ACQUISIZIONE AUTOMATICA DELLA MISURA

1. Introduzione

Un sistema automatico di misura consiste di un insieme di strumenti connessi in rete e gestiti da un calcolatore i quali possono scambiare tra loro messaggi e dati in modo coordinato per realizzare una misurazione. La strumentazione può essere di tipo tradizionale, purché dotata di interfacce di comunicazione (seriale o parallela), o essere implementata su schede residenti nel calcolatore (personal computer-PC).

Nella presente dispensa vengono descritti alcuni aspetti essenziali da prendere in considerazione durante il processo dell'acquisizione automatica della misura. In particolare è descritto il funzionamento delle schede di acquisizione e delle interfacce di comunicazione maggiormente impiegate nella moderna gestione di una misura automatizzata.

Per comprendere come l'accuratezza del risultato della misura possa essere limitata in un processo di acquisizione automatica sono richiamati i concetti generali sul campionamento e sulla conversione analogico/digitale e digitale/analogica. Infatti l'acquisizione automatica e la successiva elaborazione del risultato di misura richiedono inevitabilmente un segnale di tipo digitale.

2. Conversione Analogica Digitale (AD) e Digitale Analogica (DA)

La conversione di un segnale di tipo analogico (continuo nel tempo e nelle ampiezze) in un segnale di tipo digitale (discreto nel tempo e nelle ampiezze) è il procedimento base per poter elaborare un segnale attraverso l'impiego di metodi digitali che consentono di gestire l'informazione contenuta nel segnale stesso in maniera decisamente più versatile di quanto realizzabile con tecniche analogiche (l'integrazione su larga scala consente la realizzazione di elettronica digitale con elevato rapporto prestazioni/costi). Le tecniche di conversione AD necessitano di un'adeguata "preparazione" del segnale analogico da convertire. La fase fondamentale consiste nel campionamento del segnale analogico che, se ben effettuato, consente di conservare tutta l'informazione contenuta nel segnale originario (informazione = contenuto armonico).

2.1 Richiami sul campionamento di un segnale

Sia *x(t)* un segnale continuo avente una trasformata di Fourier *X(f)* limitata in frequenza ($-f_{max} \le f \le +f_{max}$). Il segnale campionato con una delta di Dirac, δ, con periodo di campionamento T_c, è esprimibile come

$$x_c(t) = x(t)\sum_{k=-\infty}^{+\infty}\delta(t-kT_c)$$

La trasformata di Fourier del segnale campionato è quindi data da

$$X_c(f) = f_c\sum_{n=-\infty}^{+\infty}X(f-nf_c)$$

e consiste nella replica, con cadenza $f_c=1/T_c$, dello spettro originale *X(f)*.

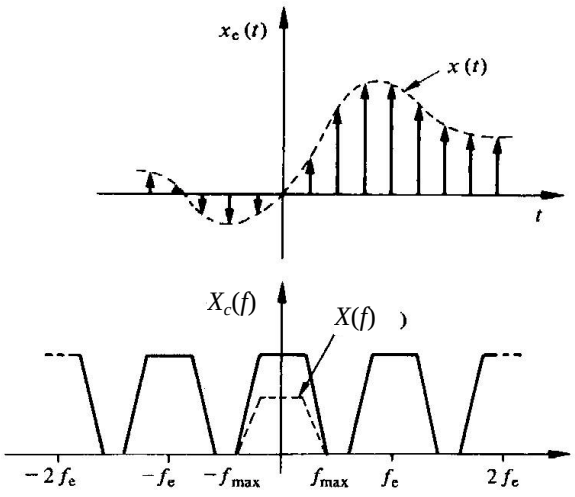


Figura 1. Spettro segnale campionato

Per evitare sovrapposizioni tra le varie ripetizioni (*aliasing*) che inducono una distorsione dello spettro del segnale originario, la frequenza di campionamento deve essere maggiore di $2f_{max}$ (teorema di *Shannon*). Nel caso di segnali con contenuto armonico maggiore della metà della frequenza di campionamento per limitare il fenomeno dell'*aliasing* è necessario prefiltrare il segnale con un filtro passa-basso la cui banda, *B*, sia $<0.5\ f_c$.

Distorsioni dello spettro

Nel caso reale, la funzione con cui si campiona il segnale non è una δ di Dirac ma è assimilabile ad un rettangolo temporale di larghezza finita *D*. Considerando quindi l'effetto di tale larghezza, la densità spettrale del segnale campionato diviene:

$$X_c(f) = f_c D\frac{\mathrm{sen}(\pi fD)}{(\pi fD)}\sum_{n=-\infty}^{+\infty}X(f-nf_c)e^{-j\pi fD}$$

ripetizione periodica dello spettro originale *X(f)* pesata dalla funzione sinc(πfD) (trasformata di Fourier del rettangolo). La presenza della funzione sinc(πfD) introduce quindi nello spettro *X*(*f*) una distorsione (in figura 2 è riportato un esempio per $D=T_c$).

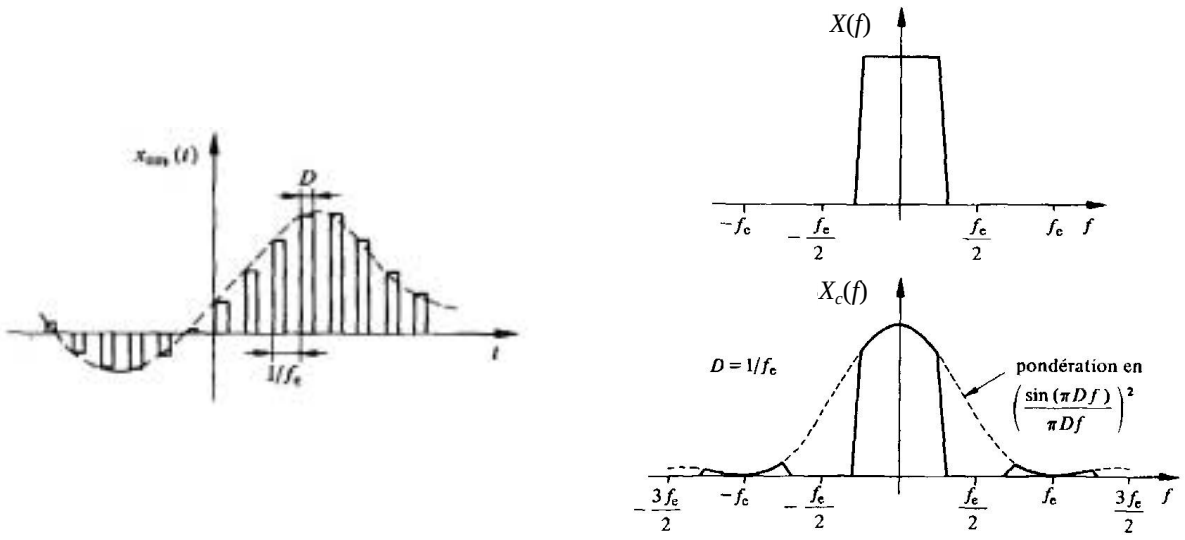


Figura 2. Distorsione dello spettro di un segnale campionato con un rettangolo di lunghezza temporale *D* finita.

Inoltre, l'acquisizione del campione avviene in un tempo seppur piccolo ma finito *T*, che introduce un'ulteriore modifica (effetto di media) sullo spettro del segnale campionato:

$$X_c(f) = f_c D \frac{\operatorname{sen}(\pi f D)}{(\pi f D)} \sum_{n=-\infty}^{+\infty} X(f - nf_c) e^{-j\pi f D} \frac{\operatorname{sen} \pi T(f - nf_c)}{\pi T(f - nf_c)} e^{-j\pi f T}$$

Il processo di campionamento introduce quindi inevitabili distorsioni nell'informazione originaria, distorsioni causate dal fenomeno dell'*aliasing* e dalle durate temporali del campionamento stesso. Nel caso di sistemi di misura con requisiti di accuratezza (precisione) elevati tali distorsioni dovranno essere ben valutate (modellizzate) se non si vuole degradare l'accuratezza del processo di misura. In generale per ridurre gli effetti provocati dall'*aliasing* e dalla durata finita del campionamento si impiegano frequenze di campionamento ben maggiori del limite imposto dal teorema di Shannon (per esempio $f_c \geq 20B$) e filtri *antialiasing* di ordine elevato.

2.2 I campionatori

Nella realizzazione pratica dei campionatori, comunemente chiamati circuiti *sample and hold*, dovranno essere rispettati i vincoli esposti al paragrafo precedente con l'aggiunta di ulteriori requisiti imposti dalla successiva conversione AD. Infatti, durante le differenti fasi del campionamento vi sono inevitabili errori, dovuti alla non idealità dei componenti circuitali adottati, che degradano ulteriormente le prestazioni in termini di accuratezza della conversione AD. In figura 3 sono riportate alcune tipiche implementazioni circuitali di *sample and hold*.

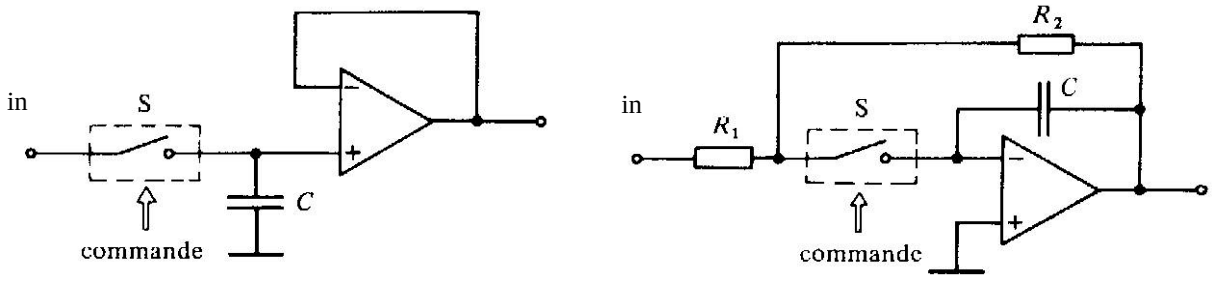


Figura 3. Circuiti pratici di *sample and hold*

In entrambe le configurazioni la capacità, *C*, agisce da elemento di memoria in cui conservare il livello di tensione del segnale di ingresso (*hold*). Quando l'interruttore *S* è chiuso il condensatore si carica alla tensione del segnale in ingresso mentre in condizioni di *S* aperto si conserva all'uscita il valore di tensione immagazzinato nel condensatore.

Quattro sono le differenti fasi in cui il processo di campionamento può essere suddiviso (figura 4): ①) transizione tra la fase di mantenimento e quella di campionamento; ②) campionamento (*sample*); ③) transizione tra la fase di campionamento e quella di mantenimento (*hold*); ④) mantenimento (*hold*).

