

1. Parallela "Centronics"

La prima interfaccia parallela che diventò uno standard di fatto fu introdotta dalla Centronics (ditta che produceva stampanti) ed esiste da prima del primo personal computer. Questo tipo di interfaccia fu adottata su tutti i primi personal computer a 8 bit e successivamente anche sul PC IBM, nel quale il caratteristico connettore che troviamo su tutte le stampanti (Amphenol 57 femmina), fu sostituito, dal lato computer da un più economico DB 25 Cannon (con collegamenti "femmina" sul computer). Questo è diventato un nuovo standard di fatto ed oggi i cavi per interfaccia parallela che si trovano sul mercato hanno sui due estremi connettori diversi, uno Centronics ed uno DB 25.

Attraverso una interfaccia parallela Centronics, si realizza un collegamento da punto a punto fra il computer ed il dispositivo collegato. Alcuni utilizzano la porta parallela del PC per comunicare con dispositivi diversi da stampanti, come scanner, cartucce magnetiche per backup, dispositivi di acquisizione dati ed anche per trasferire file fra due computer. L'interfaccia Centronics è stata progettata per le stampanti, per cui prevede un flusso di dati in una sola direzione, dal computer alla stampante. Il parallelismo dei dati è di 8 bit, su un port monodirezionale dal computer alla stampante. Dopo aver scritto il dato sulle 8 linee parallele di uscita il computer deve spedire, su una linea specifica, un segnale di "strobe" (STB) che comunica alla stampante che il dato è stabile sui fili e perciò essa lo può leggere. Solo dopo lo strobe la stampante acquisisce il dato e può stamparlo. L'interfaccia Centronics prevede che la stampante debba riconoscere il dato come valido e comunicarlo al computer attraverso una particolare linea, detta di "acknowledge" (ACK) (in USA la "receipt of acknowledgment" è la ricevuta di ritorno delle raccomandate). Il computer e la stampante realizzano così un vero e proprio handshake hardware. La velocità di comunicazione può essere cambiata a piacimento dai due dispositivi, ciascuno dei quali potrà tenere basso il proprio segnale di handshake tutte le volte che dovrà rallentare il flusso di dati. L'interfaccia Centronics prevede una comunicazione anche nel senso che va dalla stampante al computer, anche se essa è minimale. Infatti è previsto un insieme di linee, alcune pilotate dalla stampante, altre dal computer. Le linee di stato sono modificate dalla stampante e ne comunicano lo stato al computer, le linee di controllo sono modificate dal computer e cambiano il funzionamento della stampante. In particolare:

Linee di controllo:

- !STB "data strobe": abilita la lettura della stampante quando passa da alto a basso: comunica alla stampante che il dato è valido
- !AUTOLF "auto line feed" se bassa, la stampante deve inserire un Line Feed (carattere ASCII 10, nuova linea, "scroll" del foglio di carta) dopo ogni Carriage Return (carattere ASCII 13, ritorno del carrello: il carattere ASCII che corrisponde all'Enter)
- !INIT "initialize" se bassa la stampante cancella il buffer e resetta, tornando alla condizione in cui era dopo l'accensione
- !SELIN "select input", linea che va tenuta bassa per abilitare la stampa.

Linee di stato:

- !ACK "acknowledge": è la comunicazione, da parte della stampante, che il dato è stato letto
- !ERROR in logica negativa comunica se la stampante è in condizione di errore
- SEL "selected" comunica che la stampante è "on line" (abilitata alla stampa da parte dell'utente)
- PE "paper end" comunica che la stampante è rimasta senza carta
- BUSY alta quando la stampante non può ricevere dati, per qualsiasi ragione

I segnali il cui nome è preceduto da ! hanno logica negata, nelle specifiche Centronics.

I segnali presenti sui fili di un'interfaccia Centronics hanno livelli TTL. Le linee di solito non sono terminate. La lunghezza massima del cavo di solito non può superare i 5 - 6 metri, anche se non esistono specifiche al riguardo. Per proteggere gli altri dispositivi dai disturbi emessi dall'interfaccia parallela, il cavo dovrebbe essere schermato.

