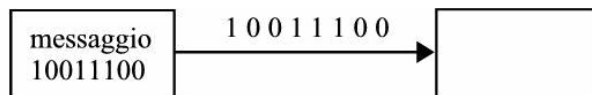
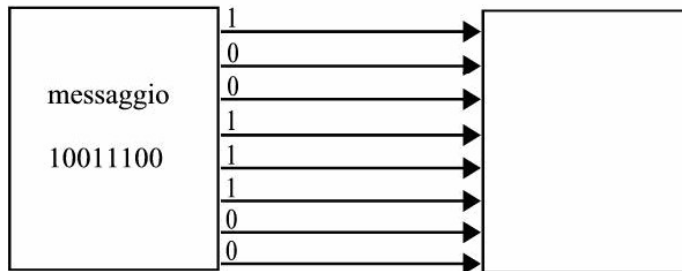


Trasmissione seriale e parallela

- La trasmissione dei dati si dice seriale, se questi sono trasmessi attraverso il link un bit alla volta, in sequenza. È questo il caso, per esempio, quando è presente un solo cavo per la trasmissione.
- La trasmissione si dice parallela quando tutti i bit di una parola sono trasmessi contemporaneamente, in parallelo. Per ottenere ciò, è necessario avere a disposizione tanti cavi (o canali) quanti sono i bit della parola.



TRASMISSIONE SERIALE



TRASMISSIONE PARALLELA

Full duplex / Half duplex / Simplex

- Una linea di comunicazione si dice full duplex se è in grado di ricevere e trasmettere contemporaneamente. Ci sono due canali: uno in ingresso l'altro in uscita.
- Si dice half duplex se non può trasmettere e ricevere allo stesso tempo. C'è un solo canale su cui i dati possono viaggiare in un senso o nell'altro.
- Se la trasmissione è sempre in un solo verso, si parla di Simplex

Nelle trasmissioni seriali si definisce la velocità di trasmissione o "**baud rate**" essa esprime il numero di bit che vengono trasmessi o ricevuti in un secondo. L'unità di misura è il BAUD.

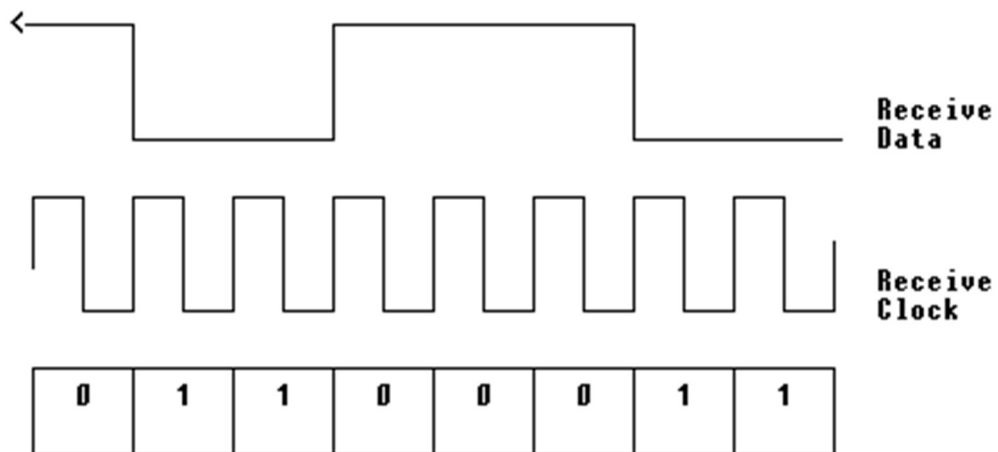
$$\text{Baud - rate} = \frac{N^\circ \text{ bit}}{\text{sec}} \quad 1 \text{ BAUD} = \frac{1 \text{ bit}}{\text{sec}}$$

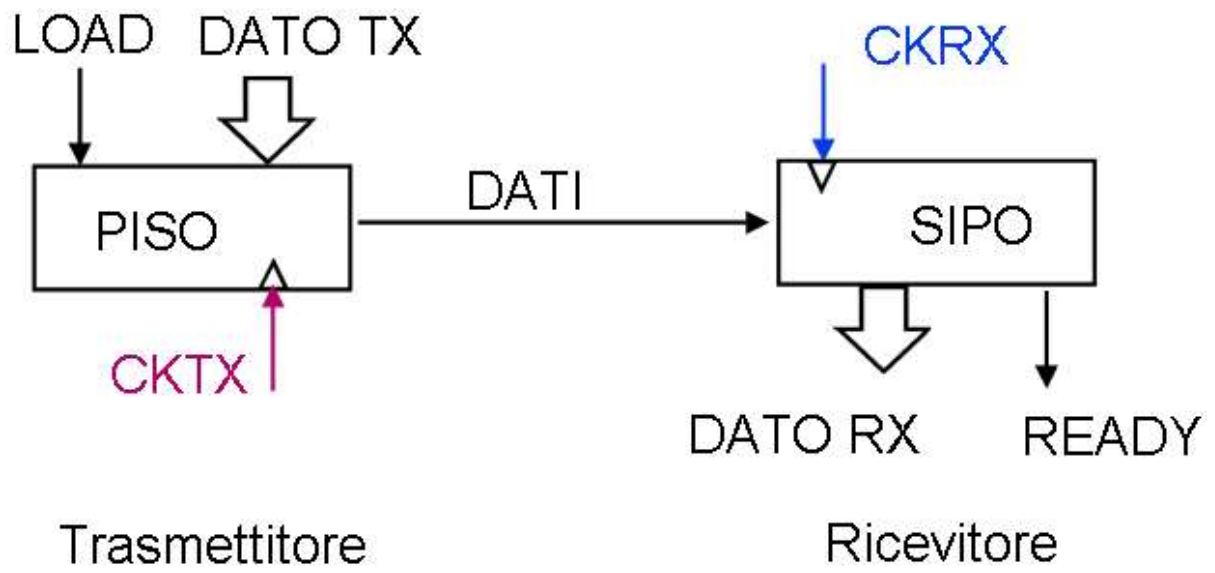
Nella tabella sono riportati alcuni fra i BAUD RATE più comuni con abbinato il tempo necessario per trasmettere un bit T-bit, di un byte T-byte e di 2000 caratteri T-pag (una pagina video di 25 righe di 80 caratteri ciascuna) su linea seriale

B_RATE	T-bit	T-byte	T-pag
150	6,67 ms	53,33 ms	106,67 s
300	3,33 ms	26,67 ms	53,33 s
600	1,67 ms	13,33 ms	26,67 s
1200	0,83 ms	6,67 ms	13,33 s
2400	0,42 ms	3,33 ms	6,67 s
4800	0,21 ms	1,67 ms	3,33 s
9600	104,17 μs	833,33 μs	1,67 s
19200	52,08 μs	416,67 μs	0,83 s
38400	26,04 μs	208,33 μs	0,42 s
57600	17,36 μs	138,89 μs	0,28 s

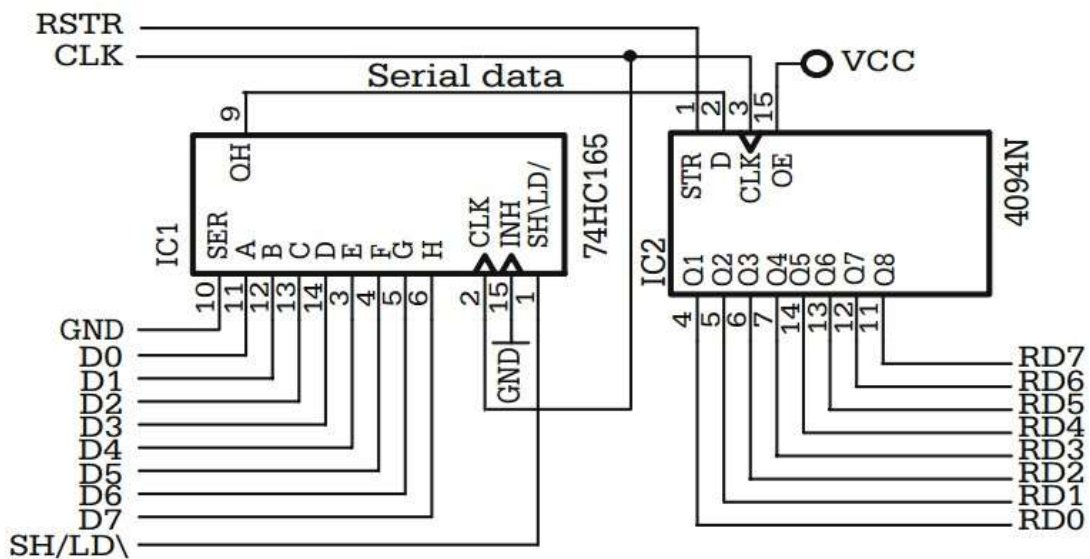
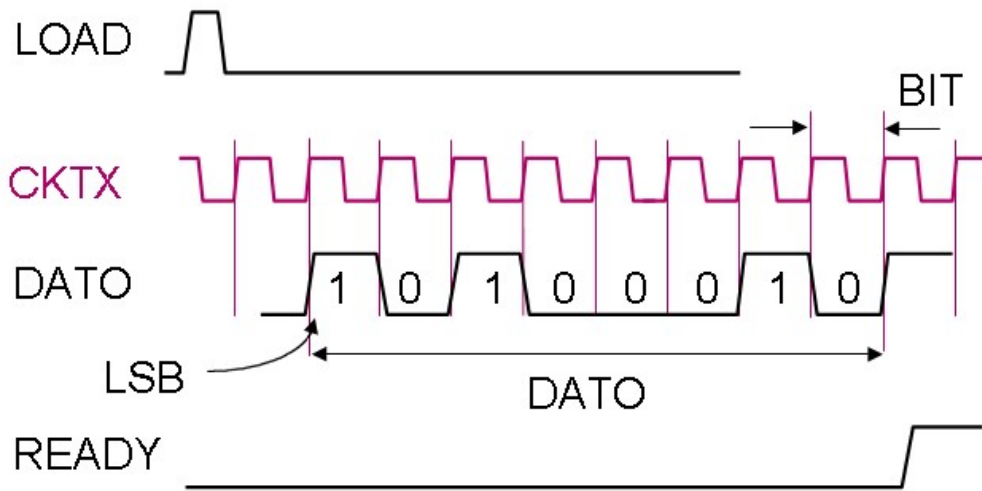
Sincronizzazione

- Sia nella trasmissione seriale sia in quella parallela, spesso viene trasmesso un segnale di sincronizzazione su un canale dedicato (clock).
- Il clock consente al ricevitore di decodificare i segnali nello stesso tempo in cui vengono inviati. Il ricevitore ed il trasmettitore sono, così, sincronizzati nel tempo di clock.
- La comunicazione, in questo, caso si dice sincrona.
- La frequenza del clock definisce anche la velocità della comunicazione che si misura in $\text{bit} \times \text{s} = \text{bps}$ o baud (in realtà baud indica la frequenza con cui la linea cambia stato che non è necessariamente pari a bps).



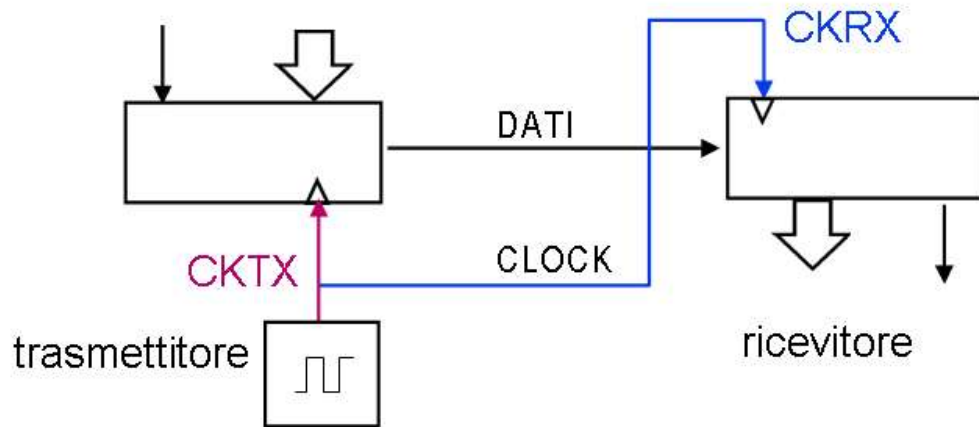


Trasmissione del dato 01000101

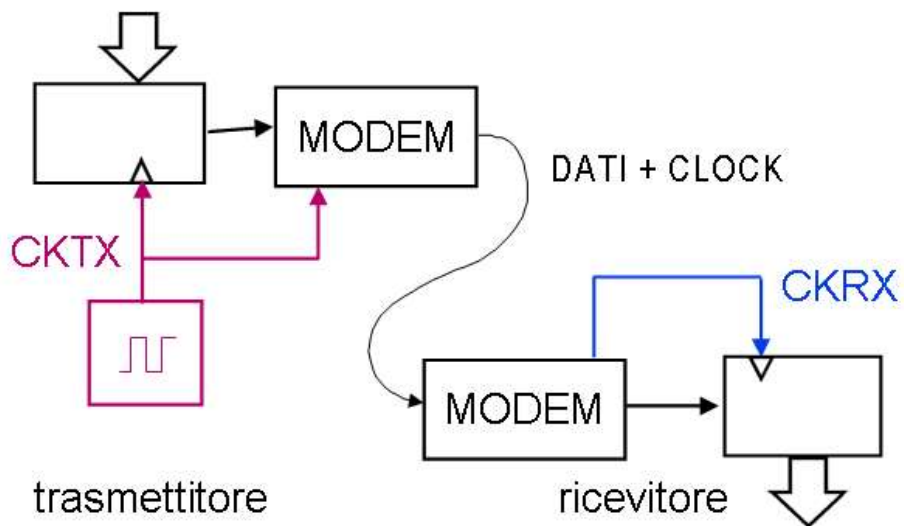


Example of synchronous serial communication circuit

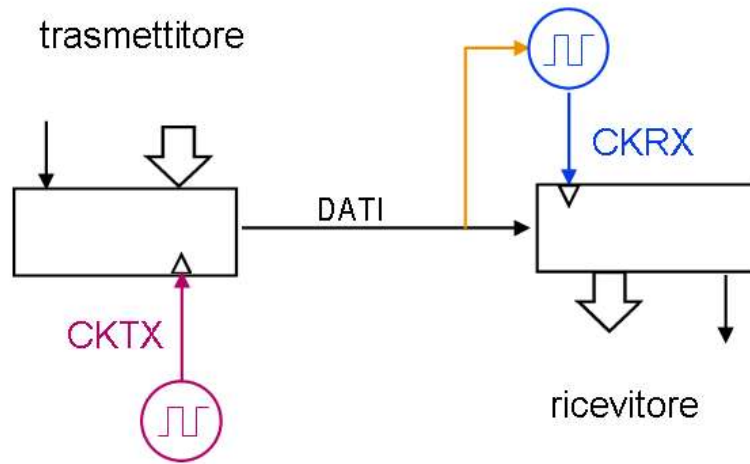
Stesso clock



Estrazione del clock dai dati

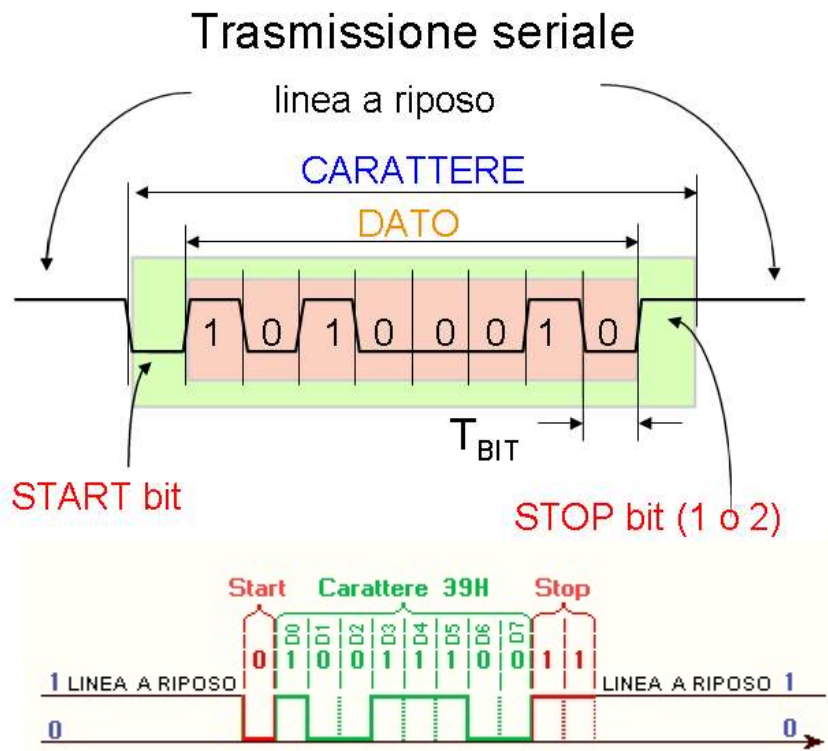


Sincronizzazione del clock lato RX



Comunicazione asincrona

- Nella comunicazione asincrona, non viene trasmesso alcun segnale di clock.
- Ai dati vengono aggiunti in testa ed in coda una serie di bit, detti di start e stop, che istruiscono il ricevitore dell'inizio e della fine di una parola.
- Il vantaggio è che trasmettitore e ricevitore hanno clock indipendenti e le parole possono essere separate da un qualsiasi intervallo di tempo.
- Il clock viene generato e sincronizzato al ricevitore
- L'inizio di un carattere viene segnalato con uno stato L.
- La linea a riposo è in uno stato definito (H).