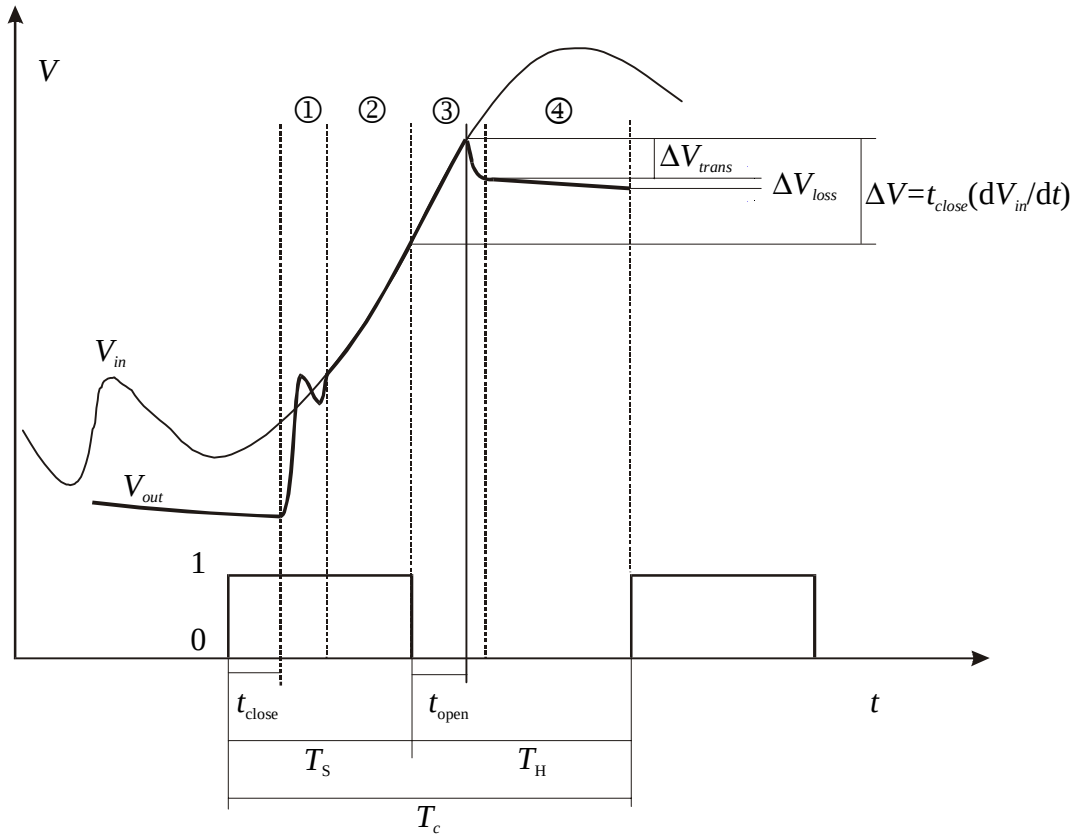


Figura 4. Fasi del campionamento di un segnale.

Nella fase ① si ha la transizione dell'interruttore dalla posizione aperta a quella chiusa (tempo ritardo di chiusura t_{close}) per caricare il condensatore al livello della tensione d'ingresso al campionatore. In questa fase il condensatore deve essere caricato il più rapidamente possibile al fine di ridurre il tempo di acquisizione del segnale (definito come tempo affinché il segnale di uscita sia uguale a quello di ingresso). Per questo motivo la resistenza equivalente ai capi del condensatore deve essere la più piccola e la corrente di carica la più alta possibile. Queste specifiche sono raggiunte impiegando un buffer in ingresso all'interruttore *S* con elevato *slew-rate* e con ridotta resistenza di uscita e un interruttore con bassa resistenza serie. Nella prima configurazione riportata in figura 3 il tempo di acquisizione è funzione della resistenza di uscita del buffer di ingresso (varia quindi al variare di essa) mentre, nel secondo schema circuitale il tempo di acquisizione è fisso ed è determinato dalla resistenza di retroazione R_2. Il tempo di acquisizione è il parametro che quindi limita la massima frequenza del segnale all'ingresso del campionatore. Esaurito il transitorio di carica il segnale ai capi del condensatore segue il segnale di ingresso (fase ②).

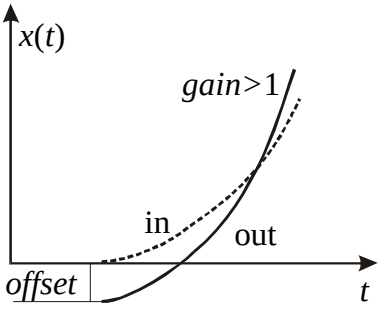


Figura 5.

Durante l'acquisizione del segnale (fase ②) gli errori sono essenzialmente dovuti all'*offset* e al guadagno

dell'amplificatore (*buffer*) impiegato per l'acquisizione del segnale (figura 5). L'*offset* produce in uscita un segnale diverso da zero anche se il valore del segnale in ingresso è nullo mentre il guadagno non perfettamente unitario produce un segnale all'uscita proporzionale all'ingresso (caso ideale uscita = ingresso). Entrambi questi errori possono essere parzialmente corretti attraverso l'impiego di potenziometri che agiscono sulla polarizzazione e sul guadagno dell'amplificatore utilizzato.

Nella fase ③ si ha la transizione dell'interruttore dalla posizione chiusa a quella aperta (tempo ritardo di apertura t_{open}) per mantenere all'uscita del campionatore il livello di tensione del condensatore. Durante questa fase il tempo di ritardo nell'apertura dell'interruttore incrementa la tensione sul condensatore proporzionalmente alla velocità di variazione del segnale di ingresso. La variazione della tensione ai capi del condensatore dovuta a t_{open}, ΔV, rispetto all'istante del comando di apertura è

$$\Delta V = t_{open} \frac{dV_{in}}{dt}$$

Inoltre, l'interruzione della corrente di carica del condensatore produce un transitorio nella carica immagazzinata che si traduce in una ulteriore variazione della tensione ai capi del condensatore ΔV_{trans}.

Nella fase ④ (*holding*) in uscita del campionatore si ha la tensione presente sulla capacità. La carica immagazzinata nel condensatore non dovrà quindi essere dispersa durante l'intero periodo T_H. Nella realtà le correnti di polarizzazione (*bias*) all'ingresso del buffer di lettura e di perdita dell'interruttore comandato e il valore finito della resistenza che il condensatore vede ai suoi capi tendono a disperdere la carica immagazzinata producendo una diminuzione della tensione ai capi del condensatore, ΔV_{loss}, al variare del tempo (andamento quasi lineare). Indicando con I_{bias} la corrente equivalente che scorre nel condensatore dovuta ai contributi dell'amplificatore e dell'interruttore e con R_{eq} la resistenza ai suoi capi

$$\Delta V_{loss} = \frac{I_{bias}}{C} T_H + \frac{V_{max}}{R_{eq} C} T_H$$

Affinché le variazioni di tensioni generate nelle fasi ③ e ④ non degradino l'accuratezza della successiva conversione AD devono risultare inferiori al passo di quantizzazione q del convertitore adottato (funzione del numero di bit). È importante notare come il contributo di variazione della tensione dovuto al ritardo di apertura t_{open} sia proporzionale alla velocità di variazione del segnale di ingresso, vale a dire al contenuto armonico del segnale. Quindi all'aumentare della frequenza l'accuratezza della conversione diminuisce.

2.3 Convertitori Analogici/Digitali (ADC)

In questo paragrafo vengono richiamate le principali caratteristiche degli ADC attualmente disponibili sul mercato con particolare riguardo alle proprietà che possono degradare l'accuratezza di una misura. Non saranno quindi presentati i principi di funzionamento dei diversi convertitori analizzati.

La conversione AD consente di rappresentare l'ampiezza del segnale all'ingresso del convertitore con un numero finito di livelli (quantizzazione). Se N è il numero di bit e ΔV è la portata del convertitore il passo di quantizzazione è $q = \frac{\Delta V}{2^N}$. Nel discretizzare un segnale si compie implicitamente un errore, detto di quantizzazione, dovuto al numero finito di livelli con i quali è

possibile rappresentare il segnale stesso. L'errore di quantizzazione è espresso come differenza tra valore della tensione di ingresso, V_{in}, e il valore della conversione realizzata dall'ADC, $n_i q$, cioè $\varepsilon_q = n_i q - V_{in}$. In Figura 6 è riportata la caratteristica ideale tra ingresso e uscita di un ADC a tre bit. Nel caso ideale ε_q è compreso tra $-q/2$ e $+q/2$ e ha valore medio nullo.

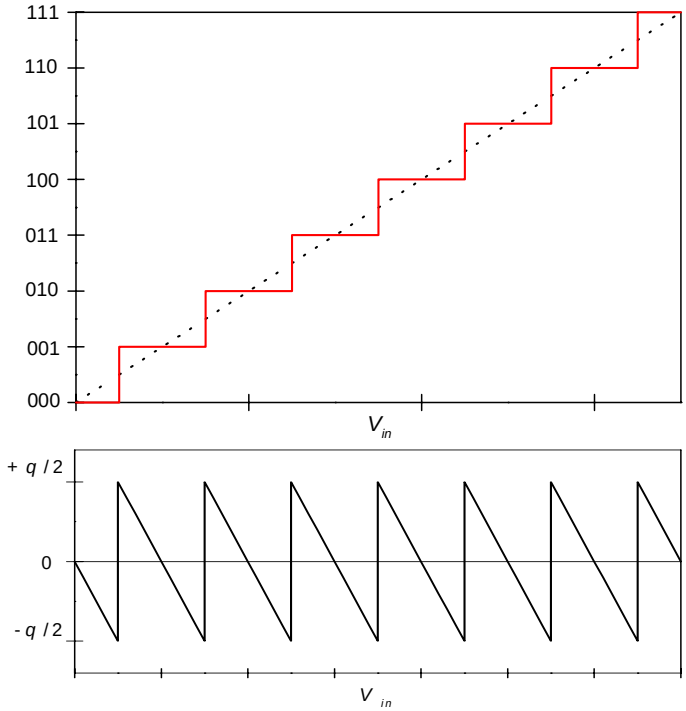


Figura 6. Caratteristica ideale di un ADC.

La situazione reale è ovviamente lontana dal caso ideale. In Figura 7 è mostrato un esempio di caratteristica reale di un ADC. Lo scostamento dal comportamento ideale è causato da differenti errori classificabili in: errore di polarizzazione (*offset*), errore di guadagno ed errore di linearità (integrale e differenziale).