1. L'interfaccia parallela del PC

Il PC IBM originale comprendeva una porta parallela Centronics per l'accesso alla stampante¹ ed il supporto software per altre due porte. Essa aveva un errore di progettazione nel meccanismo di interrupt, che non faceva riconoscere il segnale d'interruzione in alcuni casi. I produttori di software non hanno perciò potuto fidarsi troppo dell'interrupt della parallela ed è invalso l'uso di non utilizzarlo. Peraltro tutti i computer PC compatibili del giorno d'oggi funzionano regolarmente in interrupt sulla parallela. Un altro problema del progetto originario IBM è che non era possibile utilizzare le linee dati in modo bidirezionale, cosa non prevista dall'interfaccia Centronics, ma che sarebbe stata facile da realizzare (bastava utilizzare buffer di uscita "three state" o quantomeno "open collector"). Così se si vuole avere una comunicazione bidirezionale completa, è necessario fare dei "trucchi" un po' strani. Essi non sarebbero necessari con la maggior parte delle interfacce parallele in commercio al giorno d'oggi, in quanto esse sono completamente bidirezionali. Peraltro la bidirezionalità è stata di solito realizzata in maniera non standard, per cui ogni tipo di scheda può avere differenti modalità di utilizzazione. Inoltre può capitare di incontrare qualche vecchia scheda che non è bidirezionale. Questi problemi non sussistono se si usano schede parallele che rispettino gli standard EPP o ECP, stilati parecchi anni dopo la comparsa del PC IBM.

Nel primo PC non esisteva un unico circuito integrato di I/O, simile all'8255, che svolgeva le funzioni di porta parallela. Infatti dal punto di vista elettrico, l'interfaccia parallela del PC è costituita con un insieme di latch e driver di linea che

1 Il S.O. MSDOS chiamava questa porta "LPT1" (Line PrinTer numero 1))

sono pilotati dal bus di espansione del sistema. Essi sono mappati in Input Output, a tre indirizzi consecutivi. L'indirizzo base delle due porte più comuni in un PC è di solito 378h e 278h. In vecchie schede video che comprendevano l'interfaccia parallela; l'indirizzo base era 3BCh.

Se il PC ha come S.O. l'MS-DOS, all'accensione va a vedere se esiste l'hardware per l'interfaccia parallela agli indirizzi 3BCh, 378h, e 278h, nell'ordine. Se la porta esiste, assegna ad essa il "nome" di LPT1, LPT2 o LPT3, nell'ordine. L'interrupt per LPT1 è IRQ7, per LPT2 di solito IRQ5, anche se non sempre.

Si deve notare che per completare un singolo trasferimento di 8 bit ad una stampante sono richieste almeno 4 istruzioni di IN o OUT. Infatti i segnali di handshake sono gestiti dal software, perciò il driver della stampante deve:

- verificare che il pin BUSY del connettore sia basso (IN + AND di mascheramento) (il bit letto dal port dovrà essere alto, perché questo segnale è negato)
- scrivere il dato da trasferire (OUT)
- portare basso !STB (OUT)
- portare alto !STB (OUT)

Questo richiede almeno 4 cicli di I/O, ciascuno dei quali impiega poco meno di 1 μ s sul bus ISA, ciò è un considerevole "impegno" per le CPU moderne, che in 4 μ s potrebbero eseguire moltissime istruzioni. Un modo, non molto utilizzato, per velocizzare il ciclo di I/O sul bus ISA, prevede l'utilizzazione di un particolare segnale del bus di espansione del PC (NOWS no wait state) che non inserisce stati di attesa nell'accesso al port. Gli standard più moderni prevedono il trasferimento con handshake completamente controllato dall'hardware in modo tale che un'operazione di I/O si possa eseguire con una sola istruzione ed in un singolo ciclo di bus.

