

*Fondamenti sugli standard di
interfaccia RS-232 e IEEE-488*

1 Introduzione

Come noto, un dispositivo di interfaccia permette il colloquio tra la struttura hardware di un calcolatore e uno o più dispositivi esterni. Uno degli obiettivi principali di un'interfaccia consiste nel fornire un mezzo per il trasferimento di dati, garantendo livelli accettabili di affidabilità, velocità, flessibilità e, naturalmente, costo.

Con il termine standard sono indicate un insieme di specifiche tecniche sia fisiche, sia funzionali, sia software che permettono di definire l'interfaccia e ne consentono un corretto utilizzo.

La realizzazione delle interfacce di trasmissione si basa, principalmente, su due tecniche di trasferimento dati:

Trasferimento *seriale*: viene trasferito un bit per volta

Trasferimento in *parallelo*: viene trasferito un insieme di bit per volta

La scelta tra l'una e l'altra è legata a diversi fattori (tipo di dati trasmessi, dispositivi collegati, ...) e viene effettuata tenendo in considerazione pregi e difetti delle due soluzioni: la maggior velocità del trasferimento in parallelo è accompagnata da una maggior complessità della struttura di controllo e, spesso, da problemi di affidabilità su collegamenti per distanze elevate, per esempio superiori a qualche metro. Per contro, la trasmissione seriale offre, in genere, maggiore semplicità nella realizzazione del canale trasmissivo, a parità di tecnologia impiegata (cavo elettrico, fibra ottica,...), superiore distanza tra trasmettitore e ricevitore, al prezzo di una minore efficienza di trasmissione.

Nella letteratura tecnica si incontrano parecchie proposte di standard di interfaccia, sia nel caso di comunicazioni seriali, sia nel caso di trasferimenti in parallelo. Nel seguito verranno presi in esame due tra le più diffuse modalità per la connessione di strumenti elettronici di misura ad un calcolatore:

Lo standard RS-232

Lo standard IEEE 488

2 Lo Standard RS-232

Si tratta di una modalità di trasmissione molto utilizzata nel collegamento fra unità centrale e periferiche di varia natura, ad esempio stampanti, modem, dispositivi di puntamento, strumentazione elettronica, etc.. Tale modalità di comunicazione richiede una struttura del trasmettitore e del ricevitore molto semplice, mentre il canale fisico di trasmissione può essere costituito nel caso più semplice da soli tre conduttori, due per la trasmissione bidirezionale dei segnali ed il terzo per il riferimento dei potenziali elettrici. Inoltre le distanze percorribili con questo metodo di trasmissione sono, in generale, sufficienti per le normali esigenze di un sistema di misura o nella pratica industriale, dal momento che è possibile coprire facilmente una decina di metri.

2.1 Struttura dei cavi di connessione

La connessione tra periferica e calcolatore secondo lo standard RS-232 è del tipo “punto a punto” (Fig. 1): questo significa che per consentire la comunicazione sono richiesti cavi e connettori distinti per ogni coppia periferica-calcolatore (Fig. 1).

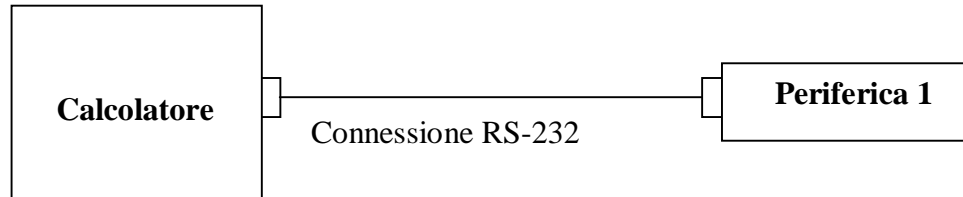


Fig. 1 Connessione secondo lo standard RS-232 (connessione “punto a punto”)

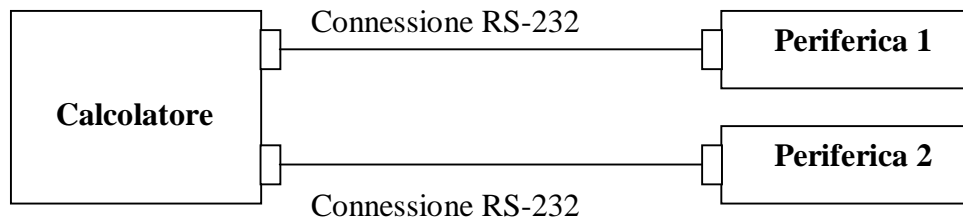


Fig. 2 Connessione secondo lo standard RS-232 nel caso di più periferiche

Il cavo di collegamento tra periferica e calcolatore è costituito da più conduttori: lo standard fissa il significato assunto da ciascuno di essi. Al contrario, non stabilisce univocamente un tipo di connettore da utilizzare, ma il modello DB-25 a 25 pin (Fig. 3) è quello più usato, insieme alla versione minima DB-9 a 9 pin (Fig. 4).



Fig. 3 Connettore RS-232 DB-25 (maschio e femmina)



Fig. 4 Connettore RS-232 DB-9 (maschio e femmina)

Il connettore femmina dovrebbe, di norma, essere associato a DCE (Data Communication Equipment), cioè la **periferica**, il connettore maschio a DTE (Data Terminal Equipment), il **computer**. La seguente tabella elenca il ruolo assegnato dallo standard a ciascuna terminazione del connettore:

DB-25 Pin#	DB-9 Pin#	Nome comune	Direzione DTE-DCE	Nome formale
1		FG	-	Frame Ground
2	3	TD	→	Transmitted Data, TxD
3	2	RD	←	Received Data, RxD
4	7	RTS	→	Request to Send
5	8	CTS	←	Clear To Send
6	6	DSR	←	Data Set Ready
7	5	SG	-	Signal Ground, GND
8	1	DCD	←	Data Carrier Detect
9		--	-	+P
10		--	-	-P
11		--	-	Non assegnato
12		SDCD	←	Secondary Data Carrier Detect
13		SCTS	←	Secondary Clear To Send
14		STD	→	Secondary Transmitted Data
15		TC	←	Transmission Signal Element Timing
16		SRD	←	Secondary Received Data
17		RC	→	Receiver Signal Element Timing
18		--	-	Non assegnato
19		SRTS	→	Secondary Request To Send
20	4	DTR	→	Data Terminal Ready
21		SQ	←	Signal Quality detector
22	9	RI	←	Ring Indicator
23		--	→	Data Signal Rate Selector
24		--	←	Transmitter Signal Element Timing
25		--	-	Non assegnato

Tab. 1 Segnali associati ai pin dei connettori DB-25 e DB-9 dallo standard RS-232

Qui di seguito si descrive brevemente la funzionalità associata ai segnali inviati su ciascuno dei conduttori che costituiscono il cavo di connessione.

Nota: Si osservi che un segnale è considerato attivo quando è a livello logico 1. La corrispondenza tra tensione e livelli logici è descritta in un paragrafo seguente.

Protective Ground

Normalmente collegato alla struttura esterna di uno dei dispositivi, il DCE o il DTE, e opportunamente collegato a terra. L'unico scopo di tale connessione è di proteggere il sistema da shock elettrici accidentali. È considerato opzionale.

Transmit Data

Linea di trasmissione dei bit di informazione dal DTE (periferica) a DCE (computer). Il DTE mantiene tale linea al valore logico 1 quando *non* ci sono dati da trasmettere; la trasmissione del dato su questa linea è possibile solo se i segnali *Request To Send*, *Clear To Send*, *Data Set Ready* e *Data Terminal Ready*, quando presenti, assumono valore logico 0.

Receive Data

Linea di trasmissione dei bit di informazione dal DCE (computer) a DTE (periferica). Il dato (bit) primario viene inviato su questa linea dal DCE al DTE. Questo segnale viene mantenuto ad un valore logico 1 quando DCE non trasmette dati e viene portato a 0 per un breve intervallo di tempo dopo una transizione della linea *Request To Send* da 1 a 0, per consentire il completamento della trasmissione.

Request To Send

Abilita i circuiti di trasmissione. DTE utilizza questo segnale quando intende trasmettere dati a DCE. Questo segnale, in combinazione con *Clear To Send*, coordina il trasferimento dati da DTE a DCE. Un valore logico 0 su questa linea mantiene DCE in modalità di trasmissione; DCE riceverà i dati da DTE e li trasmetterà attraverso il canale di comunicazione. Una transizione da 1 a 0 su questa linea segnala a DCE di completare la trasmissione dati in corso e di portarsi nella modalità di ricezione.

Clear To Send

Segnale di risposta a DTE. Quando attivo, indica a DTE che la trasmissione può iniziare (sulla linea *Transmit Data*). Se CTS è attivo contemporaneamente ai segnali *Request To Send*, *Data Set Ready* e *Data Terminal Ready* i dati provenienti da DTE vengono inviati lungo il canale di trasmissione. La non attività del segnale CTS viene interpretata da DTE come non disponibilità di DCE a ricevere dati (DTE, quindi, attende ad inviare dati).

Data Set Ready

Con questa linea DCE avvisa DTE che il canale di comunicazione è disponibile, cioè che DCE è pronto a trasmettere o a ricevere.

Signal Ground

Riferimento di tensione per tutti gli altri segnali.

Receive Line Signal Detect (or Data Carrier Detect)

DCE utilizza questa linea per segnalare a DTE che sta ricevendo un “buon segnale”, cioè una portante analogica in grado di assicurare una demodulazione dei dati ricevuti priva di errori.

+P

E' mantenuto ad una tensione di +12 Volts DC a scopo di test.

-P

E' mantenuto ad una tensione di -12 Volts DC a scopo di test.

Secondary Receive Line Signal Detect

Questo segnale è attivo quando il canale di comunicazione secondario sta ricevendo un “buon segnale”, cioè una portante analogica in grado di assicurare una demodulazione dei dati ricevuti priva di errori. Questo segnale è analogo a *Receive Line Signal Detect*.

Secondary Clear To Send

Segnale di risposta a DTE. Quando attivo, indica a DTE che può dare inizio alla trasmissione sul canale secondario (linea Transmit Data)

Secondary Transmitted Data

Linea di trasmissione dati secondaria, utilizzata sia da DTE a DCE, sia da DCE a ATE. Questo segnale equivale a *Transmitted Data*.

Transmission Signal Element Timing

Segnale di clock inviato da DCE a DTE, in modo che DTE sia in grado di sincronizzare il proprio circuito di output che pilota la linea Transmitted Data. La frequenza del segnale di clock dipende dal bit- rate associato alla linea Transmitted Data. La transizione da 1 a 0 denota il punto centrale del tratto di segnale corrispondente ad un bit sulla Transmitted Data.

Secondary Receive Data

Linea impiegata per l'invio dei dati (bit) da DCE a DTE. Quando il canale secondario è utilizzato a scopo diagnostico o per interrompere il flusso di dati sul canale primario, questo segnale non viene fornito.

Receiver Signal Element Timing

Segnale di clock inviato da DCE a DTE in modo che DTE sia in grado di sincronizzare il proprio circuito di ricezione che pilota la linea Received Data. La frequenza del segnale di clock dipende dal bit-rate della trasmissione sulla linea Received Data. La transizione da 1 a 0 indica il punto centrale del tratto di segnale corrispondente ad un bit sulla Received Data.

Secondary Request To Send

Richiesta di dati effettuata sul canale secondario da DTE e rivolta a DCE. Questo segnale equivale a Request To Send.

Data Terminal Ready

Se questo segnale è a livello logico 1, DCE viene informato che DTE è pronto per la ricezione. Il segnale DTR deve essere attivo prima che DCE attivi il segnale Data Set Ready, indicando così di essere connesso al canale di comunicazione. Se il segnale DTR assume il valore logico 0, DCE interrompe la trasmissione in corso.

Signal Quality Detector

Linea usata da DCE per indicare se c'è o meno una elevata probabilità che si verifichi un errore nella ricezione dei dati. Viene posta ad un valore logico 0 se la probabilità di errore è elevata.

Ring Indicator

Linea usata da DCE per segnalare a DTE che sta per giungere una richiesta di collegamento. Il segnale *Ring Indicator* viene mantenuto sempre a livello logico 0, tranne quando DCE riceve un segnale di chiamata in arrivo.

Data Signal Rate Selector

Linea utilizzata per selezionare il bit-rate di trasmissione del DCE. In caso di connessione sincrona il bit-rate può assumere uno tra due valori possibili; se la trasmissione è di tipo asincrono, il bit-rate può assumere un valore compreso all'interno di due intervalli specificati dallo standard.