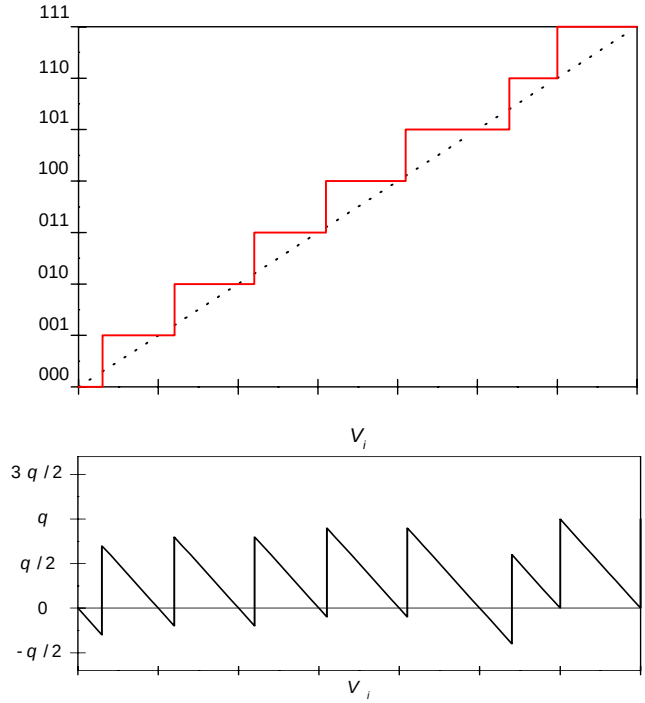


Figura 7. Caratteristica reale di un ADC.

L'errore di polarizzazione (*offset*) è determinato da un livello di tensione diverso da zero anche se all'ingresso dell'ADC vi è una tensione uguale a zero. Questo errore si evidenzia con una traslazione lungo l'asse delle ascisse della caratteristica ingresso/uscita dell'ADC e con un valore medio non nullo nell'errore di quantizzazione (figura 8).

L'errore di guadagno va invece a modificare la pendenza della caratteristica ingresso/uscita rispetto all'andamento ideale, pendenza = 1, riducendo o aumentando il passo di quantizzazione. In questo caso, l'errore di quantizzazione varia linearmente con la tensione da convertire.

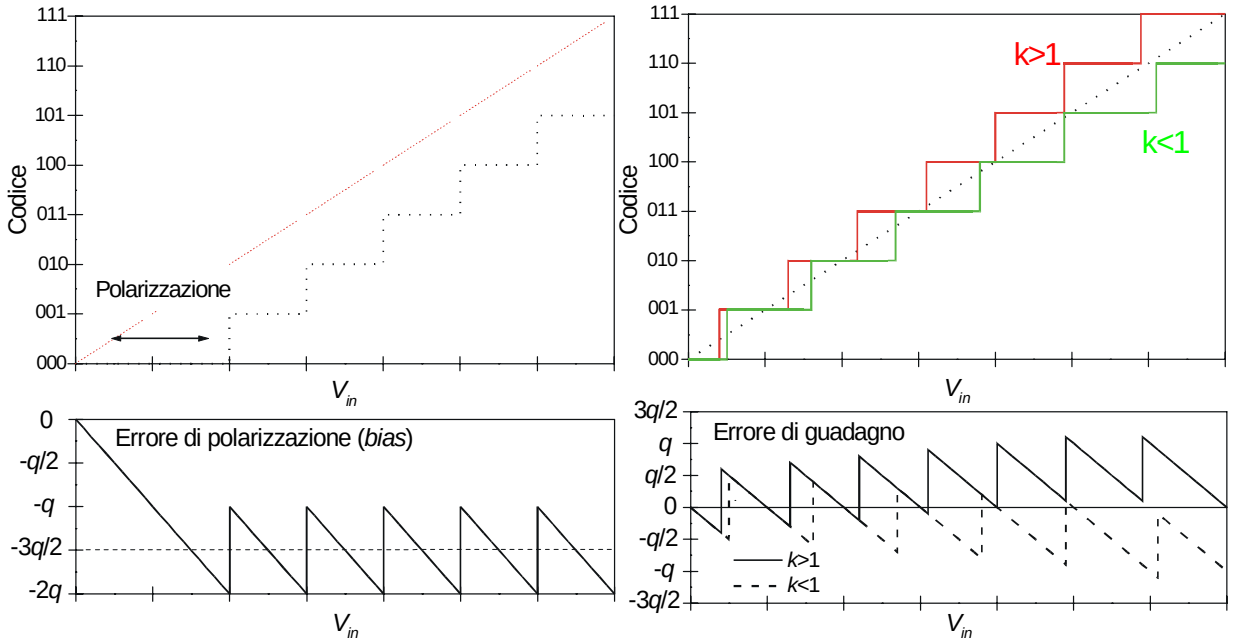


Figura 8. Errori di polarizzazione e di guadagno.

A seconda della definizione data, l'errore di linearità è detto di tipo integrale o di tipo differenziale. Si parla di linearità integrale (figura 9) quando l'errore è definito come differenza tra il valore di tensione che effettivamente produce una transizione dell'uscita del convertitore meno il valore di tensione ideale. L'errore di linearità differenziale è definito come la differenza tra il passo di quantizzazione reale (ampiezza reale di un gradino) e passo di quantizzazione ideale. Se l'errore è superiore al passo di quantizzazione *q*, allora si perde una codifica (figura 10).

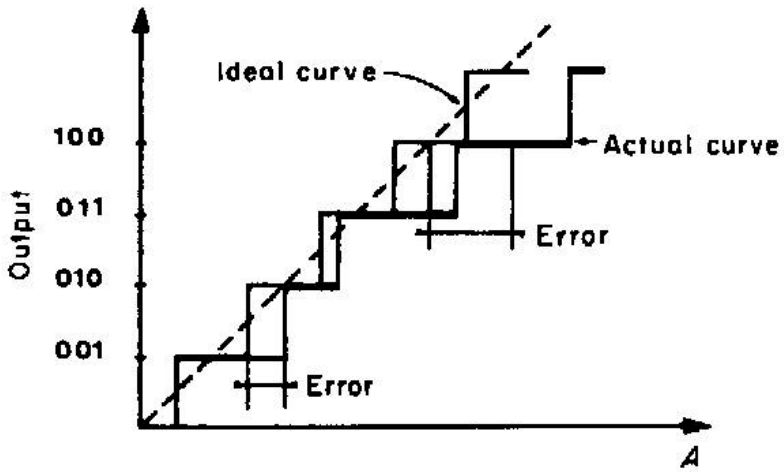


Figura 9. Errore di linearità integrale.

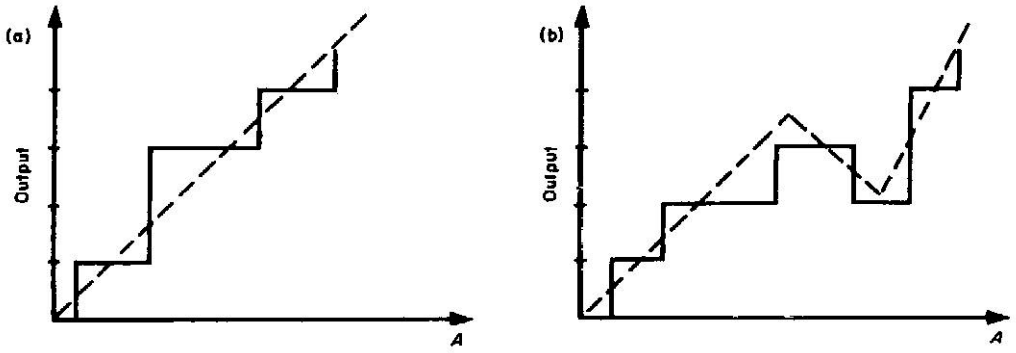


Figura 10. Errori di linearità differenziale.

Oltre all'accuratezza della conversione, determinata dai vari tipi di errori e dal rumore elettronico nell'ADC, un ulteriore parametro fondamentale è il tempo con il quale l'ADC effettua la conversione tra segnale analogico e digitale. Tale parametro, tempo di conversione, definisce la velocità con la quale l'ADC converte il segnale analogico in segnale digitale e quindi impone un limite sulla massima banda del segnale da convertire. Esso è sempre legato alla risoluzione dell'ADC, vale a dire al numero N di bit (ad eccezione del convertitore parallelo), e aumenta all'aumentare della risoluzione (andamento lineare o esponenziale con N). In tabella 1 sono riportati l'accuratezza normalizzata al valore di fondo scala dell'ADC (accuratezza relativa) e il tempo di conversione degli ADC comunemente disponibili sul mercato.

Tipo di convertitore A/D	**Accuratezza**	**Tempo di conversione**
inseguimento	10^{-4}	$t_{conv}=2^N T_c$
approssimazioni successive	10^{-4}	$t_{conv}=N T_c$
rampa	10^{-3}	$t_{conv}=(2^N-1)T_c$
doppia rampa	10^{-5}	$t_{conv}=2(2^N-1)T_c$
trasferimento di carica	10^{-3}	$t_{conv}=N T_c$
conversione V/F	10^{-6}	$t_{conv}=2^N T_c$
parallelo (o flash)	10^{-2}-10^{-3}	$t_{conv}=T_c$

Tabella 1.

Prima di concludere la trattazione degli ADC è importante illustrare come il rumore in un convertitore limita l'accuratezza della conversione. Dato un ADC con N bit e dinamica ΔV caratterizzato da un rumore di tensione $e_n(t)$, è possibile valutare l'effetto che il rumore ha sulla conversione confrontando la potenza di rumore $\overline{e_n^2}$ con il rumore introdotto dalla quantizzazione. Il rumore introdotto dalla quantizzazione è $\sqrt{\overline{e_q^2}} =$. La situazione ottimale è quella in cui il rumore $\overline{e_n^2}$ è inferiore ad $\overline{e_q^2}$. In questo caso la conversione viene realizzata effettivamente con tutti i bit a disposizione. Nel caso in cui i due rumori sono confrontabili o peggio ancora $\overline{e_n^2}$ è maggiore di $\overline{e_q^2}$, il contenuto di informazione presente nel segnale digitale è influenzato dal rumore del convertitore e quindi non tutti i bit sono utili alla conversione del segnale. In questo caso è opportuno definire quale è il

Figura 11. ADC rumoroso

numero di bit che effettivamente contengono l'informazione del segnale da convertire (bit equivalenti) e quali sono invece degradati dal rumore. Il numero dei bit equivalenti, N_{eq}, si ottiene eguagliando la somma dei rumori di quantizzazione e di tensione dell'ADC con il rumore di quantizzazione prodotto da N_{eq}.

$$\left[\overline{e_q^2}\right]_{eq} = \left[\frac{\Delta V}{2^{N_{eq}}}\Big/\sqrt{12}\right]^2 = \overline{e_n^2} + \left[\frac{\Delta V}{2^N}\Big/\sqrt{12}\right]^2$$

equivalente a

$$\left(\frac{1}{2^{N_{eq}}}\right)^2 = \left(\frac{1}{2^N}\right)^2\left(1+\frac{\overline{e_n^2}}{\overline{e_q^2}}\right)$$

Passando al logaritmo in base due si ricava l'espressione di N_{eq}

$$N_{eq} = N - \frac{1}{2}\log_2\left(1+\frac{\overline{e_n^2}}{\overline{e_q^2}}\right).$$

2.4 Convertitori Digitali/Analogici (DAC)

Anche in questo paragrafo si descriveranno le principali caratteristiche dei convertitori DA attualmente disponibili sul mercato con particolare enfasi alle proprietà che possono degradare l'accuratezza di una misura, senza discutere i principi di funzionamento dei diversi convertitori analizzati.