Nei PC attuali spesso tutte le porte parallele e seriali sono comandate da un unico chip, che risiede sulla scheda madre del sistema. In questo caso non esisteranno schede di espansione per le interfacce parallele e seriali. Inoltre il comportamento del chip che controlla la porta parallela è stabilito dai parametri memorizzati nella memoria CMOS di sistema, che possono essere cambiati eseguendo il programma "BIOS CMOS setup", di solito premendo il tasto "cancella" prima che il computer esegua la sequenza di inizializzazione. Il programma di setup di solito permette di far funzionare la porta parallela in una sola delle modalità standard che vedremo (SPP, ECP, EPP).

1. Uso non convenzionale della interfaccia parallela del PC

Dimenticando che serve anche per la stampante, l'interfaccia parallela può essere vista come un'insieme di 17 circuiti TTL, qualcuno di ingresso, altri di uscita, altri ancora utilizzabili in ingresso o in uscita. Essi sono disponibili per collegarsi a qualsiasi dispositivo di I/O digitale, microinterruttori, LED, sensori di prossimità, relais, traguardi ottici, e tanto di altro.

La porta parallela viene utilizzata anche per trasferimenti di file anche se, visto che non tutte sono completamente bidirezionali, necessita di software particolare e di un cavo speciale (di solito questi cavi vengono detti "cavi LapLink", dal nome del programma che li utilizzò per primi). Windows supporta l'uso dei cavi LapLink per il trasferimento dei file, dalla versione 95 in poi. Lo stesso è vero anche per Linux, che ha il supporto per il protocollo PLIP (Parallel Line Internet Protocol).

1. I/O tramite la porta parallela

I tre registri della porta parallela sono:

- un registro dati (DR, Data Register)
- un registro di stato (SR, Status Register)
- un registro di controllo (CR, Control Register)

Il registro dati è un normale registro da 8 bit utilizzabile solo in uscita, che serve per trasmettere alla stampante il carattere da stampare. Gli altri registri della parallela hanno significato solo bit per bit. In SR il computer legge la condizione attuale della stampante; SR si può utilizzare solo in ingresso. Con CR, da utilizzare in uscita, il computer controlla il funzionamento della stampante.

L'interrupt della porta parallela è attivato quando il segnale ACK, nel registro SR, viene portato basso dal dispositivo.

L'interrupt viene generato solo se il bit 4 del CR è alto (enable).

Nella parallela del PC alcune linee sono negate, perciò, se in esse si vorrà ottenere un'uscita alta, si dovrà scrivere 0 nel bit corrispondente del suo registro. Il fatto che le linee della parallela del PC siano negate non ha a che fare con il fatto che alcuni segnali Centronics sono in logica negativa.

Registro Dati (DR), solo in uscita, indirizzo: (BASE)

Peso del bit	7	6	5	4	3	2	1	0
Funzione Centronics	Data 7	Data 6	Data 5	Data 4	Data 3	Data 2	Data 1	Data 0
Direzione	O	O	O	O	O	O	O	O
Logica del segnale (PC)	normale	normale	normale	normale	normale	normale	normale	normale
Pin DB25 PC	9	8	7	6	5	4	3	2
Bit application board	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

Registro di stato (SR), solo in ingresso, indirizzo: (BASE + 1)

Peso del bit	7	6	5	4	3	2	1	0

Funzione Centronics	BUSY	!ACK	PE	SEL	!ERROR	*	non usato	non usato
Direzione	I	I	I	I	I			
Logica del segnale (PC)	invertita	normale	normale	normale	normale			
Pin DB25 PC	11	10	12	13	15			
Bit application board	PA3	fotodiodo	PA2	PA1	PA0			

(*) in alcune schede il bit 2 significa "l'IRQ è stato lanciato"

Registro di controllo (CR), in ingresso o in uscita, indirizzo: (BASE + 2)

Peso del bit	7	6	5	4	3	2	1	0
Funzione Centronics	non usato	non usato	input mode (#)	interrupt enable	!SELIN	!INIT	!AUTOLF	!STB
Direzione			O	O	I/O	I/O	I/O	I/O
Logica del segnale (PC)				normale	invertita	normale	invertita	invertita
Pin DB25 PC					17	16	14	1
Bit application board (*)					PA7	PA6	PA5	PA4

Uscita normale: 1 = alto , 0 = basso

Uscita invertita: 1 = basso, 0 = alto

I piedini (pin) da 18 a 25 del connettore DB 25 sono collegati alla massa del segnale.

