' prima di spegnerla o comunque di staccarla fisicamente, per non rischiare di perdere i dati. Se il file system principale è innestato da un'unità USB, in presenza di un kernel Linux molto vecchio, potrebbe succedere che lo spegnimento che conclude la procedura di arresto del sistema avvenga prima che l'aggiornamento del file system stesso possa essere stato completato effettivamente. Per ovviare a questo grave inconveniente, si potrebbe disabilitare la funzionalità attraverso un parametro di avvio del kernel:

```
apm=off
```

In alternativa, in modo più specifico:

```
apm=no-power-off
```

## 9.7.2 Sequenza di caricamento dei moduli del kernel

Quando la gestione del bus USB è demandata a dei moduli del kernel Linux, può crearsi una situazione, apparentemente innocua, che invece è bene tenere in considerazione.

Il problema da tenere in evidenza riguarda i moduli di gestione del tipo di unità di controllo USB, la quale può essere di tipo OHCI, UHCI o EHCI. I moduli in questione sono `'ohci-hcd.ko'`, `'uhci-hcd.ko'` ed `'ehci-hcd.ko'`, ovvero `'ohci_hcd'`, `'uhci_hcd'` ed `'ehci_hcd'`.

Può succedere che un elaboratore contenga un'unità di controllo USB che può essere pilotata con due tipi alternativi di moduli; per esempio può essere compatibile con il tipo UHCI e anche EHCI, pertanto potrebbero essere usati i moduli `'uhci_hcd'` ed `'ehci_hcd'`. Se i moduli vengono caricati in modo automatico, c'è da attendersi che la scelta del modulo più appropriato avvenga correttamente; se invece si deve fare questo in modo manuale, o comunque con propri script, il caricamento del modulo del kernel «migliore» potrebbe semplicemente prendere il sopravvento su un altro modulo compatibile e già caricato in una fase precedente.

A titolo di esempio, si suppone che ci sia la possibilità di caricare sia il modulo `'uhci_hcd'`, sia `'ehci_hcd'`, come già ipotizzato, ma in questo caso si aggiunge l'accesso a un'unità di memorizzazione esterna (solida o a disco). Si comincia caricando il modulo `'uhci_hcd'`; quindi si innesta il file system contenuto nell'unità di memorizzazione:

```
# modprobe uhci_hcd [Invio]
```

```
# lsusb [Invio]
```

```

Bus 004 Device 002: ID 1307:0163 Transcend Information, Inc.
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000

```

```
# mount -t auto /dev/sda1 /mnt/sda1 [Invio]
```

A questo punto si carica anche il modulo `'ehci_hcd'` e succede che poi, l'innesto ipotizzato nella directory `'/mnt/sda1/'`, non è più efficace:

```
# modprobe ehci_hcd [Invio]

# lsusb [Invio]

Bus 005 Device 002: ID 1307:0163 Transcend Information, Inc.
Bus 005 Device 001: ID 0000:0000
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

Come si vede, l'unità «1307:0163» è passata dal bus «004» al bus «005». Se si tenta di leggere il contenuto della directory `'/mnt/sda1/'`, dove dovrebbe trovarsi l'innesto dell'unità di memorizzazione esterna, questa risulta semplicemente vuota (ovvero inaccessibile).

```
# rmmod ehci_hcd [Invio]

# umount /mnt/sda1 [Invio]

# rmmod uhci_hcd [Invio]
```

Per risolvere il problema, volendo caricare entrambi i moduli `'uhci_hcd'` ed `'ehci_hcd'`, dovrebbe bastare procedere in senso opposto:

```
😊 # modprobe ehci_hcd [Invio]

# lsusb [Invio]

Bus 001 Device 002: ID 1307:0163 Transcend Information, Inc.
Bus 001 Device 001: ID 0000:0000

# mount -t auto /dev/sda1 /mnt/sda1 [Invio]

# modprobe uhci_hcd [Invio]

# lsusb [Invio]

Bus 005 Device 001: ID 0000:0000
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 002: ID 1307:0163 Transcend Information, Inc.
😊 Bus 001 Device 001: ID 0000:0000
```

Come si vede, questa volta l'unità «1307:0163» non ha cambiato bus e l'unità innestata precedentemente continua a essere raggiungibile regolarmente.

Come si può intuire, una situazione come quella descritta è poco probabile che si presenti in condizioni «normali», perché è logico attendersi che tutti i moduli che si vogliono caricare vengano utilizzati prima di innestare delle unità esterne di memorizzazione. La situazione più verosimile per un problema di questo tipo riguarda l'avvio da unità USB, utilizzando un kernel modulare. Infatti, in questo caso occorre utilizzare un disco RAM iniziale, contenente un sistema GNU/Linux provvisorio. Durante il funzionamento di questo sistema provvisorio potrebbe essere caricato un modulo come `'uhci_hcd'`, con il quale andare poi a innestare il file system principale presso l'unità di memorizzazione USB. Successivamente, con l'abbandono del sistema provvisorio per passare a quello contenuto nell'unità esterna, ci potrebbe essere una fase di autorilevamento dell'hardware (che potrebbe essere prodotta anche dall'attivazione di `udev`), con la quale viene caricato automaticamente il modulo `'ehci_hcd'`: da quel punto in poi, verrebbe meno l'accesso al file system principale, bloccando la procedura di avvio.

## 9.8 Introduzione al BIOS

Il firmware degli elaboratori x86 e derivati è noto con il nome BIOS (*Basic input output system*). Originariamente il suo scopo era soltanto quello di fornire delle funzioni essenziali, in modo da consentire a «sistemi operativi» da microelaboratore, come il Dos, di avviarsi e di funzionare.

Nel tempo, il BIOS si è evoluto e normalmente costituisce un piccolo sistema di gestione dell'hardware, indipendente da altro software. In particolare, diventa lo strumento primario di configurazione delle varie componenti fisiche che sono incorporate nella scheda madre (conservando la configurazione in una piccola memoria C/MOS, sostenuta da una batteria).

### 9.8.1 Collocazione fisica del BIOS

Rispetto alla terminologia tradizionale, il BIOS (compresa la porzione della memoria C/MOS), costituisce la memoria «ROM» dell'elaboratore. In questo senso (a parte la porzione «C/MOS»), il BIOS è contenuto normalmente all'interno di integrati che hanno una funzione del genere (memorie ROM vere e proprie, oppure memorie cancellabili e riprogrammabili in vario modo).

Figura 9.72. Integrato riprogrammabile degli anni 1980: sotto l'etichetta si intravede la finestrella per la cancellazione, che richiede l'esposizione a raggi ultravioletti. L'etichetta avrebbe anche lo scopo di proteggere la memoria da una cancellazione accidentale.



Figura 9.73. Memoria «flash» degli anni 1990, riprogrammabile senza bisogno di estrarre l'integrato dalla scheda madre.

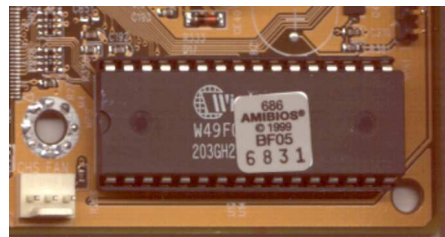
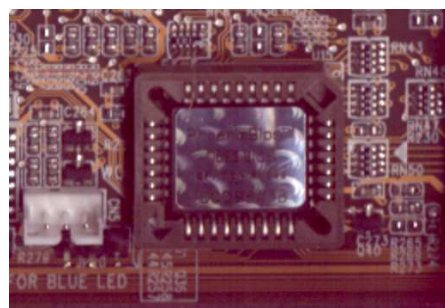


Figura 9.74. Memoria «flash» compatta, usata alla fine degli anni 1990 e all'inizio degli anni 2000.



In base alla tradizione, come si vede nelle figure, l'integrato che contiene il BIOS dovrebbe essere estraibile, per facilitarne la sostituzione o la riprogrammazione. Tuttavia, le schede madri tendono a disporre di un BIOS contenuto in un integrato saldato e pertanto non rimpiazzabile.



## 9.8.2 Blocco di avvio

Quando il BIOS risulta danneggiato per qualche ragione, può darsi che il codice del blocco di avvio (*bootblock*) contenuto del BIOS, sia ancora integro. In tal caso, dovrebbe essere possibile avviare un sistema Dos di emergenza dall'unità a dischetti, anche se lo schermo e la tastiera possono risultare inutilizzabili. In questi casi, ammesso che se ne possa disporre, serve un dischetto contenente il sistema operativo minimo e gli strumenti indispensabili al ripristino del BIOS.

In un'altra sezione di questo capitolo, viene descritto UniFlash per salvare e ripristinare il BIOS. A titolo di esempio, volendo usare questo programma allo scopo di ripristinare un BIOS che è in grado di eseguire solo il codice del blocco di avvio, occorre inserire nel dischetto il file `'UNIFLASH.EXE'`, la copia del BIOS da ripristinare e un file `'AUTOEXEC.BAT'` realizzato in modo da eseguire automaticamente `'UNIFLASH.EXE'`:

```
@UNIFLASH -E file_bios_da_ripristinare
```

Nel caso particolare di un BIOS AMI, il blocco di avvio dovrebbe contenere delle istruzioni speciali, attraverso le quali viene caricato automaticamente dal dischetto il file `'AMIBOOT.ROM'` (senza che il dischetto debba contenere un sistema Dos o altro), oppure un altro nome specifico nel caso di schede madri speciali. Eventualmente, può essere necessario premere la combinazione di tasti [*Ctrl+Inizio*] ([*Ctrl+Home*]) per forzare il caricamento del file all'avvio (nel caso di una scheda madre ASUS, si può trattare della combinazione [*Alt F2*]).

Quando si realizza un dischetto Dos con gli strumenti necessari per il ripristino del BIOS, è necessario che non venga avviato nulla di superfluo. Pertanto, è meglio sopprimere il file `'CONFIG.SYS'` e mettere nel file `'AUTOEXEC.BAT'` esclusivamente il comando o i comandi di ripristino. In particolare, non vanno avviati programmi come `'HIMEM.SYS'` o simili.

Se il BIOS è stato danneggiato, ma il blocco di avvio è integro e non si vede nulla sullo schermo, si può provare a installare una scheda VGA nel bus ISA, oppure, in mancanza di un bus ISA si può tentare con una scheda per il bus AGP.

## 9.8.3 Modifica del BIOS

A seconda del tipo di tecnologia adottato, si possono utilizzare delle tecniche differenti per la modifica del contenuto del BIOS. Nei primi elaboratori era necessario estrarre l'integrato contenente il BIOS, cancellarlo (seguendo la procedura prevista per quel tipo di componente) e riscriverlo, attraverso un'apparecchiatura appropriata. Successivamente si è introdotto l'uso di memorie «flash» che consentono la riscrittura senza dover estrarre l'integrato dalla scheda madre.

La facilità attuale con cui è possibile riprogrammare la memoria del BIOS, consente di aggiornarlo, di solito per attivare funzionalità nuove, non previste originariamente per quella tale scheda madre.

Teoricamente, l'aggiornamento del BIOS dovrebbe avvenire con software realizzato appositamente per il tipo di scheda madre di cui si dispone, dal momento che non esiste una procedura unica standardizzata (si veda comunque il programma UniFlash). Tuttavia, questo software specifico è disponibile normalmente dai siti delle case produttrici, assieme alle varie versioni del BIOS esistenti per ogni scheda.

Il procedimento di riscrittura del BIOS ha un solo inconveniente: se l'operazione fallisce (di solito per un'interruzione dell'energia elettrica) o viene eseguita con un file non adatto, si rischia di non poter più avviare l'elaboratore; in pratica, si rende inutilizzabile la scheda madre.

## 9.8.4 BIOS di «marca»

Le case produttrici di elaboratori x86 completi, più conosciute, hanno utilizzato principalmente un BIOS prodotto internamente, più o meno derivato da quello originale di IBM. Al contrario, i produttori di schede madri hanno preferito affidarsi ad altre aziende specializzate, contribuendo così allo sviluppo di BIOS più raffinati e generalizzati.

Si ricordano solitamente tre nomi importanti di aziende produttrici di BIOS: AMI Phoenix e Award. Tuttavia, nel tempo le cose sono cambiate; per esempio Award è stata acquisita da Phoenix, ma anche senza avvenimenti così significativi, si può avvertire che nel tempo la configurazione di un certo tipo di BIOS cambia radicalmente.

Ogni produttore di BIOS mette a disposizione i propri strumenti per la modifica e la sostituzione del codice relativo, quando la memoria usata consente la riscrittura senza l'estrazione dalla scheda madre. Tuttavia, l'assenza di uno standard uniforme, richiede sempre una ricerca per l'acquisizione del software necessario alla versione del BIOS di cui si dispone, compatibile con la propria scheda madre.

Anche se il BIOS è realizzato in modo abbastanza generalizzato, ogni modello di scheda madre richiede delle variazioni specifiche che dipendono in pratica dai componenti installati effettivamente. Di conseguenza, l'aggiornamento del BIOS richiede di disporre di un file realizzato appositamente per la propria scheda madre, anche se gli strumenti di aggiornamento, eventualmente, possono essere sempre gli stessi.

Eccezionalmente, i BIOS modulari consentono la sostituzione di un certo modulo, senza interferire con il resto, ma si tratta di un'operazione delicata che richiede maggiore esperienza.

Figura 9.76. Programma di configurazione di un BIOS Award con menù semplice: menù principale.<sup>2</sup>

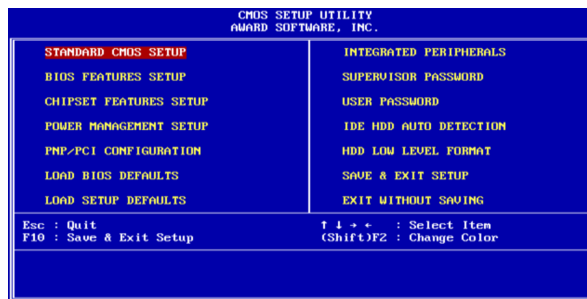


Figura 9.77. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù principale.<sup>2</sup>

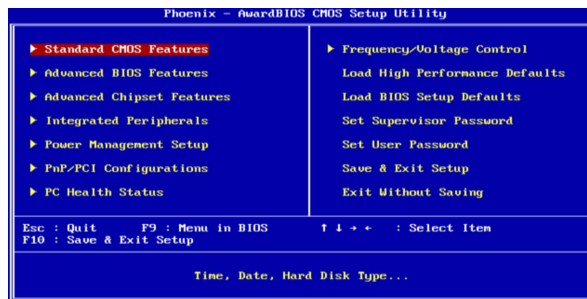


Figura 9.78. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù standard.<sup>2</sup>

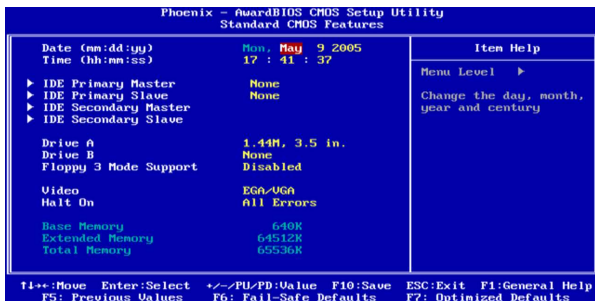


Figura 9.79. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù avanzato.<sup>2</sup>

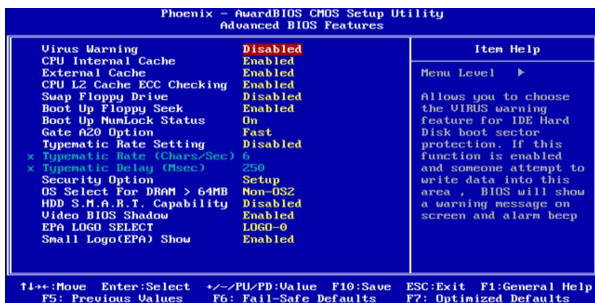


Figura 9.80. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù chipset.<sup>2</sup>

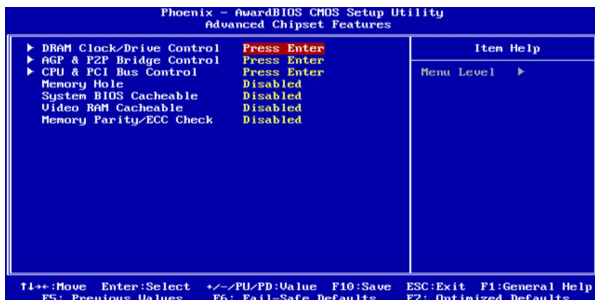


Figura 9.81. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù dispositivi integrati.<sup>2</sup>

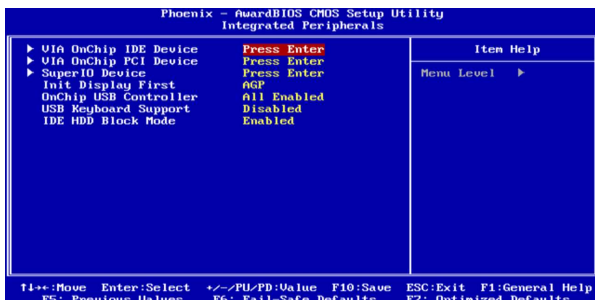


Figura 9.82. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù alimentazione.<sup>2</sup>

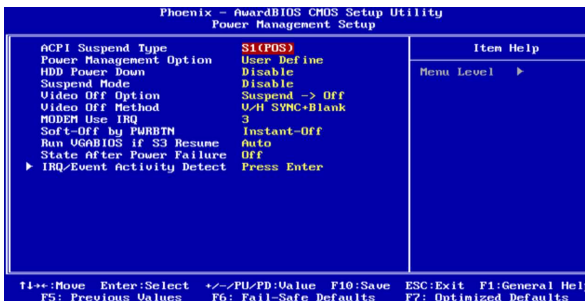


Figura 9.83. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù Plug & Play.<sup>2</sup>

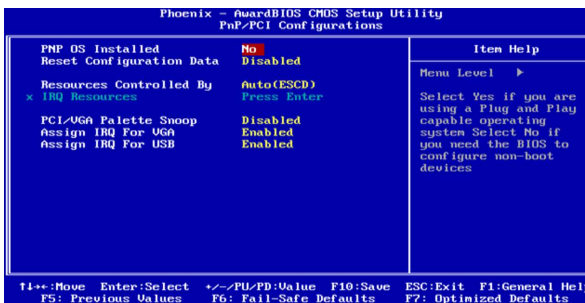


Figura 9.84. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù controllo della temperatura.<sup>2</sup>

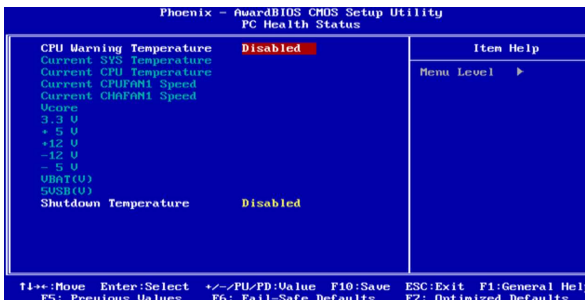


Figura 9.85. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù principale.<sup>2</sup>

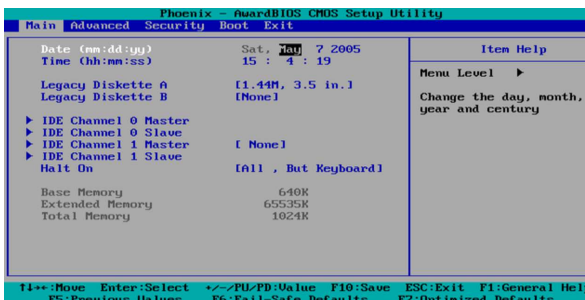


Figura 9.86. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù avanzato.<sup>2</sup>

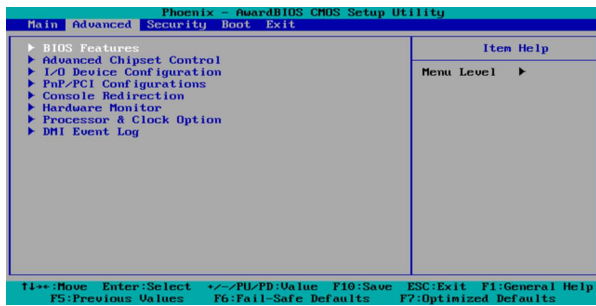


Figura 9.87. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù sicurezza.<sup>2</sup>

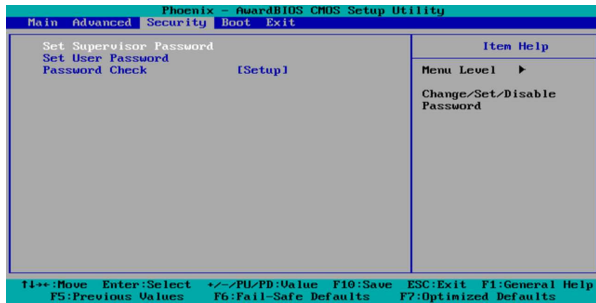


Figura 9.88. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù avvio.<sup>2</sup>

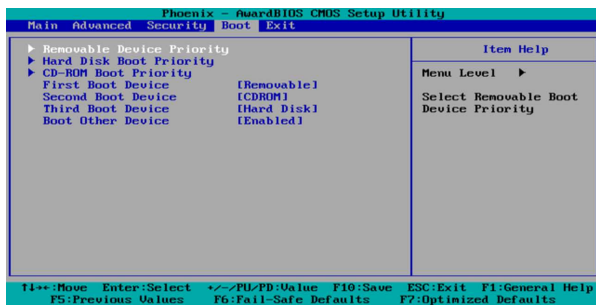


Figura 9.89. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù uscita.<sup>2</sup>

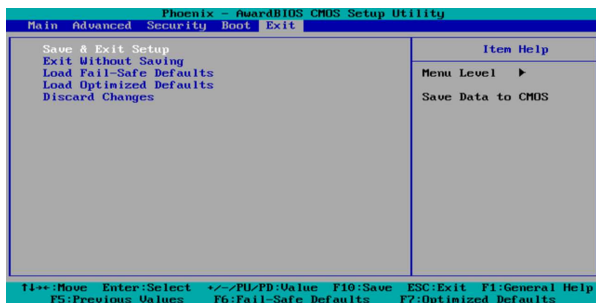


Figura 9.90. Programma di configurazione di un BIOS AMI con menù a tendina: menù principale.<sup>2</sup>



Figura 9.91. Programma di configurazione di un BIOS AMI con menù a tendina: menù avanzato.<sup>2</sup>



Figura 9.92. Programma di configurazione di un BIOS AMI con menù a tendina: menù sicurezza.<sup>2</sup>



Figura 9.93. Programma di configurazione di un BIOS AMI con menù a tendina: menù alimentazione.<sup>2</sup>



Figura 9.94. Programma di configurazione di un BIOS AMI con menù a tendina: menù avvio.<sup>2</sup>

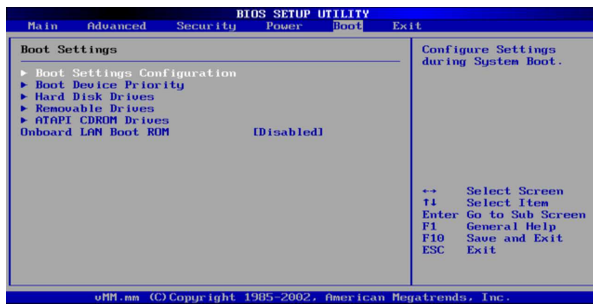


Figura 9.95. Programma di configurazione di un BIOS AMI con menù a tendina: menù uscita.<sup>2</sup>

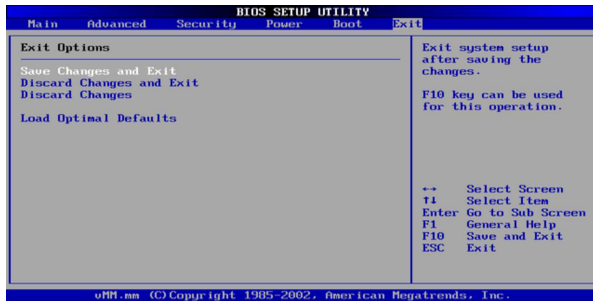


Figura 9.96. Programma di configurazione di un BIOS AMI con menù grafico: menù principale.<sup>2</sup>

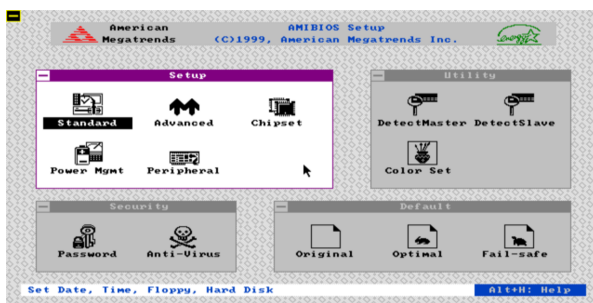


Figura 9.97. Programma di configurazione di un BIOS AMI con menù grafico: menù avanzato.<sup>2</sup>



Figura 9.98. Programma di configurazione di un BIOS AMI con menù grafico: menù chipset.<sup>2</sup>

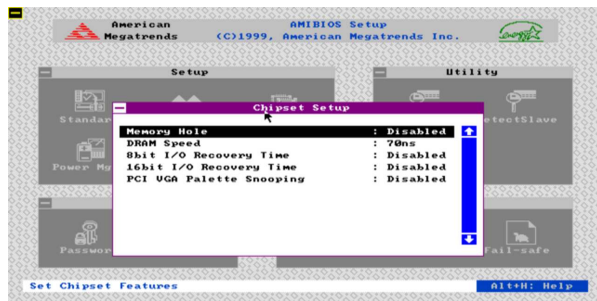


Figura 9.99. Programma di configurazione di un BIOS AMI con menù grafico: menù alimentazione.<sup>2</sup>

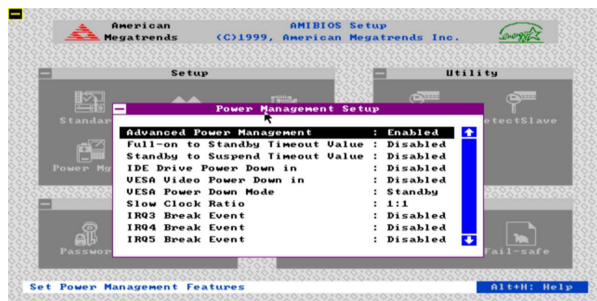


Figura 9.100. Programma di configurazione di un BIOS AMI con menù grafico: menù dispositivi integrati.<sup>2</sup>

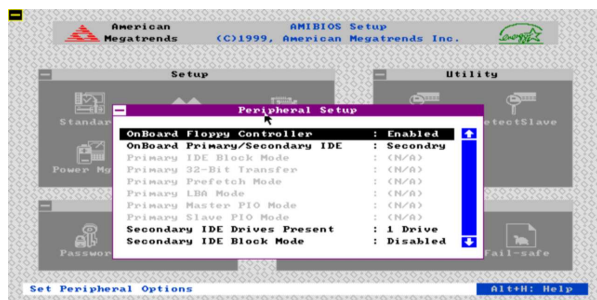
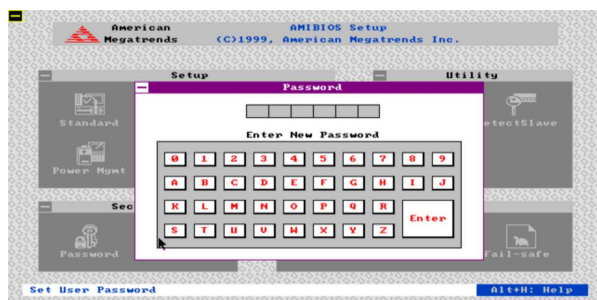


Figura 9.101. Programma di configurazione di un BIOS AMI con menù grafico: modifica della parola d'ordine.<sup>2</sup>



### 9.8.5 Salvare e riscrivere il BIOS

Per salvare e riscrivere il BIOS, ogni casa produttrice mette a disposizione i propri strumenti. Nella migliore delle ipotesi si tratta di software per sistemi Dos, che dovrebbero funzionare anche con FreeDOS. In tal caso, rimane però la necessità di disporre di un'unità a dischetti per poter operare.

È importante che il sistema Dos installato nel dischetto, assieme al programma di servizio che si vuole usare, sia ridotto al minimo (il kernel e la shell 'COMMAND.COM'), per garantire la massima

disponibilità di memoria e per lasciare lo spazio necessario ai file temporanei.

Prima di riscrivere il BIOS, occorre accertarsi di disporre degli strumenti adatti alla propria scheda madre, perché gli strumenti generici che offrono i produttori dei BIOS potrebbero non essere adeguati. Oltre a questo occorre considerare che la riscrittura è sempre un'operazione rischiosa che, se fallisce o se viene eseguita utilizzando file inadatti, rende inutilizzabile la scheda madre. Pertanto, **chi esegue una tale operazione**, oltre che essere competente, **deve accettare questo rischio a proprie spese**.

### 9.8.6 BIOS Award

Il programma AwdFlash consente il salvataggio e la riscrittura del BIOS Award; si può tentare una ricerca con Google in questo modo: <http://www.google.com/search?q=award+awdfash.exe+download>. A titolo di esempio, la versione 8.22 del programma AwdFlash, può essere usata così:

- A:\> **AWDFLASH /?** [Invio]  
mostra una guida rapida all'uso del programma;
- A:\> **AWDFLASH BIOS.ORG /pn /sy** [Invio]  
salva una copia del BIOS nel file 'BIOS.ORG';
- A:\> **AWDFLASH BIOS.NEW /py /sn** [Invio]  
riscrive il BIOS utilizzando il contenuto del file 'BIOS.NEW';
- A:\> **AWDFLASH BIOS.NEW /py /sn /cd /cp /cc** [Invio]  
riscrive il BIOS utilizzando il contenuto del file 'BIOS.NEW', azzerando però delle informazioni, tra cui la memoria C/MOS, costringendo così a rifare la configurazione.

Il programma Biosview consente di visualizzare la configurazione di un file contenente la copia di un BIOS Award. Si può tentare una ricerca con Google in questo modo: <http://www.google.com/search?q=award+biosview.exe+download>. A titolo di esempio, la versione 2.00.00 del programma Biosview, può essere usata così:

- A:\> **BIOSVIEW** [Invio]  
avvia il programma in modalità interattiva;
- A:\> **BIOSVIEW BIOS.BIN** [Invio]  
avvia il programma per visualizzare la configurazione del BIOS contenuta nel file 'BIOS.BIN'.

### 9.8.7 BIOS AMI

Per salvare una copia del BIOS AMI in un file, si può tentare di utilizzare un programma generico, come UniFlash; a ogni modo, disponendo di un file contenente un BIOS AMI, la riscrittura della memoria «flash» contenuta nella scheda madre si ottiene semplicemente realizzando un dischetto in formato Dos-FAT, contenente esclusivamente il file 'AMIBOOT.ROM', che costituisce la copia del BIOS da ripristinare o da aggiornare. Con questo dischetto, si deve riavviare l'elaboratore, premendo eventualmente la combinazione [Ctrl Inizio] ([Ctrl Home]) per forzare l'aggiornamento.

Con le schede madri ASUS il comportamento può essere diverso: la combinazione dei tasti può essere [Alt F2] e il nome del file dipende al modello della scheda madre (viene evidenziato durante il tentativo di caricare il file dal dischetto).

### 9.8.8 UniFlash

UniFlash<sup>3</sup> è un programma generalizzato, per salvare e memorizzare tutti o quasi tutti i tipi di BIOS (<http://www.rainbow-software.org/uniflash/>), a partire da un sistema Dos, da avviare attraverso un dischetto.

UniFlash è costituito dal file eseguibile 'UNIFLASH.EXE', che può essere utilizzato in modo interattivo, oppure specificando delle opzioni nella riga di comando:

```
UNIFLASH [opzioni]
```

Se non si utilizzano argomenti nella riga di comando, il programma funziona in modo interattivo. Questa modalità di funzionamento si presta per salvare il BIOS o parte di esso, oppure per aggiornare un BIOS quando tutto funziona ancora regolarmente.

Quando il BIOS risulta danneggiato per qualche ragione, può darsi che il codice del blocco di avvio (*bootblock*) contenuto del BIOS, sia ancora integro. In tal caso, si deve preparare un dischetto contenente un sistema Dos essenziale, il programma 'UNIFLASH.EXE', la copia del BIOS da ripristinare e uno script 'AUTOEXEC.BAT' realizzato in modo da eseguire automaticamente 'UNIFLASH.EXE':

```
@UNIFLASH -E file_bios_da_ripristinare
```

Figura 9.103. Il menù principale durante il funzionamento interattivo di UniFlash.<sup>2</sup>

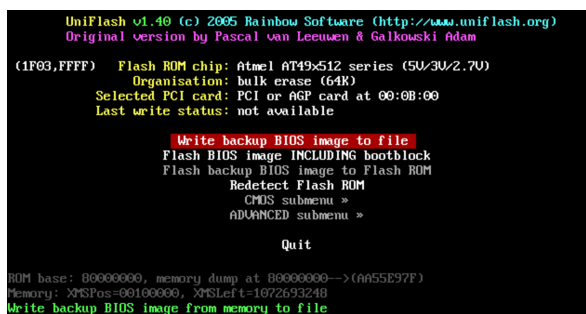


Figura 9.104. Il menù CMOS.

```
-----
| Save CMOS data to file |
| Restore CMOS data from file |
| Clear CMOS date |
| Restore backup CMOS data |
|
| < Back to main menu |
|-----
```

Figura 9.105. Il menù ADVANCED.

```
-----
| Flash BIOS image WITHOUT bootblock |
| Write boot block to file |
| Flash bootblock to Flash ROM |
|
| < Back to main menu |
|-----
```

Tabella 9.106. Alcune opzioni.

Opzione	Descrizione
-H	Mostra la guida rapida all'uso del programma.
-?	
-E <i>file</i>	Sovrascrive la memoria «flash» con il contenuto del file indicato, senza fare alcuna richiesta di conferma.
-LOG	Registra le operazioni eseguite nel file 'UNIFLASH.LOG'.
-MONO	Serve per l'uso interattivo, allo scopo di non usare i colori, nel caso lo schermo non consenta una visualizzazione soddisfacente.
-CHIPLIST	Mostra l'elenco degli integrati di memoria che sono gestiti dal programma.
-SAVE <i>file</i>	Salva una copia del BIOS nel file indicato.

Opzione	Descrizione
-CMOSS <i>file</i>	Salva una copia della memoria CMOS nel file indicato.
-CMOSR <i>file</i>	Recupera una copia della memoria CMOS dal file indicato.

### 9.8.9 Recupero disperato di un BIOS

È possibile danneggiare il BIOS attraverso un'operazione di aggiornamento che fallisce, oppure sovrascrivendo dei dati errati o inadatti, ma oltre a questo, anche un virus (come qualunque altro tipo di programma malizioso) che ottenga dal sistema operativo i privilegi necessari, potrebbe cancellare o alterare la memoria del BIOS.

Danneggiare il BIOS significa perdere l'uso della scheda madre. Se ne vale la pena, a pagamento, si può chiedere aiuto al venditore per ottenere una copia del BIOS su un altro integrato, o sullo stesso integrato che è stato cancellato o modificato incautamente. Ci sono anche aziende specializzate che offrono un servizio di ripristino del BIOS a pagamento (per esempio <http://www.jumperz.de>). In alternativa, se si dispone di una scheda madre identica, si può valutare la possibilità di fare una copia in casa: attraverso gli strumenti appropriati alla scheda madre, si deve fare una copia del BIOS funzionante, quindi, con la scheda madre in funzione, occorre estrarre l'integrato buono e sostituirlo con quello da riprogrammare, per poi sovrascrivergli la copia del BIOS precedente.

Questo procedimento, noto come *hot-flash*, è molto rischioso, perché l'estrazione dell'integrato a scheda accesa, implica il pericolo di creare qualche corto circuito nel momento dell'intervento; inoltre, l'inserimento dell'integrato da riprogrammare, deve avvenire in modo esatto, perché invertendo l'orientamento lo si distrugge, a causa dell'alimentazione che giunge nei piedini sbagliati.

Disponendo di software appropriato, come potrebbe essere UniFlash, l'operazione di scrittura del BIOS a caldo può essere fatta anche con una scheda madre differente, purché utilizzi lo stesso integrato di memoria.

Non è il caso di elencare quali difficoltà si creano; è evidente che per una cosa del genere occorre esperienza e l'accettazione totale del rischio di danneggiare anche la scheda madre che ancora risulta funzionante. Pertanto, il procedimento viene menzionato in questa sezione, solo come idea, che però non è mai stata verificata in pratica dall'autore.

## 9.9 Modifica di un BIOS modulare

I BIOS più sofisticati sono «modulari», in quanto realizzati con componenti sostituibili in funzione delle caratteristiche della scheda madre.

Dal punto di vista dell'utente comune e anche da quello di chi aggiorna il BIOS della propria scheda madre, la modularità del BIOS non comporta alcuna differenza, perché si tratta al massimo di prelevare un BIOS aggiornato e completo dal sito della casa produttrice; tuttavia, la possibilità di sostituire un modulo, con qualcosa di non previsto, può diventare importante, soprattutto per rendere avviabile il sistema attraverso la rete, con una vecchia scheda madre che non sia già in grado di farlo.

Se l'aggiornamento del BIOS è un'operazione delicata e piena di rischi, la sostituzione dei moduli è un'operazione ancora più delicata e pericolosa. Pertanto, solo chi non ha paura di perdere l'uso della scheda madre si può cimentare in questo tipo di esperimenti.

### 9.9.1 BIOS modulari Award

Per intervenire nei moduli di un BIOS modulare Award, occorre fare una copia del BIOS in un file (solitamente con il programma AwdFlash, descritto nella sezione 9.8.6), quindi si modifica questo file attraverso strumenti appropriati. Normalmente si utilizza il programma Cbrom, oppure Awardmod che però può essere usato soltanto attraverso un sistema MS-Windows (a 32 bit). Si può tentare la ricerca di Cbrom o di Awardmod con Google: <http://www.google.com/search?q=award+cbrom.exe+download>, <http://www.google.com/search?q=award+awardmod.exe+download>.

La sintassi per l'utilizzo di Cbrom è complessa, in quanto non si presenta in modo logico. Viene mostrato un esempio, attraverso dei passaggi, che però non è generalizzato a sufficienza. Si tratta del tentativo di sostituire il logo EPA con un'immagine diversa (si osservi che se il file usato in sostituzione non è compatibile, si può ottenere un'alterazione dell'immagine dello schermo all'avvio).