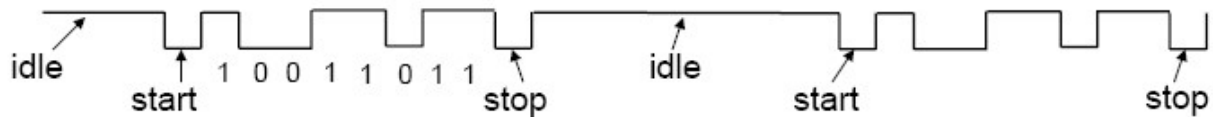
- I circuiti che gestiscono collegamenti seriali asincroni sono detti UART (Universal Asynchronous Receiver/Transmitter)
- L'UART comprende i registri P-S e S-P, e ricava il sincronismo di bit (dallo START) e verifica il formato dei caratteri (STOP)

Sincronizzazione nell' UART

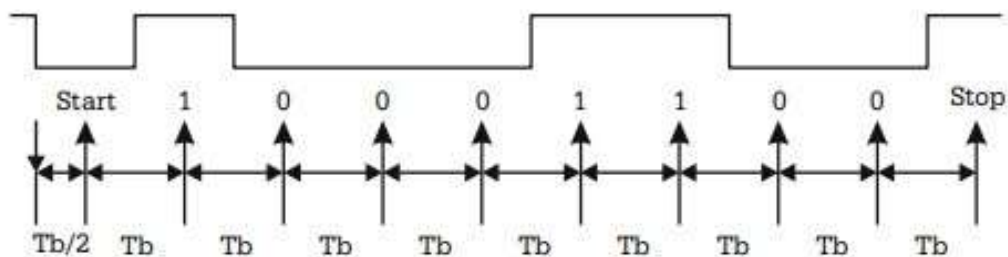
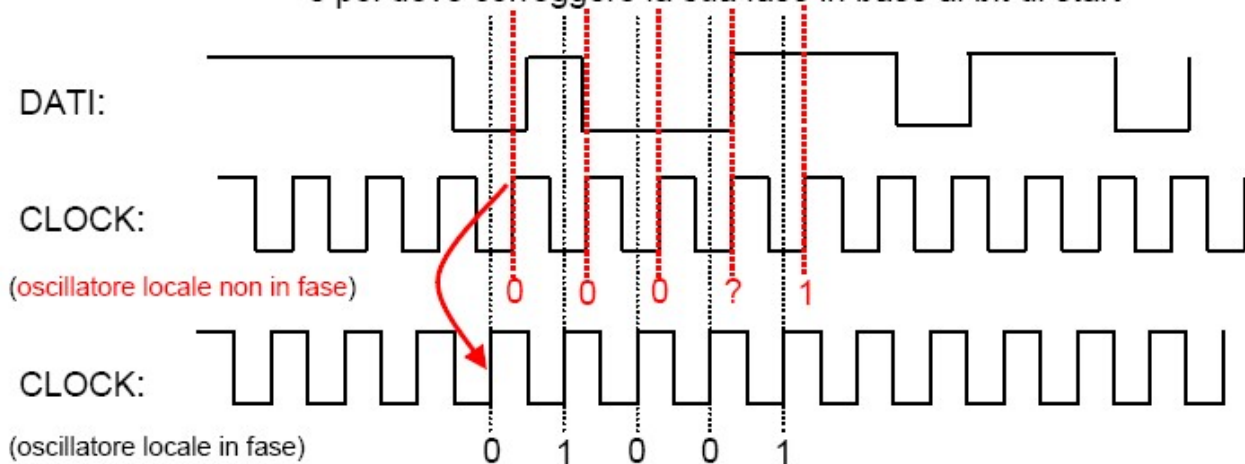
- Si verifica lo stato della linea
- Se diventa bassa, e rimane in questo stato per $T_{BIT}/2$, e' uno start bit
- sincronizza il clock di bit
- inizia il conteggio dei bit (periodi di CKRX)
- Al termine del carattere verifica la presenza di STOP bit

Comunicazione Seriale Asincrona

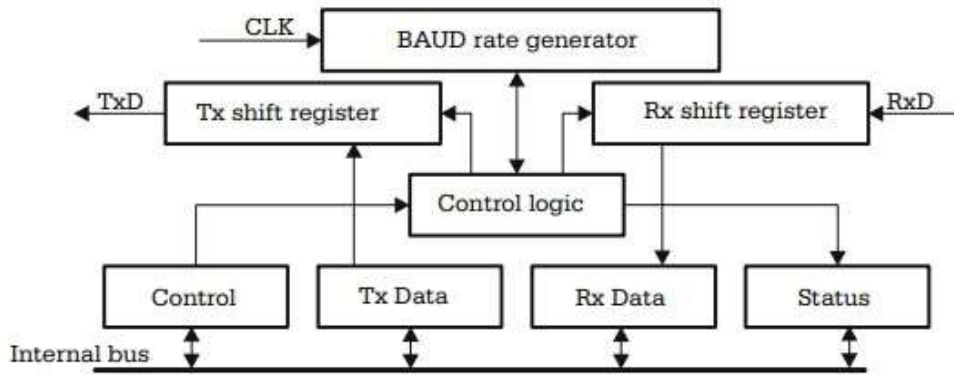
- stato di idle, bit start e bit stop



baud rate → L'oscillatore locale deve agganciarsi alla frequenza corretta e poi deve correggere la sua fase in base al bit di start



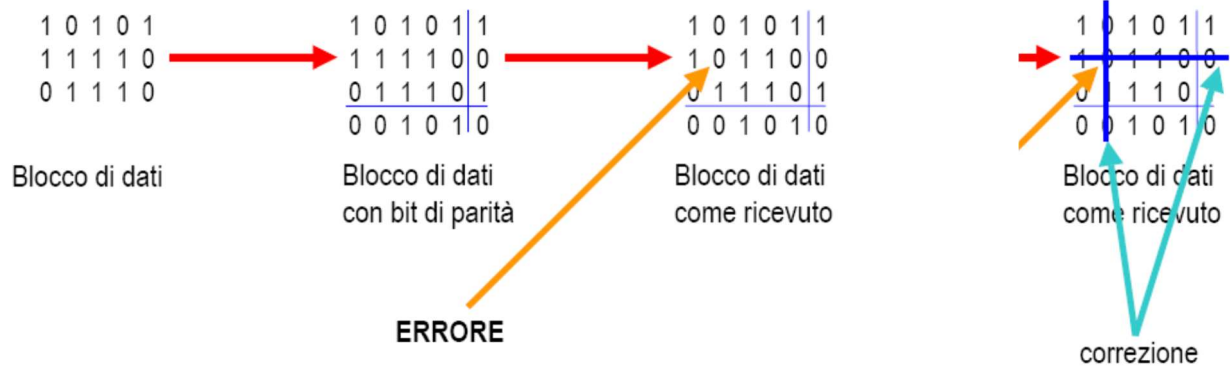
Sampling data line in an asynchronous serial communication



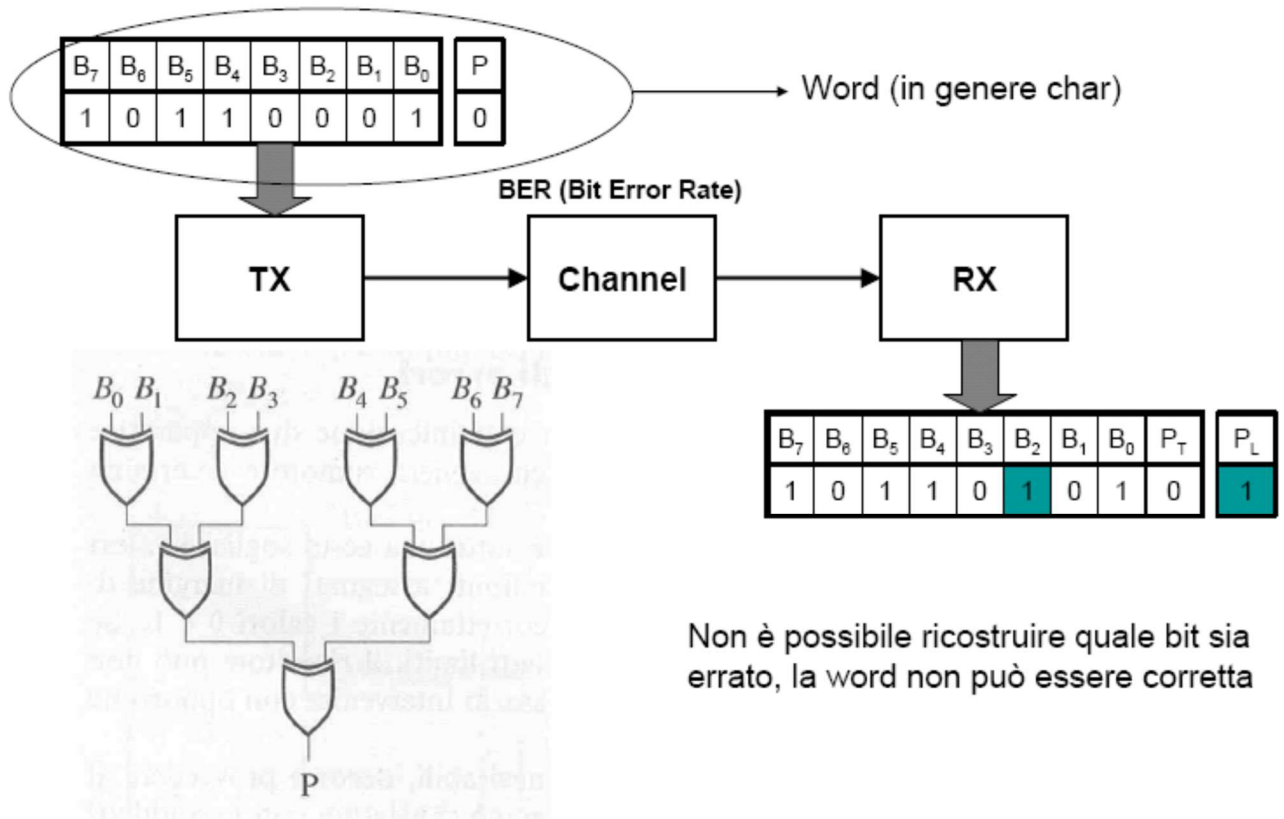
General block diagram of the asynchronous serial communication interface

Rivelazione degli errori - bit di parità

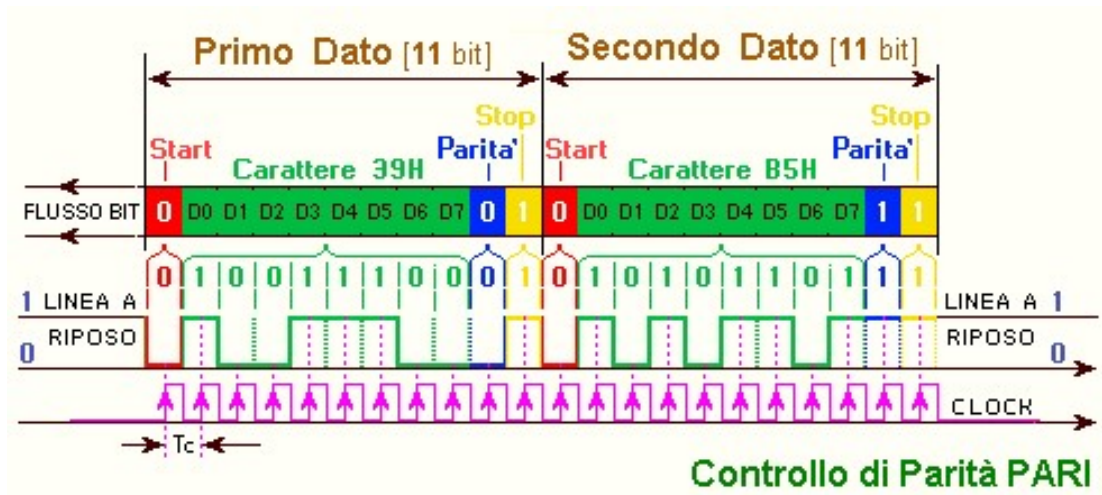
- Può capitare che i bit trasmessi risultino alterati al momento della ricezione, a causa di interferenze o errori nella trasmissione.
- C'è bisogno di un metodo per permettere la rivelazione di errori.
- Si possono trasmettere assieme ai dati uno o più bit di controllo.
- Un metodo comunemente usato è il bit di parità: esso vale 1 se il numero di bit con valore 1 compreso il bit di parità è pari (parità even), 0 se dispari (parità odd).
- Se in ricezione il bit di parità calcolato è diverso da quello ricevuto, un bit deve essere cambiato nel tragitto o erroneamente trasmesso.
- Non funziona se cambia più di un bit ma la probabilità di due errori nella trasmissione di un singolo byte è bassa.



Codice di Parità



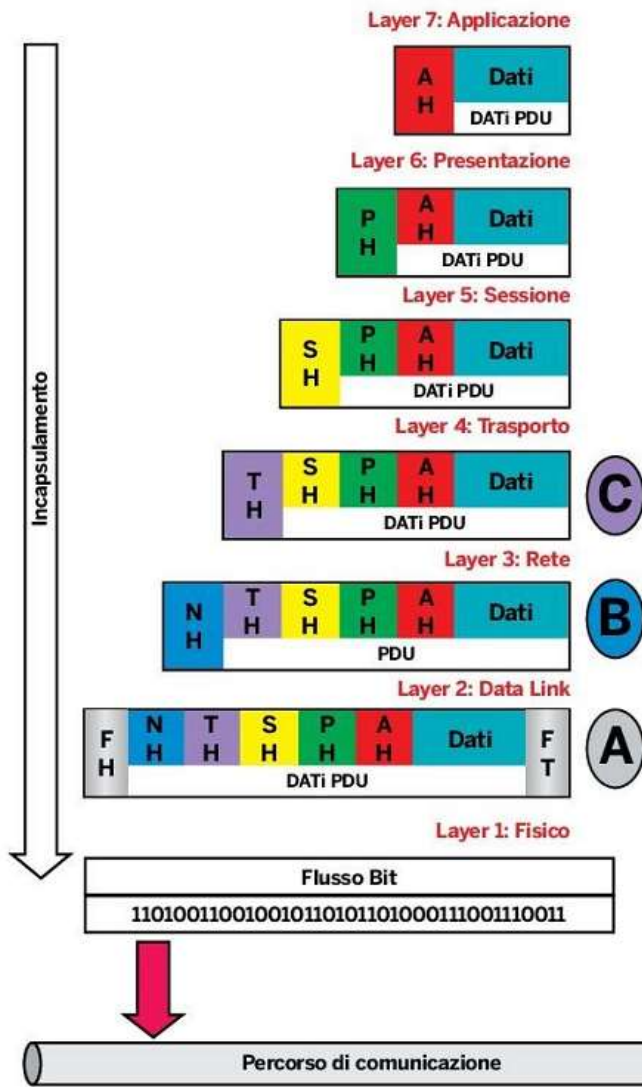
START	sempre 1	SPACE	0
DATO	5 - 6 - 7 - 8 bit	-	
PARITA'	N - E - O		
STOP	1 - 1.5 - 2	MARK	1



Bus di campo (fieldbus) è il termine fissato in ambito IEC (International Electrotechnical Commission) per indicare in un processo automatizzato lo standard di comunicazione "seriale" tra i diversi dispositivi (nodi) costituenti il processo, quali:

- dispositivi di campo (sensori, attuatori, ecc.)
- dispositivi di controllo (PLC, DCS, ecc.)
- La comunicazione tra i nodi è gestita secondo un protocollo che è caratteristico di ogni tipo di bus di campo
- Il modello ISO/OSI è il riferimento per i protocolli

Il modello Open System Interconnection (OSI)



C

2	Porta Sorgente
2	Porta Destinazione
4	Numero Sequenza
4	Numero Acknowledgment
2	Offset, bit riservati, flag
2	Dimensione finestra
2	Checksum
2	Puntatore urgente
4	Opzioni e Padding
Dati	

Segmento TCP

B

1	Versione e intestazione
1	Tipo di servizio
1	Controllo di trasporto
2	Lunghezza del datagramma
2	Identificazione
2	Flag e offset frammento
1	Time to Live
1	Protocollo
2	Checksum Header
4	Indirizzo sorgente
4	Indirizzo destinazione
4	(Opzionale) Opzioni e Padding
Dati	