Le linee di controllo del port BASE + 3 (CR) sono tipicamente di uscita ma l'elettronica è fatta in modo che ogni bit di controllo può essere anche letto, per scopo di verifica e di diagnostica.

Nella parallela "originale" queste linee di controllo erano anche di tipo "open collector"; questo fatto si può sfruttare, solo per le linee di controllo e non per le linee dati, anche per leggere segnali esterni, pilotati dalla "stampante" invece che dal computer.

Infatti negli stati di uscita open collector, quando viene comandata la scrittura di un 1, il relativo transistor di uscita si "esclude" ed il segnale viene portato alto da circuiteria esterna (resistore di "pull-up"). Se a questo punto un dispositivo esterno mette "bassa" la linea, è possibile leggere uno zero su quel bit. Ciò significa che la linea è sotto il controllo dell'"altro" dispositivo ed è diventata una linea "di ingresso". La linea può perciò essere pilotata dal dispositivo ("stampante").

Perciò, prima di fare un input sul port CR, è necessario portare alti i driver di uscita (p.es. così: OUT (378h + 2), 0100b (si tenga presente che alcuni segnali sono invertiti!)).

Per l'output di solito si usa il registro dati, che non ha nessun pin invertito, perciò non ci sono problemi. Per l'input si possono usare le quattro linee di controllo e le quattro di stato, ma ad ogni input sarà necessario aggiustare i bit invertiti con una maschera in XOR e metterli insieme in un byte con un AND e un OR.

Se si vogliono usare SR e CR per ricevere numeri di 8 bit, sarà necessario operare degli shift su SR, perché c'è il "buco" di ACK, che di solito non si usa come input "generale". In CR invece non sarà necessario fare degli shift, perché i bit sono già consecutivi.

Nota: Nel collegamento all'application board del laboratorio 13 ITI Cesena sarà necessario anche invertire l'ordine dei nibble, perché, come si vede dallo schema, il connettore è stato fatto "a rovescio".

La vecchia interfaccia parallela del PC, che ha gli stadi di uscita del port CR di tipo "open collector", viene indicata come SPP (Standard Parallel Port).

La velocità di trasferimento massima che si può raggiungere con una porta parallela standard, con un estremo sovraccarico software alla CPU, è di circa 150 kByte/s. In condizioni normali la velocità massima potrà essere anche 10 volte inferiore. Questa velocità è insufficiente per le moderne stampanti grafiche, specie per quelle economiche che lasciano alla CPU del computer il compito del calcolo delle "immagini" da stampare e che richiedono perciò lo scaricamento dei testi non più come sequenza di caratteri ASCII, ma sotto forma di un gran numero di pixel.

Per questo l'interfaccia parallela è stata sostituita, in molte stampanti, dalla seriale USB, di più moderno concetto.

2. Cavo Laplink

Laplink era, ed è in parte ancora, un programma per trasferire file rapidamente da un PC all'altro.

Nelle sue prime versioni questo trasferimento avveniva tramite la porta parallela dei due PC, per garantire maggiore velocità di trasferimento rispetto alla porta seriale. Il software veniva venduto insieme ad un cavo, che collega i pin di uscita sul port BASE ai pin di ingresso sul port BASE + 1, in modo da poter collegare con lo stesso cavo due porte parallele identiche. Il pinout del cavo venduto con il programma Laplink è indicato nella tabella seguente.