```
A:\> CBROM215 [Invio]
```

In questo caso il file eseguibile si chiama 'CBROM215.EXE' e viene avviato senza argomenti per ottenere il riepilogo della sintassi:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.
Syntax:
A:\CBROM215.EXE InputFile [/other] [8000:0] [RomFile[Release[Extract]
A:\CBROM215.EXE InputFile [/D]logo[vga...]] [RomFile[Release[Extract]
InputFile : System BIOS to be added with Option ROMs
/D : For display all combined ROMs informations in BIOS
/epa|epal-7 : Add EPA LOGO BitMap to System BIOS
/logo|logol-7: Add OEM LOGO BitMap to System BIOS
/oem0-7 : Add special OEM ROM to System BIOS
/err : Return error code after executed
/btvga : Add VGA ROM to Boot Rom Block Area.
/isa : Add ISA BIOS ROM to System BIOS.(/isa Filename [xxxx:0])
/vga, /logo, /pci, /awdf flash, /cpucode, /epa, /acpitbl, /vsa, /hpm
/hpc, /fnt0 - 5, /ros, /nnprom, /mib, /group

RomFile : File name of option ROM to add-in
Release : Release option ROM in current system BIOS
Extract : Extract option ROM to File in current system BIOS

<<< Examples >>>
A:\CBROM215.EXE 2a4ib000.bin /D
A:\CBROM215.EXE 2a4ib000.bin /VGA filename
A:\CBROM215.EXE 2a4ib000.bin /VGA filename /ERR
```

```
A:\> CBROM215 BACKUP.BIN /D [Invio]
```

Si fa leggere al programma il file 'BACKUP.BIN', contenente una copia di un BIOS Award, allo scopo di elencarne le componenti:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

***** BACKUP.BIN BIOS component *****

No. Item-Name Original-Size Compressed-Size Original-File-Name
-----
0. System BIOS 20000h(128.00K) 13A20h(78.53K) original.tmp
1. XGROUP CODE 08C90h(35.14K) 05F4Eh(23.83K) awardext.rom
2. CPU micro code 02000h(8.00K) 017C9h(5.95K) CPUCODE.BIN
3. ACPI table 023C9h(8.95K) 00F31h(3.80K) ACPITBL.BIN
4. EPA LOGO 0168Ch(5.64K) 002AAh(0.67K) AwardBmp.bmp
5. YGROUP ROM 02760h(9.84K) 01B7Fh(6.87K) awardeyt.rom

Total compress code space = 33A20h(206.53K)
Total compressed code size = 1DE91h(119.64K)
Remain compress code space = 15B8Fh(86.89K)

** Micro Code Information **
Update ID CPUID | Update ID CPUID | Update ID CPUID | Update ID CPUID
-----
PPGA 11 0681 | PPGA 10 0683 | PPGA 08 0686 | PPGA 03 0665
```

```
A:\> CBROM215 BACKUP.BIN /EPA EXTRACT [Invio]
```

Richiede l'estrazione di una copia del modulo «EPA», che in pratica contiene il logo EPA. Si tratta di un file grafico in un formato speciale; il programma richiede di specificare un nome che comunque viene suggerito:

```
Enter an extract file name: (AwardBmp.bmp) [Invio]
```

Avendo confermato si ottiene il file con il nome suggerito: 'AWARDBMP.BMP'.

```
A:\> CBROM215 BACKUP.BIN /EPA RELEASE [Invio]
```

Rimuove il modulo «EPA» dal file 'BACKUP.BIN'.

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.
[EPA] ROM is release
```

• A:\> CBROM215 BACKUP.BIN /D[Invio]

Si fa rileggere al programma il file 'BACKUP.BIN', con la nuova situazione:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

***** BACKUP.BIN BIOS component *****

No. Item-Name      Original-Size      Compressed-Size  Original-File-Name
=====
0. System BIOS    20000h(128.00K)   13A20h(78.53K)   original.tmp
1. XGROUP CODE    08C90h(35.14K)    05F4Eh(23.83K)   awardext.rom
2. CPU micro code 02000h(8.00K)     017C9h(5.95K)    CPUCODE.BIN
3. ACPI table     023C9h(8.95K)    00F31h(3.80K)    ACPITBL.BIN
4. YGROUP ROM     02760h(9.84K)    01B7Fh(6.87K)    awardeyt.rom

Total compress code space = 33A20h(206.53K)
Total compressed code size = 1DBE7h(118.98K)
Remain compress code space = 15E39h(87.56K)

** Micro Code Information **
Update ID CPUID | Update ID CPUID | Update ID CPUID | Update ID CPUID
-----
PPGA 11 0681 | PPGA 10 0683 | PPGA 08 0686 | PPGA 03 0665
```

• A:\> CBROM215 BACKUP.BIN /EPA PINGU2.EPA[Invio]

Richiede la sostituzione del modulo «EPA» utilizzando il file 'PINGU2.EPA' (che deve essere del formato appropriato).

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.
Adding pingu2.epa 42.6%
```

• A:\> CBROM215 BACKUP.BIN /D[Invio]

Si fa rileggere al programma il file 'BACKUP.BIN', con la nuova situazione:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

***** BACKUP.BIN BIOS component *****

No. Item-Name      Original-Size      Compressed-Size  Original-File-Name
=====
0. System BIOS    20000h(128.00K)   13A20h(78.53K)   original.tmp
1. XGROUP CODE    08C90h(35.14K)    05F4Eh(23.83K)   awardext.rom
2. CPU micro code 02000h(8.00K)     017C9h(5.95K)    CPUCODE.BIN
3. ACPI table     023C9h(8.95K)    00F31h(3.80K)    ACPITBL.BIN
4. YGROUP ROM     02760h(9.84K)    01B7Fh(6.87K)    awardeyt.rom
5. EPA LOGO       004CBh(1.20K)     00231h(0.55K)    pingu2.epa

Total compress code space = 33A20h(206.53K)
Total compressed code size = 1DE18h(119.52K)
Remain compress code space = 15C08h(87.01K)

** Micro Code Information **
Update ID CPUID | Update ID CPUID | Update ID CPUID | Update ID CPUID
-----
PPGA 11 0681 | PPGA 10 0683 | PPGA 08 0686 | PPGA 03 0665
```

## 9.9.2 BIOS modulari AMI

<

Per intervenire nei moduli di un BIOS modulare AMI, occorre fare una copia del BIOS in un file (solitamente con il programma AwdFlash, descritto nella sezione 9.8.6), quindi si modifica questo file attraverso strumenti appropriati. Normalmente si utilizza il programma Amibcp, che richiede un sistema operativo Dos (si può tentare la ricerca di Amibcp con Google: <http://www.google.com/search?q=ami+amibcp.exe+download>).

## Riquadro 9.113. Problemi di compatibilità.

È da osservare che il programma Amibcp può creare dei problemi con alcune schede madri. Presso *AMIBCP Repository*, un tempo disponibile presso <http://www.stormpages.com/crazyape/amibcp.html>, appariva questa nota:

*The AMIBCP program contains a bug in is colour menu mode that will corrupt the BIOS on MSI motherboards (I'm sure other brands are an issue as well though I've not had any reports as yet), this bug does not occur when the program is used in command line mode. Use the following command from the command line to replace the built in module with the modified one, then flash your modified BIOS into the motherboard.*

```
A:\> AMIBCP bios.bin /R fullhack.bin 20 ↵
↵ /M 0D30 105A[Enter]
```

Si osservi, comunque, che l'esempio fa riferimento a un modulo particolare e i parametri indicati non sono validi in generale.

Vengono mostrate alcune schermate sul funzionamento di Amibcp nelle figure successive.

Figura 9.114. Inserimento iniziale del nome del file contenente il BIOS da modificare.



Figura 9.115. Selezionando la voce *Edit BIOS Modules* si passa all'elenco dei moduli del BIOS.

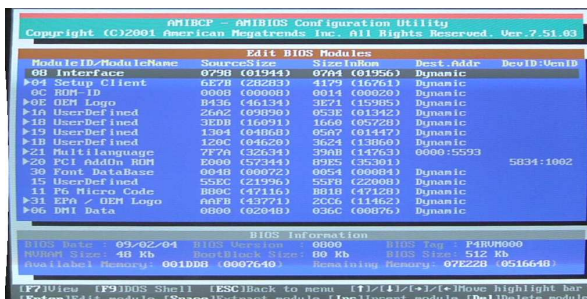


Figura 9.116. Premendo il tasto [Ins], si richiede l'aggiunta di un modulo.

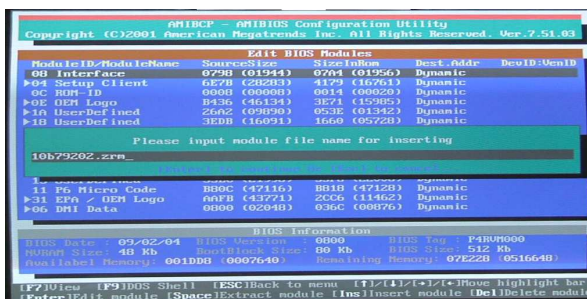


Figura 9.117. Dopo l'indicazione del modulo, viene richiesto di confermarne le caratteristiche. Si completa l'inserimento del modulo con il tasto [F10].



Figura 9.118. Al termine il modulo inserito appare nell'elenco e si può poi ritornare al menù iniziale (con il tasto [Esc]), da dove si può richiedere di salvare il file modificato.



## 9.10 Memoria centrale difettosa

Perché un sistema operativo possa funzionare correttamente, è necessario che la memoria centrale sia perfettamente efficiente, in quanto di norma non vengono applicate tecniche di ridondanza in questo ambito. Purtroppo, spesso la memoria centrale presenta invece dei difetti che si manifestano, in modo casuale, in alcune porzioni della stessa. Per fortuna, tra i parametri di avvio di un kernel Linux è possibile escludere le aree che hanno dimostrato di essere difettose. Naturalmente, per mettere alla prova la memoria centrale serve un programma diagnostico, da usare al di fuori del sistema operativo.

Va comunque osservato che l'utilizzo di un disco RAM iniziale o la dimensione particolarmente grande del file del kernel, può essere incompatibile con l'indicazione di alcune aree di memoria difettose.

### 9.10.1 Esclusione di altri fattori

Quando si verifica il funzionamento della memoria centrale, per esempio con programmi come quello descritto nella sezione successiva, gli errori che si possono riscontrare possono essere determinati da fattori estranei alla memoria stessa. Per esempio possono dipendere da difetti nella logica intermedia, tra la memoria e il microprocessore; possono dipendere da una configurazione errata nel BIOS; possono dipendere dallo sporco accumulato all'interno dell'elaboratore.

Quando si verifica la presenza di problemi con un programma diagnostico, la prima cosa che conviene fare è pulire l'elaboratore, per quanto possibile; soprattutto è bene rimuovere la memoria (ammesso che sia rimovibile) e pulire in particolare sia i moduli di memoria, sia i connettori dove questi vanno inseriti. Se dopo tale pulizia gli errori permangono, si può provare anche a scambiare i moduli tra i connettori, oppure si può provare a utilizzare qualche connettore che prima era libero. Se si dispone di altri elaboratori si può scambiare la memoria con quelli e ripetere l'esame diagnostico lì. Evidentemente, se in situazioni differenti si presentano gli stessi errori, oppure ne compaiono altri in zone vicine, è molto probabile che il difetto dipenda effettivamente dal modulo di memoria.

### 9.10.2 Memtest86+

Memtest86+<sup>4</sup> è un programma diagnostico per la memoria centrale che può funzionare con la maggior parte degli elaboratori x86. Il sito originale di distribuzione Memtest86+ è <http://www.memtest.org/>.

Questo programma, per eseguire il proprio lavoro, deve essere avviato autonomamente, al di fuori del sistema operativo, per esempio con GRUB 1 o sistemi di avvio simili, oppure da un dischetto o da un CD/DVD autoavviabile.

Si osservi che Memtest86+ richiede pochissima memoria per funzionare, quindi è possibile eseguire la verifica anche con moduli singoli, in modo da escludere i problemi causati dall'incompatibilità reciproca di moduli di memoria differenti.

Generalmente sono disponibili dei file-immagine già pronti, sia per dischetto, sia per CD/DVD. Eventualmente si può estrapolare facilmente il file binario 'memtestp.bin' e lo si può far partire, per esempio con GRUB 1, usando, nel file 'menu.lst', direttive simili a quelle seguenti:

```
...
title Memtest86+
kernel (hd0,0)/boot/memtestp/memtestp.bin
...
```

Nel caso di altri sistemi di avvio, può darsi che l'uso diretto del file 'memtestp.bin' non sia possibile; in tal caso, si può predisporre un file-immagine di un dischetto, dove il contenuto di 'memtestp.bin' è riportato, tale e quale, nei primi settori (dal primo in poi). Ciò è necessario in particolare con SYSLINUX. Supponendo di avere realizzato il file-immagine 'memtestp.dsk', il file di configurazione di SYSLINUX potrebbe contenere le direttive seguenti, dove va osservato l'uso particolare del file 'memdisk' (si tratta di un modulo di SYSLINUX) che deve essere presente nella stessa directory:

```
LABEL mem
KERNEL memdisk
APPEND initrd=memtestp.dsk
```

Appena avviato, Memtest86+ inizia a scrivere e a rileggere la memoria con vari tipi di modelli, segnalando gli indirizzi dove la lettura riporta un risultato diverso da quanto scritto. Gli indirizzi vengono indicati in esadecimale, con una traduzione approssimativa in mebi-byte (simbolo Mibyte) che però non è esatta nella porzione decimale. Memtest86+ continua a funzionare senza sosta fino a quando l'elaboratore rimane acceso, oppure quando viene premuto il tasto [Esc] che provoca il riavvio del sistema. Eventualmente sono disponibili anche altri comandi, annotati alla base della schermata.

A titolo di esempio, potrebbe essere riportato un elenco di errori come quello seguente, nell'ambito di una memoria da 512 Mibyte:

```
0D05DF2816, 0D05DF6816, 0D05E32816, 0D05E42816,
0E05DCAC16, 0E05DF0816, 0E05DF4816, 0E05E30816, 0E05E40816,
1CBBFFBC16
```

Dal momento che Memtest86+ ripete continuamente la propria ricerca, è normale che gli indirizzi difettosi compaiano in modo disordinato; di conseguenza il riordino va fatto a mano.

Gli indirizzi rappresentano i byte che in memoria hanno manifestato il problema. Per utilizzare queste informazioni occorre mettersi a tavolino e tradurre i valori in base dieci (un programma come Gbase è ottimo per questo scopo) e poi può essere necessario calcolare gli spazi utilizzabili, ma considerando che conviene prendersi un po' larghi e prevedere che gli errori possono presentarsi in futuro in celle di memoria vicine a quelle che sono già state rilevate. Per esempio, dati i valori già ipotizzati, si potrebbe preparare una tabellina come quella seguente:



Indirizzo in base sedici	Indirizzo in base dieci	Indirizzo 1048576 (indirizzo mebibyte) diviso in
0D05DF28 <sub>16</sub>	218488616	208,37
0D05DF68 <sub>16</sub>	218488680	208,37
0D05E328 <sub>16</sub>	218489640	208,37
0D05E428 <sub>16</sub>	218489896	208,37
0E05DCAC <sub>16</sub>	235265196	224,37
0E05DF08 <sub>16</sub>	235265800	224,37
0E05DF48 <sub>16</sub>	235265864	224,37
0E05E308 <sub>16</sub>	235266824	224,37
0E05E408 <sub>16</sub>	235267080	224,37
1CBBFFBC <sub>16</sub>	482082748	459,75

A questo punto si comprende che gli errori si addensano negli intervalli 208-209 Mibyte, 224-225 Mibyte e 459-460 Mibyte.

### 9.10.3 Parametri di avvio del kernel

Per dire al kernel Linux che deve utilizzare solo una certa porzione della memoria centrale, si usa il parametro `memmap=exactmap`, seguito da altri parametri del tipo:

```
memmap=lunghezza [K|M|G]@inizio [K|M|G]
```

Per esempio, per dire al kernel di utilizzare lo spazio dei primi 640 Kibyte e poi di partire dal primo mebibyte (simbolo Mibyte) per arrivare alla fine dei 512 Mibyte (saltando quindi lo spazio successivo ai 640 Kibyte e precedente al primo mebibyte), si deve scrivere una cosa così:

```
memmap=exactmap memmap=640K@0 memmap=511M@1M
```

Precisamente, ciò significa che deve essere utilizzato lo spazio che inizia dall'indirizzo zero e si estende in avanti per 640 Kibyte, quindi si deve prendere quello che inizia da 1 Mibyte e si estende per altri 511 Mibyte. Come si comprende, anche per questo occorre mettersi a tavolino e fare bene i calcoli. Riprendendo l'esempio già descritto nella sezione precedente, la tabellina potrebbe essere estesa così:

Indirizzo in base sedici	Indirizzo in base dieci	Indirizzo in mebibyte	Parametro del kernel
0D05DF28 <sub>16</sub>	218488616	208,37	memmap=207@1M
0D05DF68 <sub>16</sub>	218488680	208,37	
0D05E328 <sub>16</sub>	218489640	208,37	
0D05E428 <sub>16</sub>	218489896	208,37	
0E05DCAC <sub>16</sub>	235265196	224,37	memmap=15M@209M
0E05DF08 <sub>16</sub>	235265800	224,37	
0E05DF48 <sub>16</sub>	235265864	224,37	
0E05E308 <sub>16</sub>	235266824	224,37	
0E05E408 <sub>16</sub>	235267080	224,37	
1CBBFFBC <sub>16</sub>	482082748	459,75	memmap=234M@225M
			memmap=52M@460M

Per esempio con GRUB 1, nel file di configurazione per l'avvio si potrebbe scrivere così:

```
...
title memoria difettosa
kernel (hd0,0)/vmlinuz root=/dev/hd1 ro memmap=exactmap ←
↔ memmap=207@1M memmap=15M@209M memmap=234M@225M ↔
↔ memmap=52M@460M
...
```

Come si vede, per semplicità è stata omessa anche la porzione di memoria dei primi 640 Kibyte.

## 9.11 Restauro

Il recupero di un vecchio elaboratore richiede prima di tutto che questo venga smontato, ripulito e riparato per quanto possibile. È molto probabile che si svolgano queste operazioni senza la disponibilità di una documentazione adeguata, pertanto occorre annotare tutto ciò che serve a rimontare i vari componenti, prima di procedere alla loro separazione. È molto importante annotare anche la configurazione dei ponticelli sulla scheda madre ed eventualmente sulle altre schede presenti.

### 9.11.1 Ripulitura

La ripulitura di un vecchio elaboratore va eseguita normalmente con un aspirapolvere, usando poca potenza, con l'aiuto eventuale di aria compressa. Tuttavia, se si tratta di sporco grasso, come può essere il deposito di catrame creato dal fumo, non resta altro che il lavaggio letterale di quasi tutti i componenti. Il lavaggio è comunque un'operazione rischiosa, pertanto va fatto solo quando non ci sono alternative ragionevoli migliori.

Per lavare i componenti di un elaboratore si può usare una lavastoviglie, programmando un ciclo breve a 50 gradi centigradi. Per il lavaggio si usano le stesse sostanze che servono normalmente (sale per lavastoviglie, brillantante, detersivo).

Non si possono lavare le unità a disco e a nastro (eccezionalmente, le unità a dischetti magnetici da 9 cm si possono lavare, come viene descritto in una sezione apposita) e in generale occorre avere il buon senso di capire cosa non si può infilare in lavastoviglie.

I componenti che dispongono di una batteria rimovibile, specie le schede madri con le note batterie a bottone, devono essere privati della batteria stessa, perché si esaurirebbe inutilmente, producendo immediatamente dell'acido che danneggia inizialmente l'alloggiamento della batteria, per poi espandersi sulla scheda madre. Per lo stesso motivo, indipendentemente dal fatto che si lavino o no le schede in questione, le batterie vanno tolte quando i componenti devono essere conservati inutilizzati per qualche tempo.

Terminato il ciclo di lavaggio, i componenti vanno estratti dalla lavastoviglie e posti in un luogo dove possano rimanere ad asciugare per un paio di giorni. Prima di essere riposti ad asciugare, conviene comunque scuoterli per fare uscire l'acqua rimasta imprigionata nelle fessure.

Figura 9.124. Un alimentatore, prima e dopo il lavaggio; quello sulla destra è ancora umido.

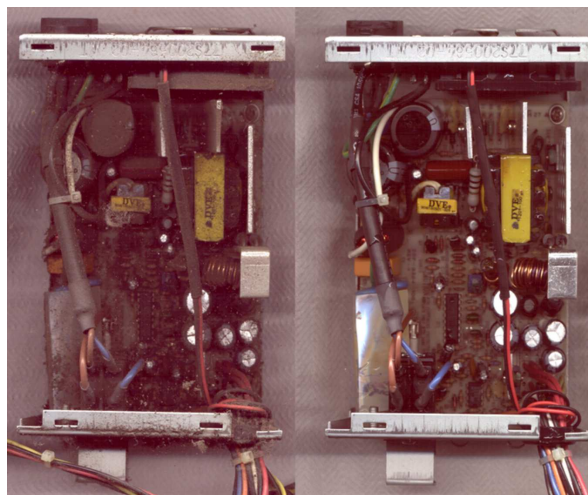


Figura 9.125. Una scheda di rete, prima e dopo il lavaggio; quella sulla destra è ancora umida.

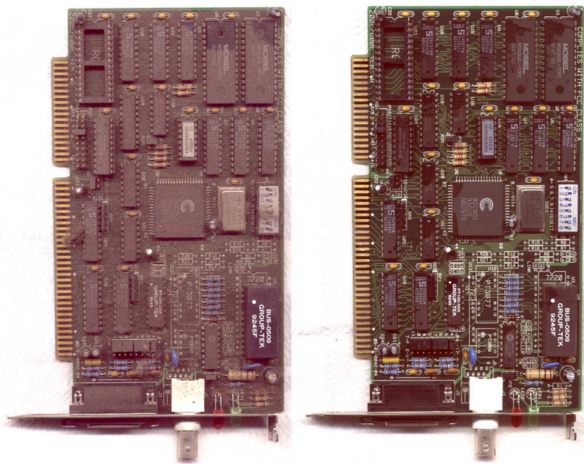
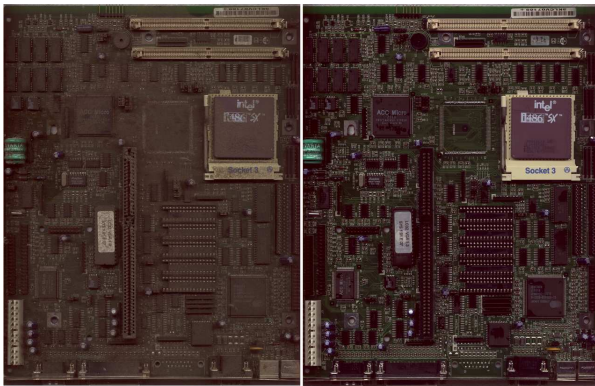


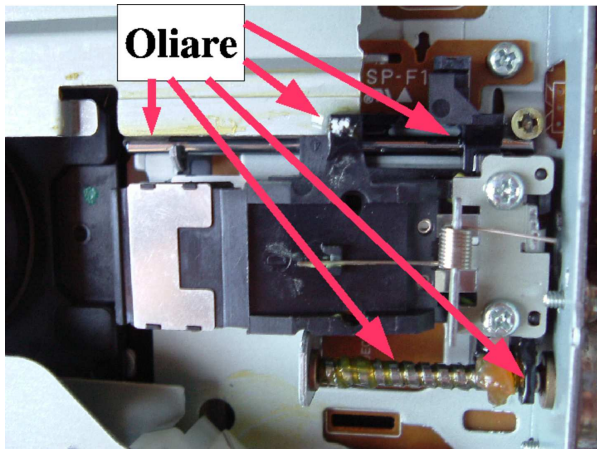
Figura 9.126. Una scheda madre, prima e dopo il lavaggio; quella sulla destra è ancora umida.



### 9.11.2 Lavaggio di unità a dischetti

In generale, le unità di memorizzazione a disco non possono essere lavate. Eccezionalmente, le vecchie unità a dischetti da 5,25 in e da 3,5 in (9 cm) possono essere lavate, avendo l'accortezza, dopo il lavaggio, di oliare o ingrassare alcune parti, come si vede nella figura successiva, usando olio o grasso adatti (per esempio, olio o grasso di vaselina, puri).

Figura 9.127. I punti da oliare o ingrassare. Si deve avere l'accortezza di non esagerare ed evitare che le testine entrino in contatto con queste sostanze.



Riquadro 9.128. Perché non si possono lavare le unità a CD o DVD.

Le unità ottiche (o magneto-ottiche), come per esempio i lettori o masterizzatori CD e DVD, non possono essere lavati, perché le lenti sono rivestite da una sostanza che viene rimossa con il lavaggio. Evidentemente la presenza di questa sostanza è indispensabile, perché dopo il lavaggio le unità, pur funzionando meccanicamente, non leggono più i dischi.

### 9.11.3 Pulizia dei monitor

I monitor a colori a tubo sono molto difficili da smontare, perché la loro costruzione richiede un intreccio preciso di fili per il controllo dei campi magnetici; tuttavia si può fare ugualmente una piccola pulizia togliendo il coperchio posteriore.

Si osservi che per operare con un monitor occorre maggiore prudenza, sia per la presenza del tubo in sé, sia per il fatto che al suo interno si producono tensioni molto elevate, che possono stazionare anche dopo lo spegnimento e il distacco del cavo di alimentazione dalla rete elettrica. Pertanto, prima di intervenire sul gruppo di alta tensione conviene iniziare toccando con un cacciavite (naturalmente deve avere il manico ben isolato), osservando se si formano piccole scariche elettrostatiche.

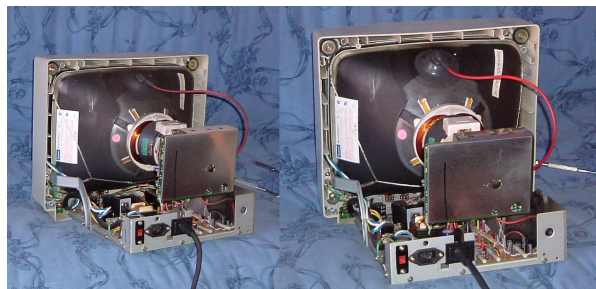
Figura 9.129. Lavaggio del coperchio posteriore di un monitor, con l'aiuto di una doccia. Sulla sinistra si vede la sporcizia maggiore accumulata in corrispondenza del cavo di alta tensione.



Una volta smontato il coperchio posteriore, il suo lavaggio è la cosa più semplice da fare, che si può eseguire in una vasca da bagno o su un piatto doccia. Per la pulizia si possono usare prodotti per la casa di uso comune, purché se ne conosca l'uso.

Per la pulizia di materiali plastici e di materiali in gomma, **non si può usare varechina** o prodotti simili che contengono composti del cloro, perché la gomma si scioglierebbe irrimediabilmente.

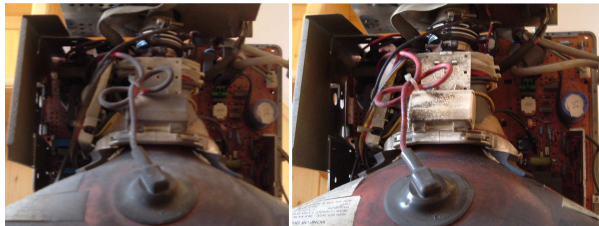
Figura 9.130. Pulizia all'interno del monitor. Sulla sinistra si vede il cavo di alta tensione (quello rosso) e altri componenti in plastica di colore chiaro velati da uno strato di polvere (attaccata elettrostaticamente); sulla destra, lo stesso monitor dopo una pulizia superficiale del cavo e delle superfici accessibili senza danneggiare i componenti elettronici.



La pulizia dell'interno del monitor può essere fatta con uno straccio un po' umido, ma **solo dopo aver verificato che non ci sono cariche elettrostatiche**, con l'aiuto di cacciavite ben isolato. La pulizia del cavo di alta tensione può fare emergere dei difetti, come punti di

scarica, che si mostrano con un puntino nero. A ogni modo, la pulizia delle parti di alta tensione accessibili riduce le perdite di energia, cosa che dovrebbe tradursi in un'immagine migliore.

Figura 9.131. Pulizia di un monitor molto sporco (a sinistra), dove il risultato è poco soddisfacente, anche a causa del fatto che probabilmente si è insinuato dell'inchiostro in polvere proveniente da una stampante laser usata in prossimità del monitor stesso.



Dove non si può arrivare con lo straccio, o semplicemente non lo si può usare, si può fare un tentativo con un pennello morbido, anche se questo significa riempire di polvere l'ambiente circostante.

#### 9.11.4 Pulizia azzardata del monitor

Nel caso non ci fossero alternative, si può tentare il recupero di un monitor a tubo, attraverso un lavaggio, da attuare però con l'ausilio di una doccia. Si tratta comunque di un lavoro che difficilmente può produrre un risultato positivo. Va comunque fatta molta attenzione all'asciugatura, la quale deve essere perfetta, altrimenti il generatore di alta tensione si distruggerebbe alla prima accensione.

Nelle figure successive si vedono le fasi del lavaggio di un monitor a tubo, molto grande. L'intento del lavaggio era la rimozione di tutte le particelle solide che si erano insinuate ovunque, le quali potevano produrre la perdita di tensione sull'anodo del tubo RC. Tuttavia, anche se alla fine il monitor si è acceso, il problema non è stato risolto e, al contrario, si è aggravato leggermente.

Figura 9.132. Tentativo di lavaggio di un monitor.

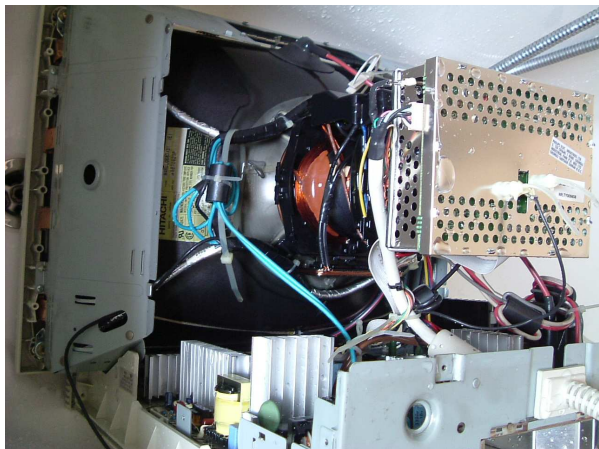
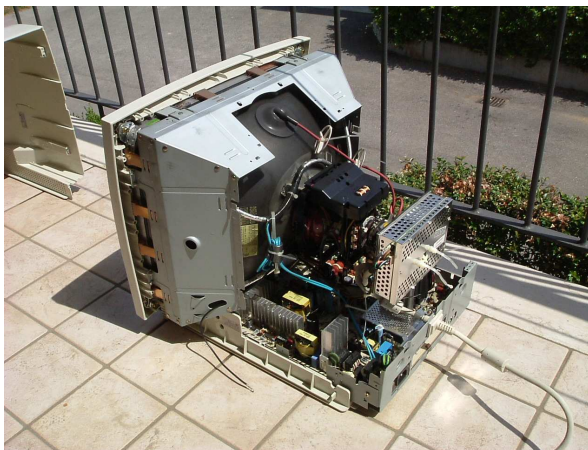


Figura 9.133. Asciugatura al sole.



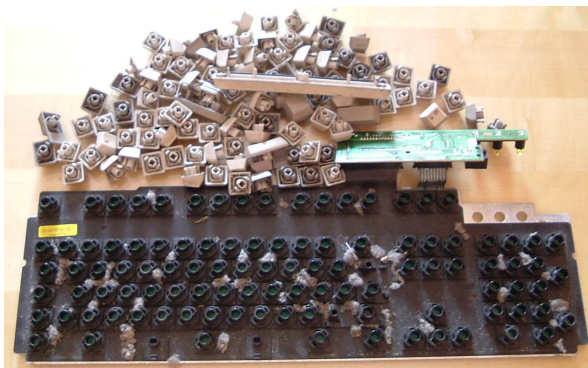
#### 9.11.5 Pulizia della tastiera

La pulizia della tastiera richiede un lavoro piuttosto noioso, perché, ammesso che ciò sia possibile, richiede la rimozione di tutti i tasti e la pulizia di questi uno per uno.

Figura 9.134. Una tastiera Olivetti degli anni 1990, prima della pulizia, dopo avere sollevato la cornice superiore.

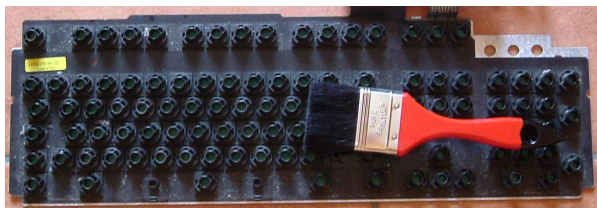


Figura 9.135. Una tastiera Olivetti degli anni 1990, prima della pulizia, dopo l'estrazione di tutti i tasti. Viene messo in evidenza quanta sporcizia si accumuli sotto i tasti.



Una volta estratti i tasti dalla tastiera, conviene passare a una pulizia a secco, eventualmente con un pennello morbido. In questo caso non si può usare l'acqua, se non per le parti esterne in plastica, come la cornice superiore e la base inferiore della tastiera, perché l'acqua questa potrebbe danneggiare la membrana in gomma della tastiera, ma soprattutto perché sarebbe molto difficile asciugare perfettamente questa parte.

Figura 9.136. Un pennello morbido è più che sufficiente per la pulizia di questo tipo di tastiera.



Una volta puliti i componenti esterni (probabilmente con l'aiuto di una doccia e con un detergente sgrassatore adatto), dopo averli asciugati si passa a rimontare la tastiera, pulendo tasto per tasto prima di inserirlo nuovamente nella sua sede.

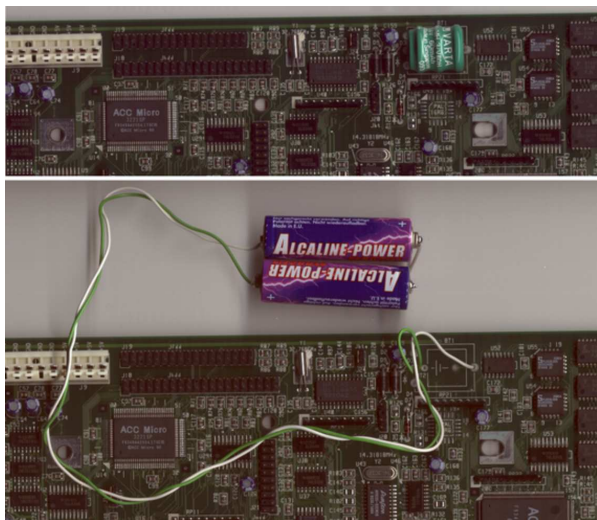
Figura 9.137. Ricostruzione della tastiera.