Transmitter Signal Element Timing

Linea usata da DTE per inviare a DCE un segnale di clock. La transizione da 1 a 0 indica il punto centrale del tratto di segnale corrispondente ad un bit sul *Transmitted Data*.

Nota: i nomi dei segnali assegnati ai pin sono gli stessi per DCE e per DTE. *Transmit Data* è una linea di trasmissione nel punto di connessione del cavo RS-232 a DTE e di ricezione

nel punto di connessione del cavo RS-232 a DCE; *Data Set Ready* è una linea di ricezione nel DTE e di trasmissione nel DCE, e così via.

2.2 Caratteristiche elettriche dei segnali

Lo standard RS-232 fissa anche le caratteristiche elettriche dei segnali relativi ai dati e al controllo della trasmissione, secondo le specifiche di seguito riportate.

Livelli di tensione definiti nello standard

Si ricorda che i segnali di controllo sono: *Transmit Data*, *Receive Data*, *Secondary Transmit Data*, *Secondary Receive Data*.

I segnali relativi ai dati sono: *Request To Send*, *Clear To Send*, *Data Set Ready*, *Data Carrier Detect*, *Secondary Data Carrier Detect*, *Secondary Clear To Send*, *Secondary Request To Send*, *Data Terminal Ready*, *Ring Indicator*, *Data Signal Rate Selector*.

Tipo segnali	Estremo	Commento	Livello logico 0	Livello logico 1
Dati				
	Trasmittitore	Tensione imposta	Da 5 a 15 Volts	Da -5 a -15 Volts
	Ricevitore	Tensione letta	Da 3 a 25 Volts	Da -3 a -25 Volts
Controllo				
	Trasmittitore	Tensione imposta	Da -5 a -15 Volts	Da 5 a -15 Volts
	Ricevitore	Tensione letta	Da -3 a -25 Volts	Da 3 a 25 Volts

Tab. 2 Livelli di tensione definiti dallo standard RS-232

Margine di rumore

Le tensioni lette in corrispondenza al ricevitore sono diverse da quelle imposte al trasmettitore: tale definizione dei livelli di tensione permette di compensare le cadute di tensione lungo il cavo (canale di trasmissione). I segnali, infatti, nella propagazione lungo il mezzo di trasmissione sono soggetti ad attenuazione e distorsione di entità proporzionale alla lunghezza del cavo. Questi effetti sono dovuti per lo più alla capacità elettrica associata al cavo. Lo standard impone una capacità massima di carico pari a 2500 pF: poiché un metro di cavo presenta tipicamente una capacità di circa 130 pF, la lunghezza massima accettabile risulta attorno ai 17 m. Chiaramente si tratta di un calcolo estremamente approssimato; nella pratica si riescono a coprire distanze massime dell'ordine dei 30 m, purché si impieghino cavi con bassa capacità oppure nel caso di ridotte velocità di trasmissione o ancora impiegando tecniche software di correzione degli errori di trasmissione.

2.3 Compatibilità RS-232 di un dispositivo

Mentre alcuni segnali nelle interfacce RS-232 sono universalmente implementati nei microcomputer, altri sono gestibili arbitrariamente da parte dei costruttori. Affinché un dispositivo si possa definire compatibile con lo standard devono essere rispettate le seguenti caratteristiche:

Caratteristiche elettriche

Livelli di tensione associati ai valori logici 0 e 1

Funzioni associate ad alcuni conduttori (Pin# 2, Pin# 3, Pin# 7)

Uso di un terminale come DTE e non come DCE

Uso di un modem come un DCE e non come un DTE

3 Lo standard 488

3.1 Introduzione

La semplicità costruttiva e di controllo delle interfacce basate sul protocollo RS-232 è il motivo principale per il quale esse sono state utilizzate per prime allo scopo di dotare strumenti numerici della capacità di comunicare con un'unità centrale. Si deve però ricordare che il sistema di comunicazione seriale non è stato progettato tenendo presente le caratteristiche offerte dagli strumenti di misura, quanto piuttosto in vista della connessione tra un calcolatore ed una periferica con funzionalità più limitate (ad esempio un plotter o un modem).

Ne consegue che alcuni strumenti, se collegati ad un calcolatore per mezzo di una linea seriale, non vengono sfruttati al pieno delle loro capacità, sia in termini di funzionalità disponibili, sia in termini di velocità di trasmissione.

Per superare le precedenti difficoltà sono stati proposti vari tipi di canali di interfaccia, organizzati appositamente per la gestione di strumenti di misura. Tra queste riveste un ruolo di particolare importanza l'interfaccia IEEE 488, nota anche come protocollo GP-IB (General Purpose Interface Bus) o ancora come HP-IB, dal nome di due costruttori che hanno tra i primi aderito allo standard. La linea di collegamento tramite la quale più dispositivi elettronici interconnessi possono comunicare tra loro è un bus: storicamente è stato progettato dai tecnici della Hewlett-Packard per permettere il controllo e lo scambio di dati a distanza tra gli strumenti programmabili HP. Tale bus è stato subito impiegato nella connessione tra computer e strumenti anche non HP, per la sua elevata velocità massima nel trasferimento dei dati (fino a 1Mbyte/sec), lo standard successivamente è stato ridefinito con il nome "IEEE 488.1"o, per brevità, "488.1". Esso stabilisce le caratteristiche fisiche del bus di collegamento e descrive una serie di funzioni di interfaccia che i costruttori devono implementare sui dispositivi perché questi possano utilizzare il bus nella trasmissione dei dati. La documentazione corrispondente elenca quindi tutti i comandi utilizzabili sul bus per il controllo del sistema di interfaccia e la gestione della trasmissione dei dati.

Nello standard 488.1 le funzioni di interfaccia sono chiaramente distinte dalle funzioni svolte dai dispositivi; i costruttori sono liberi di implementare queste ultime a loro

discrezione utilizzando, per il comando dei dispositivi, opportune istruzioni inviate sul bus, le quali possono variare a seconda del dispositivo specifico.

Una successiva revisione dello standard ha avuto come risultato una nuova versione dello stesso, denominata IEEE 488.2. In tale nuovo insieme di norme, sono definiti anche i protocolli per la comunicazione tra i dispositivi e l'host, il formato della trasmissione dei dati sul bus ed infine sono fornite utili linee guida alla programmazione dei dispositivi. A tale proposito si ricorda la presenza delle indicazioni fornite dal consorzio SCPI (Standard Commands for Programmable Instruments), le quali sono perfettamente compatibili con le norme dello standard 488.2. Le indicazioni fornite da SCPI riguardano la sintassi e il significato associato ai comandi scambiati tra host e strumenti.

3.2 Caratteristiche fisiche

Generalità

La struttura del sistema è del tipo a bus in modo da consentire la trasmissione in parallelo di più bit; risulta semplice, inoltre, apportare variazioni nel numero di strumenti collegati al sistema. Il bus è realizzato raggruppando tra loro 24 linee in un unico cavo (party-line bus). Per raggiungere il massimo delle prestazioni in termini di velocità di trasferimento dei dati sul bus (1 Mbyte/s), la distanza tra i dispositivi e la lunghezza dei cavi di collegamento tra gli stessi devono essere limitate. In particolare si consiglia:

Una distanza massima di 4 metri tra un dispositivo ed un altro e una distanza media di 2 metri tra i dispositivi

Una lunghezza totale del bus non superiore a 20 m

Non più di 15 dispositivi connessi al bus di cui almeno due terzi autoalimentati, cioè tali da non prelevare l'alimentazione dal bus stesso

Tenendo presenti questi vincoli si può organizzare un sistema di misura automatico avente una discreta complessità, con strumenti disposti non troppo lontani gli uni dagli altri, in modo non eccessivamente sparso. Si noti che la limitazione di distanza è in realtà solo apparente. Infatti esistono sul mercato opportuni dispositivi (REPEATER) che consentono di realizzare collegamenti più lunghi. Al limite, attraverso convertitori di protocollo 488-ETHERNET, è possibile connettere due sottostazioni di strumenti, collegate al bus 488, mediante rete ethernet, in modo da coprire distanze dell'ordine delle diverse decine di metri.

Le connessioni

I connettori 488 hanno una particolare struttura meccanica, rigorosamente stabilita dallo standard, e presentano 24 terminazioni o pin. Il connettore su uno strumento è di tipo "femmina", mentre i due connettori alle estremità dei cavi di collegamento sono entrambi sia "maschio", sia "femmina", rendendo semplice la realizzazione di connessioni multiple. Come illustrato in figura sono possibili, principalmente, due topologie di collegamento: *lineare* (Fig. 5 A) e *a stella* (Fig. 5 B).

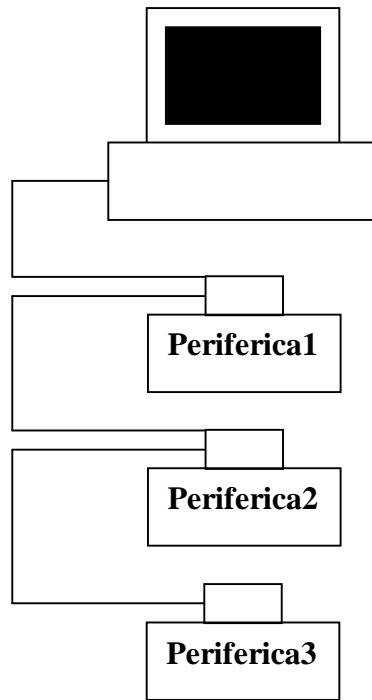


Fig. 5. A Topologia di connessione lineare sul bus 488

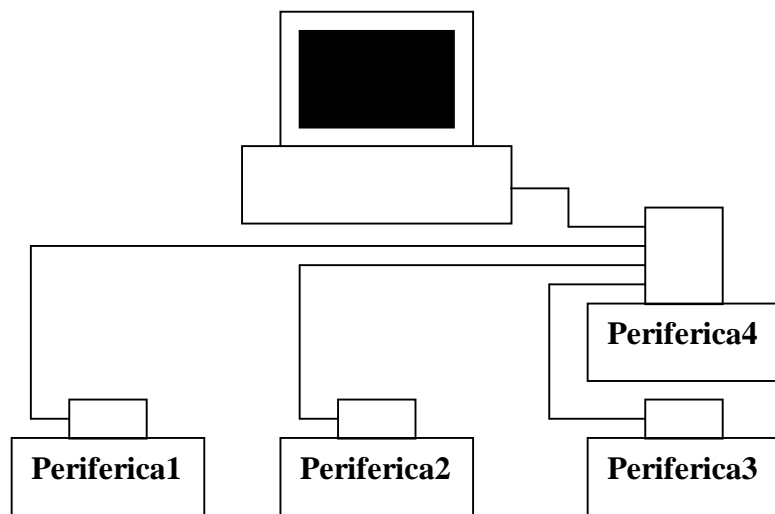


Fig. 5. B Topologia di connessione a stella sul bus 488

Sono anche disponibili topologie risultanti dall'unione delle due precedenti (Fig. 6):

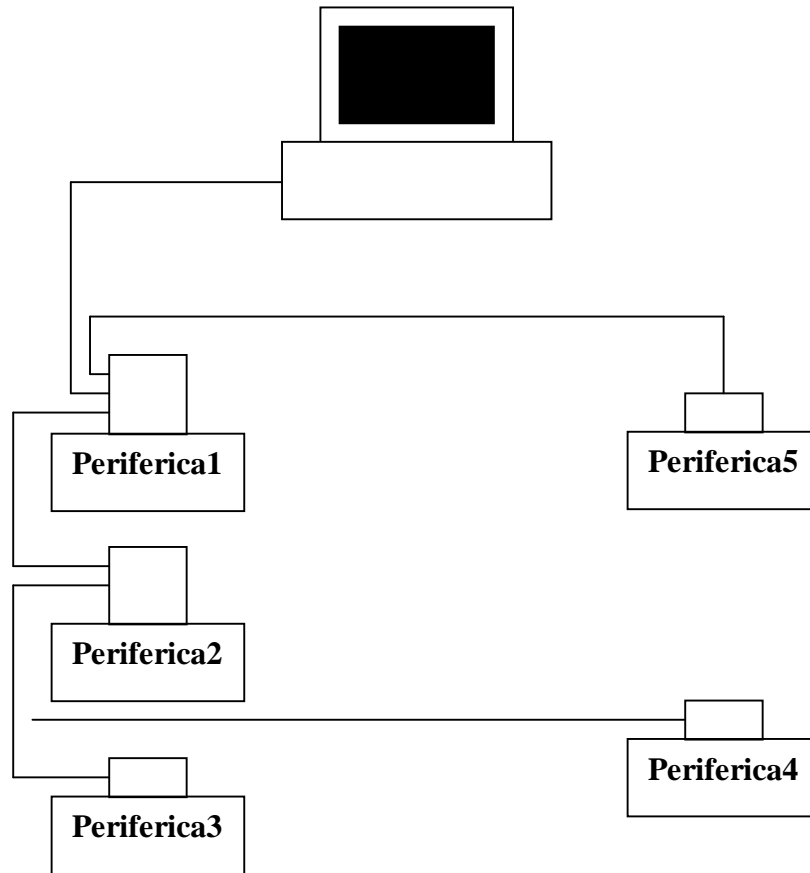


Fig. 6 Esempio di connessione in topologia “mista”

La stabilità meccanica delle connessioni è assicurata, oltre che dall’elasticità dei contatti, anche da opportune viti di fissaggio. Un altro tipo di connettore, più semplice e molto diffuso, è spesso accettato. E’ il connettore IEC 625, il quale presenta 25 pin: per usare tale connettore risulta necessario l’uso di un adattatore.