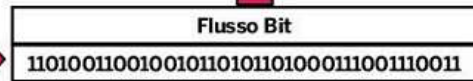
Pacchetto IP

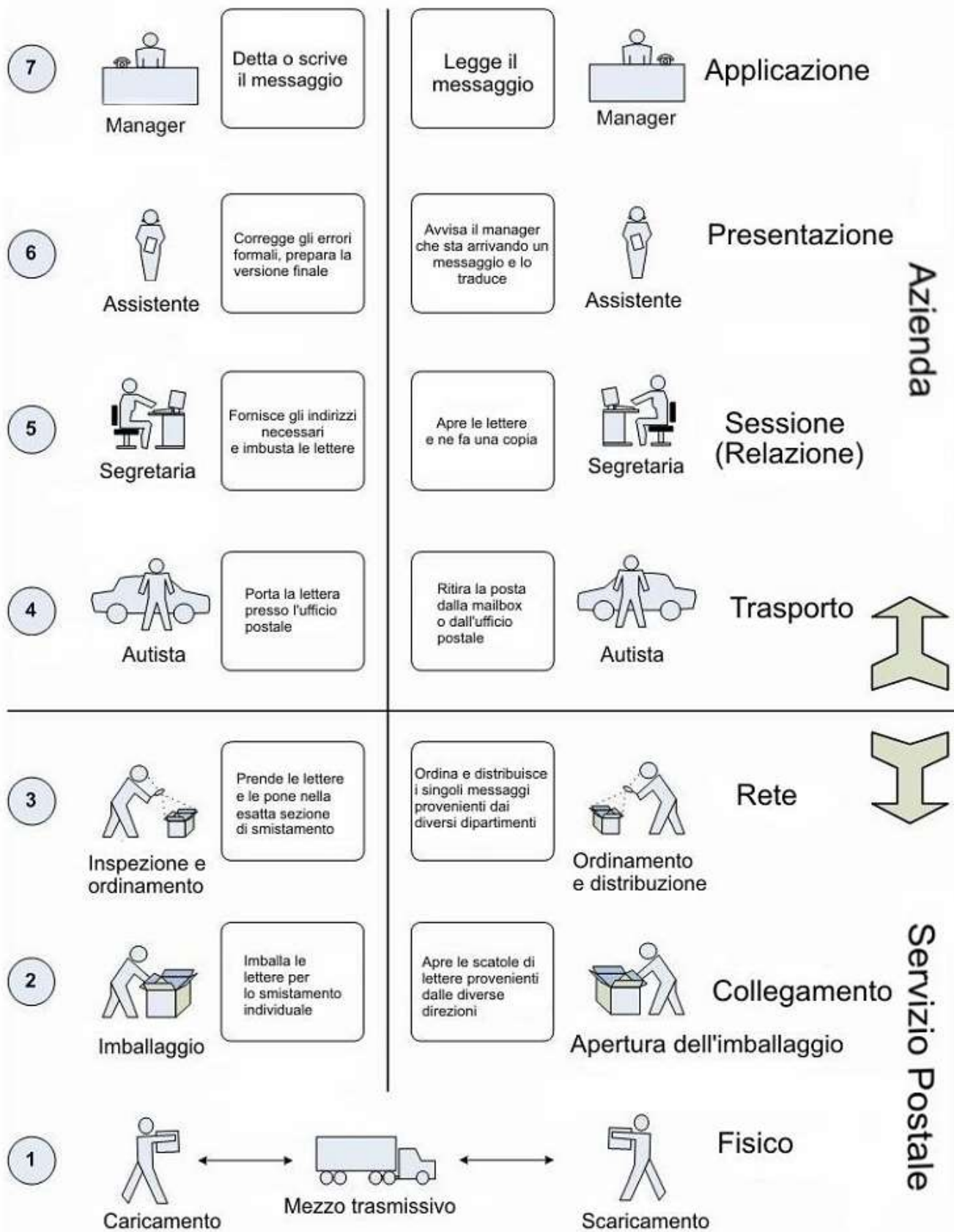
A

7	Preambolo
1	Delimitatore inizio Frame
6	Indirizzo Destinazione
6	Indirizzo Sorgente
2	Tipo
1	DSAP
1	SSAP
1	(o 2) Controllo
Dati	
4	Sequenza di controllo

Frame 802.2

De-incapsulamento



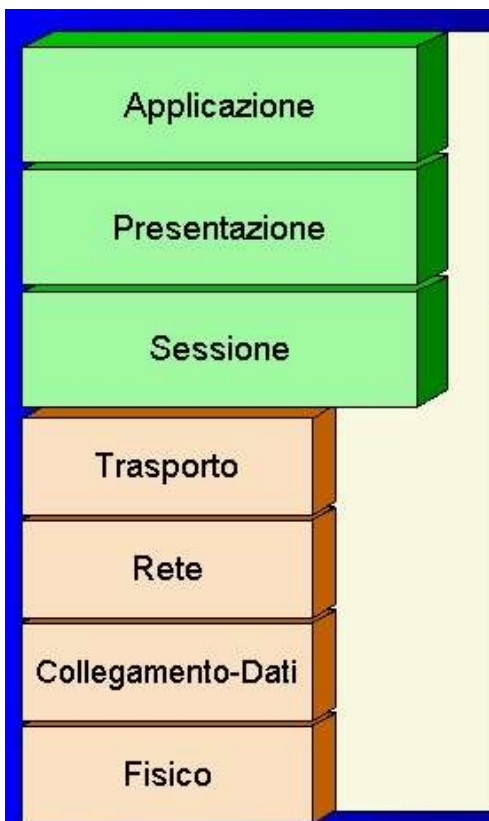


Parallelo tra una lettera e modello ISO - OSI

Il modello di riferimento OSI

Livello 7 Application	Fornisce servizi di uso comune agli utenti (file transfer, posta elettronica, ..)
Livello 6 Presentation	Esegue conversioni sui dati al fine di fornire un'interfaccia standard; svolge funzioni di compressione, formattazione, ...
Livello 5 Session	Fornisce gli strumenti per controllare il dialogo tra applicazioni residenti in sistemi diversi.
Livello 4 Transport	Permette il trasferimento ordinato di dati implementando funzioni di controllo degli errori, controllo di flusso, riordino pacchetti, ...
Livello 3 Network	Maschera i livelli sottostanti, instrada i pacchetti sulle reti fino al destinatario, ...
Livello 2 Data link	Si occupa del trasferimento di blocchi di dati (frame) sul mezzo fisico, implementa le funzioni di accesso al mezzo, il controllo e correzione errori.
Livello 1 Physical	Stabilisce le tecniche di trasmissione dei dati sul mezzo fisico, le caratteristiche del mezzo, parametri fisici (tensioni, velocità, ...), connettori, ...

- La trasmissione delle informazioni tra i componenti di un medesimo elaboratore o fra più elaboratori avviene tramite un mezzo fisico (cavi di rame, fibra ottica, radio, etc.) mediante il quale si ottiene una connessione (link).
- Il modo in cui i segnali ed i dati sono organizzati e si propagano attraverso il link definisce un protocollo di comunicazione.
- Protocolli di comunicazione: regole semantiche e sintattiche che definiscono il comportamento degli elementi di servizio durante lo scambio di informazioni.
- I protocolli possono sommarsi e dar vita a protocolli più complessi (pile di protocollo) e permettere all'utente finale di trasmettere dati nello stesso modo su differenti link fisici (un esempio di pila di protocolli è TCP/IP su Ethernet, o su Fibre Channel)
- All'aumentare dei livelli utilizzati aumenta il grado di complessità in termini di codifica del software, overhead (informazione allegata a un messaggio in rete in modo da assicurare un trasferimento senza errori al destinatario giusto) e tempi di risposta.
- Date le caratteristiche dei fieldbus, in genere, vengono implementati solo i livelli 1, 2 e 7.



Livello Applicazione

Qui si ha la più diretta interazione con l'utente. Il livello descrive come le applicazioni implementano servizi di rete, sincronizzano la comunicazione, determinano con chi comunicare e la disponibilità delle risorse. Un client di posta elettronica, un web browser, una sessione telnet sono tipici esempi di protocolli appartenenti a questo livello.

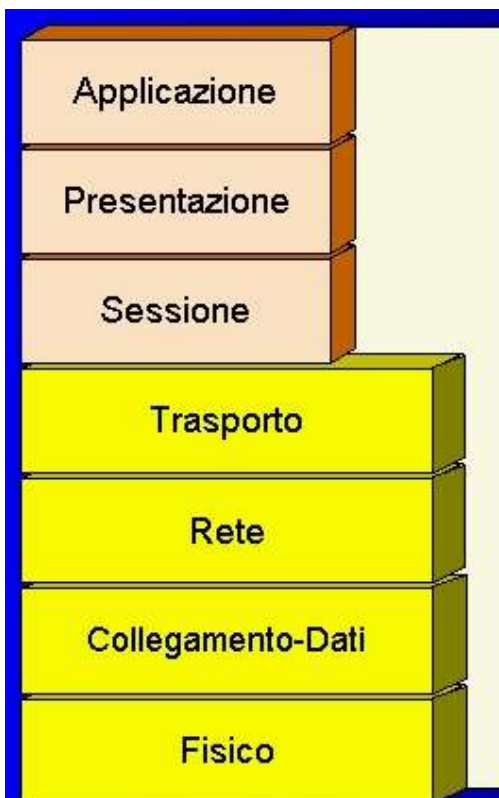
Livello Presentazione

Il livello di Presentazione si occupa della formattazione e conversione dei dati, in modo che le informazioni provenienti dal livello di applicazione di un sistema, siano correttamente leggibili dall'altro.

Il livello si occupa anche della criptazione e decrittazione dei dati (es. SSL). Pensiamo ai vari tipi di codificazione dell'informazione (ASCII, EBCDIC, Real Audio, Gif, Jpeg, ecc): essi devono essere opportunamente convertiti perchè risultino comprensibili al sistema con cui si comunica.

Livello Sessione

Questo livello gestisce la comunicazione tra i livelli di presentazione: la inizia, mantiene e termina secondo opportune modalità.



Livello Trasporto

Definisce le modalità ed i protocolli necessari a gestire una comunicazione affidabile o non affidabile (best effort), il recupero degli errori, il controllo del flusso dati.

Vengono creati i segmenti di dati.

Esempi di protocollo (TCP, UDP)

Livello Rete (Networking)

A questo livello operano tipicamente dispositivi di instradamento quali i router. Il livello definisce le modalità di instradamento, di indirizzamento logico, di determinazione del percorso. Vengono creati i pacchetti di dati.

Esempi di protocollo: IP, ICMP, ARP.

Livello Collegamento-Dati (Data-Link)

I dati vengono combinati in trame (frames). Si definiscono le modalità di accesso al medium (es. CSMA/CD), si rivelano (ma non correggono) gli errori.

A questo livello operano dispositivi quali i bridge, gli switch, le NIC.

Livello Fisico

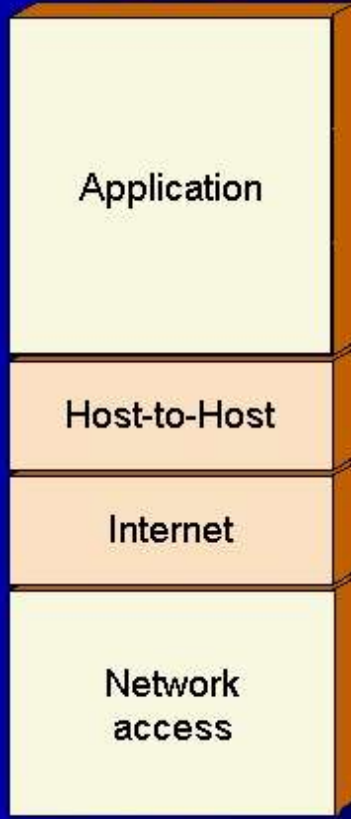
Il livello si occupa del trasferimento dei bit tra i dispositivi, specificandone le caratteristiche elettriche e meccaniche (tipo di codifica, tensioni, cavi, ecc).

A questo livello operano dispositivi come gli hub ed i repeater.

OSI



TCP/IP



Telnet, FTP, HTTP,
DNS, SMTP, POP

TCP, UDP

IP, ICMP, ARP, IPX

Ethernet,
(IEEE802.3),
Frame Relay, PPP

Modello ISO-OSI e Dispositivi

Rete



Router

Data - Link



NIC



switch



bridge

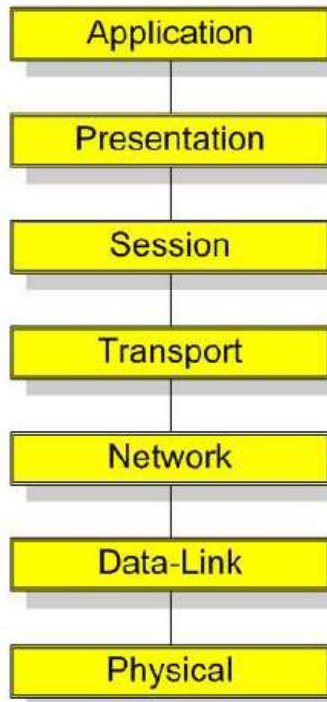
FISICO



RJ 45



hub



Organizzazione del Sistema postale



- Una serie di passi
- Ad ogni passo viene eseguito un particolare compito, su un **messaggio** che viene integrato e trasferito ad un altro agente, seguendo specifiche regole di esecuzione

Organizzazione del servizio postale: altra visione

ROSSI	TAKAMURA
Segretaria	Segretaria
Fattorino	Fattorino
Postino	Postino
Smistamento	Smistamento
Stazione	Stazione

Viaggio

- Strati:** ogni strato implementa un servizio
- attraverso le sue azioni interne allo strato
 - confida sui servizi forniti dagli strati sottostanti
 - ha uno strato corrispondente e reciproco



Modello a strati

- Elementi fondamentali del modello stratificato:
 - Flusso dati
 - Servizi
 - Protocolli
 - Interfacce

Definizioni

■ Strato

- E' un modulo interamente definito attraverso i servizi, protocolli e le interfacce che lo caratterizzano
- Nota: e' spesso indicato con il nome di LIVELLO

■ Servizio

- insieme di primitive (operazioni) che uno strato fornisce ad uno strato soprastante.

Definizioni

■ Protocollo

- insieme di regole che governano il formato e il significato dei frame, dei pacchetti o dei messaggi che vengono scambiati tra gli strati paritari di due entità diverse

■ Interfaccia

- insieme di regole che governano il formato e il significato dei frame, dei pacchetti o dei messaggi che vengono scambiati tra due strati successivi della stessa entità

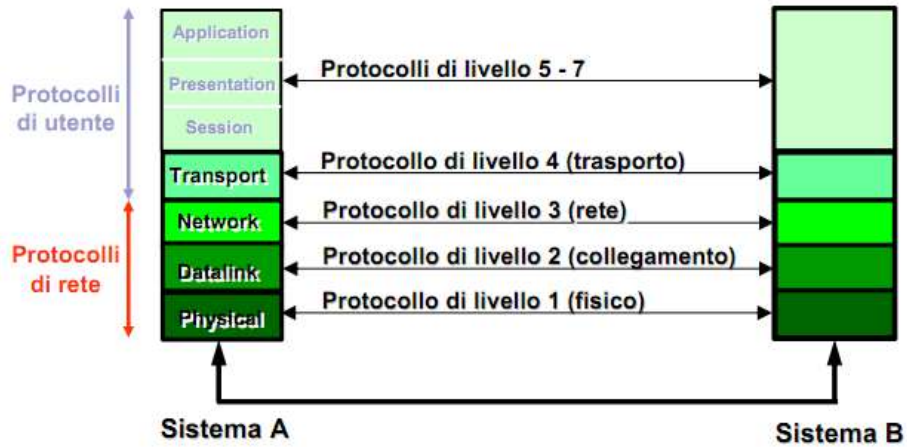
Modello ISO/OSI

- Modello ISO/OSI (1970)
- **International Standardization Organization / Open System Interconnection**
 - Normativa per la definizione dell'interconnessione di sistemi *aperti* cioè capaci di comunicare fra loro indipendentemente:
 - dal costruttore;
 - dal mezzo di comunicazione.
 - Protocolli
 - insieme di procedure che governano il flusso informativo fra entità della rete;
 - definiscono il formato e l'ordine dei messaggi scambiati tra entità della rete;
 - descrivono le azioni intraprese in corrispondenza della trasmissione e ricezione dei messaggi.
 - Livelli
 - astrazione di una funzione logica ben definita.
 - Interfacce
 - regole per lo scambio di informazioni fra due livelli adiacenti.