La caratteristica ingresso/uscita di un DAC è del tutto analoga a quella degli ADC a patto di invertire gli assi: in ascissa ci sarà il codice binario e in ordinata il corrispondente livello di tensione. Il tempo di conversione e gli errori possibili in un DAC sono definiti nel medesimo modo di quelli degli ADC.

Nel caso di un DAC c'è un problema specifico causato dalla differenza di velocità di chiusura degli interruttori comandati che determinano la conversione del segnale. Tale differenza può far nascere dei transitori di tipo impulsivo, *glitches*, che incrementano il rumore all'uscita del convertitore, quindi spesso si inseriscono dei filtri passa-basso all'uscita del convertitore per limitare questo inconveniente.

In tabella 2 sono riportati l'accuratezza e il tempo di conversione dei DAC disponibili sul mercato, con l'accuratezza normalizzata al valore di fondo scala del DAC (accuratezza relativa).

Tipo di DAC	**Accuratezza**	**Tempo di conversione**
seriale	10^{-5}	$t_{conv}=N\,T_c$
parallelo a resistenze pesate	10^{-4}	$t_{conv}=T_c$
parallelo a R-2R	10^{-5}	$t_{conv}=T_c$
parallelo a divisione di tensione	10^{-4}	$t_{conv}=T_c$

Tabella 2

3 Schede di acquisizione dati (DAQ)

Oggigiorno, è sempre più comune l'impiego di schede per l'acquisizione dei dati (*Data AcQuisition system*) comandate da un personal computer o direttamente presenti sul PC. La struttura generale di una moderna scheda di acquisizione per personal computer è illustrata nella figura 11.

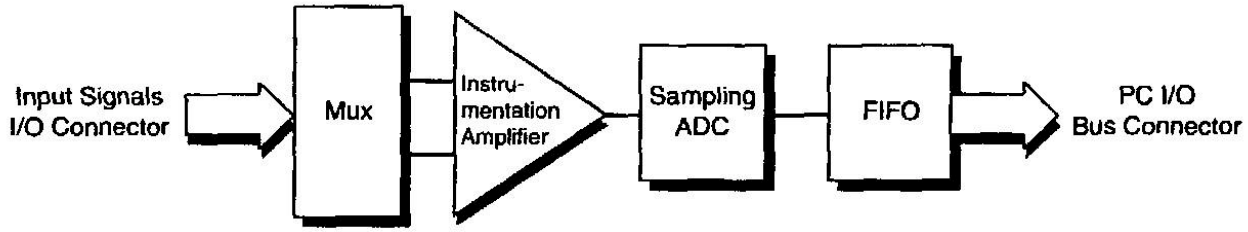


Figura 11. Schema a blocchi di una scheda di acquisizioni dati.

In figura 11 si possono riconoscere quattro blocchi circuitali fondamentali:
il multiplexer (Mux) in ingresso alla scheda consente di avere a disposizione *n* canali indipendenti; l'amplificatore per strumentazione legge il segnale all'uscita del Mux e lo invia con un'opportuna amplificazione (selezionabile dall'utente) al blocco di campionamento e di conversione AD in modo da sfruttarne il più possibile la dinamica; il cuore della scheda è il blocco di campionamento e ADC che converte il segnale analogico in un segnale digitale; infine è presente un registro FIFO (*First In First Out*) dove vengono memorizzati i dati che sono successivamente resi disponibili al *bus* dati del PC o direttamente alla memoria RAM del PC (*Direct Memory Access*).

La maggior parte delle schede dispone anche di uscite analogiche programmabili da PC attraverso quindi dei DAC, di segnali di sincronismo sia di tipo digitale (*trigger*) sia di tipo analogico (*timer*) e delle linee di ingresso/uscita digitali (*I/O lines*).

Caratteristiche essenziali di una scheda sono quindi:

- numero di canali di ingresso analogici;
- velocità di campionamento espressa in campioni al secondo (Sa/s);
- risoluzione ADC, ovvero numero di bit;
- numero di linee digitali I/O e *timer*;
- numero di uscite analogiche.

I canali di ingresso analogici possono essere di tipo singolo (single-ended) o di tipo differenziale. Nel primo caso (figura 12 a) tutti i canali sono riferiti alla massa comune della scheda, AIGND (*Analog Input GrouND*), e la configurazione è generalmente adottata quando i segnali di ingresso sono ampi (dell'ordine del volt) e devono essere riferiti ad una massa comune. Nel caso differenziale (figura 12 b) ogni canale ha una sua linea di comune differente dalla AIGND. La configurazione differenziale consente una reiezione ai disturbi e ai rumori di modo comune superiore alla configurazione *single-ended*. Poiché il numero di linee di ingresso è determinato dal multiplexer, se *M* sono gli ingressi del Mux il numero di ingressi single-ended coincide con il valore *M* mentre il numero dei canali differenziali è *M*/2.

Quando non è specificata, la massima velocità di campionamento è definita considerando l'acquisizione di un singolo canale. All'aumentare del numero dei canali acquisiti, poiché il blocco di acquisizione è costituito da un unico campionatore seguito dall'ADC, la velocità di acquisizione scala proporzionalmente al numero di canali letti. Quindi acquisendo per esempio dieci canali differenti con una scheda DAQ avente velocità di campionamento di 1 MSa/s, la velocità con la quale viene campionato ognuno dei dieci canali è 100 kSa/s. L'acquisizione viene, infatti, realizzata scandendo nel tempo i differenti canali di ingresso attraverso il multiplexer.

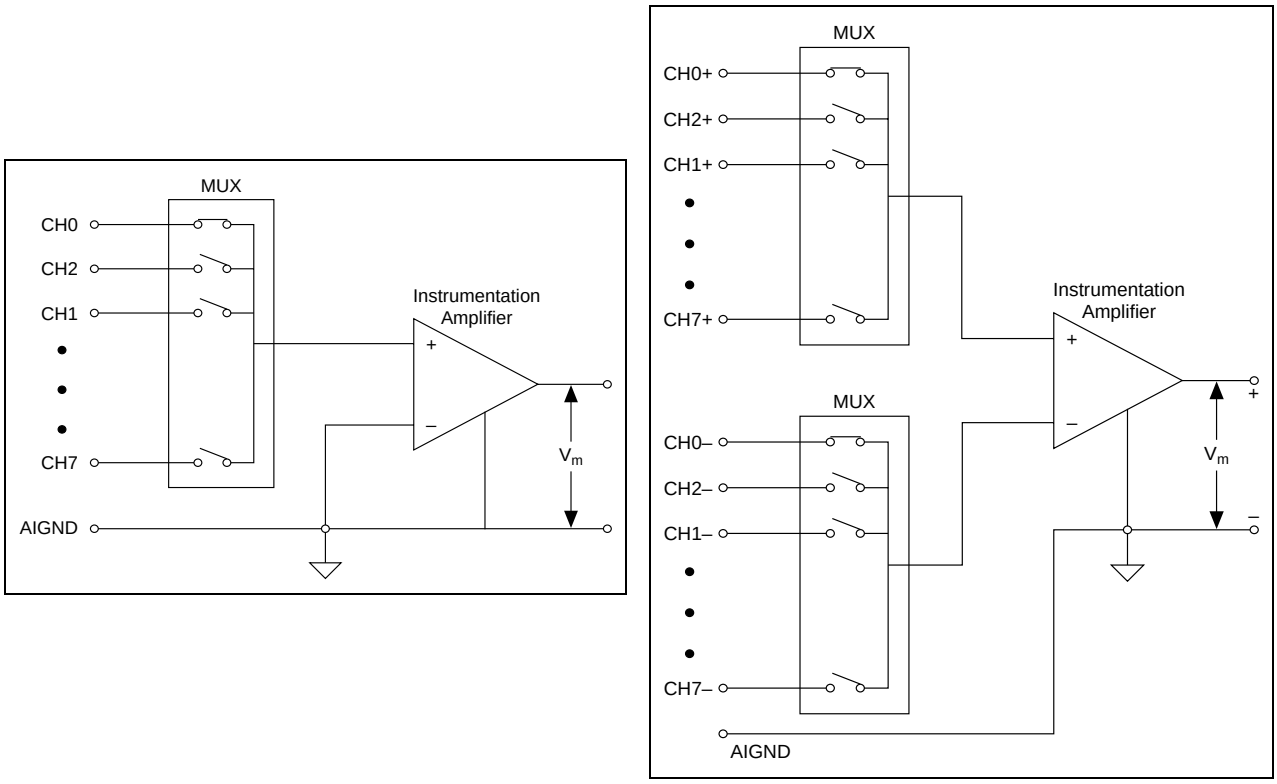


Figura 12. Canali di ingresso di tipo singolo (*single-end*) e differenziale.

Poiché l'ambiente elettromagnetico in un PC è particolarmente ricco di disturbi sia a bassa frequenza (50 Hz e sue armoniche, frequenze di *refresh* video comprese tra i 80 Hz e i 70 kHz, frequenza dell'ordine di qualche kilohertz per gli alimentatori di tipo *switching*) sia ad alta frequenza (frequenze di clock e armoniche), le schede DAQ devono essere progettate in modo tale da assicurare una reiezione alle interferenze più elevata possibile al fine di non degradare il numero di bit equivalenti disponibili all'utente e quindi la risoluzione del sistema ($q=V_{max}/2^N$). Le schede sono quindi realizzate impiegando basette multistrato (fino a 12 strati) dove è possibile ricavare opportuni piani di massa e schermature per aumentare la reiezione ai disturbi del PC.

A scopo dimostrativo si riportano in figura 13 alcune misure realizzate dalla *National Instruments* su di una scheda DAQ a 16 bit da loro prodotta.

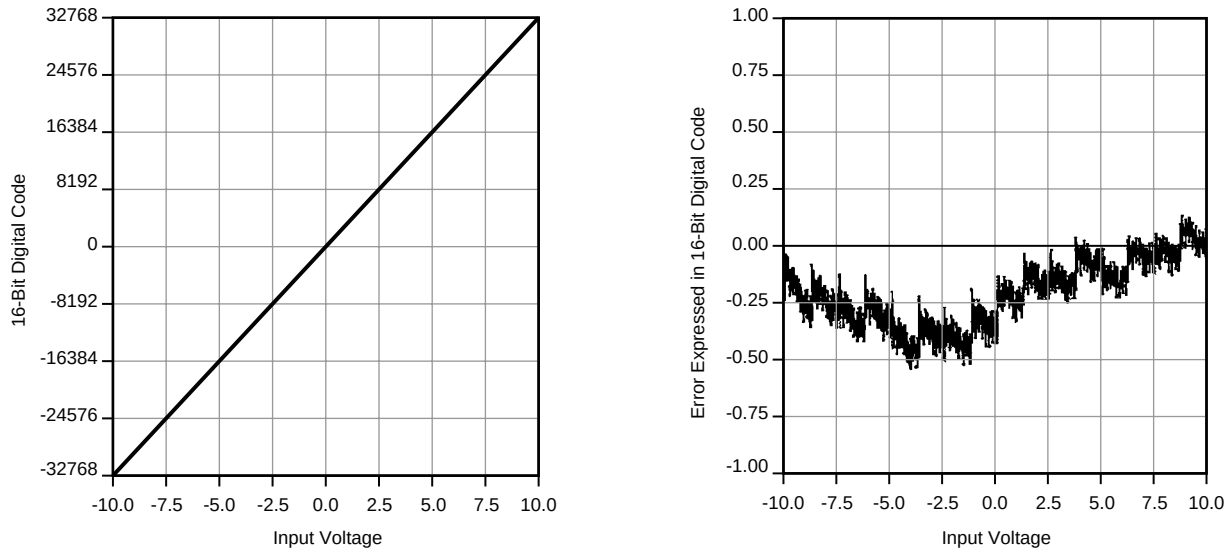


Figura 13. Misura della caratteristica ingresso/uscita e dell'errore di quantizzazione per una scheda commerciale a 16 bit (*National Instruments*).