#Usato solo nelle porte completamente bidirezionali. Se vi si scrive un 1 i driver di uscita del port dati (indirizzo BASE) sono messi in alta impedenza permettendo di effettuare delle IN su di esso.

*Questi nomi fanno riferimento ai bit di I/O delle porte di una scheda didattica in uso all'ITI di Cesena. Sono qui solo come riferimento per gli allievi dell'ITI di Cesena.

Oggi in commercio sono disponibili cavi chiamati "laplink" o "null printer", che sono simili, o uguali al cavo venduto con il programma "Laplink".

Pin DB25 PC lato 1	Port e Bit nel PC	Direzione (rispetto al Lato 1)	Funzione Centronics (Lato1)	Pin DB25 PC lato 2	Port e Bit nel PC
2	DR0	OUT	Data 0	15	SR3
3	DR1	OUT	Data 1	13	SR4
4	DR2	OUT	Data 2	12	SR5
5	DR3	OUT	Data 3	10	SR6
6	DR4	OUT	Data 4	11	SR7
10	SR6	IN	!ACK	5	DR3
11	SR7	IN	BUSY	6	DR4
12	SR5	IN	PE	4	DR2
13	SR4	IN	SEL	3	DR1
15	SR3	IN	!ERROR	2	DR0
25	GND	-	GND	25	GND
1*	CR0	IN/OUT §	!STB	1*	CR0
14*	CR1	IN/OUT §	!AUTOLF	14*	CR1
16*	CR2	IN/OUT §	!INIT	16*	CR2
17*	CR3	IN/OUT §	!SELIN	17*	CR3

(*) collegamento presente solo in cavi Laplink "estesi", diversi da quelli originali.

(§) IN solo se i bit sono stati prima messi a 1 in OUT (stadi open collector).

Se il cavo è schermato, lo schermo deve essere collegato alla parte metallica esterna del connettore **su un lato solo del cavo**.

I pin che non sono indicati in tabella non sono collegati. La colonna "funzione Centronics" è solo per riferimento, in un cavo fatto così i nomi dei segnali Centronics non hanno senso!

Con un cavo Laplink possiamo dunque avere: 5 bit in uscita, 5 bit in ingresso e, se il cavo è "esteso" come illustrato in tabella, 4 altri bit che possono essere usati in ingresso oppure in uscita, dato che le uscite sul port CR sono open collector. In totale si hanno quindi 14 I/O.

Questo tipo di cavo viene detto anche cavo "Null Printer" e viene usato in protocolli di comunicazione quali PLIP (Parallel Line Internet Protocol) tramite il quale due computer possono comunicare attraverso la parallela con gli stessi protocolli che si usano in Internet (TCP/IP).

Volendo si possono avere più I/O digitali di quelli ottenibili usando in un cavo laplink. Se si fa un collegamento come quello indicato nella seguente tabella, si possono avere 8 uscite, 8 ingressi e 1 segnale in ingresso oppure in uscita, per un totale di 17 I/O. Peraltro il cavo che si ottiene non è "standard" e non si può comprare già fatto.

Cavo "Monti" per 17 I/O senza handshake:

Pin DB25 PC lato 1	Port e Bit nel PC	Direzione (rispetto al Lato 1)	Pin DB25 PC lato 2	Port e Bit nel PC
2	DR0	OUT	15	SR3
3	DR1	OUT	13	SR4
4	DR2	OUT	12	SR5
5	DR3	OUT	10	SR6
6	DR4	OUT	11	SR7
10	SR6	IN	5	DR3
11	SR7	IN	6	DR4
12	SR5	IN	4	DR2
13	SR4	IN	3	DR1
15	SR3	IN	2	DR0
25	GND	-	25	GND
7	DR5	OUT	1	CR0
8	DR6	OUT	14	CR1
9	DR7	OUT	16	CR2
1	CR0	IN §	7	DR5
14	CR1	IN §	8	DR6
16	CR2	IN §	9	DR7
17	CR3	IN/OUT §	17	CR3

(§) IN solo se i bit sono stati prima messi a 1 in OUT (stadi open collector).