Da un punto di vista logico (Fig. 7) il bus può essere suddiviso in *data bus* e *control bus*; la trasmissione è di tipo asincrono, scandita, cioè, da procedure di inizio e fine trasmissione rigorosamente fissate dallo standard. Tali procedure richiedono l’impiego di tre segnali di handshaking (presenti sul control bus).

IL BUS 488

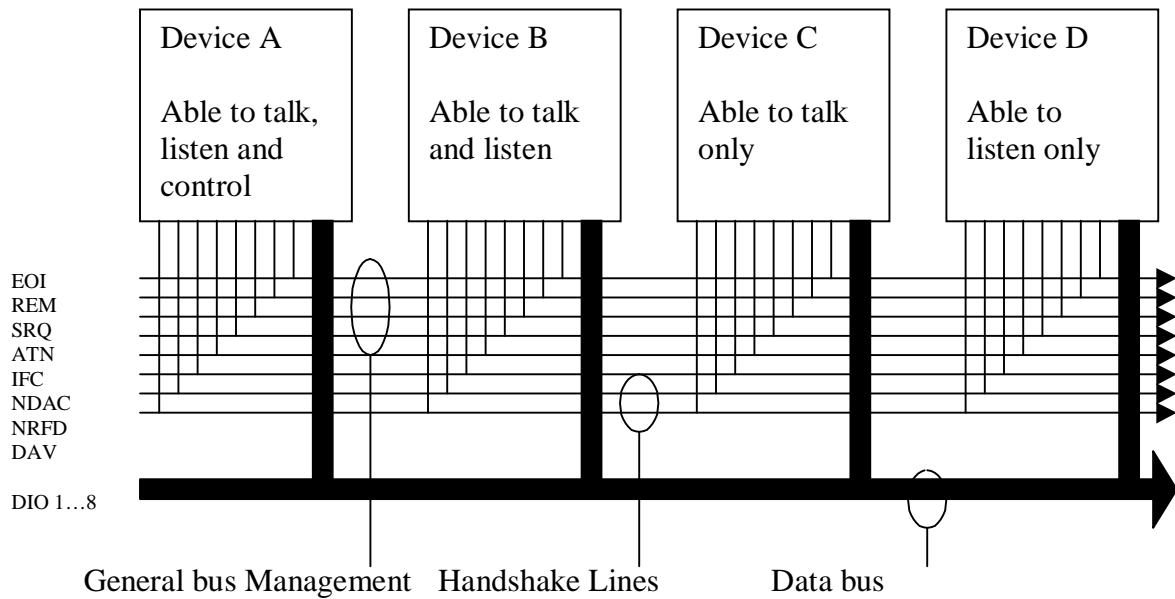


Fig. 7 Struttura del bus 488

Delle 24 linee (o 25, nel caso di connettori IEC 625) che formano il cavo di collegamento, 8 sono utilizzate come linee dati (Data I/O: DIO), 8 come linee di controllo, una serve da riferimento dei potenziali per i vari segnali, una è impiegata come linea di sicurezza per il collegamento a terra degli involucri degli strumenti. Le restanti 6 linee (o 7) realizzano una parziale funzione di schermo per attenuare gli effetti dei rumori di varia origine, che possono alterare l'informazione trasmessa. La funzione di protezione è ottenuta avvolgendo ognuna di queste restanti linee con una linea del control bus in modo da ridurre in particolare i flussi e le forze elettromotrici concatenate.

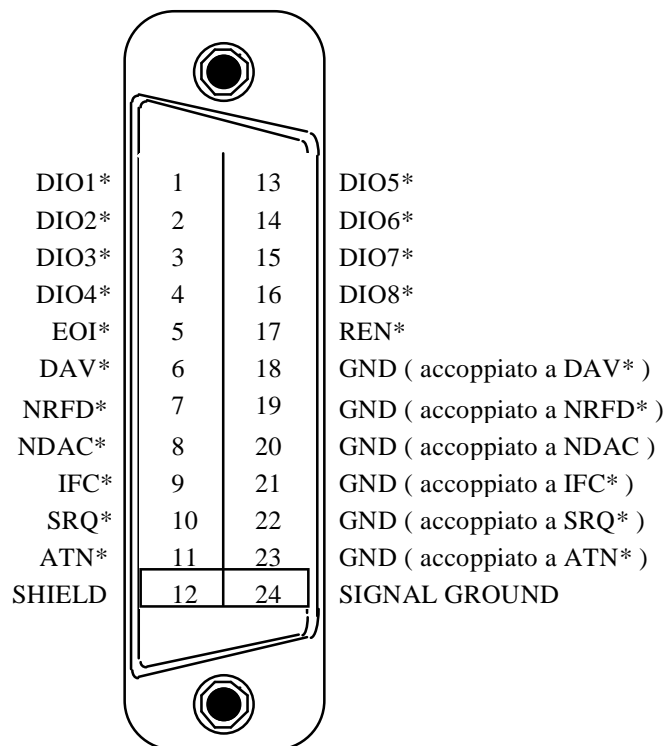


Fig. 8 Disposizione dei pin nel connettore HP-IB a 24 poli

La disposizione dei vari segnali sul connettore è dovuta ad alcuni accorgimenti:

Le linee del control bus che richiedono una maggiore affidabilità di trasmissione sono collocate in prossimità della linea utilizzabile come schermo

Le linee utilizzate come riferimento dei potenziali o per sicurezza sono poste ad un'estremità del connettore in modo da attenuare l'effetto che le correnti in esse circolanti possono avere sulle altre linee del bus

Le linee non schermate si trovano ad una distanza maggiore dalla linea di massa, fonte probabile di disturbi

3.3 Caratteristiche elettriche

Generale

I livelli di tensione adottati per lo scambio di informazioni sul bus 488 sono compatibili TTL. La relazione tra valori logici e di tensione è specificata nella seguente tabella:

<i>Livello logico</i>	<i>Livello di tensione</i>
0 (False)	> +2.4 Volts (High)
1 (True)	< +0.8 Volts (Low)

Tab. 3 Livelli di tensione definiti dallo standard IEEE 488.1

Come mostra la tabella, nello standard 488 si è adottata una logica negativa. Ciò significa che al livello di tensione alto (H), cioè superiore a 2.4 Volts, si attribuisce il valore logico 0 (False), mentre al livello di tensione inferiore a 0.8 Volts (L) si attribuisce il valore logico 1 (True). Una giustificazione di questa scelta si può ricercare nel fatto che ad alcuni trasmettitori è imposta la struttura Open Collector, per la quale è necessario che un segnale sia attivo quando si trova a livello logico basso.

Nel diagramma di Fig. 8 sono indicati i legami che devono essere soddisfatti tra le tensioni e le correnti sulle linee del bus 488.

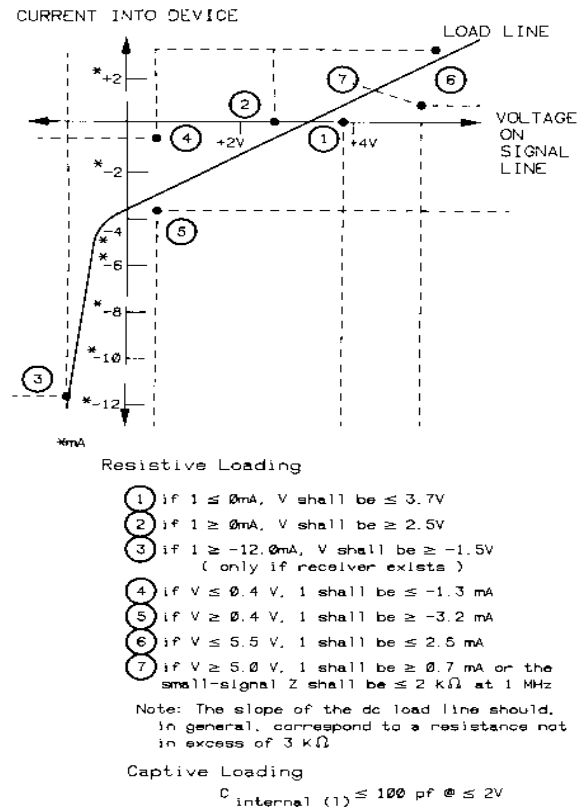


Fig. 8 Specifiche elettriche: caratteristica tensione-corrente

Nota: le tensioni sono relative al potenziale di riferimento presente nella linea collegata al pin 24 del connettore, mentre le correnti si riferiscono all'ingresso di un ricevitore. Si fa riferimento alla convenzione secondo cui la corrente è uscente dalla linea a potenziale positivo del trasmettitore, ed entrante nella corrispondente linea del ricevitore. Inoltre la corrente di un trasmettitore è pari alla somma delle correnti entranti in tutti i ricevitori collegati a quel trasmettitore.

Non viene specificata una ben precisa curva tensione-corrente, ma sono fissate alcune condizioni di disequaglianza tali da poter essere soddisfatte da parecchi tipi di componenti elettronici, utilizzati nella costruzione dei trasmettitori e dei ricevitori.

Trasmettitori e ricevitori

Per l'ingresso dei ricevitori e l'uscita dei trasmettitori vengono fornite indicazioni in termini di circuito equivalente, specificando, in particolare, impedenze di ingresso e di uscita (Fig. 9).

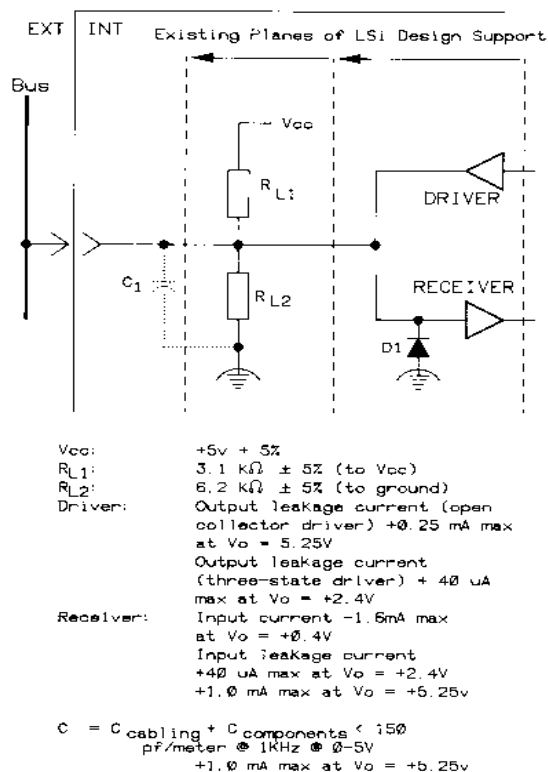


Fig. 9 Schema elettrico

Un trasmettitore deve presentare le caratteristiche circuitali indicate in tabella (Tab. 4):

Nome Segnale	Circuito in uscita
SRQ	Open Collector
NRFD	“ “
NDAC	“ “
ATN	Open Collector / Tristate
IFC	“ “
REN	“ “
EOI	“ “
DAV	“ “
DIO 1-8	“ “

Tab.4 Caratteristiche circuitali delle uscite di un trasmettitore

L'uscita di tipo Tristate è utilizzata per raggiungere velocità di trasferimento dati superiori a 250 Kbytes/s.