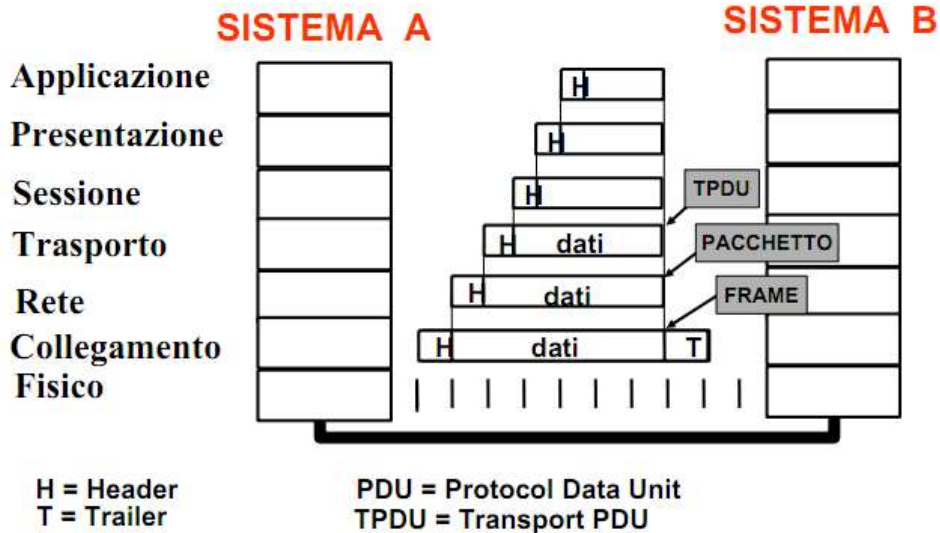
Modello ISO/OSI

- Nel modello OSI sono previsti 7 livelli
 - 3 LIVELLI DI RETE
 - fisico,
 - link,
 - rete
 - 4 LIVELLI DI UTENTE
 - trasporto,
 - sessione,
 - presentazione,
 - applicazione

Protocolli di comunicazione



Incapsulamento dei dati

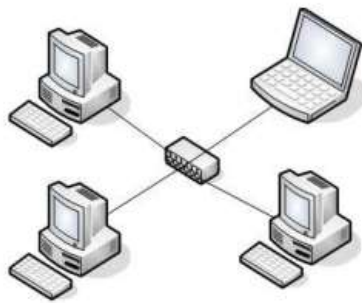


Organizzazione del servizio postale: estensione

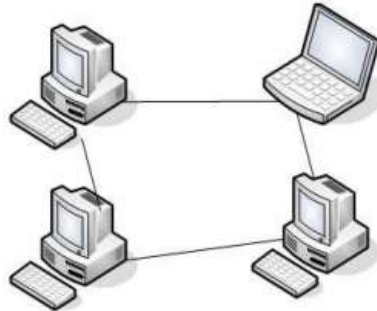


Trasporto: una o più stazioni intermedie

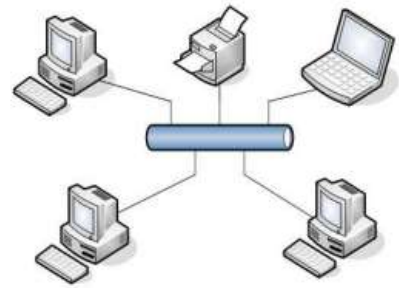
- accettano lettere da più mittenti
- le inoltrano verso la destinazione finale



a) Rete a stella



b) Rete ad anello



c) Rete a bus

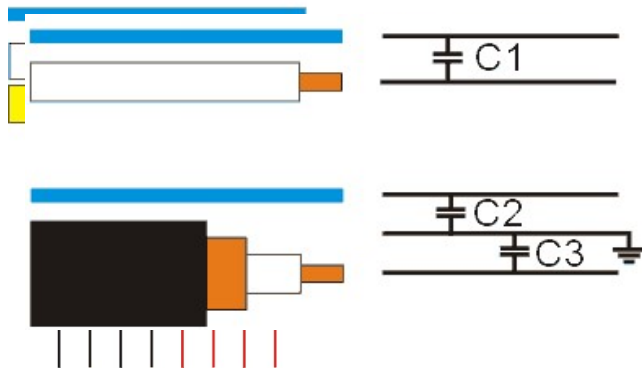
Criteri di selezione

- **Ambiente di lavoro:** fieldbus che possono operare in luoghi con pericolo di esplosione (industria chimica) perché operanti a sicurezza intrinseca cioè non sono in grado di generare scintille in grado di innescare incendi; presenza di temperature elevate; ambiente con forti disturbi di natura elettromagnetica (stazioni di saldatura, ...); ...
- **Topologia del bus:** i bus ad anello potrebbero essere scomodi da installare in certe applicazioni in cui i dispositivi sono fisicamente disposti in modo lineare.
- **Lunghezza del bus:** si va dal bus che collega i dispositivi di una singola macchina (< 100m) al bus che collega un'intera linea di produzione (< 1000m) ai bus usati per l'automazione di edifici (>> 1000m). Esiste sempre un legame fra lunghezza e velocità di trasmissione. Esistono dispositivi di amplificazione (ripetitori) in grado di estendere la lunghezza del bus: occupano spazio e devono essere alimentati!
- **Numero di dispositivi collegabili:** si va da pochi elementi di I/O a migliaia, quello che interessa è il numero di dispositivi direttamente connessi al bus (esiste sempre un limite fisico o di indirizzamento).
- **Facilità di cablaggio:** è importante se vi sono molti dispositivi collegati al bus. Per alcuni fieldbus vi sono particolari cavi a perforazione di isolante molto comodi anche se costosi. Vi sono fieldbus in cui l'alimentazione dei dispositivi viene prelevata direttamente dal bus.
- **Possibilità di avere più master del bus:** si hanno bus in cui vi è uno e un solo master, bus in cui vi sono un numero limitato di master e bus in cui tutti i dispositivi connessi possono essere master (peer to peer).
- **Velocità di trasmissione:** è importante non tanto la velocità di picco ma il tempo di ritardo massimo (detto anche tempo di risposta) fra il verificarsi di un evento in campo e l'acquisizione dell'evento da parte del controllore. E' difficile da calcolare perché dipende da tanti fattori. Si va da tempi << 1ms per loop di regolazione assi, a tempi < 10ms per dispositivi di I/O digitali su macchine o per messaggi di allarme, a tempi < 200ms per l'interfaccia con l'operatore a tempi anche nell'ordine dei secondi per la regolazione di temperature in forni industriali.
- **Modalità di interrogazione degli slave:** deterministico o event-driven. In alcune applicazioni (loop di regolazione veloci, ...) è necessario ottenere le informazioni con scadenze temporali fisse, determinabili a priori e con jitter limitato (rapide variazioni in un segnale). In altre (gestione veloce di allarmi, ...) è necessario ottenere l'informazione non appena possibile senza attendere il relativo slot temporale.
- **Dimensione massima del blocco dati:** si va da pochi byte per i dispositivi di I/O a molti Kbyte per i dati di produzione.

- **Il livello fisico**

Mezzi di trasmissione

- **Cavi elettrici:** ciascun fieldbus fornisce le caratteristiche elettriche dei cavi elettrici utilizzabili per il collegamento dei vari dispositivi. Viene definito il tipo di cavo (coassiale, a coppie twistate, con schermatura parziale e/o totale, ...), numero di conduttori (2, 4, ...), sezione minima dei conduttori (0,5 mm², ...), l'impedenza caratteristica del cavo (120 ohm). Spesso viene fornito un elenco dei cavi certificati. Se non vengono usati cavi adatti non si riescono ad ottenere le prestazioni dichiarate oppure ci possono verificare molti problemi sulle comunicazioni. Attenzione ai cablaggi (stub, ...) e alle resistenze di terminazione, se previste
- **Fibre ottiche:** vengono usate raramente per i fieldbus e quasi sempre per collegare i ripetitori fra loro. Sono immuni ai disturbi elettromagnetici, quindi sono molto comode per collegamenti a lunga distanza, anche se più costose dei cavi. Molti fieldbus usano protocolli di livello fisico standard (RS485, ...) e spesso si utilizzano ripetitori standard, non specifici per il fieldbus, per collegare fra loro più spezzoni di bus distanti fra loro.

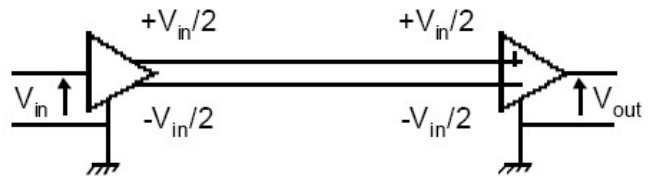


Trasmissione sbilanciata



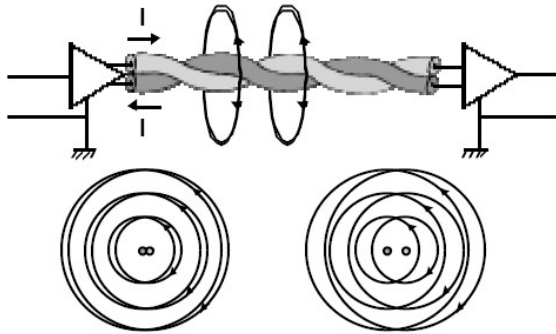
Lo schermo trasporta la tensione di riferimento dei segnali

Trasmissione bilanciata

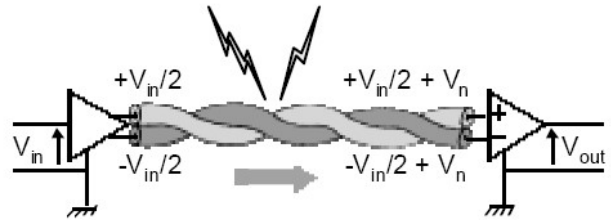


$$V_{out} = \frac{V_{in}}{2} - \left(-\frac{V_{in}}{2} \right) = V_{in}$$

Emissione di disturbi elettromagnetici



Immunità ai disturbi elettromagnetici



$$V_{out} = \frac{V_{in}}{2} + V_n - \left(-\frac{V_{in}}{2} + V_n \right) = V_{in}$$

Trasmissione bilanciata riduce emissioni di disturbi e.m.

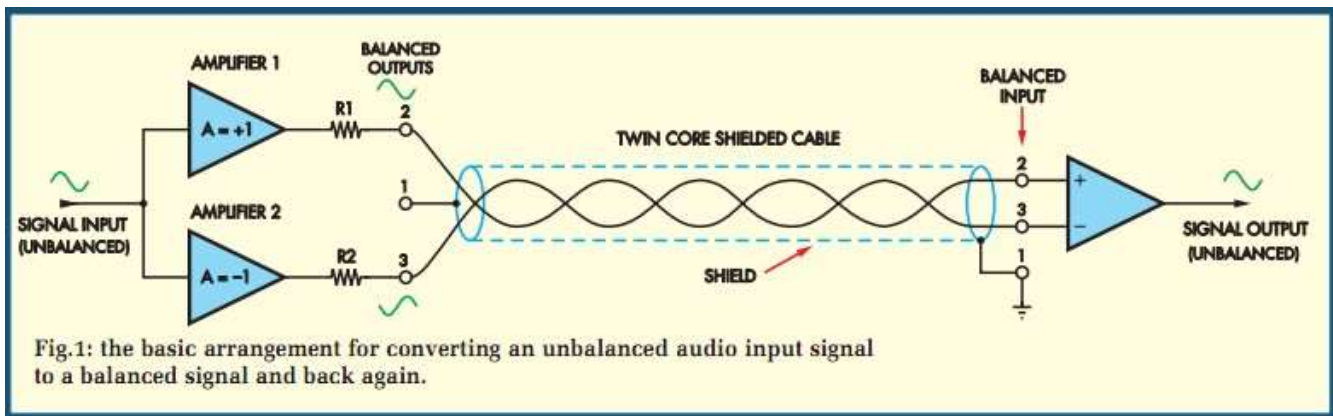
- - correnti nei due conduttori di uguale intensità e verso opposto
- - generano campi magnetici opposti che tendono ad annullarsi

Immunità ai disturbi esterni per amplificazione differenziale del segnale

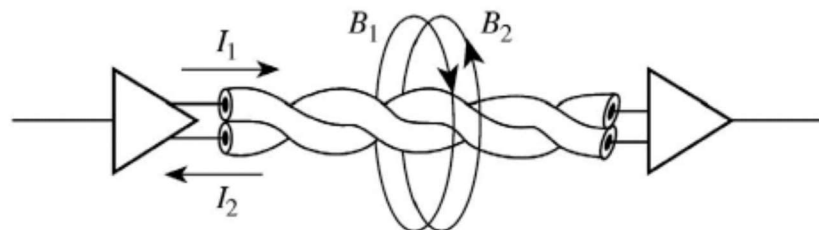
- - un'onda elettromagnetica induce medesima tensione di disturbo
- - tale disturbo scompare nella differenza che genera segnale di uscita

Presupposto fondamentale per la trasmissione bilanciata è che due conduttori siano perfettamente simmetrici rispetto a qualsiasi punto dello spazio, in modo da annullare sia l'emissione che la sensibilità ai disturbi elettromagnetici. La perfetta simmetria potrebbe essere raggiunta soltanto se i due conduttori coincidessero, cosa irrealizzabile per il limite minimo nelle dimensioni fisiche e per la necessità di interporre fra essi del materiale isolante, ma può essere approssimata ritorcendo i due conduttori. Si realizza così il "doppino ritorto" (Twisted Pair). La trasmissione bilanciata su TP riduce le emissioni di disturbi elettromagnetici in quanto le correnti che attraversano i due conduttori sono di uguale intensità e verso opposto, e generano campi magnetici che tendono ad annullarsi.

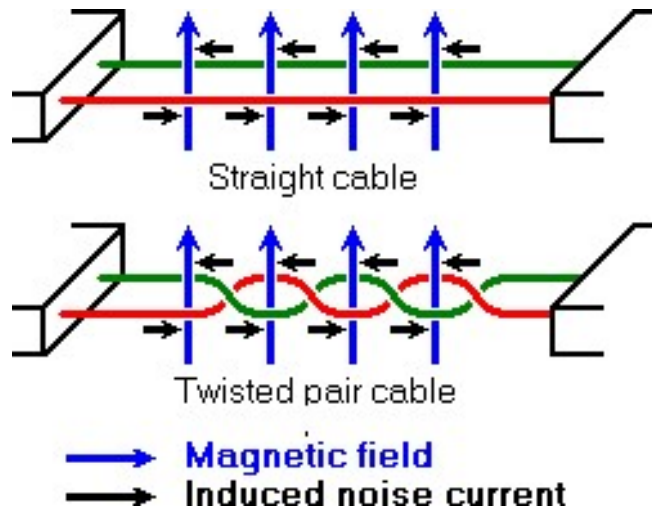
Mentre la ridotta emissione di disturbi è dovuta alla simmetria nella trasmissione, l'immunità ai disturbi esterni è dovuta all'amplificazione differenziale del segnale nel ricevitore. Grazie alla simmetria del sistema onde elettromagnetiche provenienti dall'esterno inducono la medesima tensione di disturbo in entrambi i conduttori, e tale termine scompare nella differenza che genera il segnale d'uscita.



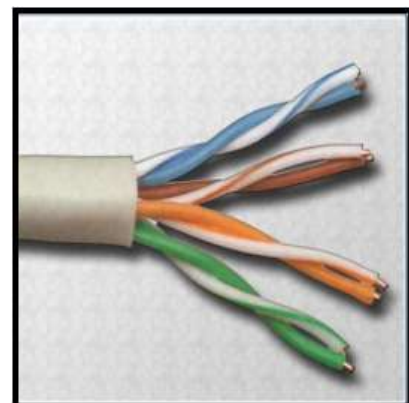
- ❑ Trasmissione bilanciata riduce emissioni di disturbi e.m.
 - correnti nei due conduttori di uguale intensità e verso opposto
 - generano campi magnetici opposti che tendono ad annullarsi
- ❑ Immunità ai disturbi esterni per amplificazione differenziale del segnale
 - un'onda elettromagnetica induce medesima tensione di disturbo
 - tale disturbo scompare nella differenza che genera segnale di uscita



- ❑ Soltanto recentemente la tecnologia ha permesso l'applicazione del TP a sistemi per trasmissione dati ad alta velocità
- ❑ Le difficoltà consistono nel produrre doppi a geometria altamente regolare, in cui gli effetti del rumore risultano simmetrici nei due fili
- ❑ E nel realizzare amplificatori differenziali ad elevata reiezione del modo comune (cioè che misurino la differenza dei segnali con un residuo molto basso di segnale comune), in grado di funzionare a centinaia di MHz.