Per sfruttare tutta la dinamica dell'ADC è opportuno condizionare il segnale di tensione all'uscita del Mux prima di inviarlo al blocco di conversione. Tale condizionamento è realizzato per mezzo dell'amplificatore per strumentazione che ha un guadagno programmabile (valori tipici sono 1, 2, 5, 10, 20, 50 e 100). Se per esempio l'ADC ha 12 bit e una dinamica di ±5 V, la sua risoluzione è $q_{ADC}=10/2^{12}=2.44$ mV (se tutti i bit sono significativi). Se il guadagno dell'amplificatore per strumentazione è impostato a 100, la risoluzione sul segnale all'ingresso della scheda DAQ risulta $q_{in}=q_{ADC}/100=24.4\ \mu$V. Per non limitare l'accuratezza della conversione l'amplificatore per strumentazione deve avere un elevato *slew-rate* e un ridotto tempo di assestamento soprattutto quando il guadagno è elevato (queste caratteristiche impongono quindi un elevato prodotto guadagno-banda). Infatti, durante l'acquisizione di più canali, ad ogni colpo di *clock* che consente la scansione dei canali da acquisire, l'uscita dell'amplificatore deve raggiungere il più rapidamente possibile il livello corrispondente alla tensione presente sul canale abilitato all'acquisizione. Per guadagni elevati le escursioni della tensione presenti all'uscita dell'amplificatore, dettati dalla scansione dei canali, possono essere molto ampie anche se i segnali all'ingresso dei canali sono costanti (basta infatti che abbiano livelli sufficientemente diversi). In figura 14 è riportato l'andamento della tensione in ingresso di un amplificatore per strumentazione durante l'acquisizione di 40 canali con segnali continui. Tale segnale ha un contenuto armonico molto elevato, anche se i segnali ai singoli canali sono continui, e affinché l'amplificatore non introduca distorsioni che vanno a degradare l'accuratezza della conversione deve appunto avere elevato *slew-rate* e un ridotto tempo di assestamento.

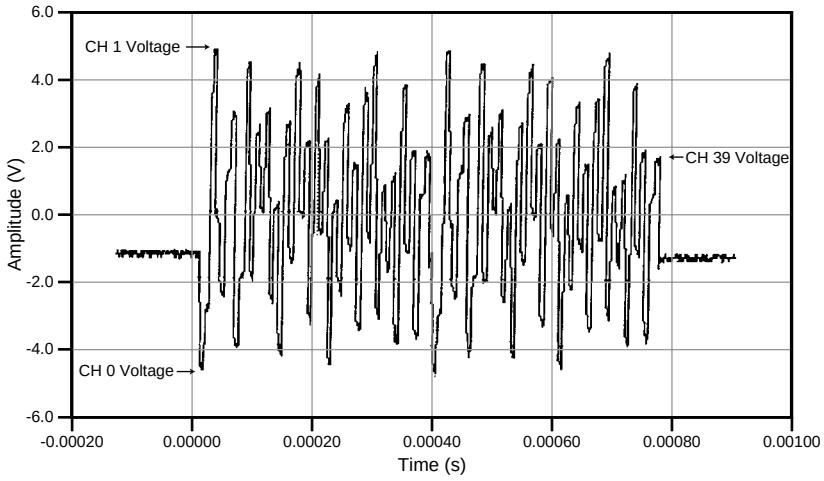


Figura 14. Uscita di un amplificatore per strumentazione per una scheda commerciale a 12 bit (*National Instruments*) durante l'acquisizione di 40 segnali continui.

Dopo aver presentato la struttura di una scheda DAQ, il successivo passo consiste nel descrivere brevemente l'architettura per l'acquisizione di un generico fenomeno fisico. Dallo schema generale di una acquisizione, riportato in figura 15, sono evidenti quattro differenti fasi del processo di acquisizione:

1. trasduzione del segnale. In questa fase il misurando deve essere convertito in una grandezza elettrica (corrente, tensione, resistenza, capacità) attraverso l'impiego di uno o più trasduttori (sensori);
2. condizionamento del segnale trasdotto. Con questo termine si indicano le elaborazioni analogiche da effettuare sul segnale proveniente dal sensore atte a linearizzare se possibile la risposta del sensore, a garantire o aumentare l'isolamento galvanico tra sensore e scheda DAQ, ad adattare il segnale alla dinamica della scheda DAQ;
3. scheda DAQ;

4. personal computer per l'impostazione del processo di misura (programmazione della scheda DAQ) e l'elaborazione in tempo reale o a processo di misurazione concluso dell'informazione di misura.

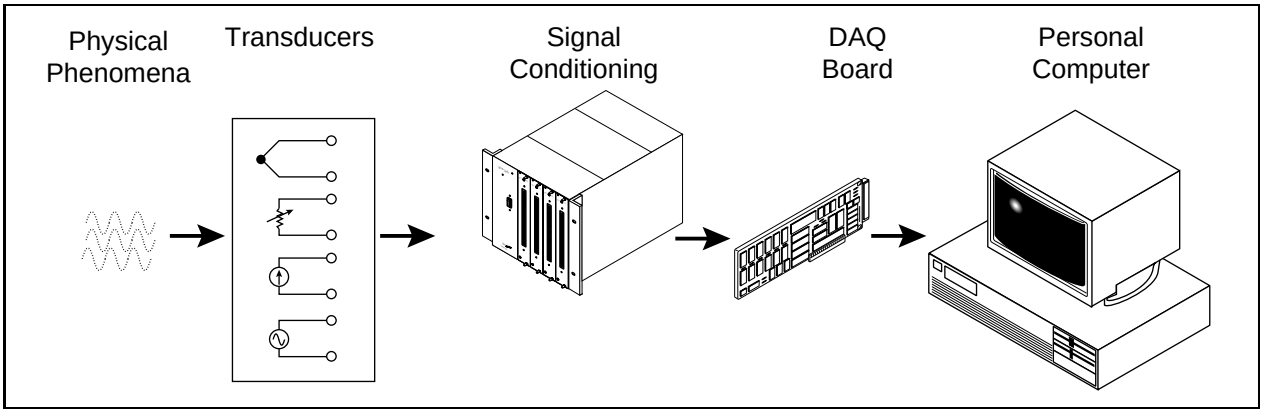


Figura 15. Fasi di un'acquisizione.

4 Porte e protocolli di comunicazione nella moderna strumentazione

Nella moderna strumentazione di misura sono oramai sempre presenti porte di comunicazione, di tipo seriale e parallelo, per consentire l'acquisizione della misura e l'impostazione della strumentazione attraverso l'impiego di un personal computer (PC). Inoltre i protocolli impiegati consentono di collegare più dispositivi tra loro attraverso una "rete di acquisizione" gestita da un controllore (quasi sempre un PC) realizzando in questo modo quello che viene comunemente chiamato "sistema automatico di misura".

In questo capitolo si presentano le porte e i protocolli di comunicazione oggigiorno più comuni.

4.1 Interfaccia seriale - RS-232

Il protocollo di comunicazione seriale RS-232 è ampiamente impiegato e diffuso da essere oramai sempre presente su di un normale personal computer (PC-IBM) per la connessione di stampanti, mouse e modem. Tale protocollo ha quindi trovato ampio utilizzo anche nella moderna strumentazione (la maggior parte degli strumenti possiedono una porta di comunicazione seriale). Il successo della RS-232 deriva dalla sua semplicità di comunicazione: i dati vengono trasferiti e ricevuti, in formato di *byte*, bit per bit. Ovviamente tale protocollo è piuttosto lento (velocità tipica di 9.6 kbit/s) rispetto a quello parallelo (*byte* per *byte*) ma consente di trasferire dati anche su distanze relativamente lunghe (fino a 200 m impiegando cavi schermati). I dati trasmessi serialmente sono tipicamente ASCII (per dettagli si veda la Tabella 3).

La comunicazione avviene attraverso tre linee: (1) RX ricezione, (2) TX trasmissione i cui livelli sono riferiti alla linea di massa GND (3). Altre linee possono essere disponibili ma in generale non sono richieste. Poiché il protocollo è asincrono la porta può ricevere e inviare dati nello stesso tempo.

Parametri fondamentali in un protocollo seriale sono: *baud rate*, *data bits*, *stop bit* e *parity bit*.

- *baud rate*, indica la velocità della comunicazione, cioè il numero di bit trasferiti in un secondo. Le velocità di trasmissione sono di 9.6 kbit/s (tipica per la maggior parte della strumentazione), 14.4 kbit/s, 28.8 kbit/s, 33.6 kbit/s. Al crescere della velocità si riduce la distanza tra i dispositivi connessi.
- *data bits*, sono i bit relativi al dato trasmesso o ricevuto. I dati sono costituiti da pacchetti di 5, 6, 7 e 8 bit a seconda dell'informazione da trasferire. Per esempio lo standard ASCII impiega 7 bit (valori da 0 a 127) mentre il formato ASCII esteso usa 8 bit (valori da 0 a 255).
- *stop bit*, sono bit (uno o due) che indicano la fine della comunicazione di un singolo messaggio (pacchetto di informazione). Poiché la trasmissione è asincrona ogni dispositivo ha una sua temporizzazione interna (stabilita dalla velocità di trasmissione), e questi bit forniscono anche una indicazione sugli errori di trasmissione dovuti a differente velocità di trasferimento (più bit sono usati maggiore è il controllo per la sincronizzazione della comunicazione ma minore è la velocità di trasmissione).
- *parity bit*, bit usato per il controllo di parità (è inserito dopo i bit di dato). Quattro sono i controlli di parità: pari, dispari, *marked* e *spaced*. Nei primi due il bit di parità è impostato per avere un numero di bit del pacchetto nello stato alto pari o dispari. Nelle modalità *marked* e *spaced* il bit è sempre nello stato rispettivamente alto o basso. In questo modo si possono valutare gli effetti del rumore sulla trasmissione valutando il cambiamento di stato di questo bit.