3.4 Lo standard IEEE 488.1

Indirizzamento

Consideriamo un sistema automatico di misura, costituito, per esempio, da un PC dotato di scheda di interfaccia 488 e da un certo numero di strumenti connessi al bus 488. Il flusso di informazioni tra i diversi componenti, in generale, avviene attraverso lo scambio di messaggi opportunamente codificati: se, da un lato, alcuni di questi messaggi sono destinati a tutti i dispositivi collegati (Universal Multiline Commands), altri sono rivolti ad un preciso dispositivo (Addressed Commands). L'operazione di indirizzamento consiste nel selezionare, per mezzo di un *indirizzo*, il destinatario dell'informazione che si intende inviare o ricevere lungo le linee del bus. Ad ogni dispositivo periferico (scheda di interfaccia compresa) connesso al bus 488 deve essere associato un indirizzo univoco, impostabile dal pannello frontale o posizionando opportunamente degli switch situati sul retro dello strumento.

Formato di un indirizzo 488

Un indirizzo 488, come tutte le informazioni inviate attraverso il data bus, viene espresso secondo il codice ASCII a 7 bit. I 5 bit meno significativi rappresentano l'indirizzo primario (**primary address**) del dispositivo; può assumere solo i valori compresi tra 0 e 30, mentre il valore 31, corrispondente alla configurazione "11111", è riservato. Il significato attribuito ai bit restanti è illustrato nella seguente tabella (Tab. 5):

Bit Position	7	6	5	4	3	2	1	0
Meaning	0	TA	LA	Primary Address (range 0-30)				

Tab. 5 Gli 8 bit di un indirizzo GP-IB

Il bit più significativo del comando 488 di indirizzamento è sempre posto a zero. Il controller setta il bit LA per indirizzare un dispositivo come listener, il bit TA per indirizzare un dispositivo come talker; è possibile, inoltre, assegnare ad un dispositivo contemporaneamente i ruoli di listener e talker, settando entrambi i bit TA e LA. Il controller utilizza normalmente l'indirizzo 0.

Con alcuni dispositivi è possibile utilizzare, in aggiunta all'indirizzo primario, un indirizzo secondario (**secondary address**) opzionale, rappresentato da un numero intero compreso tra 96 e 126 in codice ASCII. Quando il controller utilizza entrambi gli indirizzi per selezionare un dispositivo, invia sulle linee del bus prima il byte corrispondente all'indirizzo primario (nel formato talker o listener) e quindi il byte corrispondente all'indirizzo secondario. Tale possibilità risulta utile, per esempio, nella gestione di strumenti organizzati, al loro interno, in sottounità individualmente indirizzabili. In questo caso l'indirizzo primario indica lo strumento nel suo complesso, l'indirizzo secondario la singola sottounità.

Le funzioni di interfaccia

Come in tutte le strutture a bus, per un corretto funzionamento è necessario che in ogni intervallo di tempo sia attivo un solo trasmettitore, mentre si possono avere contemporaneamente più ricevitori destinatari dell'informazione inviata dal trasmettitore. Per garantire tale fatto è necessaria un'opportuna gestione del bus stesso in modo da poter stabilire, in ogni istante di tempo, chi sia il trasmettitore e quali siano i ricevitori. A tal fine, ogni dispositivo che rispetti lo standard 488 deve essere in grado di svolgere uno o più dei seguenti ruoli:

Listener

Dispositivo in grado di ricevere dati attraverso l'interfaccia, quando indirizzato. In un bus 488 possono essere presenti un massimo di 14 dispositivi attivi contemporaneamente come listener

Talker

Dispositivo in grado di trasmettere dati attraverso l'interfaccia, quando indirizzato. In un bus 488 un solo dispositivo può essere attivo come talker in un certo istante

Controller

Dispositivo in grado di stabilire quale chi è il talker (uno) e quali sono i listener (uno o più) in un trasferimento di dati. In un bus 488 possono essere presenti più dispositivi controller; in tal caso uno (e solo uno di essi) assume il ruolo di *system controller*.

Il controller si occupa di gestire il flusso delle informazioni spedendo comandi a tutti i dispositivi; decide, in ogni istante, quale dispositivo talker possa trasmettere dati e quali dispositivi listener possano riceverne, prima di iniziare il trasferimento dei dati stessi sulle linee del bus. Alcune configurazioni del bus 488 non richiedono la presenza di un controller perché di solito è presente un unico dispositivo che può trasmettere dati, mentre tutti gli altri possono solo riceverne. Normalmente le funzioni di controller sono assunte dal computer sul quale è montata la scheda di interfaccia 488 che pilota il bus. Il computer con l'interfaccia 488 può assumere tutti i ruoli di listener, talker e controller.

E' possibile che più dispositivi connessi al bus GP-IB siano in grado di svolgere le funzioni di controller. Chiaramente in ogni momento solo un dispositivo può essere attivo come controller il quale, in tal caso, viene chiamato **controller-in-charge** (CIC). Il controller-in-charge può decidere di passare il controllo del bus ad un altro controller ma allora non riuscirà più ad assumere il controllo del bus di propria iniziativa: sarà infatti necessario che il nuovo controller-in-charge gli restituisca il controllo.

Solo un dispositivo connesso al bus può divenire autonomamente controller-in-charge; tale dispositivo viene chiamato **system controller** ed è normalmente costituito dalla scheda di interfaccia 488 montata su computer.

Si noti che il progettista è libero di scegliere quali di queste funzioni un generico dispositivo o strumento sia in grado di svolgere, in base allo scopo per cui è realizzato. Lo standard raccomanda perciò di indicare le funzioni supportate dal dispositivo stesso per mezzo di appositi codici serigrafati in prossimità del connettore.

Handshaking

Il protocollo di handshake consente di coordinare il trasferimento dei dati lungo il data bus dalla sorgente (talker o controller) al destinatario (uno o più listener), assicurandone la correttezza.

Si osserva che, a differenza di quanto avviene nel caso di un bus interno ad un calcolatore, lo standard 488 non impone ai dispositivi stringenti specifiche temporali circa i segnali di controllo o i dati inviati sul bus. Ciò significa che possono essere contemporaneamente connessi al bus anche dispositivi che presentano velocità di trasferimento dati molto diverse. La particolare modalità di gestione delle linee di handshaking consente, tuttavia, che un trasmettitore, il quale voglia inviare dati contemporaneamente a più ricevitori, regoli in modo automatico la propria velocità di scrittura sul bus in modo che il dispositivo più lento sia in grado di ricevere correttamente i dati inviati sul bus stesso. Per questo motivo le linee di handshake previste dal protocollo 488 sono tre:

DAV – Data Valid

Utilizzata per segnalare la presenza di informazione sulle linee dati. Viene portata al valore logico 1 (tensione bassa) dal talker quando il dato è disponibile e la linea NRFD si trova ad un valore logico 0 (tensione alta).

NRFD – Not Ready For Data

Utilizzata per segnalare la capacità di un dispositivo ad acquisire dati. Un ricevitore imposta la linea NRFD ad un valore logico 1 per indicare che non è in grado di accettare i dati. In caso contrario la linea assume un valore logico 0. Dal momento che i circuiti di uscita di tutti i dispositivi connessi al bus sono del tipo “Open Collector”, essi realizzano un circuito combinatorio “Wired Or” in logica negata. Pertanto questa linea assume il valore logico 0 solo quando tutti i ricevitori coinvolti hanno accettato i dati.

NDAC – Not Data Accepted

Utilizzata per segnalare l'accettazione dei dati da parte dei dispositivi. Il ricevitore imposta il valore logico 1 per indicare che non ha acquisito i dati. La linea NDAC viene posta ad un valore logico 0 quando sono stati accettati i dati dalle linee corrispondenti (Digital I/O, DIO). In ogni caso, analogamente alla linea NRFD, il segnale NDAC assume il valore logico 0 solo dopo che anche l'ultimo dei ricevitori coinvolti ha accettato i dati.

La Fig. 10 illustra il diagramma temporale del protocollo di handshaking.

Si supponga che il controller abbia selezionato il talker che può inviare un byte sul data bus ed i ricevitori che devono acquisirlo. Una volta immesso tale byte nelle linee DIO, il talker deve segnalare ai destinatari che il dato è disponibile; allo scopo controlla innanzi tutto che

la linea NRFD del control bus, gestita dai ricevitori, sia al livello logico 1. Poiché lo standard impone che tutti i ricevitori siano collegati con tale linea con driver di tipo Open Collector, se anche uno solo dei listener non è pronto per la ricezione, la linea NRFD si trova al livello logico 0, bloccando ogni iniziativa del talker.

Verificato il livello logico 1 sulla linea NRFD, il talker attiva la segnalazione di presenza di un dato valido sul data bus, portando la linea DAV al livello logico 0. Tutti i listener, ricevono questa informazione e si predispongono ad acquisire il nuovo dato. L'acquisizione del nuovo dato rende i ricevitori non più disponibili a riceverne degli altri; questa temporanea condizione dei ricevitori viene segnalata portando la linea NRFD al livello logico 0. Si noti che la linea NRFD viene attivata dal ricevitore più veloce.

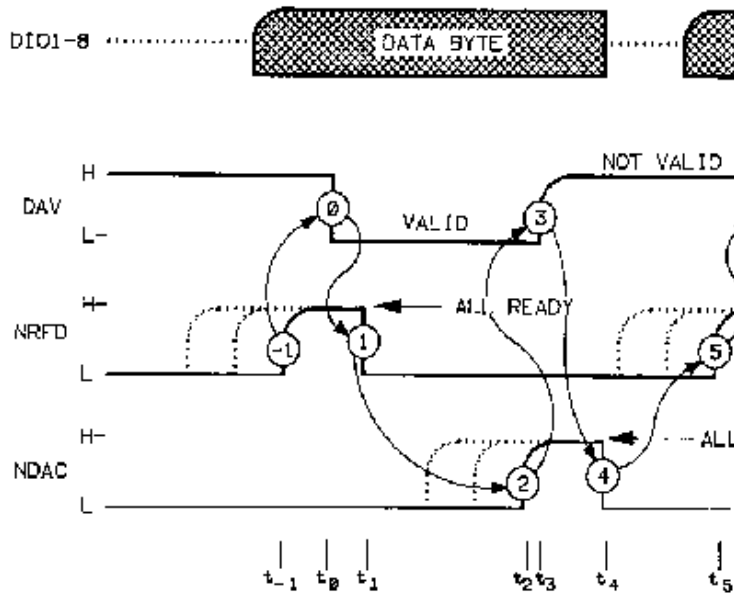


Fig. 10 Diagramma temporale del protocollo di handshaking

I listener acquisiscono quanto presente sul data bus, ognuno con la sua tempistica; quando un ricevitore non necessita ulteriormente della presenza del dato sulle linee del bus, perché lo ha già utilizzato, o perché lo ha memorizzato in un buffer interno, diventa disponibile ad accettare un eventuale nuova informazione: questa situazione viene segnalata portando al livello logico 1 la linea NDAC. La disattivazione di questa linea è condizionata da tutti i driver ad essa collegati. Solo quando il più lento dei listener ha completato l'attività di acquisizione del nuovo dato, la linea NDAC può assumere il livello logico 1, segnalando al talker che si può dare inizio ad una nuova attività. Il trasmettitore porta al livello logico 0 la linea DAV, ad indicare che i valori presenti sul data bus non rappresentano informazioni utili, e i ricevitori si predispongono ad una eventuale nuova acquisizione portando al livello logico 0 la linea NDAC. La procedura prosegue ripetendo le operazioni appena descritte.

Linee di controllo del bus

Cinque linee del bus 488 sono utilizzate per gestire un generico flusso d'informazioni attraverso un'interfaccia:

ATN (ATteNtion)

Linea posta al livello logico 1 dal controller corrente quando il controller stesso desidera utilizzare le linee del bus per inviare ai dispositivi dei messaggi (sia di tipo *device dependent*, sia di tipo *interface managing*). Un generico dispositivo in stato talker, prima di inviare dati, deve verificare che la linea ATN sia a livello logico 0.

IFC (InterFace Clear)

Linea utilizzata dal system-controller per inizializzare tutte le linee del bus e diventare controller-in-charge.

REN (Remote ENable)

Linea pilotata dal system-controller per portare i dispositivi connessi al bus in uno stato d'impostazione remota (cioè via 488) o locale (cioè tramite il pannello frontale dello strumento).

SRQ (Service ReQuest)

Linea utilizzata dai dispositivi per comunicare al controller-in-charge una richiesta di servizio. Ad esempio quando uno di essi desidera trasmettere dati sul bus il controller dovrebbe gestire il trasferimento impostando i dispositivi come Talker o Listener nel modo appropriato.

EOI (End Or Identify)

Ha due funzioni: 1) i dispositivi Talker utilizzano tale linea per indicare ai dispositivi Listener la fine della stringa trasmessa sul bus; 2) il controller utilizza questa linea per invitare tutti i dispositivi a rispondere ad una richiesta del tipo parallel poll (descritta in seguito).