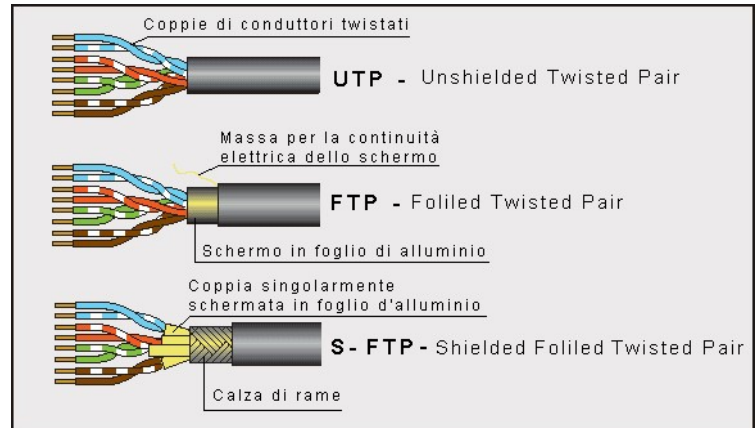


- **UTP (Unshielded Twisted Pair):** è costituito da una coppia di conduttori intrecciati.
 - Un cavo è usualmente costituito da almeno 4 coppie UTP
 - La qualità del cavo è identificata dalla categoria:
 - Cat. 1: per telefonia analogica
 - Cat. 2: per telefonia digitale
 - Cat. 3: per reti locali sino a 10 Mbit/s, banda passante 16 MHz
 - Cat. 4: per reti locali sino a 16 Mbit/s, banda passante 20 MHz
 - Cat. 5: per reti locali e metropolitane sino a 100 Mbit/s, banda passante 100 MHz (attenuazione 20 dB per 100 m)
 - Cat. 5E: sino a 1 Gbps, banda passante 100 MHz
 - Cat. 6: banda passante 250 MHz
 - Cat. 7: banda passante 600 MHz
- **S-UTP (Shielded UTP):** il cavo è avvolto da uno schermo globale metallico
- **STP (Shielded Twisted Pair):** oltre allo schermo globale, esiste una schermatura per ciascuna coppia UTP

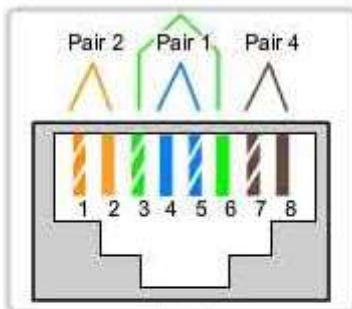




STP (Shielded Twisted Pair)
 24 AWG, 4 coppie, 100 Ω,
 schermatura individuale delle coppie
 schermatura globale in calza di rame



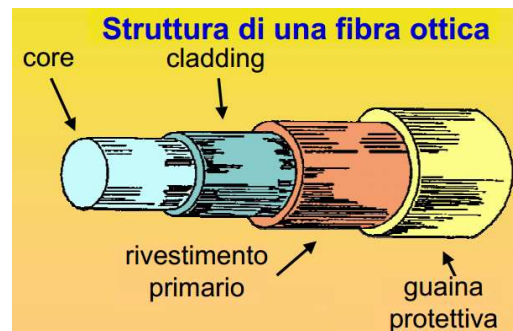
RJ-45 T568B Termination
 Pair 3



T568B

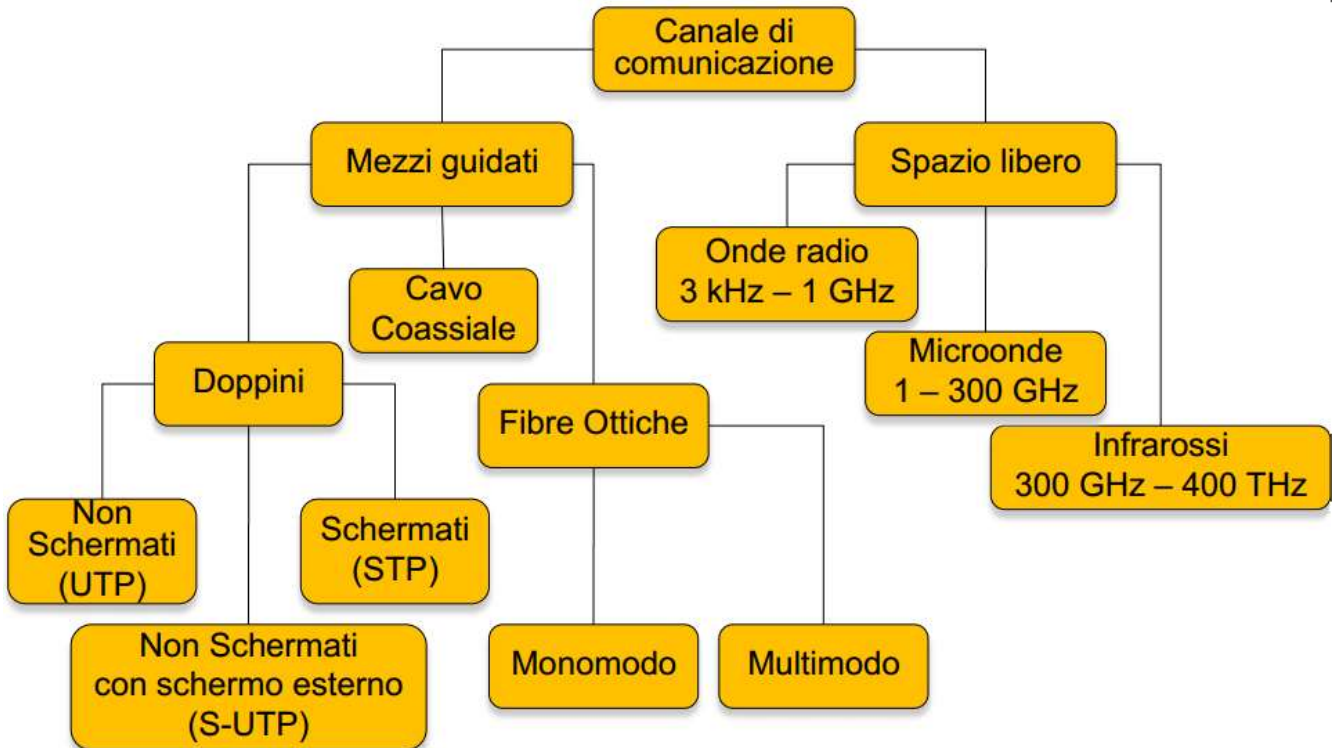
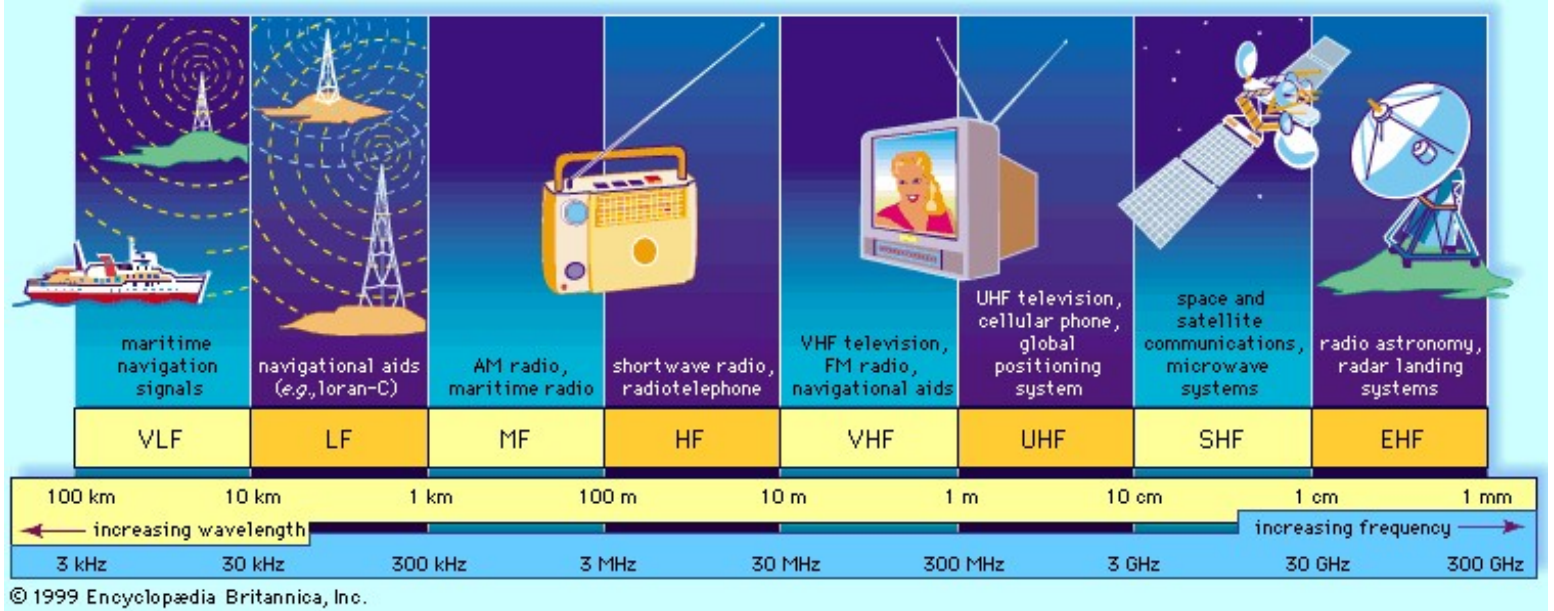


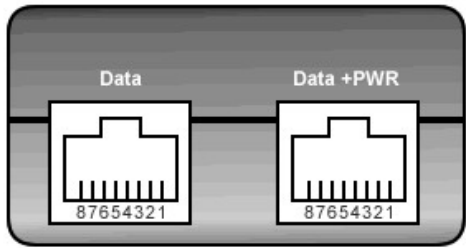
T568B
 (Top View)



Caratteristiche delle fibre ottiche

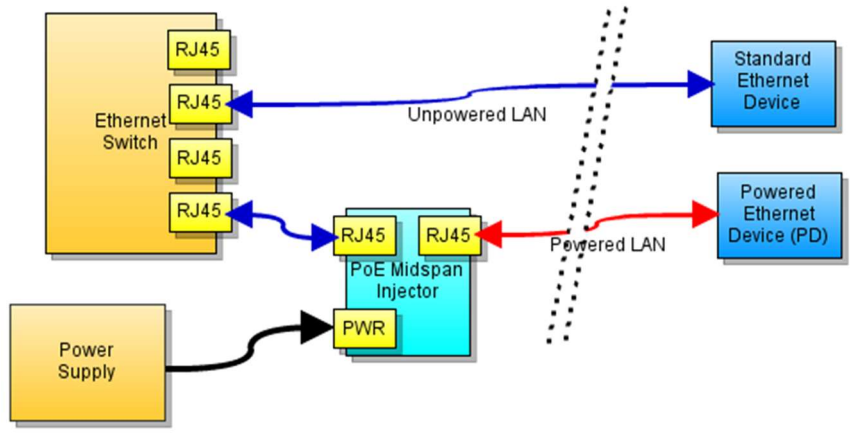
- Numero di fibre nel cavo
- Struttura del cavo (tight, loose)
- Tipo di rivestimento (per interni, per esterni, stagno, ignifugo, ecc.)
- Minimo raggio di curvatura: tipicamente 10 volte il diametro del cavo
- Tipo di trasmissione (monomodale, multimodale)

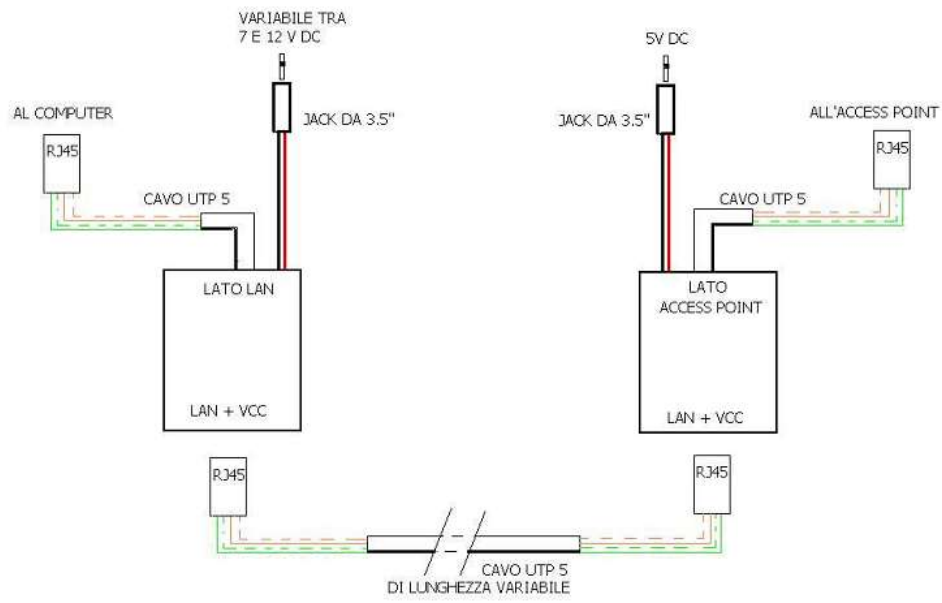




J1 Pins	↔	J2 Pins
1 Data Pair 1	↔	1 Data Pair 1
2 Data Pair 1	↔	2 Data Pair 1
3 Data Pair 2	↔	3 Data Pair 2
4 No Connection		4 + VDC
5 No Connection		5 + VDC
6 Data Pair 2	↔	6 Data Pair 2
7 No Connection		7 - VDC
8 No Connection		8 - VDC

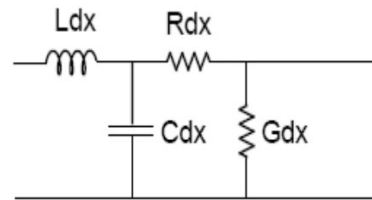
INIETTORE: uscita RJ-45 (dati e potenza)		
Pin	Simbolo	Descrizione
1	RX+	ricezione dati
2	RX-	ricezione dati
3	TX+	trasmissione dati
4	Vo+	potenza di alimentazione (+)
5	Vo+	potenza di alimentazione (+)
6	TX-	trasmissione dati
7	Vo-	potenza di alimentazione (-)
8	Vo-	potenza di alimentazione (-)



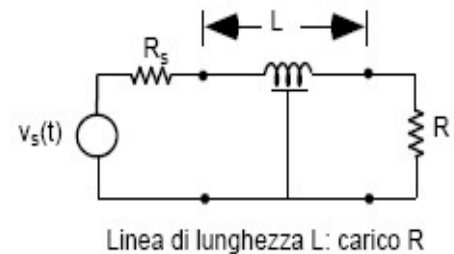


Per linea di trasmissione si intende in genere una coppia di conduttori attraverso cui si propaga un segnale. I due conduttori possono essere paralleli (come in una piattina), coassiali, o composti da un conduttore ed un piano di massa (come nei circuiti stampati).

Un tratto infinitesimo di linea si può schematizzare come in figura, in cui i parametri L , R , C e G sono intesi per unità di lunghezza

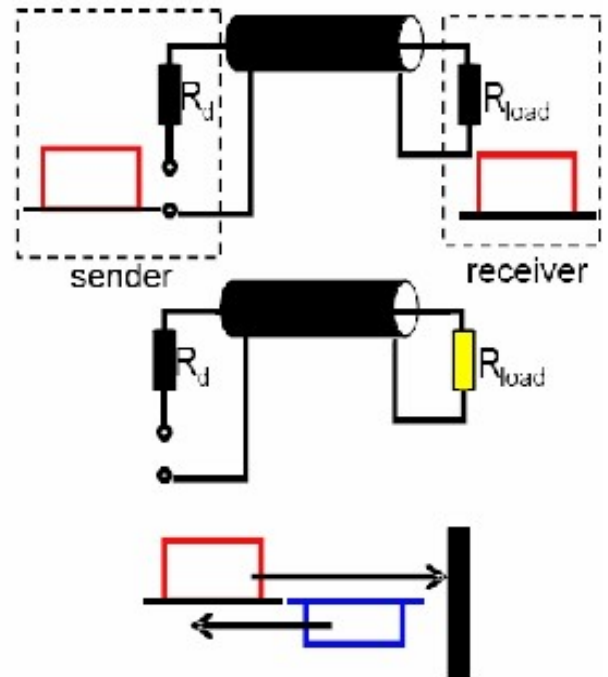


Se R e G sono trascurabili (linea non dissipativa), indicando con $R_0 = \sqrt{L/C}$ l'impedenza caratteristica della linea. Se chiudiamo la linea su una resistenza R , nel punto alla fine della linea, la corrente sarà $v/R \Rightarrow i(L, t) = 1/R_0[v_s - \rho v_s] = 1/R[v_s + \rho v_s]$. Da cui si ricava il coefficiente di riflessione: $\rho = (R - R_0)/(R + R_0)$. Se $R = R_0$ non c'è riflessione: Linea adattata. R è la terminazione.



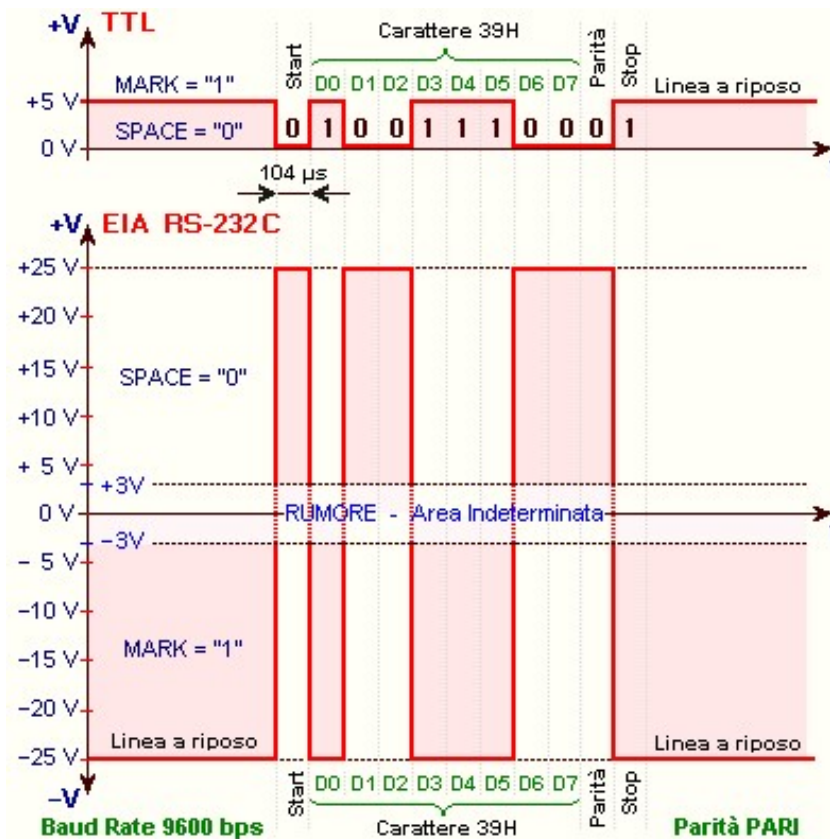
Adattamento di impedenza

Se l'impedenza è adattata il cavo appare infinitamente lungo: nessun segnale riflesso. Se l'impedenza è disadattata un segnale riflesso al termine della linea si sovrappone al segnale inviato.