Ovviamente un singolo messaggio è un pacchetto costituito da i bit di dati, stop e parità. In figura 16 è rappresentato il diagramma dei livelli di una trasmissione del carattere ASCII "A". Il livello di tensione corrispondente allo stato alto (1) è compreso tra +3 V e +12 V mentre lo stato logico basso (0) è tra -3 V e -12 V.

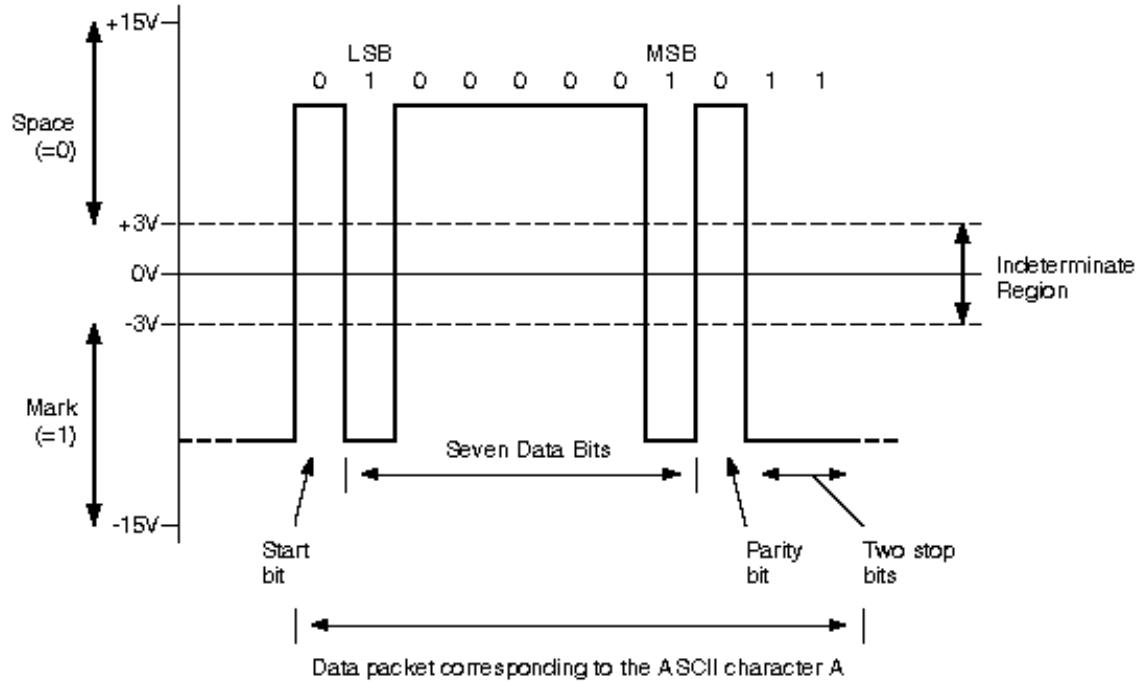
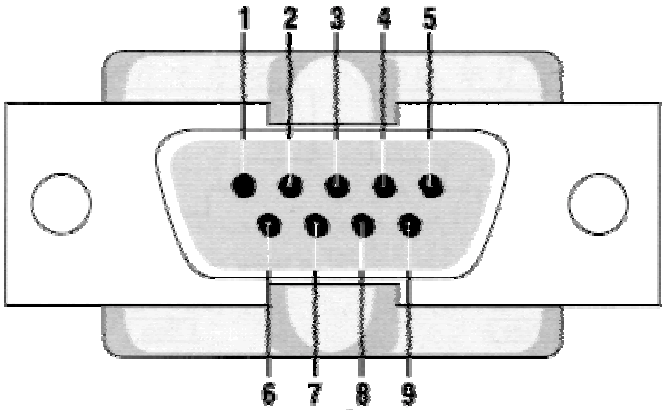


Figura 16. Pacchetto di un singolo messaggio (carattere ASCII "A") seriale con il protocollo RS-232.

Uno schema completo delle connessioni di una porta a nove poli RS-232, con relativa foto, è mostrato in figura 17.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figura 17. Connettore a nove poli RS232 con relativa tabella di connessione.

L'analogo protocollo seriale impiegato nei personal computer Macintosh è denominato RS-422. Questo standard impiega una connessione di tipo differenziale (contro quella *single-ended* utilizzata dalla RS-232) che migliora notevolmente l'immunità al rumore e incrementa quindi la distanza tra i dispositivi connessi (vantaggi fondamentali in ambienti industriali). La trasmissione differenziale deve comunque dedicare due linee per la trasmissione e due linee per la ricezione dei dati. Una evoluzione dello standard RS-422 è costituito dal protocollo RS-485 che consente di collegare un maggiore numero di dispositivi (da 10 a 32).

Sebbene la connessione con il protocollo seriale necessita solo di tre linee (RX, TX e GND) per ridurre i problemi relativi alla sincronizzazione e al controllo della comunicazione si ricorre all'uso del *handshaking* (linee di controllo) che nel caso della RS-232 può essere di tipo software o hardware.

- **Software Handshaking** - Questo metodo impiega i *byte* di dati per trasmette/ricevere dei caratteri di controllo. La comunicazione avviene sempre usando le tre linee (RX, TX e GND) perché il comando di controllo è analogo ad un dato. Due sono i comandi di *software handshaking*: XOFF (carattere ASCII decimale 19 o esadecimale 13) e XON (carattere ASCII decimale 17 o esadecimale 11). Questi comandi sono inviati dal ricevitore al trasmettitore per interrompere temporaneamente (*pause*) e riprendere (*resume*) la comunicazione. Il grande svantaggio di tale metodologia di controllo consiste nella indisponibilità dei caratteri decimali 19 e 17 (esadecimali 13 e 11) come valori assegnabili ad un dato.
- **Hardware Handshaking** - Questo metodo impiega delle linee di trasmissione dedicate. Le linee RTS/CTS e DTR/DSR sono analoghe alle linee RX/TX.
  RTS (*Request To Send*) - La linea è asserita (stato alto) dal ricevitore quando è pronto per ricevere un dato. Il trasmettitore imposta allora la linea CTS (*Clear To Send*) per confermare l'invio del dato.
  DTR (*Data Terminal Ready*) e DSR (*Data Set Ready*) sono comunemente impiegate per le comunicazioni via modem. Quando il modem è pronto per inviare un dato asserisce la linea DTR. Il PC risponde abilitando la linea DSR che conferma l'invio del dato.

Dec	Hex	Binary	Value	Dec	Hex	Binary	Value
000	000	00000000	NUL (Null char.)	064	040	01000000	@
001	001	00000001	SOH (Start of Header)	065	041	01000001	A
002	002	00000010	STX (Start of Text)	066	042	01000010	B
003	003	00000011	ETX (End of Text)	067	043	01000011	C
004	004	00000100	EOT (End of Transmission)	068	044	01000100	D
005	005	00000101	ENQ (Enquiry)	069	045	01000101	E
006	006	00000110	ACK (Acknowledgment)	070	046	01000110	F
007	007	00000111	BEL (Bell)	071	047	01000111	G
008	008	00001000	BS (Backspace)	072	048	01001000	H
009	009	00001001	HT (Horizontal Tab)	073	049	01001001	I
010	00A	00001010	LF (Line Feed)	074	04A	01001010	J
011	00B	00001011	VT (Vertical Tab)	075	04B	01001011	K
012	00C	00001100	FF (Form Feed)	076	04C	01001100	L
013	00D	00001101	CR (Carriage Return)	077	04D	01001101	M
014	00E	00001110	SO (Serial In)	078	04E	01001110	N
015	00F	00001111	SI (Serial Out)	079	04F	01001111	O
016	010	00010000	DLE (Data Link Escape)	080	050	01010000	P
017	011	00010001	DC1 (XON) (Device Control 1)	081	051	01010001	Q
018	012	00010010	DC2 (Device Control 2)	082	052	01010010	R
019	013	00010011	DC3 (XOFF)(Device Control 3)	083	053	01010011	S
020	014	00010100	DC4 (Device Control 4)	084	054	01010100	T
021	015	00010101	NAK (Negative Acknowledgement)	085	055	01010101	U
022	016	00010110	SYN (Synchronous Idle)	086	056	01010110	V
023	017	00010111	ETB (End of Trans. Block)	087	057	01011111	W
024	018	00011000	CAN (Cancel)	088	058	01011000	X
025	019	00011001	EM	089	059	01011001	Y
026	01A	00011010	SUB	090	05A	01011010	Z
027	01B	00011011	ESC (Escape)	091	05B	01011011	[
028	01C	00011100	FS (File Separator)	092	05C	01011100	\
029	01D	00011101	GS	093	05D	01011101	]
030	01E	00011110	RS (Request to Send)	094	05E	01011110	^
031	01F	00011111	US	095	05F	01011111	_
032	020	00100000	SP (Space)	096	060	01100000	`
033	021	00100001	!	097	061	01100001	a
034	022	00100010	"	098	062	01100010	b
035	023	00100011	#	099	063	01100011	c
036	024	00100100	$	100	064	01100100	d
037	025	00100101	%	101	065	01100101	e
038	026	00100110	&	102	066	01100110	f
039	027	00100111	'	103	067	01100111	g
040	028	00101000	(	104	068	01101000	h
041	029	00101001	)	105	069	01101001	i
042	02A	00101010	*	106	06A	01101010	j
043	02B	00101011	+	107	06B	01101011	k
044	02C	00101100	,	108	06C	01101100	l
045	02D	00101101	-	109	06D	01101101	m
046	02E	00101110	.	110	06E	01101110	n
047	02F	00101111	/	111	06F	01101111	o
048	030	00110000	0	112	070	01110000	p
049	031	00110001	1	113	071	01110001	q
050	032	00110010	2	114	072	01110010	r
051	033	00110011	3	115	073	01110011	s
052	034	00110100	4	116	074	01110100	t
053	035	00110101	5	117	075	01110101	u
054	036	00110110	6	118	076	01110110	v
055	037	00110111	7	119	077	01110111	w
056	038	00111000	8	120	078	01111000	x
057	039	00111001	9	121	079	01111001	y
058	03A	00111010	:	122	07A	01111010	z
059	03B	00111011	;	123	07B	01111011	{
060	03C	00111100	<	124	07C	01111100	|
061	03D	00111101	=	125	07D	01111101	}
062	03E	00111110	>	126	07E	01111110	~
063	03F	00111111	?	127	07F	01111111	DEL

Tabella 3. Codisce ASCII

4.2 Interfaccia IEEE-488 (HP-IB o GP-IB)

L'interfaccia IEEE-488 è stata originariamente sviluppata nel 1965 dalla Hewlett Packard (HP) con il nome di interfaccia HP-IB (Hewlett Packard Interface Bus) per collegare e controllare la strumentazione HP. Nel 1975, l'Istituto degli ingegneri elettrici ed elettronici (IEEE) ha pubblicato la norma ANSI/IEEE Standard 488-1975 che descrive le specifiche meccaniche, elettriche e funzionali dell'interfaccia per la programmazione della strumentazione denominata IEEE-488 (nota anche come GPIB, *General Purpose Interface Bus*). Con lo scopo di aumentare la compatibilità e le modalità di configurazione nell'impiego del bus IEEE-488 è stato pubblicato un supplemento, IEEE-488.2, sui codici, formati, protocolli e comandi comuni impiegati dall'interfaccia IEEE-488 (qui rinominata IEEE-488.1). La IEEE-488.2 non sostituisce la IEEE-488.1 tanto che molti dispositivi sono standardizzati solo alla IEEE-488.1.