I comandi del bus

Questi comandi controllano le operazioni del bus e non quelle dei dispositivi ad esso connessi e possono essere raggruppati in 4 classi: *Universal*, *Uniline*, *Addressed* e *Secondary*.

Universal Multiline Commands

Ci sono cinque comandi del tipo Universal Multiline e sono elencati nella seguente tabella (Tab. 6):

Multiline Command	Mnemonic	Decimal Code	Hexadecimal Code	Octal Code	ASCII/ISO Character
Device Clear	DCL	20	14	24	DC4
Local Lockout	LLO	17	11	21	DC1
Serial Poll Enable	SPE	24	18	30	CAN
Serial Poll Disable	SPD	25	19	31	EM
Parallel Poll Unconfigure	PPU	21	15	25	NAK

Tab. 6 I comandi Universal Multiline

Segue una breve descrizione delle funzionalità associata ai singoli comandi:

Device Clear Command (DCL)

Riporta tutti gli strumenti, che riconoscono il comando stesso, ad uno stato predefinito dal costruttore, dipendente dallo strumento e specificato nel manuale; il dispositivo deve essere in modalità remota o preventivamente indirizzato.

Local Lockout Command (LLO)

Disabilita il controllo “return-to-local” del dispositivo (in pratica disabilita il pannello frontale dello strumento). Il dispositivo deve essere indirizzato, in modalità remota o locale. Per abilitare nuovamente il controllo “return-to-local” la linea fisica REN deve essere portata ad un valore logico 0: in questo modo il dispositivo viene impostato per il controllo locale (attivazione del pannello frontale dello strumento).

Serial Poll Enable Command (SPE)

Effettua un Serial Polling di tutti i dispositivi indirizzabili come Talker; ciascuno di essi ritorna uno status-byte, il quale rappresenta in forma opportunamente codificata lo stato del dispositivo (richiesta di servizio, dati disponibili, situazioni di errore,...), come descritto in un paragrafo successivo.

Serial Poll Disable Command (SPD)

Conclude la modalità Serial Poll e riporta i dispositivi allo stato di Talker, secondo il quale gli strumenti forniscono dati di tipo device-dependent piuttosto che informazioni di stato.

Parallel Poll Unconfigure Command (PPU)

Porta tutti i dispositivi ad uno stato di “non risposta” ad un parallel poll.

A questi si aggiungono i comandi di Untalk e Unlisten che vengono classificati come “indirizzi” (Tab. 7):

<i>Command</i>	<i>Mnemonic</i>
Untalk	UNT
Unlisten	UNL

Tab. 7 I comandi Untalk e Unlisten

Untalk Command (UNT)

Disabilita il dispositivo Talker corrente e lo porta in uno stato disabilitato. Un risultato identico si può ottenere selezionando come Talker un altro dispositivo, oppure un indirizzo non utilizzato.

Unlisten Command (UNL)

Disabilita tutti i dispositivi Listener correnti. Si noti che non è possibile disattivare un singolo Listener; questo comando può essere convenientemente impiegato per disabilitare tutti i dispositivi listener, prima di abilitare i soli listener che si desidera siano impostati come tali.

Uni-Line Commands

I quattro comandi di tipo Uniline sono indicati in tabella (Tab. 8):

<i>Uniline command</i>	<i>Interface Management Line</i>
Interface Clear	IFC
Remote Enable	REN
Attention	ATN
Identify (IDY)	EOI \wedge ATN

Tab. 8 I comandi Uni-Line

Alcuni di questi comandi sono già stati descritti (IFC, REN, ATN), gli altri (IDY) saranno discussi congiuntamente all’operazione Parallel Poll.

Addressed Commands

La tabella Tab. 9 elenca i comandi del tipo Addressed:

Addressed Command	Mnemonic	Decimal Code	Hex Code	Octal Code	ASCII
Group Execute Trigger	GET	08	08	10	BS
Selected Device Clear	SDC	04	04	04	EOT
Go To Local	GTL	01	01	01	SOH
Parallel Poll Configure	PPC	05	05	05	ENQ
Take Control	TCT	09	09	11	HT

Tab. 9 I comandi Addressed

Segue una breve descrizione delle funzionalità dei singoli comandi:

Group Execute Trigger Command (GET)

Segnala a tutti i dispositivi attivi come Talker, in grado di gestire il comando, di dare inizio ad un'azione in precedenza definita per ogni dispositivo. Il comando GET fornisce un mezzo per inviare un segnale di sincronizzazione simultaneamente a di più dispositivi.

Selected Device Clear (SDC)

Riporta tutti i dispositivi indirizzati come Listener ad uno stato device-dependent predefinito. Equivale al comando DCL (vedi Universal Multiline Commands).

Go To Local (GTL)

Questo comando riporta tutti i dispositivi indirizzati come Listener alla modalità di controllo locale (uscita dallo stato remoto). Il dispositivo tornerà alla modalità di funzionamento remota se attivato come Listener e la linea REN sarà al valore logico 1.

Parallel Poll Configure (PPC)

Tutti i dispositivi indirizzati come Listener vengono configurati in accordo al comando che segue immediatamente PPC.

Take Control Talker (TCT)

Abilita il dispositivo correntemente attivo come Talker ad assumere il ruolo di Bus controller.

Secondary Commands

I comandi di tipo Secondary sono utilizzati come indirizzi secondari estesi (di tipo talk e listen) e come comandi di parallel poll secondario (Tab. 10).

Secondary Command	Mnemonic	Hex Code	Octal Code	Decimal Code	ASCII/ISO Character
Parallel Poll Enable	PPE	60-6F	140-157	96-111	' thru o
Parallel Poll Disable	PPD	70	160	112	p

Tab. 10 I Secondary Commands

Segue una breve descrizione delle funzionalità dei singoli comandi:

Parallel Poll Enable (PPE)

Configura i dispositivi che hanno ricevuto il comando PPC in modo tale da rispondere al Parallel Poll su una particolare linea di dati (DIO), con un certo valore numerico predefinito.

Parallel Poll Disable (PPD)

Disabilita i dispositivi che hanno ricevuto il comando PPC a rispondere al Parallel Poll.

Polling

Ci sono due possibili procedure di polling sul bus, il *Serial Poll* e il *Parallel Poll*.

Serial Poll

Il *Serial Poll* è una procedura che consente al controller di ricevere informazioni sullo stato di un dispositivo: in particolare il controller può stabilire se uno o più dispositivi hanno

effettuato una richiesta di servizio. I dispositivi in grado di rispondere ad un Serial Poll ritornano al controller uno *Status Byte*, il quale descrive il proprio stato.

La procedura di interrogazione mediante Serial Poll prevede l'invio del comando SPE seguito dall'abilitazione sequenziale di ciascun dispositivo collegato al bus a rispondere al comando Serial Poll inviando il byte di stato (*Status Byte*). Il controller, una volta conclusa la scansione dei dispositivi per la lettura dello status byte, deve inviare i comandi SPD e UNT (la maggior parte dei controller permettono l'esecuzione di tutti i precedenti passi automaticamente, attraverso chiamate a funzioni di libreria di alto livello).

Parallel Poll

Lo scopo di tale procedura è di ottenere informazioni sullo stato dei dispositivi connessi al bus. I dispositivi, individualmente o collettivamente, ritornano uno *Status Bit* su una delle linee DIO. L'assegnazione di una linea DIO ad un dispositivo viene effettuata attraverso interruttori, jumpers o dal controller per mezzo del comando PPC. Quando più dispositivi rispondono collettivamente, la lettura delle linee DIO fornisce al controller l'AND logico (se il livello logico 1 corrisponde a tensione alta) o l'OR logico (se il livello logico 1 corrisponde a tensione bassa) dei bit di stato.

Lo standard 488 specifica che i dispositivi coinvolti debbano rispondere al Parallel Poll entro 200 ns; il controller deve leggere la risposta 2 μ s dopo lo scadere dei 200 ns di attesa seguente al Parallel Poll.

3.5 Lo standard IEEE 488.2

Lo standard IEEE 488.2 costituisce un'estensione dello standard 488 appena illustrato, rinominato 488.1, per distinguerlo, appunto dalla nuova versione. In accordo a tale estensione dello standard, un'interfaccia per strumenti di misura può essere descritta come costituita da un insieme di livelli funzionali, come illustrato in Fig. 11.

Il livello più basso (**Remote Interface Messages**) rappresenta l'interfaccia fisica, ovvero realizzata seguendo le specifiche del bus IEEE 488.1 (connettore, cablaggi, segnali elettrici, protocollo di handshaking, etc.). Lo standard IEEE 488.2 definisce i due strati intermedi:

Il livello delle **Syntax and Data Structures**, che definisce le modalità di intercomunicazione tra gli strumenti nonché il formato dei dati scambiati.

Il livello dei **Common Commands and Queries**.

Il livello **Device Dependent Messages** rappresenta l'insieme dei comandi che possono essere inviati ad uno strumento affinché svolga operazioni utili, ed è definito dal costruttore del dispositivo.

Si noti che il protocollo 488.1 non definisce alcuni elementi associati a un'interfaccia, con conseguenti possibili difficoltà nello scambio di informazioni. Tra questi, 488.1 non definisce:

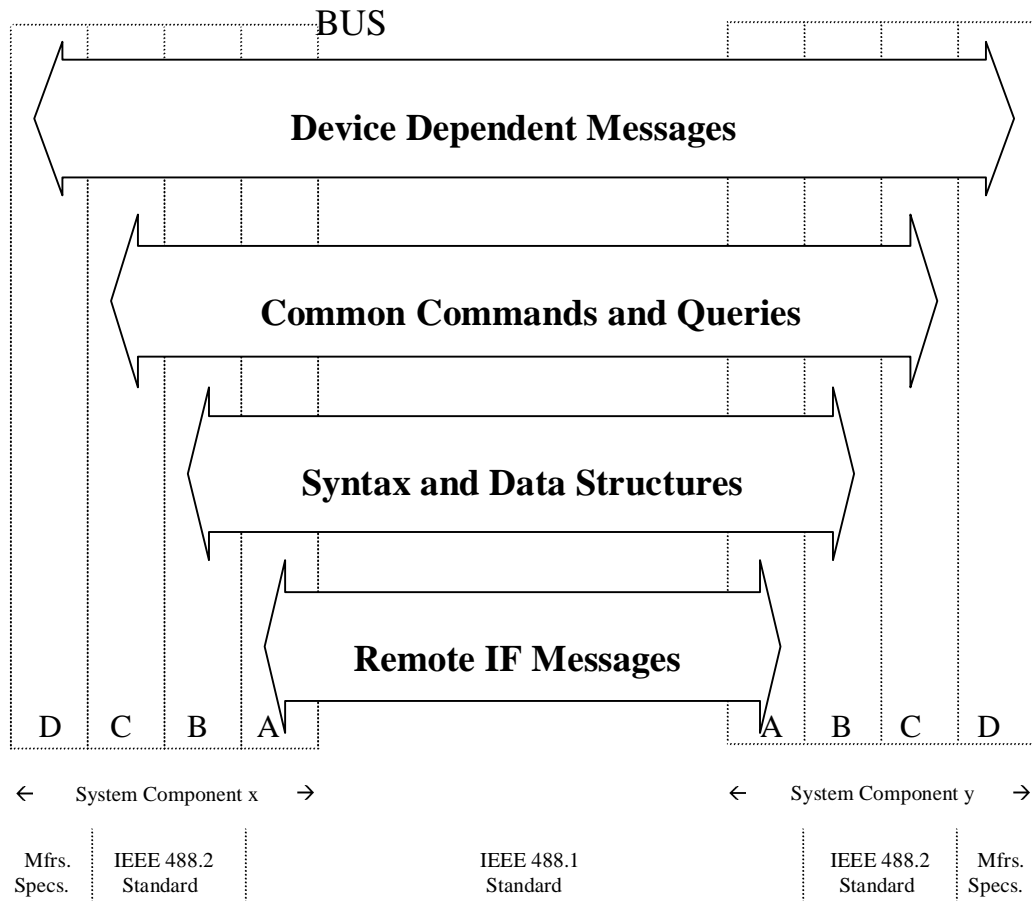


Fig. 11 Suddivisione in livelli funzionali dello scambio di informazioni

Funzioni di interfaccia di uno strumento

Formato comune di dati

Standardizzazione dei messaggi

Insieme di comandi comune a tutti gli strumenti

Significato e gestione dello Status Byte di un dispositivo

Lo standard IEEE 488.2 risolve questi problemi definendo:

Insiemi di funzioni di interfaccia che devono essere rese disponibili da parte di uno strumento

Formato e sintassi per i dati scambiati

Protocollo relativo ai device-message

Modello per il formato dello Status Byte fornito da dispositivo in seguito ad una interrogazione

Funzioni di interfaccia obbligatorie nel protocollo 488.2

La seguente tabella (Tab. 11) elenca l'insieme di funzioni di interfaccia che uno strumento può supportare:

Funzionalità	Codice	Grado di funzionalità
Source Handshake	SH1	Piena
Acceptor Handshake	AH1	Piena
Talker	T(TE)5, o T(TE)6	Basic Talker, Serial Poll, untalk sul segnale MLA
Listener	L(LE)3, o L(LE)4	Basic Listener, unlisten sul segnale MTA
Service Request	SR1	Piena
Device Clear	DC1	Piena
Remote Local	RL0 o RL1	Nessuna o piena
Parallel Poll	PP0 o PP1	Nessuna o piena
Device Trigger	DT0 o DT1	Nessuna o piena
Controller	C0, o C4, with C5, C7, C8 o C11	Nessuna o Risposta al segnale SRQ, Invio di messaggi di interfaccia, cessione, acquisizione del controllo
Electrical Interface	E1 o E2	Open Collector o Tristate

Tab. 11 Funzioni di interfaccia supportabili da uno strumento compatibile 488.2

In sostanza, tutti i dispositivi sono in grado di inviare e ricevere dati, richieste di servizio e di rispondere al comando Device Clear. La tabella specifica anche le funzioni minime che lo strumento deve poter gestire per svolgere il ruolo di controller, per rispondere ad un Parallel Poll e per operare nelle modalità remota e locale.