### 9.11.6 Batteria

Dopo qualche anno, la batteria che mantiene la memoria C/MOS deve essere sostituita, anche se si tratta di un accumulatore. Se la batteria è saldata, occorre usare un saldatore a stagno con una punta sottile e probabilmente è necessario predisporre una batteria alternativa, che offra una tensione **non superiore** a quella della batteria estratta.

Figura 9.138. Sostituzione di una batteria, originariamente da 3,6 V, con una coppia di batterie stilo da 1,5 V, messe in serie (per un totale di 3 V).



Dal momento che le batterie che si vanno a mettere non sono ancorate alla scheda madre, è necessario isolarle e fissarle da qualche parte nel contenitore dell'elaboratore. La figura mostra delle batterie non isolate per fare comprendere esattamente il procedimento.

### 9.11.7 Osservazioni sulla configurazione BIOS di elaboratori molto vecchi

La configurazione del BIOS attraverso una memoria C/MOS, degli elaboratori con architettura x86, viene introdotta precisamente dai

microprocessori i286. Originariamente questa memoria veniva alimentata da un piccolo accumulatore NiCd (nichel-cadmio), destinato comunque a una rapida distruzione. Purtroppo, il programma di configurazione del BIOS dei primi elaboratori di questo tipo, non faceva parte del BIOS stesso, oppure le funzioni incorporate erano estremamente limitate.

Le tabelle successive riepilogano i comportamenti di alcuni elaboratori o di BIOS più noti. Come si può osservare esiste una grande quantità di possibilità e sicuramente i casi non si esauriscono a quelli descritti. Nelle situazioni in cui i vari tentativi falliscono, si può cercare di provocare una situazione per cui è il BIOS stesso a suggerire la configurazione: staccando la tastiera ed eventualmente anche tutte le unità, escluso ciò che serve per poter visualizzare qualcosa sullo schermo.

Tabella 9.139. Accesso alla configurazione del BIOS in base alla combinazione di tasti.

Comando	Elaboratore o BIOS tipico	Note
[ Canc ]	BIOS AMI e BIOS Award	
[ Esc ]	BIOS DTK	
[ Ins ]	IBM	
[ F1 ]	IBM	
[ F2 ]	Packard Bell	
[ F10 ]	Compaq	Si preme il tasto quando appare un rettangolino lampeggiante all'angolo superiore destro dello schermo. Il programma per la modifica della configurazione risiede probabilmente nel disco fisso.
[ Ctrl Ins ]	IBM	Quando un cursore appare all'angolo superiore destro dello schermo.
[ Ctrl s ]	BIOS Phoenix	
[ Ctrl Alt ]	IBM	
[ Ctrl Alt Invio ]	Dell	
[ Ctrl Alt Esc ]	AST	
[ Ctrl Alt Ins ]	Zenith	
[ Ctrl Alt ? ]	IBM	
[ Ctrl Alt s ]	BIOS Phoenix	
[ Ctrl Alt a ]	Fortiva	
[ Mautscole Ctrl Esc ]	Tandon	
[ Ctrl Alt u ]	Olivetti M4	Il programma per la modifica della configurazione risiede probabilmente nel disco fisso.

Tabella 9.140. Accesso alla configurazione del BIOS in base al nome del BIOS.

BIOS	Comando
AMI	[ Canc ], [ F1 ], [ F2 ]
Award	[ Canc ], [ Ctrl Alt Esc ]
DTK	[ Esc ]
Phoenix	[ F1 ], [ F2 ], [ Ctrl s ], [ Ctrl Alt Esc ], [ Ctrl Alt Ins ], [ Ctrl Alt s ]
Opus	[ Ctrl Alt Esc ]

Tabella 9.141. Accesso alla configurazione del BIOS in base al nome dell'elaboratore.

Elaboratore	Comando	Note
Acer	[ Ctrl Alt Esc ]	
ALR	[ Ctrl Alt Esc ], [ F2 ]	
Aptiva	[ F1 ]	
AST	[ F2 ], [ Ctrl Alt Esc ]	
Compaq	[ F10 ]	Si deve premere il tasto quando appare un rettangolino lampeggiante all'angolo superiore destro dello schermo. Il programma di configurazione è esterno.
Dell	[ F1 ], [ Canc ], [ Ctrl Alt Invio ]	Alcuni modelli richiedono la pressione del tasto di riavvio (reset) per due volte.
Fortiva	[ Ctrl Alt a ]	
HP	[ F1 ], [ F2 ]	
IBM	[ F1 ], [ Ctrl Alt Ins ], [ Ins ], [ Ctrl Alt ], [ Ctrl Ins ], [ Ctrl Alt ? ]	Nel caso della combinazione [ Ctrl Ins ], questa va usata quando il cursore si posiziona all'angolo superiore destro dello schermo.
NEC	[ F2 ]	

Elaboratore	Comando	Note
Olivetti	[Mauscole Ctrl Alt] [Ctrl Alt u]	Il programma per la modifica della configurazione risiede probabilmente nel disco fisso.
Packard Bell	[F1], [F2]	
Sharp	[F2]	
Sony	[F3]	
Tandon	[Mauscole Ctrl Esc], [Ctrl Alt Esc]	
Toshiba	[Esc], [F1]	
Zenith	[Ctrl Alt Ins]	

La configurazione contenuta nella memoria C/MOS cambia in base al BIOS, ovvero in base alle caratteristiche del firmware. Quando il BIOS non contiene anche il programma per accedere alla modifica della configurazione, questo può essere stato distribuito assieme al sistema operativo (si tratta normalmente del Dos), oppure può risiedere in una partizione speciale del disco fisso.

L'utilizzazione di una porzione del disco fisso per collocare il programma di modifica è tipico degli elaboratori Olivetti e Compaq.

In mancanza di altro, quando c'è la necessità di intervenire nella configurazione del BIOS, ci si può arrischiare di azzerare completamente la memoria C/MOS, attraverso l'uso di qualche programma che consenta di farlo. Eventualmente, anche un programma che salva e recupera la configurazione può essere usato per lo scopo, modificando il file con un programma che consenta di intervenire in modo binario (di solito in esadecimale). In questo modo si spera che il BIOS attribuisca almeno una configurazione predefinita; d'altra parte, questo è ciò che succede quando si scarica la batteria, anche se l'azzeramento non porta necessariamente a buoni risultati.

Per modificare o azzerare la configurazione del BIOS si potrebbe tentare l'uso di Cmoser<sup>5</sup> che è disponibile presso <ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/cmose11.zip> ed eventualmente anche altri siti (<http://www.google.com/search?q=cmose11.zip>).

Il BIOS degli elaboratori più vecchi non è in grado di individuare da solo la geometria del disco fisso ATA. Se non c'è la possibilità di avviare un sistema GNU/Linux allo scopo di avere questa informazione, si può usare il programma Dug\_ide,<sup>6</sup> disponibile presso <ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/dugide21.zip> ed eventualmente anche attraverso altri siti (<http://www.google.com/search?q=dugide21.zip>).

Oltre che salvare il contenuto della configurazione, può essere utile fare una copia del BIOS in un file, da conservare da qualche parte. Teoricamente, quando la memoria ROM programmabile dovesse alterarsi, dovrebbe essere possibile reperire un programmatore di ROM in grado di riprodurne una copia fisica. A tale proposito possono essere utili due programmi: Bios,<sup>7</sup> il quale dovrebbe essere disponibile presso <http://www.bockelkamp.de/software/discontinued/bios1351.zip>, e Biosutil,<sup>8</sup> disponibile presso [http://www.drivermuseum.com/files/utl/bios\\_u.html](http://www.drivermuseum.com/files/utl/bios_u.html).

## 9.12 Connettori e cablaggi

### 9.12.1 Video MGA, CGA, EGA, Hercules

Figura 9.142. Schema di collegamento del connettore DIN a sei poli di un monitor monocromatico.

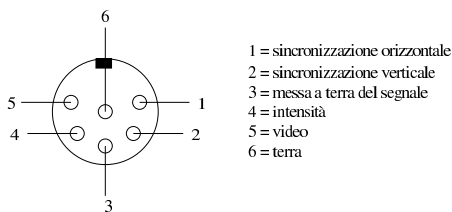


Figura 9.143. Schema del cavo di adattamento da connettore DIN a sei poli a connettore DB-9 di una scheda video grafica EGA (o CGA o Hercules).

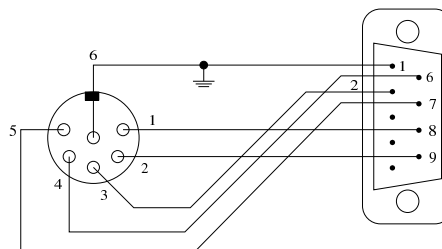
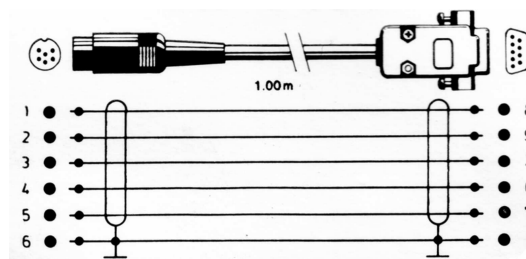
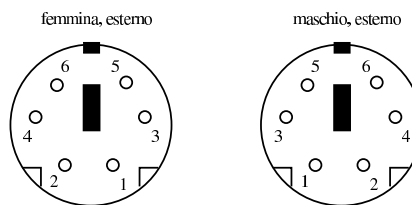


Figura 9.144. Lo schema riportato in una vecchia confezione di un cavo di adattamento.



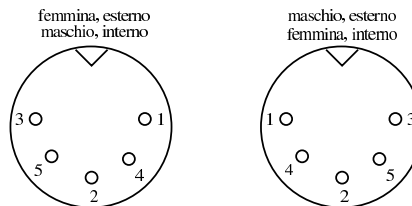
### 9.12.2 Tastiere

Figura 9.145. Connettore PS/2, ovvero «mini-DIN», di una tastiera.



Terminale	Denominazione standard	Annotazioni
1	keyboard data	Il filo potrebbe essere di colore bianco.
2	keyboard reset	Non utilizzato e non collegato.
3	ground	Il filo potrebbe essere di colore bianco giallo ed è separato elettricamente dalla schermatura esterna.
4	keyboard power supply	Il filo potrebbe essere di colore bianco rosso e viene fornita una tensione di +5 V.
5	keyboard clock	Il filo potrebbe essere di colore verde.
6	--	Non utilizzato.

Figura 9.146. Connettore AT, ovvero «DIN», di una tastiera.



Terminale	Denominazione standard	Annotazioni
1	keyboard clock	
2	keyboard data	
3	keyboard reset	Non utilizzato e non collegato.
4	ground	
5	keyboard power supply	Viene fornita una tensione di +5 V.

Tabella 9.147. Il cavo di una tastiera «AT» che termina con un connettore «DIN», normalmente, all'altra estremità è collegato al circuito stampato attraverso un connettore rettangolare a sei poli, secondo questa numerazione.

Terminale	Denominazione standard	Annotazioni
1	keyboard clock	
2	keyboard data	
3	--	Non utilizzato.
4	--	Non utilizzato.
5	ground	
6	keyboard power supply	+5 V.

9.12.3 Porte seriali

« Per realizzare un cavo Null-modem che permetta la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale utilizzando un controllo di flusso software, ovvero XON/XOFF, sono sufficienti tre fili. Nello schema della tabella 9.148 sono rappresentate le diverse possibilità di collegamento a seconda che si utilizzino connettori DB-25 o DB-9.

Tabella 9.148. Cavo seriale a tre fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
na	na	na	na	na	na
2	3	2	2	2	3
3	2	3	3	3	2
7	7	7	5	5	5

Per realizzare un cavo Null-modem che permetta la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale utilizzando un controllo di flusso hardware, ovvero RTS/CTS, sono necessari sette fili. Nello schema della tabella 9.149 sono rappresentate le diverse possibilità di collegamento a seconda che si utilizzino connettori DB-25 o DB-9 femmina.

Tabella 9.149. Cavo seriale a sette fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
na	na	na	na	na	na
2	3	2	2	3	2
3	2	3	3	2	3
4	5	4	8	7	8
5	4	5	7	8	7
6+8	20	6+8	4	6+1	4
20	6+8	20	6+1	4	6+1
7	7	7	5	5	5

Figura 9.150. Disposizione dei contatti di un connettore BD-25 femmina.

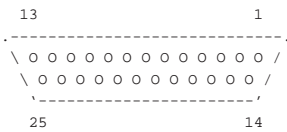