Nel 1990 nel protocollo IEEE-488.2 si inseriscono i comandi standard per la programmazione della strumentazione (SCPI *Standard Commands for Programmable Instrumentation*). Questa ulteriore standardizzazione garantisce la completa compatibilità nella programmazione di strumentazione diversa e prodotta da differenti costruttori.

I principali obiettivi per i quali lo standard IEEE-488 è stato sviluppato sono:

- definire un sistema di interconnessione tra strumenti su distanze limitate (rete locale);
- minimizzare le limitazioni che il sistema di interconnesione impone sulle prestazioni dei singoli strumenti;
- consentire uno scambio di informazioni con velocità elevate;
- rendere possibile l'integrazione tra strumenti di diversi costruttori.

Le caratteristiche elettriche essenziali dell'interfaccia IEEE488.1 sono:

- il bus è costituito da 8 linee per i dati (DIO1-8), 3 linee di *handshaking* e 5 linee di gestione interfaccia (figura 18);
- i messaggi sono trasferiti in modalità *byte* seriale, bit parallelo, asincrono controllato da *handshaking*;
- il codice utilizzato è ASCII a 7 bit più un bit di parità;
- la connessione tra i dispositivi può essere di tipo a stella o a festone (cascata) per un totale di una lunghezza massima di collegamento pari a $2 \cdot N$ metri dove N è il numero di dispositivi connessi. L'estensione totale non deve in ogni caso essere superiore a 20 m (lunghezza massima di un cavo 4 m);
- numero massimo di dispositivi collegabili al bus 15;
- velocità massima di trasmissione 1 Mbyte/s (velocità ordinarie comprese tra i 250 kbyte/s e 500 kbyte/s);
- la logica utilizzata è TTL Schottky compatibile *open collector* o *tri-state*. Tale struttura associata ad una logica negata consente la configurazione *wired-OR*. La configurazione *wired-OR* è essenziale, assieme all'utilizzo della logica negata (livello logico basso$\rightarrow \geq$2 V livello logico alto$\rightarrow \leq$0.8 V), per consentire a più *driver* di operare senza conflitti sulla stessa linea (*party-line*). Un esempio di configurazione *wired-OR* con *driver open-collector* è riportato in figura 19. La configurazione *wired-OR* (OR cablato) permette di effettuare l'operazione OR (somma) consentendo quindi di asserire una linea anche da parte di uno solo dei dispositivi connessi

Pin	Signal Name	Function	Pin	Signal Name	Function
1	DIO1	Data input/output bit 1	13	DIO5	Data input/output bit 5
2	DIO2	Data input/output bit 2	14	DIO6	Data input/output bit 6
3	DIO3	Data input/output bit 3	15	DIO7	Data input/output bit 7
4	DIO4	Data input/output bit 4	16	DIO8	Data input/output bit 8
5	EOI	End-or-identify	17	REN	Remote enable
6	DAV	Data valid	18	SHIELD	Ground (DAV)
7	NRFD	Not ready for data	19	SHIELD	Ground (NRFD)
8	NDAC	Not data accepted	20	SHIELD	Ground (NDAC)
9	IFC	Interface clear	21	SHIELD	Ground (IFC)
10	SRQ	Service request	22	SHIELD	Ground (SRQ)
11	ATN	Attention	23	SHIELD	Ground (ATN)
12	SHIELD	Chassis ground	24	SIGNAL GND	Signal ground

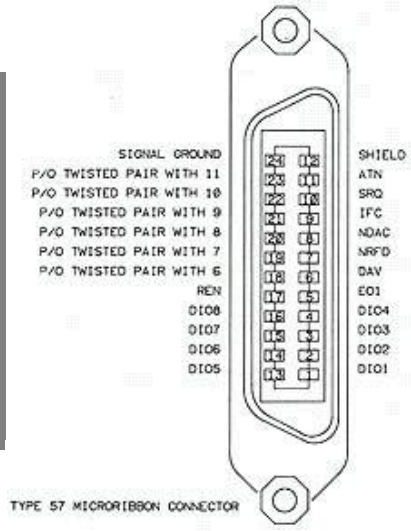


Figura 18. Connettore IEEE-488 con relativa tabella di connessione.

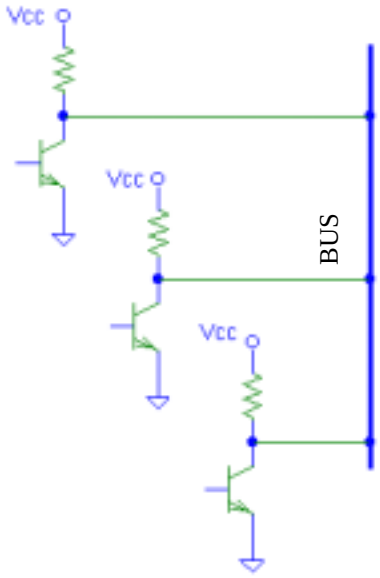


Figura 19. Connessione *wired-OR* in logica negata.

I messaggi immessi nel bus dati (DIO1-8) sono disponibili per tutti i dispositivi connessi attraverso l'interfaccia IEEE-488, poiché il sistema è di tipo parallelo. I messaggi possono essere dei codici di sistema standardizzati, comprensibili da tutti i dispositivi e quindi non dipendenti dal dispositivo che li ha messi a disposizione (tipo *device independent*), oppure dipendenti dallo strumento che li ha immessi sul bus (tipo *device dependent*), come ad esempio un dato corrispondente ad un valore di misura. Attraverso le linee di gestione dell'interfaccia si definisce il tipo di messaggio che è presente sul bus.

Per identificare in modo univoco i dispositivi connessi in una rete IEEE-488 è necessario assegnare ad ognuno di essi un indirizzo (detto indirizzo primario). Alcuni dispositivi sono composti da più sottosistemi che possono a loro volta essere indirizzati. In questo caso oltre all'indirizzo primario per identificare il sottosistema occorre fornire anche un indirizzo secondario (figura 20). Il numero massimo di indirizzi primari è 31 (realizzato con cinque bit) mentre il secondario è 961.

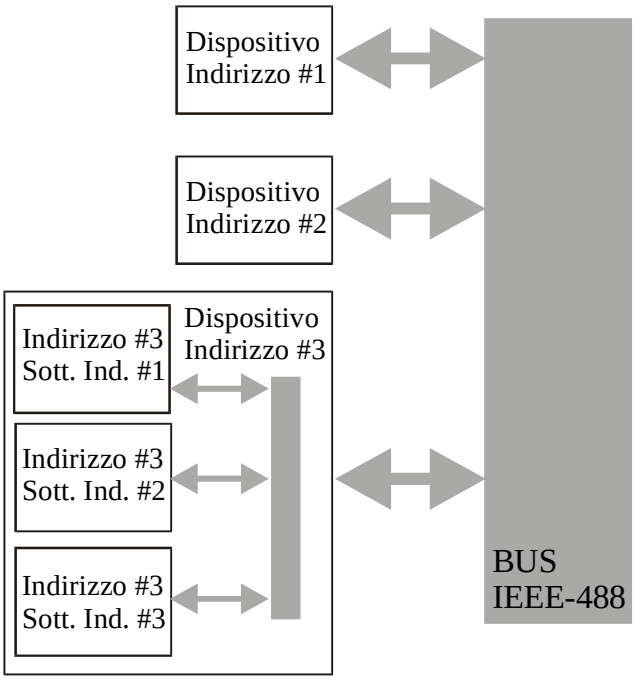


Figura 20. Struttura indirizzi primari e secondari.

Ognuno dei dispositivi collegati al bus può assumere uno dei tre ruoli (modalità) seguenti:

- ***LISTENER*** (ascoltatore - ricevitore). Tale ruolo è assegnato al dispositivo che deve ricevere i dati trasmessi sul bus. Il numero massimo di ricevitori può essere pari al numero di dispositivi connessi attraverso l'interfaccia IEEE-488 meno uno (il numero massimo è quindi uguale a 14).
- ***TALKER*** (parlatore - trasmettitore). Tale ruolo è assegnato al dispositivo che deve immettere i dati sul bus e renderli disponibili ai ricevitori (*listener*). Un solo trasmettitore alla volta può essere attivo in ciascuna delle fasi di comunicazione.
- ***CONTROLLER*** (controllore). È il dispositivo che gestisce la comunicazione sull'interfaccia IEEE-488. Il controllore indirizza e assegna ai dispositivi connessi sul bus i distinti ruoli, gestisce il corretto funzionamento della rete e le eventuali richieste di servizio dei dispositivi. In una rete possono esserci più dispositivi aventi il ruolo di controllore, ma solo uno può essere attivo in una determinata fase della gestione dell'interfaccia. Generalmente il PC assume il ruolo di controllore di sistema (*system controller*) che ha la gerarchia di controllo più alta.

In realtà è presente anche un quarto ruolo, detto ***IDLER*** (ozioso), nel quale il dispositivo non partecipa alle fasi di comunicazione, e quindi non rallenta le operazioni di trasferimento dati, ma è disponibile a leggere le informazioni di servizio del controllore come qualsiasi altro dispositivo.

Il funzionamento di una rete basata sulla IEEE-488 è possibile anche senza un controllore. In questo caso, la minima configurazione richiede un trasmettitore e un ricevitore (entrambi con il ruolo fisso - *talker only* e *listener only*). Tale configurazione vale per esempio quando uno strumento (*talker only*) trasferisce ad una stampante (*listener only*) i dati relativi alla misurazione effettuata.

4.2.1 Gestione della sincronizzazione (handshaking)

Durante una trasmissione sul bus IEEE-488, la gestione della sincronizzazione tra ricevitore, trasmettitore e controllore è realizzata impiegando le tre linee di *handshaking* (in logica *wired-OR*) DAV, *DAta Valid*; NRFD, *Not Ready For Data*; NDAC, *Not Data ACcepted*.

La linea DAV è gestita dal trasmettitore (*talker*) per validare i dati da lui immessi sul bus (asserire coincide in logica negata allo stato basso). Le linee NRFD e NDAC sono invece gestite dal ricevitore (*listener*). NRFD indica che il ricevitore non è pronto a ricevere dati; NDAC indica che il dato non è stato ancora ricevuto.

Per comprendere come avviene la sincronizzazione si riporta in figura 21 il diagramma temporale dello stato delle tre linee durante la trasmissione di un dato.