Formato dei dati

Lo standard stabilisce una serie di regole che devono essere rispettate nel formare un qualsiasi messaggio scambiato tra due dispositivi. Innanzi tutto va precisato che le informazioni trasmesse lungo il bus vengono rappresentate utilizzando un codice ASCII a 7 bit e solo in determinate e ben documentate situazioni possono essere seguite codifiche diverse.

Il principio generale adottato nello standard è di fissare norme molto più rigide per i Talker rispetto ai Listener (“*Forgiving listening. Precise talking.*”) in modo tale da consentire la compatibilità Talker-Listener tra IEEE 488.1 e IEEE 488.2. In altre parole i dati devono

rispettare un formato molto più rigido, se trasmessi, più flessibile, se ricevuti. Si consideri come esempio il formato associato ad un dato numerico che un Listener deve essere in grado di accettare.

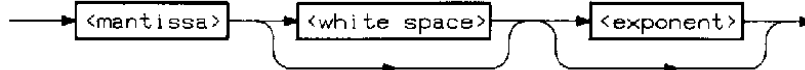


Fig. 12 Formato di un numero decimale accettato da un listener secondo IEEE 488.2

Con riferimento alla Fig. 12, si vede che un numero decimale è rappresentato utilizzando per ogni cifra la codifica stabilita dalle tabelle dei codici ASCII. E' inoltre costituito da un campo "mantissa" seguito da un campo "esponente" (opzionale), eventualmente separati da uno spazio. Lo standard indica le possibili forme alternative per specificare questi campi. Come ulteriore esempio è rappresentata in Figg. 13-14 la descrizione del formato di "mantissa" ed "esponente":

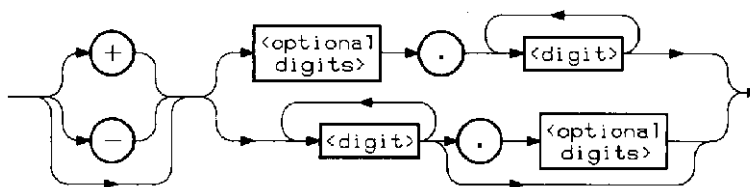


Fig. 13 Formato del campo mantissa del diagramma di Fig. 12

Nella mantissa può essere implicitamente assunto il segno positivo oppure si può premettere l'indicazione esplicita del segno; il valore numerico espresso mediante le singole cifre (codificate in ASCII) può essere composto dalla sola parte intera, dalla sola parte decimale o da entrambe.

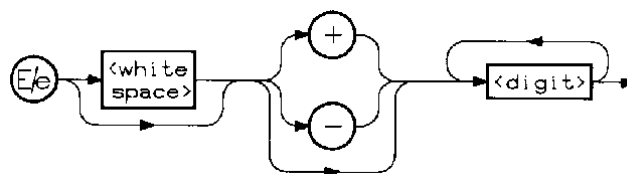


Fig. 14 Formato del campo esponente del diagramma di Fig. 12

La presentazione dell'esponente deve iniziare con la lettera "E" oppure "e" (codificate in ASCII). Sono ovviamente imposti dei limiti ai valori massimi rappresentabili.

Con la notazione introdotta, si vede come i valori numerici possono essere espressi con formati diversi, come ad esempio:

.123
 1.23
 +12.3
 .123 E -45
 +12.3e+45
 .123 E +1
 +12.3e-1
 -0.123

Per evidenziare il minor grado di flessibilità lasciato al trasmettitore dallo standard IEEE 488.2, in Fig. 15 è riportato il formato relativo all'invio di un numero floating-point. Si noti che il Talker deve obbligatoriamente usare, per la rappresentazione della mantissa, il punto decimale; non ci può essere, inoltre, uno spazio tra mantissa ed esponente. Questo può iniziare solo con la lettera "E" e deve esserne specificato il segno.

Ad esempio un talker, a differenza di un listener, può solo inviare dati del tipo seguente:

1.23E+5
 123.4E-56
 -12345.678E+90

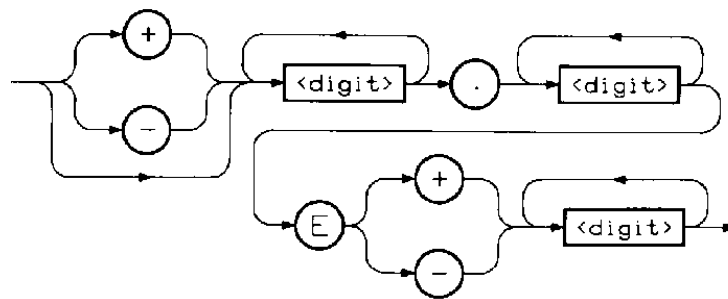


Fig. 15 Formato di un valore numerico Floating Point in trasmissione, secondo IEEE 488.2

Lo standard fissa inoltre le norme per specificare eventuali unità di misura, per rappresentare numeri con base diversa da quella decimale, per inviare sul bus una stringa alfanumerica, etc.. Tali norme sono espresse in termini di diagrammi simili alle precedenti Figg. 12-15.

Sintassi

Lo scambio di informazioni tra talker e listener, per esempio l'invio di un comando da parte del controller o di una risposta da parte di uno strumento, deve rispettare regole precise. Anche in questo caso lo standard segue uno stile del tipo "Forgiving listening.

Precise talking.”, specificato in due sotto-sezioni denominate sintassi di ricezione (**Listening Syntax**) e sintassi di trasmissione (**Talking Syntax**).

Listening Syntax

La sintassi di un comando accettato in ricezione prevede l'uso di:

Separatori: ne sono contemplati vari tipi

Program Message Separator: consente di separare comandi distinti all'interno di uno stesso messaggio; è definito dal diagramma di Fig. 16



Fig. 16 Program Message Separator

Program Header Separator: consente di separare l'intestazione del comando dai parametri ad esso associati; è definito come in Fig. 17

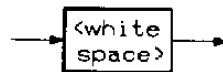


Fig. 17 Program Header Separator

Program Data Separator: consente di distinguere dati diversi all'interno di una lista; è definito come in Fig. 18

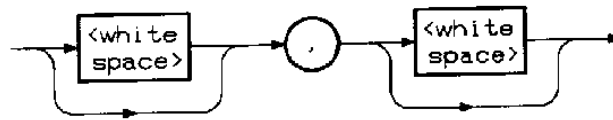


Fig. 18 Program Data Separator

Commands: la sintassi secondo la quale deve essere rappresentato un comando è definita dal command program header descritto graficamente in Fig. 19

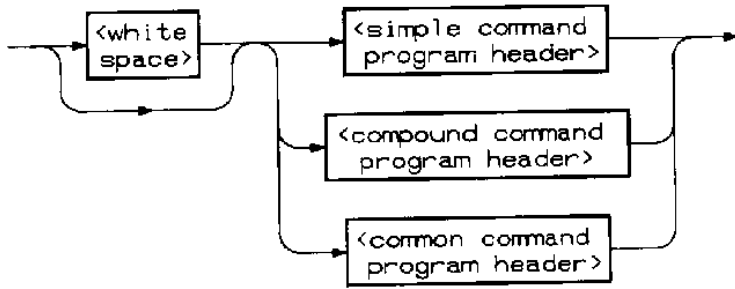


Fig. 19 Command Program Header

I simboli non terminali “simple command program header”, “compound command program header” e “common command program header” di Fig. 20 sono definiti nei tre diagrammi seguenti. In particolare “simple command program header” coincide con “program mnemonic”, descritto in Fig. 20

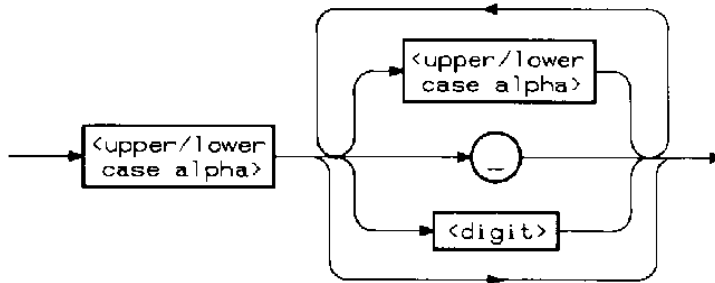


Fig. 20 Program Mnemonic, coincidente con Simple Command Program Header

“compound program header” è definito come in Fig. 21

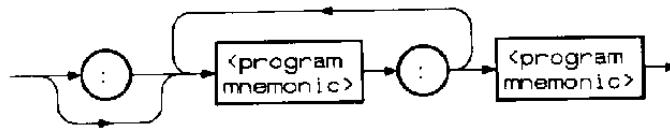


Fig. 21 Compound Program Header

“common command program header” è definito come in Fig. 22

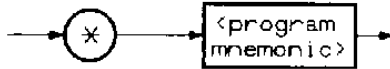


Fig. 22 Common Command Program Header

Query: rappresenta una richiesta di informazioni ad un dispositivo; la sintassi è del tutto analoga a quella di un generico comando, ma è terminata obbligatoriamente dal carattere “?”, in accordo alla seguente definizione (“query program header”) di Fig. 23

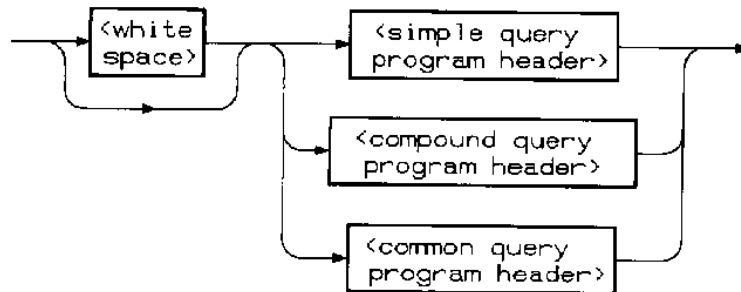


Fig. 23 Query Program Header

Con riferimento alla Fig. 23 i campi “simple query program header”, “compound query program header” e “common query program header” sono definiti nei seguenti diagrammi. In particolare, “simple query program header” è definito come in Fig. 24

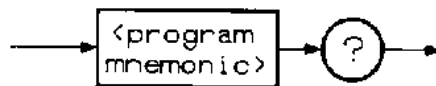


Fig. 24 Simple Query Program Header

“compound query program header” è definito come in Fig. 25

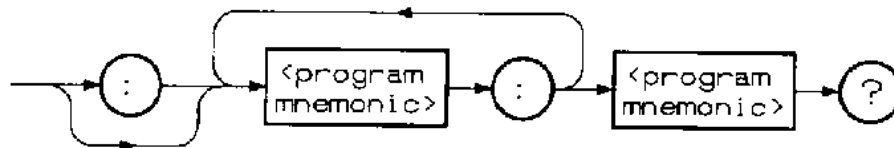


Fig. 25 Compound Query Program header

Infine, “common query program header” è definito come in Fig. 26

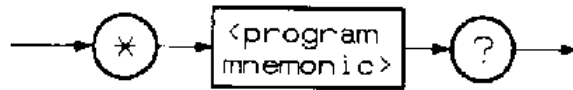


Fig. 26 Common Query Program Header

Terminatori: segnalano al dispositivo listener la fine del messaggio ricevuto. La sintassi corrispondente è definita nel diagramma “program message terminator” di Fig. 27