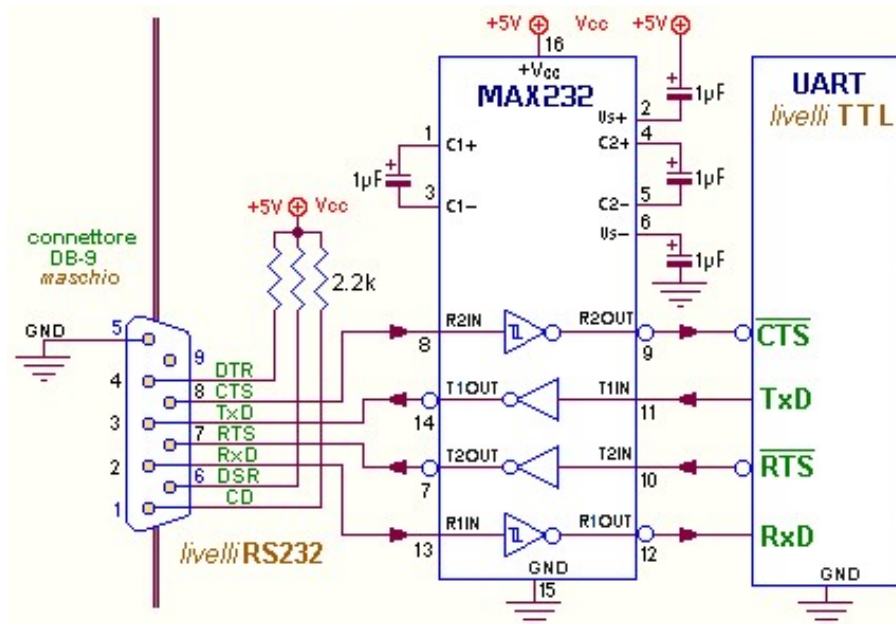
Interfaccia RS-232-C (Electronic Industries Association)

- RS-232 è uno standard per le comunicazioni seriali definito dalla Electronic Industries Association (EIA). RS sta per Recommended Standard.
- E' un protocollo di livello fisico molto comune (PC, ...) ma poco utilizzato in ambiente industriale se non per il collegamento di dispositivi locali.
- Può essere usato per collegamenti punto-punto a breve distanza, ha segnali di TX e RX e alcuni segnali di hand-shake. La massa è in comune a tutti i segnali. E' estremamente sensibile ai disturbi perché usa un collegamento non bilanciato con ritorno sulla massa comune.
- Esiste in tre diverse versioni A, B o C che definiscono i differenti voltaggi per i livelli on ed off
- La versione più utilizzata è RS-232C; i segnali vengono trasmessi in tensione: "0" (space) per $3V < \text{tensione} < 15V$ e "1" (mark) per $-3V > \text{tensione} > -15V$.

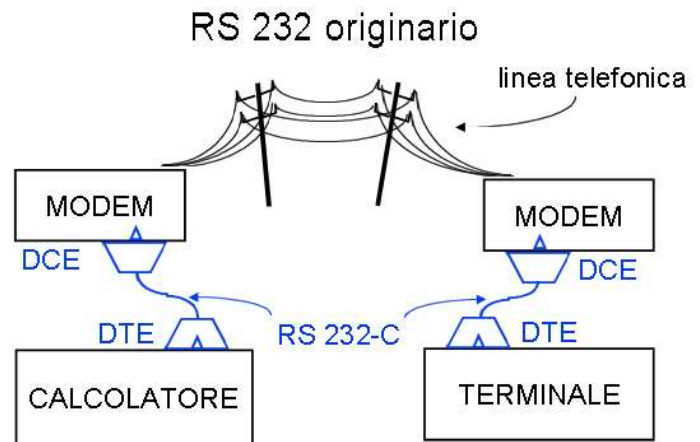


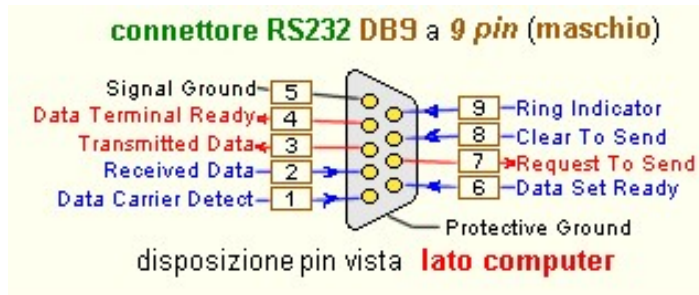
- La lunghezza massima del collegamento è 15m (meglio se < 3m in ambiente industriale) e la velocità di 115200 bps (meglio se < 19200 bps in ambiente industriale).





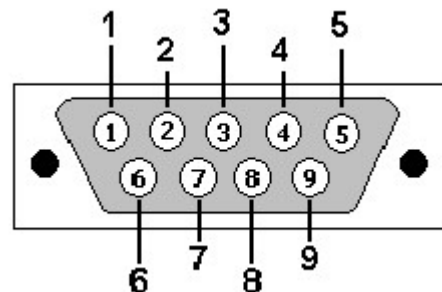
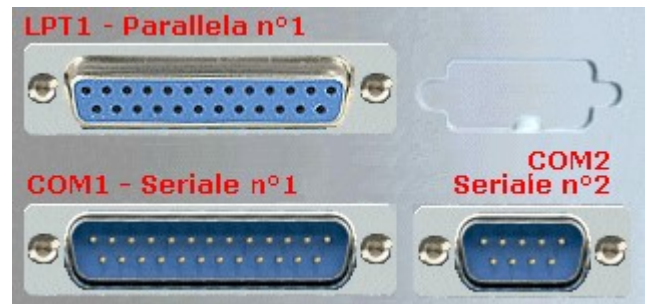
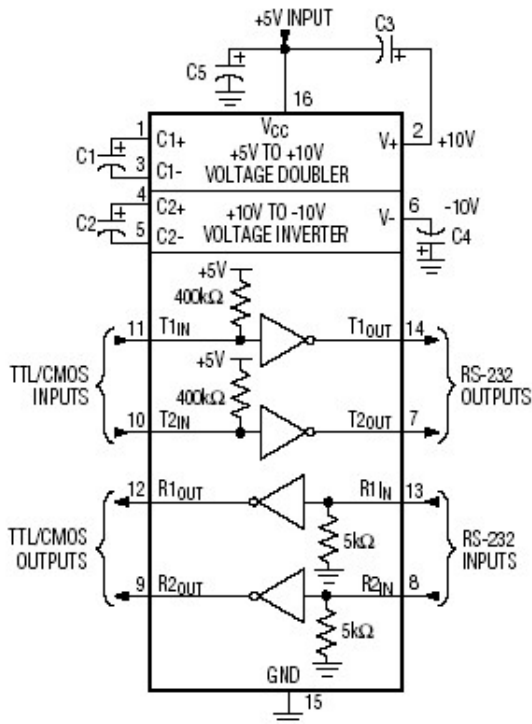
- Le porte o i dispositivi seriali sono etichettati come Data Communication Equipment (DCE) (modem) o Data Terminal Equipment (DTE) (calcolatore, stampante, terminale,...) I segnali di trasmissione e ricezione fra queste due tipologie sono invertiti di posto





DB9	DB25	DESCRIZIONE
	1	Protective Ground
3	2	Transmit Data
2	3	Received Data
7	4	Request To Send
8	5	Clear To Send
6	6	Data Set Ready
5	7	Signal Ground
1	8	Data Carrier Detect
4	20	Data Terminal Ready
9	22	Ring Indicator

- I segnali presenti sui connettori possono essere suddivisi in segnali per il trasferimento dei dati, segnali di controllo e segnali di temporizzazione per il controllo del flusso di dati, sono previsti poi specifici collegamenti di massa.
- DTR/DSR: Quando il PC è collegato per la prima volta, pone alto DTR. La periferica risponde ponendo alto DSR
- RTS/CTS: quando il PC inizia la trasmissione pone RTS alto, la periferica segnala che è pronta a iniziare la comunicazione ponendo CTS alto. Per interrompere la trasmissione la periferica pone CTS basso.
- Il connettore utilizzato è un connettore a "D" a 9/25 poli

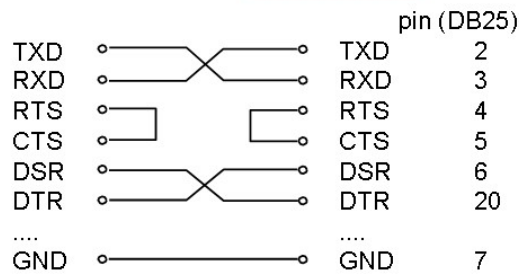


Collegamento DTE - DTE

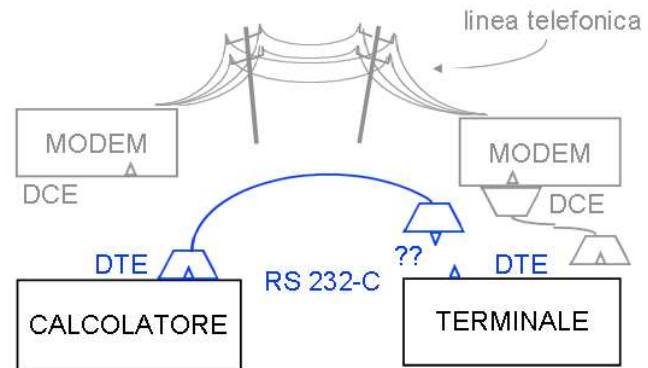
Due possibilità:

1. interporre un modulo che simula la catena Modem1 - linea - Modem2 (NULL MODEM)
2. trasformare uno degli apparati in DCE

Esempio di null-modem



Uso improprio dello RS 232



- **GND** - Logic Ground: Riferimento di tensione per tutti gli altri segnali.
- **TXD** - Transmitted 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.
- **RXD** - Received 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.
- **DCD** - Data Carrier Detect: Il segnale DCD indica che il computer o il dispositivo sono connessi o on line. DCD non è sempre usato o disponibile.
- **DSR** - Data set Ready: Su questa linea il DCE dice al DTE che il canale di comunicazione è disponibile, ma non indica che effettivamente sia stato stabilito un link con un dispositivo remoto.
- **DTR** - 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.
- **CTS** - Clear To Send: Segnale di risposta a DTE. Quando attivo, indica a DTE che può dare inizio alla trasmissione (linea TXD)
- **RTS** - Request To Send: dal DTE al DCE, disponibilità a trasmettere; quando attivo questo segnale informa il DCE che il DTE è pronto a spedire un byte.
- **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.
- **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.

Pin 1: DCD Pin 2: Rx Pin 3: Tx Pin 4:
 DTR Pin 5: GND Pin 6: DSR Pin 7:
 RTS Pin 8: CTS Pin 9: RI

DCD (Data Carrier Detect): Ingresso settato se il DCE sincronizzato.

Rx : Pin di ricezione

Tx : Pin di trasmissione

DTR (Data Terminal Ready) : comunica che il DTE è pronto a inviare un dato al DCE.

GND : Massa di riferimento per i segnali.

DSR (Data Set Ready) : segnale di risposta al DTE da parte del DCE

RTS (Request to send) : Avviso di trasmissione

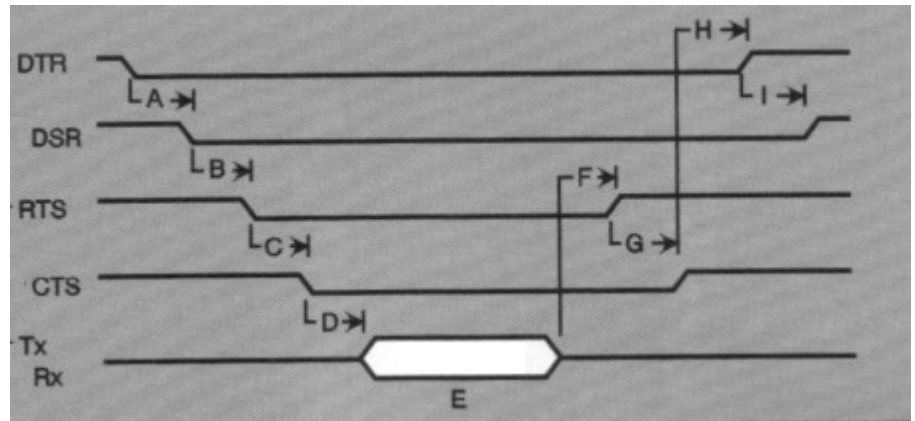
CTS (Clear to send) : Conferma per avviare la trasmissione (in risposta all RTS).

RI (Ring Indicator).

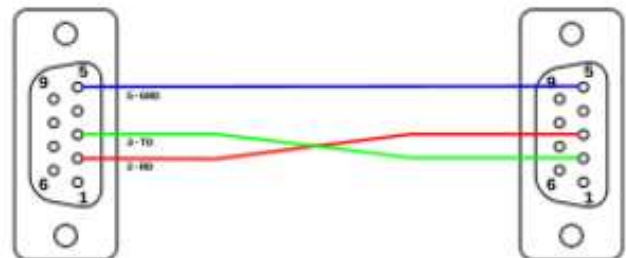
Con opportuni collegamenti è possibile creare una connessione NULL MODEM, che consente una connessione di altri dispositivi (che non siano appunto un Modem) tramite l'RS232.

Il collegamento più semplice, senza quindi alcun segnale di handshake, è semplicemente invertendo i pin Rx e Tx lato DCE.

Per ingannare i segnali di Handshake, in un null modem si collegano insieme DTR, DSR e DCD e anche RTS e CTS fra loro, facendo in modo che ogni segnale di Handshake abbia una risposta immediata essendo collegato al suo corrispettivo. Di seguito sono riportate delle configurazioni ottenibile con un'RS232:



- **Null-modem semplice asincrono:**

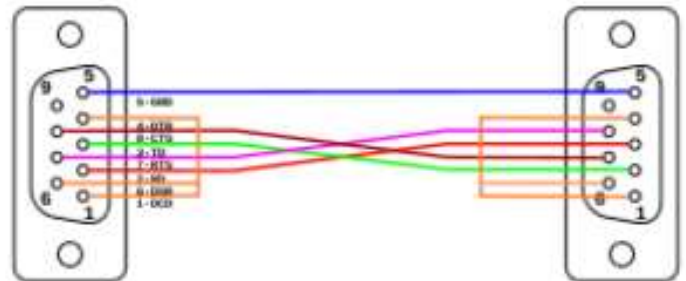
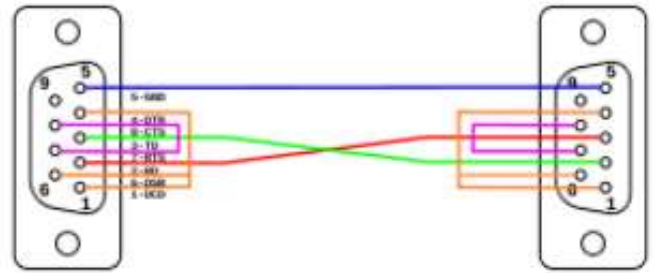


- **Null-modem a 3 fili con handshaking locale asincrono:**

Ove è necessario utilizzare una comunicazione su soli 3 fili e i DTE richiedano un controllo di flusso hardware si può adottare uno schema che gestisca un handshake locale. Un handshake locale si ottiene collegando su ciascun connettore, localmente, le linee RTS e CTS fra loro in modo che il DTE, attivando l'uscita RTS, veda il segnale CTS attivo. Allo stesso modo, collegando localmente le linee DTR, DSR e DCD, si ottiene una emulazione locale della gestione di questi segnali. In questo modo quando il DTE è pronto ad inviare i dati e attiva l'uscita DTR si vede attive le linee DSR e DCD e ritiene così che esista la portante (segnale DCD) e il DTE remoto sia operativo (DSR).

- **Null modem a 5 fili con handshaking parziale asincrono:**

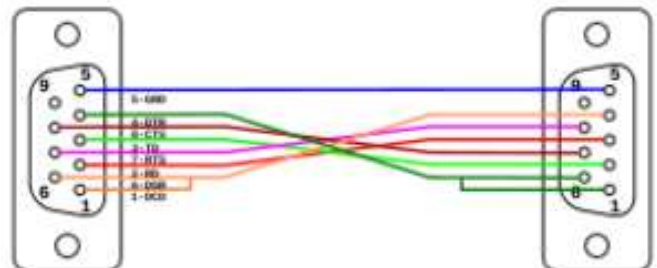
Un controllo di flusso parziale si ottiene su un cavo a 5 fili collegando la linea RTS di un DTE alla linea CTS dell'altro DTE e vice versa. In questo modo, quando il buffer di un DTE si riempie, abbassando la linea RTS si comunica all'altro DTE di sospendere la trasmissione fino a quando l'RTS non sarà nuovamente impostato su alto



- **Null modem a 7 fili con handshaking completo asincrono:**

Il controllo di flusso durante la trasmissione avviene utilizzando le linee RTS e CTS come nello schema precedente, ma un DTE può in questo modo comunicare all'altro DTE quando ci sono dei dati da trasferire alzando il segnale DTR. Il segnale DTR su un lato deve essere collegato sia al DSR del DTE remoto sia al DCD dello stesso DTE remoto. Il collegamento DTR/DSR fa sì che un DTE comunichi all'altro quando ha dei dati da trasmettere mentre il collegamento DTR/DCD emula il segnale di presenza della portante in modo che il DTE ritenga che il modem remoto (emulato dal null modem) sia in linea.