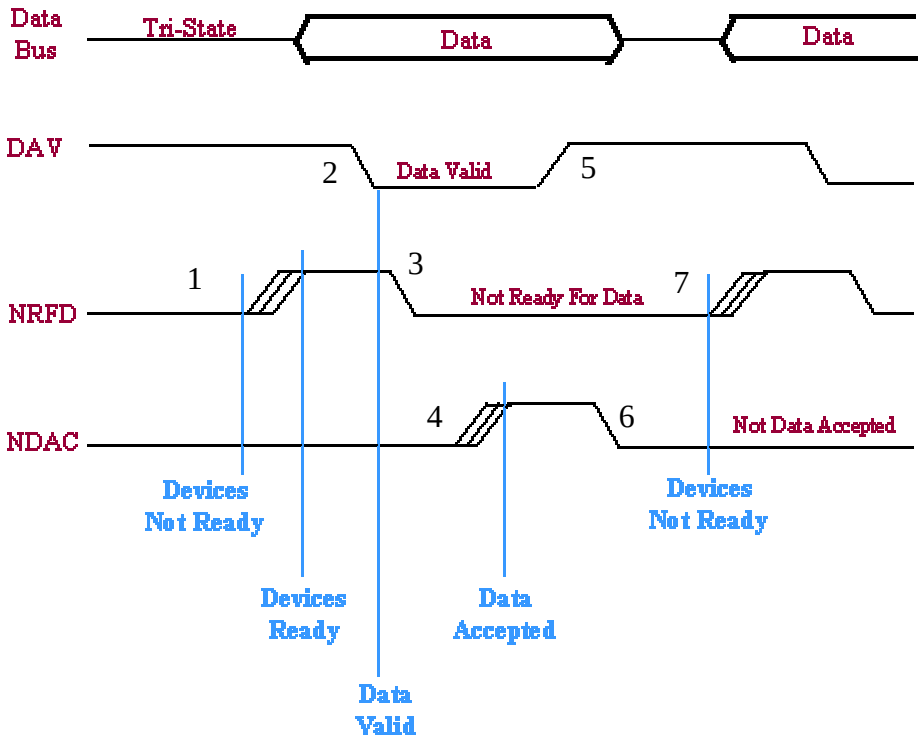


Figura 21. Andamento temporale del handshaking.

1. Quando tutti i ricevitori sono pronti a ricevere un dato la linea NRFD viene negata (stato logico alto).
2. Il trasmettitore inserisce sul bus i dati e li valida asserendo la linea DAV (stato logico basso).
3. Il primo ricevitore ad accettare il dato (il più veloce) asserisce la NRFD per indicare che non è più pronto a ricevere un dato.
4. L'ultimo ricevitore ad accettare il dato (il più lento) nega la NDAC per indicare al trasmettitore che tutti i ricevitori hanno accettato il dato.
5. In questa fase il trasmettitore nega la DAV per indicare che i dati presenti sul bus non sono più validi.

6/7. Il ricevitore più veloce asserisce la NDAC e successivamente nega la NRFD per indicare al trasmettitore che è pronto a ricevere il dato successivo.

Dal diagramma di figura 21 è evidente come la durata del ciclo di trasmissione di un dato sia limitata dal dispositivo più lento. È importante, quindi, impostare tutti i dispositivi che non

partecipano al ciclo di trasmissione nel ruolo ozioso (*idler*) per non limitare la velocità della trasmissione.

4.2.2 Messaggi unilinea

Sono i messaggi che vengono trasmessi attraverso le cinque linee di gestione del bus IEEE-488. Cinque sono quindi i messaggi unilinea: ATN, *ATteNtion*; EOI, *End Or Identify*; REN, *Remote ENable*; IFC, *InterFace Clear*; SRQ, *Service ReQuest*. Poiché la gestione del bus di interfaccia IEEE-488 è realizzata dal controllore (*system controller* o *controller*), le linee di gestione sono quasi esclusivamente utilizzate dal controllore.

ATN. È asserito dal controllore per indicare ai dispositivi collegati al bus IEEE-488 che sta immettendo dei messaggi *device independent* (ad esempio un indirizzo con relativa assegnazione di ruolo). Tutti i dispositivi partecipano indifferentemente alla fase di handshaking successiva (anche i dispositivi oziosi). ATN è negato per permettere al trasmettitore (*talker*) di trasmettere i dati sul bus (a questa fase non partecipano i dispositivi oziosi).

EOI. È asserito dal trasmettitore (*talker*) simultaneamente all'invio dell'ultimo byte trasferito sul bus IEEE-488 per indicare la fine del messaggio. Quando però ATN è asserito, la linea EOI è gestita dal controllore per rilevare quale dispositivo è responsabile di una richiesta di servizio attraverso un *polling* di tipo parallelo (si veda il paragrafo gestione delle richieste di servizio).

REN. È asserito solo dal *system controller* per impostare gli strumenti che riconoscono questo messaggio in modalità programmazione remota (*remote*).

IFC. È asserito solo dal *system controller* per impostare tutti i dispositivi collegati all'interfaccia IEEE-488 in una configurazione iniziale di "azzeramento".

SRQ. È simile ad un *interupt*. Può essere asserito da qualsiasi dispositivo per richiedere al controllore una determinata azione (richiesta di servizio).

4.2.3 Messaggi multilinea

Sono i messaggi che vengono trasmessi attraverso le otto linee del bus dati. Tali messaggi possono essere dei comandi standardizzati (*device independent*), che devono quindi essere compresi da tutti gli strumenti con interfaccia IEEE-488, oppure dei dati (*device dependent*), cioè informazioni con significato dipendente dal tipo di strumento che li ha immessi sul bus. Come appena descritto nel paragrafo dei messaggi unilinea, è il comando ATN che stabilisce se il messaggio è un comando (ATN asserito) o un dato (ATN negato).

I comandi sono distinti in: comandi di indirizzamento, comandi universali (*broadcast*), comandi secondari e comandi indirizzati.

Comando di *Indirizzamento*- Consiste nel comando corrispondente all'indirizzo di un dispositivo connesso al bus IEEE-488 e alla sua assegnazione ad un ruolo specifico (ricevitore o trasmettitore). Degli otto bit di dati il più significativo è riservato al controllo di parità, il secondo e il terzo stabiliscono il ruolo (01 *listener* 10 *talker*) mentre i restanti cinque bit sono riservati all'indirizzo.

Comandi *Universali* (*broadcast*)- Sono sei: UNT (*UNTalk*), UNL (*UNListen*), LLO (*Local Lock Out*), SPE (*Serial Poll Enable*), SPD (*Serial Poll Disable*), PPU (*Parallel Poll Unconfigure*).

UNT - Disindirizza il trasmettitore attivo il quale si porta nello stato ozioso.

UNL - Disindirizza i ricevitori attivi i quali si portano nello stato ozioso.

LLO - Disabilita l'impostazione manuale del dispositivo da pannello di controllo (per i dispositivi in grado di riconoscere tale comando).
SPE - Abilita i dispositivi connessi ad una risposta sequenziale durante l'interrogazione da parte del controllore per verificare una richiesta di servizio.
SPD - Termina la sequenza delle operazioni di *polling seriale*.
PPU - Disabilita tutti gli strumenti ad una risposta parallela durante l'interrogazione da parte del controllore per verificare una richiesta di servizio.

Comandi *Indirizzati*- Sono recepiti solo dai dispositivi precedentemente indirizzati come ricevitori. Sono cinque: GET (*Group Execute Trigger*), SDC (*Seleted Device Clear*), GTL (*Go To Local*), PPC (*Parallel Poll Configure*), TCT (*Take ConTrol*).
GET - Da inizio all'attività dei ricevitori.
SDC - Azzera lo stato (valore predefinito) dei ricevitori attivi.
GTL - Abilita l'impostazione manuale a pannello degli strumenti precedentemente disabilitati dal comando universale LLO.
PPC - Configura il ricevitore per istruirlo sulla modalità di risposta durante una interrogazione per una richiesta di servizio di tipo parallelo.
TCT - È usato dal controllore per trasferire il controllo del bus ad un altro strumento.

Comandi di *Secondari*- A questa categoria di comandi appartengono l'indirizzo secondario di un dispositivo costituito da più sottosistemi, e i comandi PPE (*Parallel Poll Enable*) e PPD (*Parallel Poll Disable*). Sono usati in unione con i codici di indirizzamento e i codici di comando.
PPE - Abilita il dispositivo che precedentemente ha ricevuto un comando di PPC a rispondere alla interrogazione parallela per una richiesta di servizio.
PPD - Disabilita il dispositivo che precedentemente ha ricevuto un comando di PPC a rispondere alla interrogazione parallela per una richiesta di servizio..

4.2.4 Gestione delle richieste di servizio

Se durante una fase della gestione del bus IEEE-488 un dispositivo necessita di attenzione da parte del controllore per un evento di emergenza (errore) o perché ad esempio un dispositivo è pronto a trasferire dei dati, il dispositivo asserisce la linea SRQ. Il controllore allora avvia una procedura di interrogazione (*polling*) dei dispositivi allo scopo di individuare quello che ha asserito la linea SRQ. Due sono le procedure possibili nello standard IEEE-488: interrogazione sequenziale (*Serial Polling*) o interrogazione parallela (*Parallel Polling*).

Polling Seriale

In questa modalità i dispositivi generano un byte detto di stato (*status byte*) nel quale lo stato del settimo bit indica se vi è stata una richiesta di servizio (alto equivale alla richiesta), mentre gli altri bit indicano, a seconda dello strumento, il motivo della richiesta (deve essere interpretato dal controllore). Durante il *polling* seriale, il controllore interroga sequenzialmente i dispositivi connessi al bus analizzando lo stato del settimo bit dello *status byte*, fino a quando non trova il dispositivo che ha asserito la linea SRQ. A questo punto analizza tutto lo *status byte* e attiva una fase di comunicazione con il dispositivo per individuare il motivo della richiesta di servizio. La procedura termina con un comando di SPD.

Polling Parallelo

La procedura sequenziale ha l'indubbio svantaggio di essere lenta. Per ovviare a tale situazione è possibile individuare il dispositivo che ha richiesto il servizio con una procedura

parallela (una sola interrogazione). Non tutti i dispositivi hanno la possibilità di implementare la procedura di *polling parallelo*. Tale procedura richiede una fase iniziale di configurazione (PPC) nella quale ad ogni dispositivo che implementa il *polling parallelo* viene assegnata una particolare linea del bus dati che il dispositivo dovrà attivare durante l'interrogazione parallela. Il numero della linea dedicata al dispositivo viene inviato attraverso il comando PPE nel quale gli ultimi tre bit indicano il numero di linea. Poiché le linee dati sono otto, il numero massimo di strumenti configurabili in modalità *polling parallelo* è otto. è comunque possibile combinare sull'interfaccia IEEE-488 una modalità di polling mista, seriale e parallela, quando il numero di dispositivi collegati è superiore a otto.