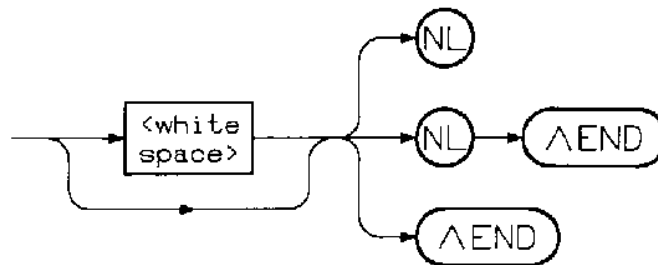


Fig. 27 Program Message Terminator

Il simbolo “^ END” segnala che, dopo l’invio dell’ultimo byte, è stata attivata la linea EOI (End Or Identify)

Talking Syntax

La sintassi per le informazioni fornite da un generico trasmettitore risulta più restrittiva rispetto a quella per le informazioni che devono essere accettate da un ricevitore. Una generica informazione fornita da un talker può contenere uno o più messaggi suddivisi nei campi seguenti:

Intestazione
Dati

La sintassi prevede l’uso di:

Separatori: possono essere del tipo seguente

Message Separator: consente di separare messaggi di risposta a query diverse, inviati in un unico messaggio. E’ definito come il carattere ‘ ; ‘

Data Separator: consente di separare i singoli dati all’interno di una lista. E’ definito come il carattere ‘ , ‘

Header Separator: consente di distinguere l'intestazione della risposta dai dati. E' definito come il carattere ' ' (spazio)

Terminatori: il generico messaggio di risposta viene inviato dallo strumento in seguito ad una query. Il terminatore di risposta, definito "response message terminator", è descritto in Fig. 28



Fig. 28 Response Message Separator

Si noti che la linea fisica EOI del bus 488 viene attivata quando sul bus è presente il carattere New Line (NL) o Line Feed (LF), corrispondenti entrambi al codice decimale 10.

Common Commands

IEEE 488.2 fornisce un insieme di comandi, noti come **Common Commands**, i quali devono poter essere gestiti da una grande varietà di strumenti, e pertanto risultano standardizzati. Alcuni (indicati con l'aggettivo inglese "Required" in Tab. 12) devono obbligatoriamente essere gestiti da tutti gli strumenti compatibili con lo standard; altri, viceversa, sono solo definiti dallo standard e possono non essere riconosciuti da tutti gli strumenti (indicati con l'aggettivo inglese "Optional" in Tab. 12).

Da un punto di vista sintattico questi comandi devono obbligatoriamente iniziare con il carattere ASCII ' * '; sono formati da tre caratteri maiuscoli, e sono seguiti dal carattere ' ? ', se il comando costituisce una richiesta di informazioni al dispositivo.

Nella seguente tabella Tab. 12 è riportato un elenco dei Common Commands, raggruppati in base alle funzioni svolte:

Gruppo	Nome Mnemonico	Descrizione	Commento
Auto Configure Commands			
	*AAD	Assign Address	Optional
	*DLF	Disable Listener Function	Optional
System Data Commands			
	*IDN?	Identification Query	Required
	*OPT?	Option Identification Query	Optional
	*PUD	Protected User Data	Optional
	*PUD?	Protected User Data Query	Optional

	*RDT	Resource Description Transfer	Optional
	*RDT?	Resource Descr. Transfer Query	Optional
Internal Operation Commands			
	*CAL	Calibration Query	Optional
	*LRN	Learn Device Setup Query	Optional
	*RST	Reset	Required
	*TST?	Self-Test Query	Required
Synchronization Commands			
	*OPC	Operation Complete	Required
	*OPC?	Operation Complete Query	Required
	*WAI	Wait to Complete	Required
Macro Commands			
	*DMC	Define Macro	Optional
	*EMC	Enable Macro	Optional
	*EMC?	Enable Macro Query	Optional
	*GMC?	Get Macro Contents Query	Optional
	*LMC?	Learn Macro Query	Optional
	*PMC	Purge Macro	Optional
Parallel Poll Commands			
	*IST?	Individual Status Query	Req. if PP1
	*PRE	Par. Poll En. Register En.	Req. if PP1
	*PRE?	Par. Poll En. Register En. Query	Req. if PP1
Status & Event Commands			
	*CLS	Clear Status	Required
	*ESE	Event Status Enable	Required
	*ESE?	Event Status Enable Query	Required
	*ESR?	Event Status Register Query	Required
	*PSC	Power on Status Clear	Optional
	*PSC?	Power on Status Clear Query	Optional
	*SRE	Service Request Enable	Required
	*SRE?	Service Request Enable Query	Required
	*STB	Read Status Byte Query	Required
Device Trigger Commands			
	*DDT	Define Device Trigger	Opt. if DT1
	*DDT?	Define Device Trigger Query	Opt. if DT1
	*TRG	Trigger	Req. If DT1

Controller Commands			
	*PCB	Pass Control Back	Req. if CIC
Stored Settings Commands			
	*RCL	Recall Instrument State	Optional
	*SAV	Save Instrument State	Optional

Tab. 12 Common Command Groups

Si noti, infine, che questi comandi sono inviati quando la linea ATN è al valore logico 0 (Data Mode): non si tratta, quindi, di comandi per il controllo dell'interfaccia dei dispositivi, ma di comandi che i dispositivi devono interpretare ed eseguire.

I registri di stato

Lo standard IEEE 488.1 prevede alcune procedure per l'acquisizione di un registro di stato (Status Byte) di un dispositivo collegato al bus 488. Nel caso più semplice del comando Serial Poll, per esempio, lo strumento segnala al Controller di dover inviare un dato settando opportunamente la linea SRQ; per riconoscere il richiedente, il Controller acquisisce lo Status Byte di ciascuno dei dispositivi collegati al bus gestendo eventuali priorità, in caso di più richieste. Sempre nello standard 488.1, il dispositivo richiedente *deve solo* impostare il bit DIO n° 7 al livello logico 1, mentre è lasciata completa libertà di utilizzazione dei restanti 7 bit: come conseguenza, ogni strumento può presentare, a parità di stato, uno Status Byte diverso in base alla codifica scelta dal costruttore per i 7 bit "arbitrari".

Lo standard IEEE 488.2, pur conservando la compatibilità con l' IEEE 488.1, richiede che ogni dispositivo collegato al bus invii le informazioni relative al proprio stato in modo uniforme, fissando una procedura di riferimento ed assegnando significati ben precisi ai vari bit. La modalità seguita in quest'operazione prevede l'uso di registri, nei quali viene indicato se, rispetto ad una situazione di riferimento, si è verificata una variazione; in questo modo, tuttavia, non si dispone dell'evoluzione dello stato dello strumento. Utilizzando una struttura di accodamento (output queue) è possibile memorizzare anche la successione degli stati (Fig. 29).

Da un punto di vista logico, all' interno del dispositivo possiamo individuare:

Condition Register: i bit di questo registro rappresentano lo stato attuale dello strumento

Event Register: in questo registro viene memorizzato se si è verificata una certa condizione nel Condition Register; essendo di tipo latch, l' EventRegister non rappresenta lo stato attuale del dispositivo

Event Enable Register: questo registro consente di stabilire quali tra i bit dell' EventRegister sono abilitati a segnalare all' esterno una propria variazione (AND logico tra bit corrispondenti dei due registri)

Summary Message: questo bit è il risultato dell' OR logico delle operazioni logiche descritte al punto precedente; viene settato se si è verificata almeno una delle transizioni previste (se abilitata ad essere segnalata)

Ad ogni strumento è imposta una *struttura minima* per indicare lo stato, che prende il nome di **struttura standard**, in cui sono fissati i significati da attribuire ai vari bit: vengono segnalate situazioni generali (per esempio il verificarsi di vari tipi di errori), ed in nessun caso si fa riferimento alle funzionalità del dispositivo. Lo standard IEEE 488.2 stabilisce che il Summary Message bit vada ad influire sul bit 6 del registro letto dal controller quando chiede allo strumento di fornire il proprio stato (Status Byte Register). Degli altri sette bit, uno è utilizzato per garantire la compatibilità con la prima versione dello standard; il bit 5, denominato Standard Event Bit (ESB), viene settato in corrispondenza all'attivazione di una delle condizioni previste nell'Event Register; il bit 4, denominato Message Available (MAV) indica invece che è presente qualche elemento da leggere nella output queue; i bit rimanenti non sono definiti neppure dalla seconda versione dello standard e possono essere gestiti dal progettista.

Nota: E' possibile effettuare una lettura dello Status Register su iniziativa del controller, anche senza che il dispositivo ne faccia richiesta tramite la procedura di Serial Poll. In tal caso in controller invia il Common Command "*STB?" a cui lo strumento risponde inviando il byte del registro di stato nel quale il bit 6 assume il significato di Master Summary Status (MSS).

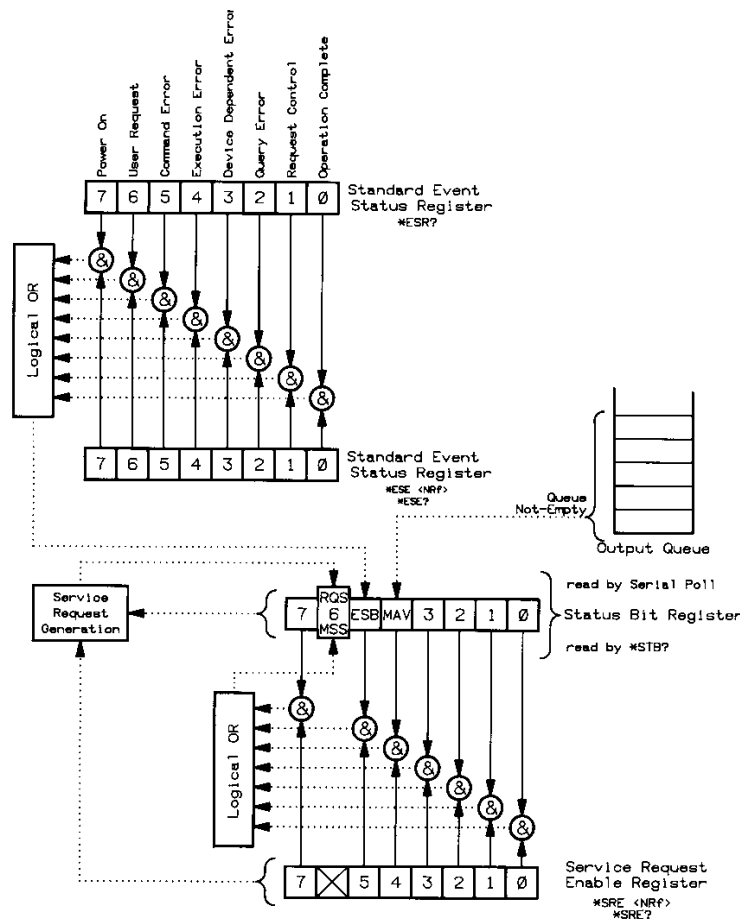


Fig. 29 Schema logico per la costruzione dello Status Register di un dispositivo

Comunicazione Talker-Listener

I dati inviati sul data bus vengono rappresentati in codice ASCII a 7 bit: i singoli caratteri, tuttavia, assumono significati diversi in relazione allo stato della linea di controllo ATN, gestita esclusivamente dal controller. Solo se questa linea è a livello logico 1 l'informazione presente sul data bus assume il significato di indirizzo, secondo quanto specificato nella seguente tabella (Tab. 13):