Con un handshaking completo si possono ottenere anche velocità di trasmissioni più alte.



Controllo di flusso software

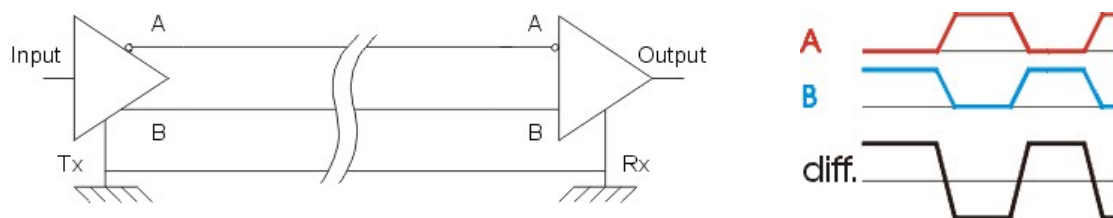
Per la trasmissione asincrona su RS232 è possibile non fare uso dei segnali di controllo per il controllo di flusso.

- A tale scopo vengono trasmessi byte particolari di start e stop. Questi sono definiti dal codice ASCII (American Standard Code for Information Exchange) come XON (binario:10001) e XOFF (10011). Si racchiude il messaggio trasmesso tra i due caratteri di controllo XON e XOFF
- Questo metodo è utile quando si trasferiscono dati in formato testo (codice ASCII), meno quando si trasferiscono altri formati.
- Il controllo di flusso software è più lento di quello hardware (handshaking).

PIN (9)	PIN (25)	NOME V.24 ITU	NOME RS-232	DESCRIZIONE
	1	C101	FG	Frame ground = Massa di protezione
3	2	C103	TxD	Trasmitted data = Dati in trasmissione
2	3	C104	RxD	Received data = Dati in ricezione
7	4	C105	RTS	Request to send = Richiesta di trasmissione
8	5	C106	CTS	Clear to send = Pronto a trasmettere
6	6	C107	DSR	Data set ready = DCE pronto
5	7	C102	GND	Ground = Massa dei segnali
1	8	C109	DCD	Data carrier detector = Portante in ricezione presente
	9			Riservato per apparecchi di collaudo
	10			Riservato per apparecchi di collaudo
	11	C126	CK	Scelta frequenza in trasmissione
	12	C122	SCF	Segnale di ricezione presente sul canale ausiliario
	13	C121	SCB	Pronto per la trasmissione sul canale ausiliario
	14	C118	SBA	Dati in trasmissione del canale ausiliario
	15	C114	TC	Transmit clock = Clock di trasmissione dal modem
	16	C119	SBB	Dati in ricezione del canale ausiliario
	17	C115	RC	Received clock = Clock di ricezione
	18			Non connesso
	19	C120	SCA	Richiesta di trasmissione del canale ausiliario
4	20	C108	DTR	Data terminal ready = DTE pronto
	21	C110	CG	Rivelatore della qualità del segnale
9	22	C125	RI	Ring indicator = Chiamata in arrivo
	23	C111	CI	Selezione velocità di trasmissione da DTE
	24	C113	DA	Clock di trasmissione da DTE
	25			Non connesso

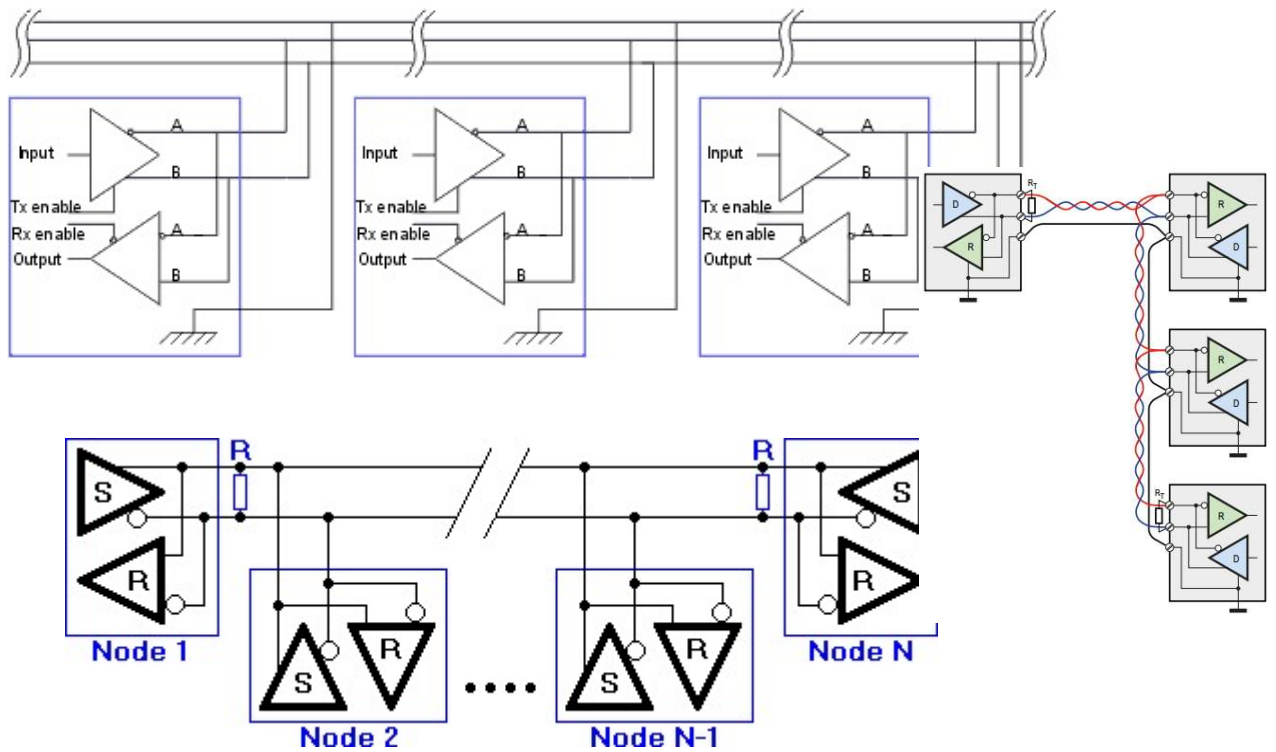
Interfaccia RS-422

- E' un protocollo di livello fisico molto utilizzato in ambiente industriale per collegamenti punto-punto fra apparecchiature. I ricevitori possono essere fino a 10.
- Ha segnali di TX e RX e alcuni segnali di hand-shake (anche se questi ultimi molto spesso non sono presenti). Usa una trasmissione bilanciata (differenziale), quindi molto immune ai disturbi di modo comune. In genere l'interfaccia ha 4 fili: RX+, RX-, TX+ e TX-.
- I segnali vengono trasmessi in tensione (tipicamente da 1 a 4 V): "0" per $RX+ > RX-$ e "1" per $RX+ < RX-$.
- La lunghezza massima del collegamento e la velocità di trasmissione sono legate fra loro: è di 1000mt per velocità $< 100\text{Kbps}$ fino a raggiungere i 20m per una velocità massima di 10 Mbps.
- Occorre prestare attenzione alle differenze di potenziale fra i dispositivi connessi: occorre usare un unico riferimento equipotenziale di massa oppure optoisolare l'interfaccia.

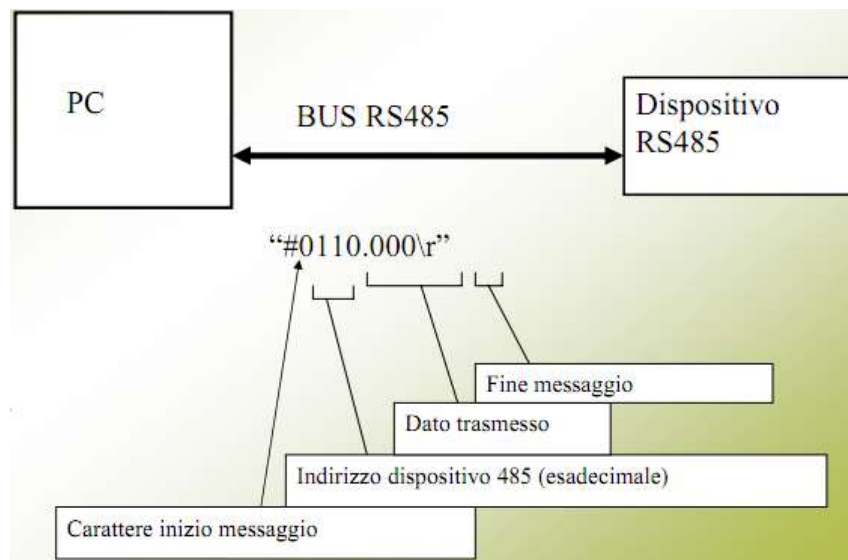


Interfaccia RS-485

- E' un protocollo di livello fisico molto utilizzato in ambiente industriale per collegamenti multipoint su bus di apparecchiature. Il bus è costituito da 2 conduttori, denominati D+ e D-, usati sia per la TX che per la RX per mezzo di trasmissione differenziale.
- Possono essere collegati sullo stesso bus fino a 32 ricevitori e fino a 32 trasmettitori con circuiti di protezione in caso di conflitto hardware.
- Un dispositivo connesso al bus può essere in ricezione (default) o trasmissione.
- I segnali vengono trasmessi in tensione (tipicamente da 1 a 4 V): "0" per $D+ > D-$ e "1" per $D+ < D-$.
- La lunghezza massima del collegamento e la velocità di trasmissione sono legate fra loro: è di 1000mt per velocità $< 100\text{Kbps}$ fino a raggiungere i 20m per una velocità massima di 10 Mbps.
- Occorre prestare attenzione alle differenze di potenziale fra i dispositivi connessi: occorre usare un unico riferimento equipotenziale di massa oppure optoisolare l'interfaccia.
- La differenza sostanziale è il supporto delle linee multi-drop, cioè linee con la coesistenza di più ricevitori e trasmettitori sulla stessa coppia di fili. Al fine di evitare conflitti è ovviamente necessario che un solo trasmettitore alla volta sia attivo. Questo implica l'uso di trasmettitori che, oltre alle uscite corrispondenti allo zero e all'uno, possano gestire anche un "terzo stato" in cui l'elettronica appare come fisicamente non collegata alla linea (stato detto ad alta impedenza, three-state o Hi-Z).

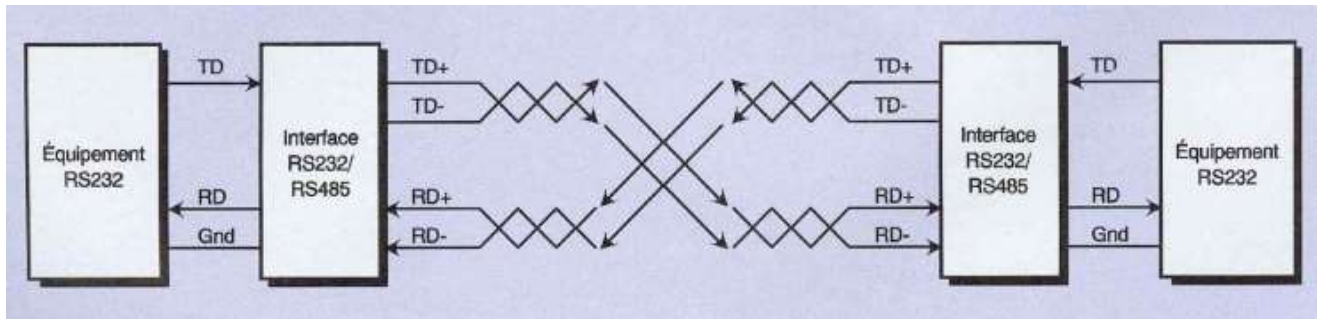


Lo standard specifica solo le caratteristiche meccaniche ed elettriche dei dispositivi RS485, ma non dice nulla riguardo al protocollo da utilizzare per la comunicazione. I costruttori di dispositivi 485 hanno implementato diversi protocolli. Attualmente il più utilizzato è il protocollo ASCII così realizzato



Characteristics of RS232, RS422, RS423 and RS485

	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers	1	1	1	32
Max number of receivers	1	10	10	32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m	20 kbs	100 kbs	10 Mbs	35 Mbs
Max speed at 1200 m	(1 kbs)	1 kbs	100 kbs	100 kbs
Max slew rate	30 V/ μ s	adjustable	n/a	n/a
Receiver input resistance	3..7 k Ω	\geq 4 k Ω	\geq 4 k Ω	\geq 12 k Ω
Driver load impedance	3..7 k Ω	\geq 450 Ω	100 Ω	54 Ω
Receiver input sensitivity	\pm 3 V	\pm 200 mV	\pm 200 mV	\pm 200 mV
Receiver input range	\pm 15 V	\pm 12 V	\pm 10 V	-7..12 V
Max driver output voltage	\pm 25 V	\pm 6 V	\pm 6 V	-7..12 V
Min driver output voltage (with load)	\pm 5 V	\pm 3.6 V	\pm 2.0 V	\pm 1.5 V



possibile full duplex su livello fisico RS485

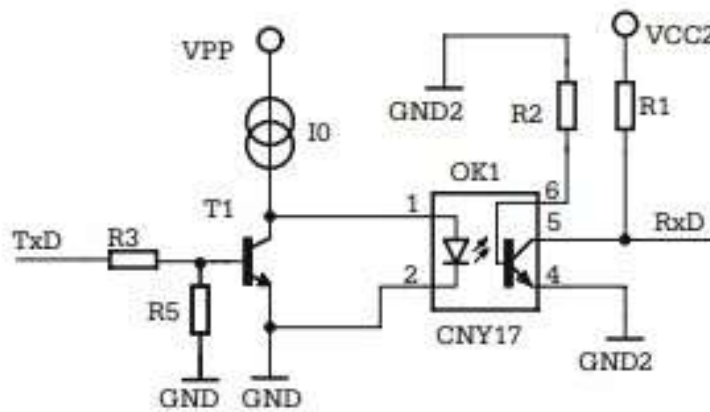
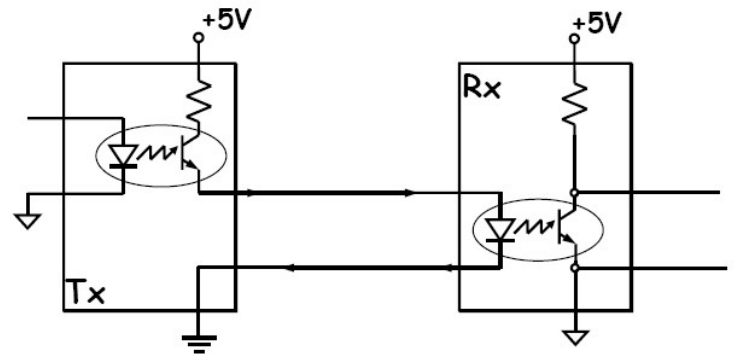
Interface	Format	Number of Devices (maximum)	Length (maximum, feet)	Speed (maximum, bits/sec.)
RS-232 (EIA/TIA-232)	asynchronous serial	2	50-100	20k (115k with some drivers)
RS-485 (TIA/EIA-485)	asynchronous serial	32 unit loads	4000	10M
IrDA	asynchronous serial infrared	2	6	115k
Microwire	synchronous serial	8	10	2M
SPI	synchronous serial	8	10	2.1M
I ² C	synchronous serial	40	18	400k
USB	asynchronous serial	127	16	12M
Firewire	serial	64	15	400M
IEEE-488 (GPIB)	parallel	15	60	1M
Ethernet	serial	1024	1600	10M
MIDI	serial current loop	2	15	31.5k
Parallel Printer Port	parallel	2, or 8 with daisy-chain support	10-30	1M

	Wi-Fi 802.11 Family	Bluetooth 802.15.1	Wireless USB	ZigBee and 802.15.4
Applications	Enterprise, networking (Internet)	PC peripherals, cable replacement	PC peripherals, cable replacement, multimedia	Sensors, home/building automation, toys
Range	50 m	10 to 100 m	3 to 10 m	50 to 100 m
Data Rate	54 Mb/s (540 Mb/s)	750 kb/s	110 to 480 Mb/s	250 kb/s
Nodes per Network	>1,000	7	127	65,000
Battery Life	Hours	Days	Hours/days	Years
Setup/Usability	Better	Good	Better	Good → Best
Frequency	2.4 GHz, 5 GHz	2.4 GHz	3 to 10 GHz	900 MHz, 2.4 GHz
Security	Best	Good	Best/good	Better

Standard Current Loop 20 mA

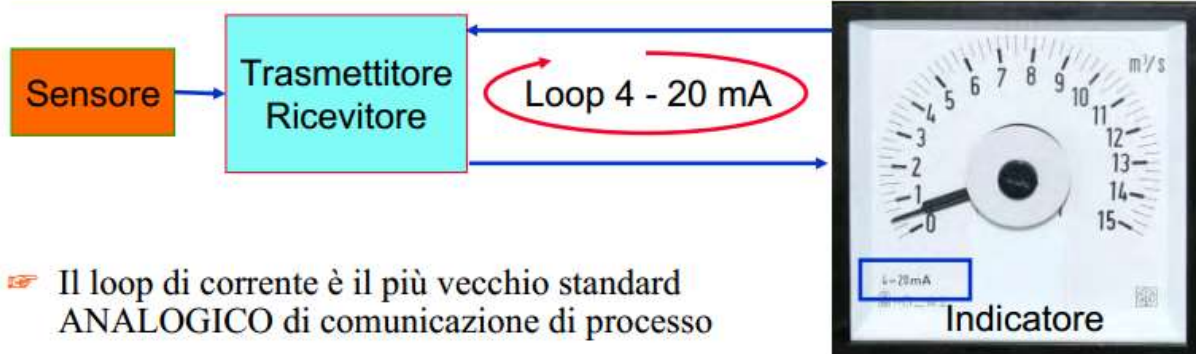
Il collegamento in loop di corrente associa il passaggio di una corrente (di 20 mA) al livello logico 1, l'assenza di corrente allo 0. Si tratta di un metodo di trasmissione molto usato a livello industriale in quanto poco sensibile al rumore elettromagnetico ambientale (massima immunità).