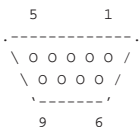


Figura 9.151. Disposizione dei contatti di un connettore BD-9 femmina.



9.12.4 Porte parallele

« Il cavo parallelo incrociato, detto anche Null-printer o Laplink o Interlink, è utilizzabile in particolare per le connessioni PLIP. Permette il collegamento attraverso porte parallele normali. L'eventuale scher-

matura metallica può essere collegata alla massa del connettore, ma solo a uno dei due capi. Controllare la documentazione contenuta nel file `'sorgenti_linux/drivers/net/README.PLIP'`.

Tabella 9.152. Cavo parallelo incrociato.

Connettore A DB-25 maschio			Connettore B DB-25 maschio
Nome	Contatto	Contatto	Nome
Data Bit 0	2	15	Error
Data Bit 1	3	13	Select
Data Bit 2	4	12	Paper Out
Data Bit 3	5	10	Acknowledge
Data Bit 4	6	11	Busy
Acknowledge	10	5	Data Bit 3
Busy	11	6	Data Bit 4
Paper Out	12	4	Data Bit 2
Select	13	3	Data Bit 1
Error	15	2	Data Bit 0
Signal Ground	25	25	Signal Ground

Il cavo parallelo bidirezionale **non è più utilizzato** con GNU/Linux: permetteva il collegamento attraverso porte parallele bidirezionali con velocità di comunicazione maggiori rispetto a un cavo incrociato normale.

La connessione di un cavo bidirezionale su elaboratori accesi comporta qualche rischio in più rispetto alla connessione di un cavo normale. Sotto questo aspetto, l'uso di un cavo del genere, anche se potrebbe fornire prestazioni doppie rispetto a un cavo incrociato normale, è **generalmente sconsigliabile**.

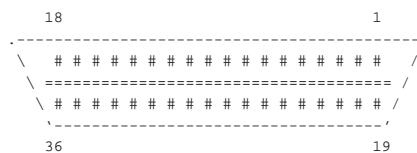
Tabella 9.153. Cavo parallelo bidirezionale **non più utilizzato** con GNU/Linux.

Connettore A DB-25 maschio			Connettore B DB-25 maschio
Nome	Contatto	Contatto	Nome
Strobe	1	11	Busy
Data Bit 0	2	2	Data Bit 0
Data Bit 1	3	3	Data Bit 1
Data Bit 2	4	4	Data Bit 2
Data Bit 3	5	5	Data Bit 3
Data Bit 4	6	6	Data Bit 4
Data Bit 5	7	7	Data Bit 5
Data Bit 6	8	8	Data Bit 6
Data Bit 7	9	9	Data Bit 7
Busy	11	1	Strobe
Paper Out	12	14	Autofeed
Select	13	17	Select
Autofeed	14	12	Paper Out
Error	15	18	Ground
Reset	16	10	Acknowledge
Ground	18	15	Error
Signal Ground	25	25	Signal Ground

Figura 9.154. Disposizione dei contatti di un connettore BD-25 maschio.



Figura 9.155. Disposizione dei contatti di un connettore Centronics a 36 poli (stampante).

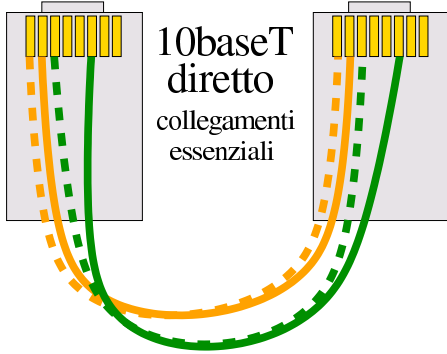


9.12.5 Reti Ethernet

Figura 9.156. Connettori RJ-45 (ISO 8877).

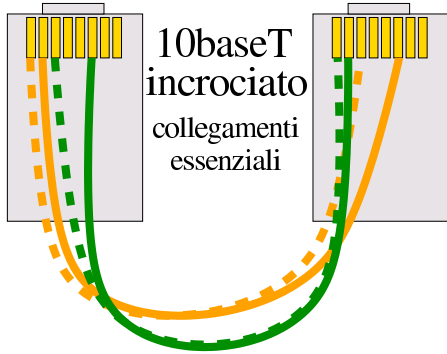


Figura 9.157. Cavo Ethernet 10baseT diretto. Le coppie 1-2 e 3-6 sono ritorte.



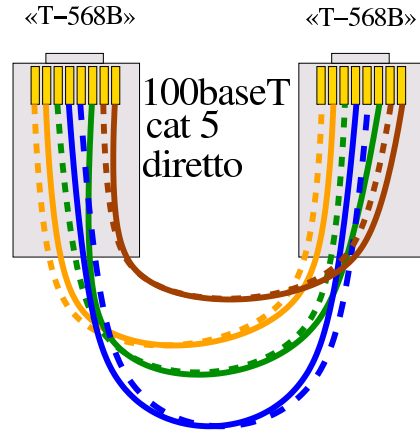
Spina RJ-45		Colore filo		Spina RJ-45	
Nome	Contatto		Contatto	Nome	
TxData +	1	bianco/arancio	1	TxData +	
TxData -	2	arancio	2	TxData -	
RxData +	3	bianco/verde	3	RxData +	
RxData -	6	verde	6	RxData -	

Figura 9.158. Cavo Ethernet 10baseT incrociato per connettere due sole stazioni senza altri componenti intermedi. Le coppie 1-2:3-6 e 3-6:1-2 sono ritorte.



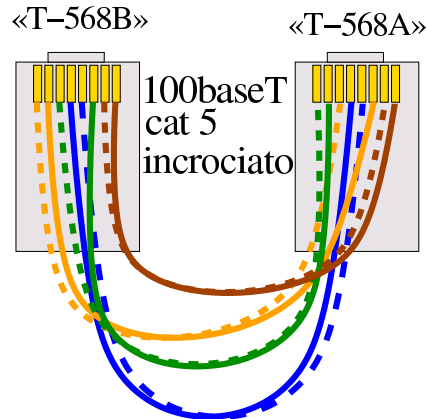
Spina RJ-45		Colore filo		Spina RJ-45	
Nome	Contatto		Contatto	Nome	
TxData +	1	bianco/arancio	3	RxData +	
TxData -	2	arancio	6	RxData -	
RxData +	3	bianco/verde	1	TxData +	
RxData -	6	verde	2	TxData -	

Figura 9.159. Cavo Ethernet 100baseT categoria 5 diretto. Le coppie 1-2, 3-6, 4-5 e 7-8 sono ritorte.



Spina RJ-45 (T-568B)		Colore filo		Spina RJ-45 (T-568B)	
Nome	Contatto		Contatto	Nome	
tx-d1+	1	arancio/bianco	1	tx-d1+	
tx-d1-	2	arancio	2	tx-d1-	
rx-d2+	3	bianco/verde	3	rx-d2+	
bi-d4+	4	blu	4	bi-d4+	
bi-d4-	5	blu/bianco	5	bi-d4-	
rx-d2-	6	verde	6	rx-d2-	
bi-d3+	7	marro-ne/bianco	7	bi-d3+	
bi-d3-	8	marrone	8	bi-d3-	

Figura 9.160. Cavo Ethernet 100baseT categoria 5 incrociato. Le coppie 1-2:3-6, 3-6:1-2, 4-5:7-8 e 7-8:4-5 sono ritorte.



Spina RJ-45 (T-568B)		Colore filo		Spina RJ-45 (T-568A)	
Nome	Contatto		Contatto	Nome	
tx-d1+	1	arancio/bianco	3	rx-d2+	
tx-d1-	2	arancio	6	rx-d2-	
rx-d2+	3	bianco/verde	1	tx-d1+	
bi-d4+	4	blu	4	bi-d4+	
bi-d4-	5	blu/bianco	5	bi-d4-	
rx-d2-	6	verde	2	tx-d1-	
bi-d3+	7	marro-ne/bianco	7	bi-d3+	
bi-d3-	8	marrone	8	bi-d3-	

Tabella 9.161. I cavi si distinguono in categorie, secondo le prestazioni offerte.

Categoria	Descrizione
categoria 1	Nessun criterio di prestazioni.
categoria 2	1 MHz (cavi telefonici).

Categoria	Descrizione
categoria 3	16 MHz (Ethernet 10baseT).
categoria 4	20 MHz (Token-Ring e Ethernet 10baseT).
categoria 5	100 MHz (Ethernet 100baseT e 10baseT).
categoria 5e	1 GHz (Ethernet 1000baseT, 100baseT e 10baseT).
categoria 6	1 GHz (Ethernet 1000baseT, 100baseT e 10baseT).

### 9.13 Riferimenti

- *How to Convert a Computer ATX Power Supply to a Lab Power Supply*, <http://www.wikihow.com/Convert-a-Computer-ATX-Power-Supply-to-a-Lab-Power-Supply>
- Intel corporation, *ATX12V Power Supply Design Guide*, 2005, [http://www.formfactors.org/developer%5Cspecs%5CATX12V\\_PSDG\\_2\\_2\\_public\\_br2.pdf](http://www.formfactors.org/developer%5Cspecs%5CATX12V_PSDG_2_2_public_br2.pdf)
- Leroy Davis, *IDE (ATA) Bus*, [http://www.interfacebus.com/Design\\_Connector\\_IDE.html](http://www.interfacebus.com/Design_Connector_IDE.html)
- *Computer hope*, *Hard disk drive definitions*, <http://www.computerhope.com/jargon/hdd.htm>
- *Computer hope*, *Computer interface help and information*, <http://www.computerhope.com/help/ide.htm>
- Charles M. Kozierok, *The PC guide, Integrated Drive Electronics / AT Attachment (IDE/ATA) Interface*, <http://www.pcguides.com/ref/hdd/if/ide/>
- *ATA/ATAPI history*, ATA-ATAPI.COM, <http://www.ata-atapi.com/hist.html>
- *The Linux gazette*, numero 57, settembre 2000, Chris Stoddard, *CD-writing with an ATAPI CDR Mini-HOWTO*, <http://linuxgazette.net/issue57/stoddard.html>
- Leroy Davis, *Serial ATA Bus*, [http://www.interfacebus.com/Design\\_Connector\\_Serial\\_ATA.html](http://www.interfacebus.com/Design_Connector_Serial_ATA.html)
- *Serial ATA (SATA)*, <http://www.ata-atapi.com/sata.html>
- *SATA-IO*, <http://www.sata-io.org/>, <http://www.serialata.org/>
- Charles M. Kozierok, *The PC guide, Older bus types*, <http://www.pcguides.com/ref/mbsys/buses/types/older-c.html>
- Steven Nikkel, *AGP Compatibility*, [http://www.ertyu.org/steven\\_nikkel/agpcompatibility.html](http://www.ertyu.org/steven_nikkel/agpcompatibility.html)
- *PCI Express*, [http://www.pantherproducts.co.uk/Articles/Motherboard/PCI\\_Express.shtml](http://www.pantherproducts.co.uk/Articles/Motherboard/PCI_Express.shtml)
- Brad Hards, *The Linux USB sub-system*, Sigma Bravo Pty, <http://www.linux-usb.org/USB-guide/book1.html>
- Willy Tarreau, *Linux-kernel archive*, 2 maggio 2004, <http://lkml.indiana.edu/hypermail/linux/kernel/0405.0/0252.html>
- *WimsBIOS*, <http://www.wimsbios.com/>
- *UniFlash*, <http://www.rainbow-software.org/uniflash/>
- *Bios-Drivers.com*, <http://www.bios-drivers.com>
- *Award BIOS upgrades*, <http://www.award-bios.com>
- *American Megatrends, Inc.*, <http://www.megatrends.com>
- *Coreboot*, [http://www.coreboot.org/Welcome\\_to\\_coreboot](http://www.coreboot.org/Welcome_to_coreboot)
- Ron Minnich, *Porting LinuxBIOS to the AMD SC520*, <http://www.linuxjournal.com/article/8120>
- *OpenBIOS*, [http://www.openfirmware.info/Welcome\\_to\\_Open\\_BIOS](http://www.openfirmware.info/Welcome_to_Open_BIOS)
- *U-BOOT*, <http://u-boot.sourceforge.net>
- *Awardmod*, <http://sourceforge.net/projects/awardmod/>
- *The ROMOS project*, <http://rayer.ic.cz/romos/romose.htm>
- *BIOS Kompendium*, <http://www.bios-info.de/download/dlprgs.htm>

- *Modifying your BIOS for fun and profit*, <http://darkness.codefu.org/wordpress/2003/01/modifying-your-bios-for-fun-and-profit/>
- Wikipedia, *Ethernet over twisted pair*, [http://en.wikipedia.org/wiki/Ethernet\\_over\\_twisted\\_pair](http://en.wikipedia.org/wiki/Ethernet_over_twisted_pair)
- *Getting into the BIOS*, <http://www.inventio.co.uk/gettingintothebios.htm>
- *Getting into the "setup" program during boot-up*, <http://www.bobjohnson.com/laptopandcomputerinfo/setup.htm>
- Hubert Gregoire, *BIOS et SETUP*, <http://www.ac-nancy-metz.fr/services/tec/bios.htm>
- *Bios central*, <http://www.bioscentral.com/>
- *Total hardware 1999*, [http://wayback.archive.org/web/\\*/http://th99.dyndns.org/](http://wayback.archive.org/web/*/http://th99.dyndns.org/)
- Jarno Niemelä, *Jargon's Driver Museum: BIOS utilities*, [http://www.drivermuseum.com/files/utls/bios\\_u.html](http://www.drivermuseum.com/files/utls/bios_u.html)
- *Mastodonic*, <http://mastodonpc.tripod.com/>
- Andreas Schwertner, *PROJECT N° 5, Z80-CPU-based micro-computer*, [http://wayback.archive.org/web/2004/quisquose.com/project\\_no5.htm](http://wayback.archive.org/web/2004/quisquose.com/project_no5.htm)
- *Bitsavers*, <http://www.bitsavers.org>
- John G. Zabolitzky e collaboratori, *Cray-Cyber*, <http://www.cray-cyber.org/general/start.php>
- John G. Zabolitzky, *Vintage Computer Festival Europe 1.0, MUNIAC*, <http://www.travelnotes.de/computer/vcfe/muniac.htm>

<sup>1</sup> **Linux PCI utilities** GNU GPL

<sup>2</sup> Questa immagine è di [borg\\_no-one](#) (≠) [gmx-net](#) e viene riutilizzata qui con il suo permesso. L'autore dell'immagine la considera di dominio pubblico. Il sito da cui proveniva questa immagine era *Borg Number One's - BIOS Pages*, attualmente non più rintracciabile.

<sup>3</sup> **UniFlash** GNU GPL

<sup>4</sup> **Memtest86+** GNU GPL

<sup>5</sup> **Cmoser** software gratuito

<sup>6</sup> **Dug\_ide** software gratuito, accompagnato dai sorgenti

<sup>7</sup> **Bios** software gratuito

<sup>8</sup> **Biosutil** shareware