BITS		CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
B7	B6	B5	B4	B3	B2	B1	B0	B7	B6
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0
0	0	0	1	0	1	1	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	1	0	0	0
0	0	1	0	1	0	0	0	0	0
0	0	1	0	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0	0
0	0	1	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	1	0	0	0
0	0	1	1	0	1	0	0	0	0
0	0	1	1	0	1	1	0	0	0
0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	1	0	0	0
0	1	0	0	1	1	0	0	0	0
0	1	0	0	1	1	1	0	0	0
0	1	0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1	0	0
0	1	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	0	0
0	1	0	1	0	1	1	0	0	0
0	1	0	1	1	0	0	0	0	0
0	1	0	1	1	0	0	1	0	0
0	1	0	1	1	0	1	0	0	0
0	1	0	1	1	1	0	0	0	0
0	1	0	1	1	1	1	0	0	0
0	1	1	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0
0	1	1	0	0	0	1	0	0	0
0	1	1	0	0	1	0	0	0	0
0	1	1	0	0	1	1	0	0	0
0	1	1	0	1	0	0	0	0	0
0	1	1	0	1	0	0	1	0	0
0	1	1	0	1	0	1	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	0	1	1	1	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0	0	0
0	1	1	1	0	1	0	0	0	0
0	1	1	1	0	1	1	0	0	0
0	1	1	1	1	0	0	0	0	0
0	1	1	1	1	0	0	1	0	0
0	1	1	1	1	0	1	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	0
1	0	0	0	0	1	1	0	0	0
1	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	1	0	0
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	0
1	0	0	0	1	1	1	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0
1	0	0	1	0	1	0	0	0	0
1	0	0	1	0	1	1	0	0	0
1	0	0	1	1	0	0	0	0	0
1	0	0	1	1	0	0	1	0	0
1	0	0	1	1	0	1	0	0	0
1	0	0	1	1	1	0	0	0	0
1	0	0	1	1	1	1	0	0	0
1	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0
1	0	1	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0
1	0	1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0	0	0
1	0	1	0	1	0	0	1	0	0
1	0	1	0	1	0	1	0	0	0
1	0	1	0	1	1	0	0	0	0
1	0	1	0	1	1	1	0	0	0
1	0	1	1	0	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0
1	0	1	1	0	0	1	0	0	0
1	0	1	1	0	1	0	0	0	0
1	0	1	1	0	1	1	0	0	0
1	0	1	1	1	0	0	0	0	0
1	0	1	1	1	0	0	1	0	0
1	0	1	1	1	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0
1	0	1	1	1	1	1	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	1	1	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	1	0	0
1	1	0	0	1	0	1	0	0	0
1	1	0	0	1	1	0	0	0	0
1	1	0	0	1	1	1	0	0	0
1	1	0	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1	0	0
1	1	0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0	0	0
1	1	0	1	0	1	1	0	0	0
1	1	0	1	1	0	0	0	0	0
1	1	0	1	1	0	0	1	0	0
1	1	0	1	1	0	1	0	0	0
1	1	0	1	1	1	0	0	0	0
1	1	0	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	1	0	0	0
1	1	1	0	0	1	0	0	0	0
1	1	1	0	0	1	1	0	0	0
1	1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	1	0	0
1	1	1	0	1	0	1	0	0	0
1	1	1	0	1	1	0	0	0	0
1	1	1	0	1	1	1	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	1	0	0
1	1	1	1	0	0	1	0	0	0
1	1	1	1	0	1	0	0	0	0
1	1	1	1	0	1	1	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	1	0	0
1	1	1	1	1	0	1	0	0	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	0	0	0

KEY: octal 25 PPU Message Mnemonic
 hex 15 NAK ASCII/ISO character
 decimal 21

Tab. 13 ASCII-7 bit Code Chart

Quando, invece, la linea ATN è al livello logico 0, quanto presente sul data bus assume il significato di un dato, la cui codifica è fornita dalla tabella stessa.

Esempio: la linea ATN è attiva e viene inviato il seguente carattere ASCII:

“D”: viene attivato come talker il dispositivo con indirizzo 4

“%”: viene attivato come listener il dispositivo con indirizzo 5

La chiave di lettura della tabella Tab. 13 è la seguente; sono chiaramente evidenziate quattro colonne:

CONTROL
NUMBER SYMBOLS
UPPER CASE
LOWER CASE

Supponiamo che *la linea ATN sia al livello logico 1.*

Se il codice inviato sulle linee del bus dati appartiene alla colonna CONTROL, e corrisponde ad un valore compreso tra 0_{10} e 15_{10} , ad esso viene associato il significato di *Addressed Command* (come indicato nella colonna stessa, in basso a sinistra). Ad esempio, al codice ASCII 'HT' è associato il comando 'TCT' (Addressed Command: Take Control).

Se il codice inviato sulle linee del bus dati appartiene alla colonna CONTROL, e corrisponde ad un valore compreso tra 16_{10} e 31_{10} , ad esso viene associato il significato di *Universal Command* (come indicato nella colonna stessa, in basso a destra). Ad esempio, al codice ASCII 'CAN' è associato il comando 'SPE' (Universal Command: Serial Poll Enable).

Se il codice inviato sulle linee del bus dati appartiene alla colonna NUMBERS SYMBOLS (corrispondente ad un valore compreso tra 32_{10} e 63_{10}), ad esso viene associato il significato di *Listen Address* (come indicato nella colonna stessa, in basso). Il valore binario dell'indirizzo si ricava dai valori $B_5B_4B_3B_2B_1$, letti alla voce BITS in corrispondenza alla riga e alla colonna individuate dalla posizione del codice ASCII inviato. Ad esempio, al codice ASCII '8' corrisponde il Listen Address 11000_2 ($B_5=1$, $B_4=1$, $B_3=0$, $B_2=0$, $B_1=0$) oppure 24_{10} .

Se il codice inviato sulle linee del bus dati appartiene alla colonna UPPER CASE (corrispondente ad un valore compreso tra 64_{10} e 95_{10}), ad esso viene associato il significato di *Talk Address* (come indicato nella colonna stessa, in basso). Il valore binario dell'indirizzo si ricava dai valori $B_5B_4B_3B_2B_1$, letti alla voce BITS in corrispondenza alla riga e alla colonna individuate dalla posizione del codice ASCII inviato. . Ad esempio, al codice ASCII 'X' corrisponde il Talk Address 11000_2 ($B_5=1$, $B_4=1$, $B_3=0$, $B_2=0$, $B_1=0$) oppure 24_{10} .

Se il codice inviato sulle linee del bus dati appartiene alla colonna LOWER CASE (corrispondente ad un valore compreso tra 96_{10} e 126_{10}), ad esso viene associato il significato di *Secondary Address* o di *Secondary Command* (come indicato nella colonna stessa, in basso). L'uso del codice corrispondente a 127_{10} è riservato.

Si osservi come i bit B_7 e B_6 assumano sempre la configurazione '01', in corrispondenza a Listen Address, e '10' nel caso di Talk Address.

Per quanto riguarda la gestione della linea ATN si assume che tutti i dispositivi, quando tale linea è al livello logico 1, devono considerare quanto loro inviato come un comando da parte del controller: l'invio di un indirizzo si può allora interpretare come un caso particolare di comando rivolto ad uno specifico strumento, con l'ordine di disporsi a funzionare nella modalità di talker o listener indicata nel comando stesso. Il dispositivo si predispone a svolgere il ruolo assegnatogli, in attesa che il controller concluda, disattivando la linea ATN, la fase di selezione e impostazione dei dispositivi coinvolti nella comunicazione. Successivamente ha inizio lo scambio di dati tra il talker e i listener selezionati, secondo la procedura di handshaking precedentemente descritta nel paragrafo 1.3.4.3.

Esempio: Si consideri il sistema di misura con la seguente semplice configurazione:

Calcolatore (PC)

Scheda di interfaccia 488 (alloggiata nel PC)

Strumento di misura connesso al bus 488 (es: multimetro digitale)

Graficamente la situazione si presenta come in Fig. 30

Dal momento che la scheda di interfaccia 488 è montata sul PC, questi può assumere il ruolo di controller, oltre che di talker e di listener. Supponiamo che il multimetro possa essere predisposto come listener e come talker, con indirizzo primario 5 e nessun indirizzo secondario; lo strumento, inoltre, è in grado di rispondere ad un Serial Poll con il proprio Status Byte.

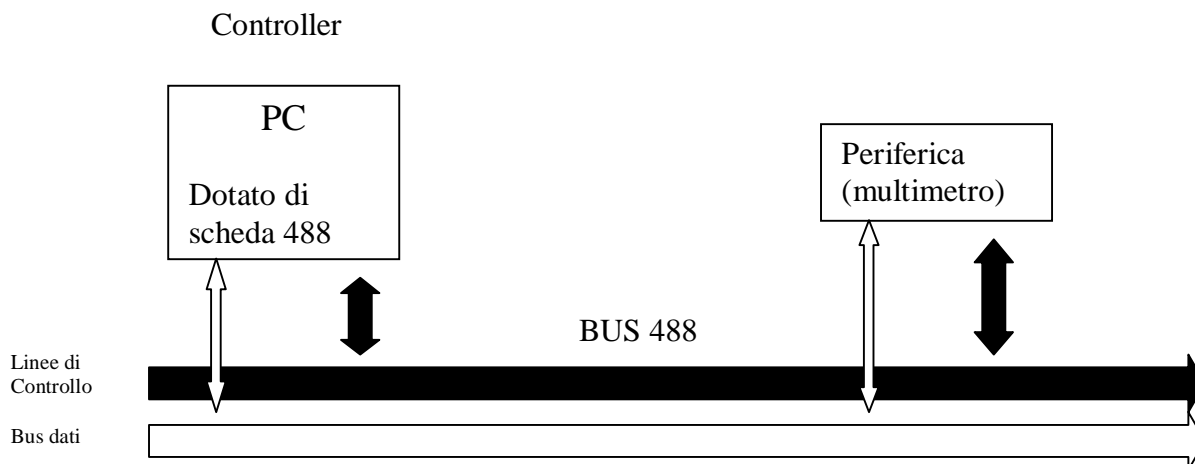


Fig. 30 Semplice configurazione di un sistema di misura

Si analizza ora la sequenza delle azioni svolte dal controller (C) e dal dispositivo (D) e i messaggi scambiati, nell'ipotesi in cui il PC richieda al multimetro una lettura della tensione misurata, attendendo poi la risposta da parte dello strumento.

Azioni svolte da controller (C) e dispositivo (D) per l'invio di un comando al dispositivo stesso

(C): attiva la linea ATN (Command Mode)

(C): invia sul bus dati il proprio indirizzo primario (come talker): le linee DIO assumono la configurazione 01000000₂

(C): invia sul bus dati l'indirizzo primario del multimetro (come listener): in codice ASCII-7bit al valore decimale 5 corrisponde il carattere "%", mentre le linee DIO assumono la configurazione 00100101₂

(C): mette a disposizione sul data bus il messaggio (per esempio: "MEAS:VOLT ?") espresso in codice ASCII, in accordo alla Talking Syntax

(D): preleva quanto presente sul data bus, secondo il protocollo di handshaking

(C): attende che si verifichi una richiesta di servizio

Avvio della procedura di Serial Poll nei confronti dei dispositivi collegati al bus

(C): invia il comando SPE (Serial Poll Enable). E' un comando di tipo Universal, diretto a tutti i dispositivi. In codice ASCII-7bit corrisponde al carattere "CAN", mentre le linee DIO assumono la configurazione 00011000₂

(C): invia sul bus dati l'indirizzo primario del multimetro (come talker): in codice ASCII-7bit corrisponde al carattere "E", mentre le linee DIO assumono la configurazione 01000101₂

(C): invia sul bus dati il proprio indirizzo primario (come listener): le linee DIO assumono la configurazione 00100000₂

Prelievo dello Status Byte di un dispositivo (procedura ripetuta per ogni dispositivo)

(C): disabilita la linea ATN (Data Mode)

(D): mette a disposizione sul data bus il proprio Status Byte

(C): preleva quanto presente sul data bus, secondo il protocollo di handshaking

Conclusione della procedura di Serial Poll

(C): attiva la linea ATN (Command Mode)

(C): invia il comando SPD (Serial Poll Disable). E' un comando di tipo Universal, diretto a tutti i dispositivi. In codice ASCII-7bit corrisponde al carattere "EM", mentre le linee DIO assumono la configurazione 0001100₂

(C): invia il comando UNT (Untalk), che disabilita il talker corrente (in questo caso il multimetro)

Predisposizione di talker (C) e listener (D), allo scopo di soddisfare la richiesta di servizio, effettuata dal dispositivo non appena è disponibile un dato di misura

(C): invia sul bus dati il proprio indirizzo primario (come listener): le linee DIO assumono la configurazione 00100000₂

(C): invia sul bus dati l'indirizzo primario del dispositivo (come talker): in codice ASCII-7bit corrisponde al carattere "E", mentre le linee DIO assumono la configurazione 01000101₂

Prelievo dei dati forniti dal dispositivo

(C): disabilita la linea ATN (Data Mode)

(D): mette a disposizione sul data bus il valore numerico corrispondente alla misura effettuata (per esempio: "1.27"), nel formato specificato dallo standard (nel caso il formato Floating Point per i *dati in risposta* descritto in Figg. 13 e 14)

(C): preleva quanto presente sul data bus, secondo il protocollo di handshaking

(C): attiva la linea ATN (Command Mode)

(C): predispone nuovamente il multimetro ad operare come listener