Questo cavo è compatibile con il cavo Laplink standard, perché i bit che collegano DR con SR sono uguali, ma è diverso da quello esteso, perché si sfruttano anche i tre bit alti di DR, che nel cavo laplink non erano usati, e si collegano a CR.

Per scrivere 8 bit in output basterà usare il port BASE, mentre per avere un intero byte in ingresso si prenderanno i 5 bit più bassi nella parte alta di SR (Base + 1) ed i 3 più alti nella parte bassa di CR (BASE + 2).

Vediamo come si può fare per metterli insieme in un unico byte, con i bit nell'ordine con cui erano stati spediti. Supponiamo di caricare SR in AL e CR in AH. La situazione iniziale sarà la seguente:

```
AL (SR)  [ b4 | b3 | b2 | b1 | b0 | ? | ? | ? ]
AH (CR)  [ ? | ? | ? | ? | ? | b7 | b6 | b5 ]
```

Con due AND:

```
AL  [ b4 | b3 | b2 | b1 | b0 | 0 | 0 | 0 ]
AH  [ 0 | 0 | 0 | 0 | 0 | b7 | b6 | b5 ]
```

Con una OR:

```
OR AL, AH  [ b4 | b3 | b2 | b1 | b0 | b7 | b6 | b5 ]
```

Con tre ROR:

```
ROR AL, 3  [ b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 ]
```

Poi si dovrà fare una XOR per aggiustare la logica dei bit invertiti dalla porta parallela:

```
XOR AL,  [ 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 ]
```

3. *Uso diretto della porta parallela nei S.O. moderni.*

Le CPU dal 386 in poi possono proteggere l'accesso a specifiche locazioni di I/O, nel senso che possono riservare l'accesso ai port di I/O solo a programmi che funzionano ad un certo livello di privilegio. Qualora un programma non sufficientemente "privilegiato" tenti l'accesso a locazioni di I/O protette, la CPU genera un'eccezione, che può essere gestita dal Sistema Operativo e che di solito termina con il "kill" del programma che ha causato l'eccezione.

Nel sistema operativo Windows 95 i port di I/O in generale non sono protetti, a meno che essi non siano bloccati da un driver di un dispositivo, durante l'uso effettivo dei port. In particolare il driver dell'interfaccia parallela non blocca sempre i port dell'interfaccia. L'accesso alla porta parallela è perciò possibile, in programmi Assembly con le istruzioni IN e OUT, in programmi C attraverso le semplici funzioni inp e outp, se esse esistono nelle librerie della versione di C di interesse.

Quando non si disponesse di funzioni come inp e outp (si pensi p.es. a Visual BASIC) si possono scrivere od utilizzare librerie DLL (Dynamic Link Library) che permettono di effettuare l'operazione equivalente.

Al contrario di Windows 95, Windows NT ed i suoi derivati (tutti da XP in poi) proteggono sempre tutti i port di I/O, compresi quelli della porta parallela. Perciò non è possibile operare direttamente sulla porta, a meno che non si voglia scrivere un driver di periferica del S.O. (VxD, Virtual Device Driver), od un modulo OCX. Questo è un processo molto complicato che si può senz'altro sconsigliare per progetti non molto impegnativi.

Esistono peraltro driver gratuiti, scaricabili da Internet, che "liberano" le porte di I/O e rendono disponibili le istruzioni IN e OUT anche agli utenti "normali"

Per esempio, il file "inpout32.dll", che si trova in Internet, permette di usare in Visual BASIC 6.0 le funzioni "Out" e "Inp", analoghe alle IN ed alla OUT. Esse possono funzionare anche nei Windows più moderni.

Nei Windows che discendono da Windows NT, se la DLL viene opportunamente registrata, nel System Registry del S.O., da un utente supervisore della macchina, può essere usata anche da utenti non supervisori.