Per la sua realizzazione si impiegano spesso driver opto-isolati, che permettono quindi di mantenere galvanicamente separati il trasmettitore ed il ricevitore, fino a differenze di potenziale di migliaia di Volt.



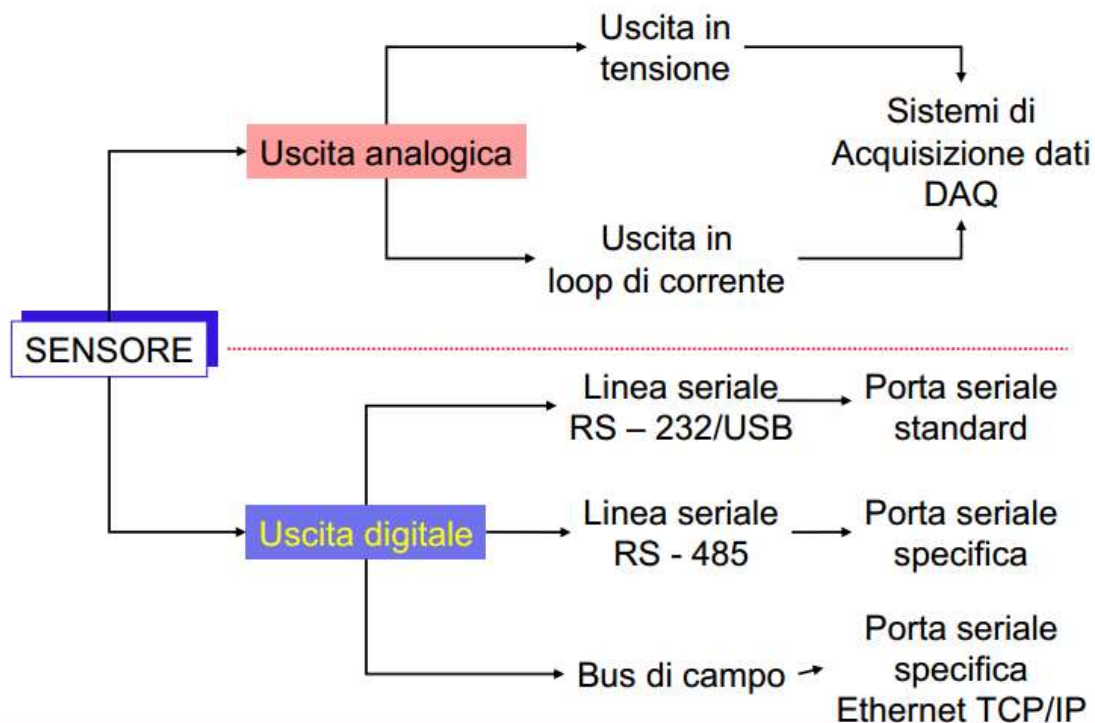
Current loop transmission circuit

Loop di corrente 4 - 20 mA



- Il loop di corrente è il più vecchio standard ANALOGICO di comunicazione di processo
- La trasmissione in corrente è preferita a quella in tensione per:
 - la bassa impedenza di linea \Rightarrow minori disturbi
 - la possibilità di inserire altri dispositivi aprendo il loop
- Il loop è generalmente 4 ÷ 20 mA dove
 - 4 mA \rightarrow 0 logico o zero analogico
 - 20 mA \rightarrow 1 logico o fondo scala analogico
 - 0 = mancanza di corrente viene interpretata come interruzione di linea
- Questo approccio è applicabile solamente a collegamenti PUNTO - PUNTO. Perciò non è idoneo per connettere molti sensori ad uno stesso sistema di controllo.

Acquisizione e trasmissione dati



Sistemi di trasmissione dati

☞ Loop di corrente 4 - 20 mA

- ⇒ Sistema analogico molto antiquato
- ⇒ solo connessione punto - punto

☞ Linea seriale RS - 232

- ⇒ La più usata
- ⇒ stessi limiti del loop di corrente: punto - punto

☞ Linea seriale RS - 485

- ⇒ Molto più potente della RS - 232
- ⇒ Alta velocità di trasmissione
- ⇒ Connessioni multi-drop (una linea - molti sensori)

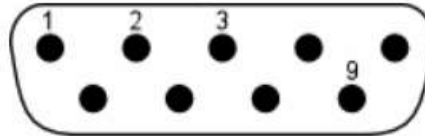
☞ Bus di campo

- ⇒ Estensione della comunicazione ad interi sistemi
 - ⇒ Multi - punto
 - ⇒ Protocolli complessi
 - ⇒ Scalabilità e Interoperabilità
-

Standard RS - 232 e suoi limiti

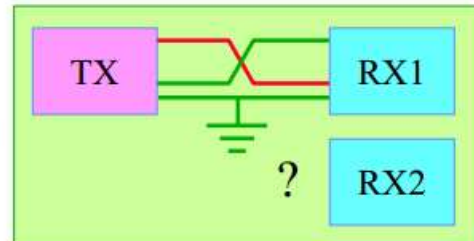
- Lo standard RS - 232 è molto diffuso perché ogni PC dispone di una porta seriale (*connettore maschio a 9 poli*).

1 = 9 = GND
2 = RX
3 = TX



- Esso è **limitato a comunicazioni punto - punto** a distanza ravvicinata con inversione dei terminali 2 e 3 (*null modem*), a meno che non si impieghi un MODEM ed una linea telefonica o un collegamento a radiofrequenza.

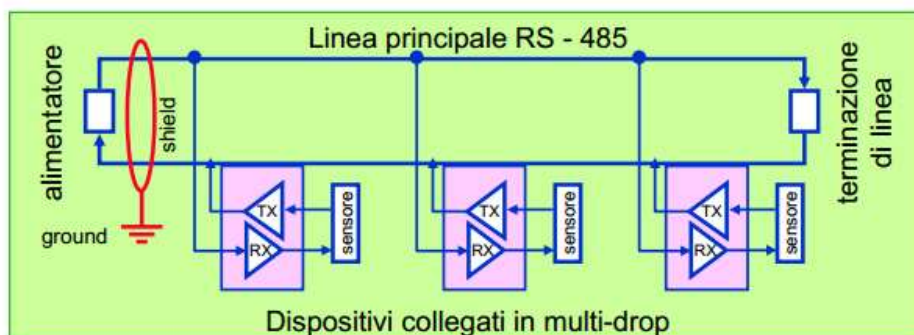
- Modo di operazione: sbilanciato (riferito a massa)
- Numero trasmettitori: 1
- Numero ricevitori: 1
- Max. lunghezza del cavo: ~ 20 m
- Velocità massima di trasmissione: ~ 56 Kbaud



Standard RS - 485

- Il protocollo RS - 485 è attualmente lo standard seriale più avanzato, consentendo comunicazioni verso molti punti di misura su grandi distanze
- Differenza con RS-232: Ricevitore e trasmettitore separati e indipendenti
- Consente la connessione di più dispositivi in parallelo sulla stessa linea (*multi drop*).

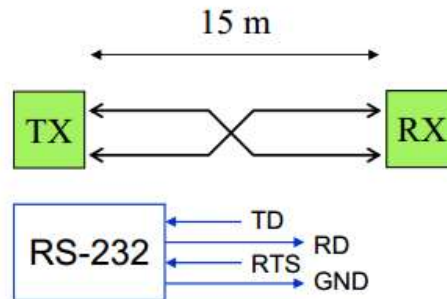
- Modo di operazione: bilanciato (non riferito a massa)
- Numero trasmettitori: 32
- Numero ricevitori: 32
- Max. lunghezza del cavo: oltre 5 km
- Velocità massima di trasmissione: 10 MBaud



Differenze fra i protocolli RS-232 e RS-485

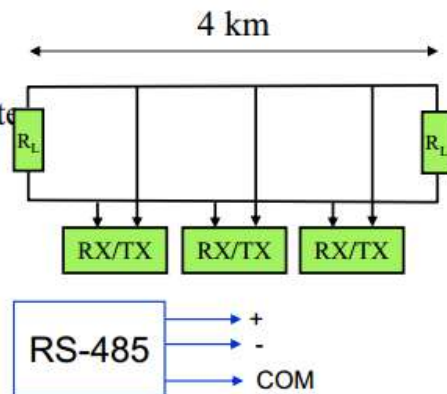
RS-232:

- ⇒ Collegamento punto - punto
- ⇒ distanza limitata (15 metri max.)
- ⇒ banda limitata (115.2 kbaud max.)



RS-485:

- ⇒ Collegamento multi-point mediante multi-drop
- ⇒ distanza notevole (4 km ed oltre)
- ⇒ larga banda (10 Mbaud ed oltre)





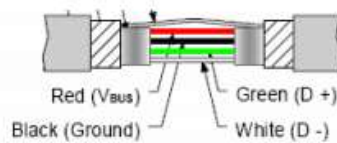
USB: Universal Serial Bus

- ☞ Erede attuale del vecchio standard seriale RS – 232
 - ⇒ Impressionante aumento di velocità: fino a 12 Mb/s (USB2)
- ☞ Il sistema USB è asimmetrico, consiste in un singolo gestore e molte periferiche collegate da una struttura simile ad un albero attraverso dei dispositivi chiamati *hub* (concentratori).
- ☞ La lunghezza massima di cavo è di 5m; oltre questo limite è necessario ricorrere ad uno o più *hub* attivi che amplifichino il segnale.
- ☞ Si possono alimentare direttamente le periferiche a basso consumo.
 - ⇒ Le periferiche che hanno richieste energetiche elevate vanno alimentate a parte.



■ Pinout del connettore USB:

Pin	Nome segnale	Colore filo
1	VBUS	ROSSO
2	D-	BIANCO
3	D+	VERDE
4	GND	NERO
Shell	SHIELD	-



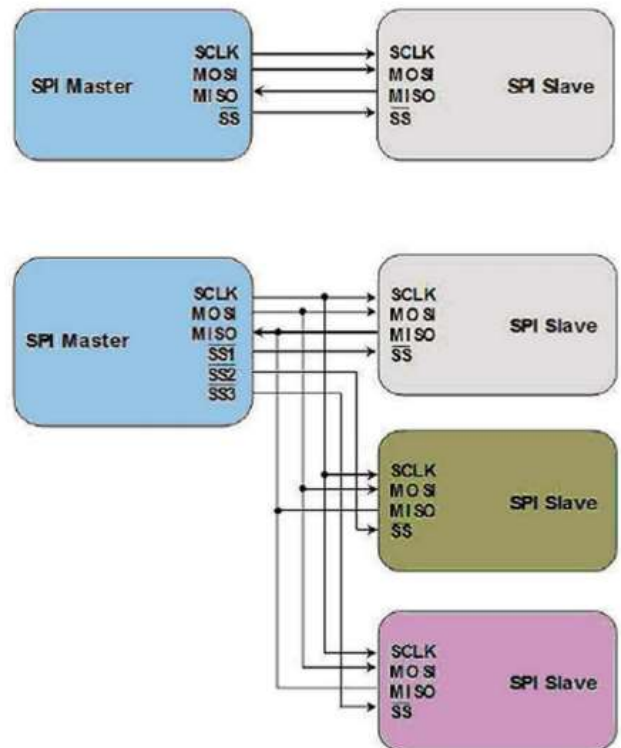
SPI

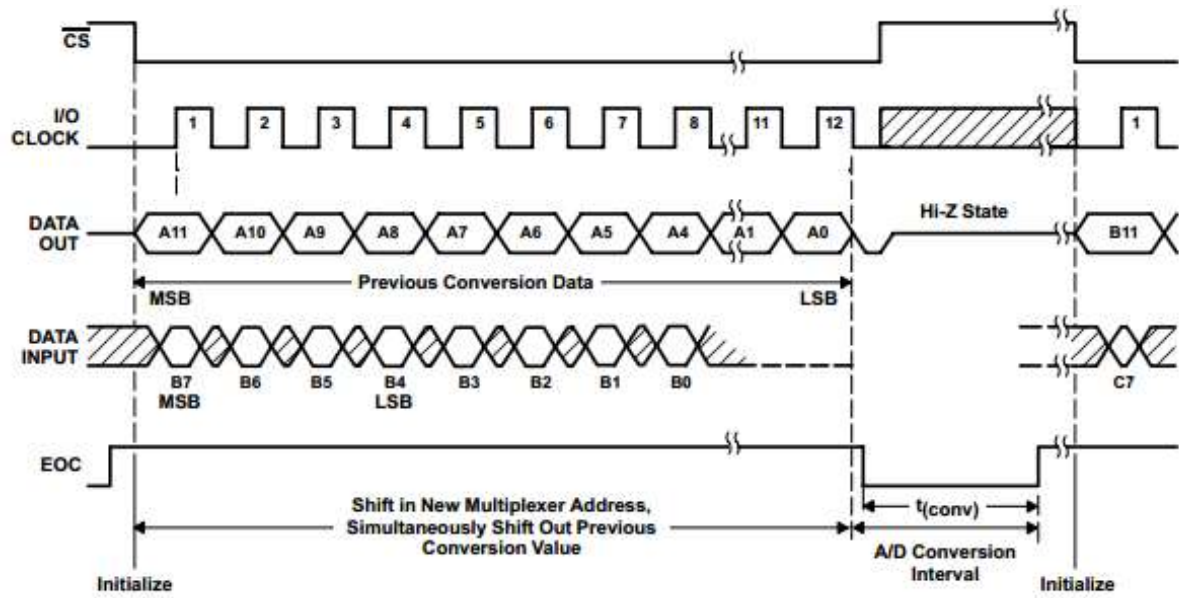
L'interfaccia SPI (Serial Peripheral Interface) è semplice da utilizzare ed affidabile e, per queste ragioni, è divenuta il sistema di comunicazione comunemente utilizzato da memorie, ADC e DAC esterni al microcontrollore.

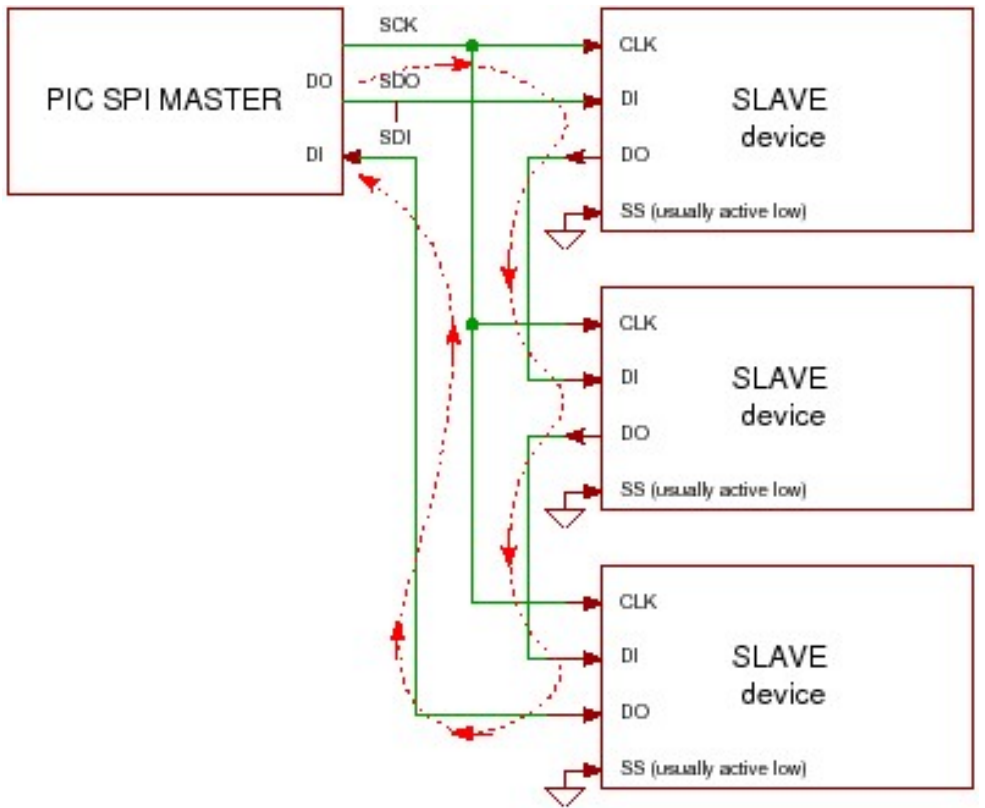
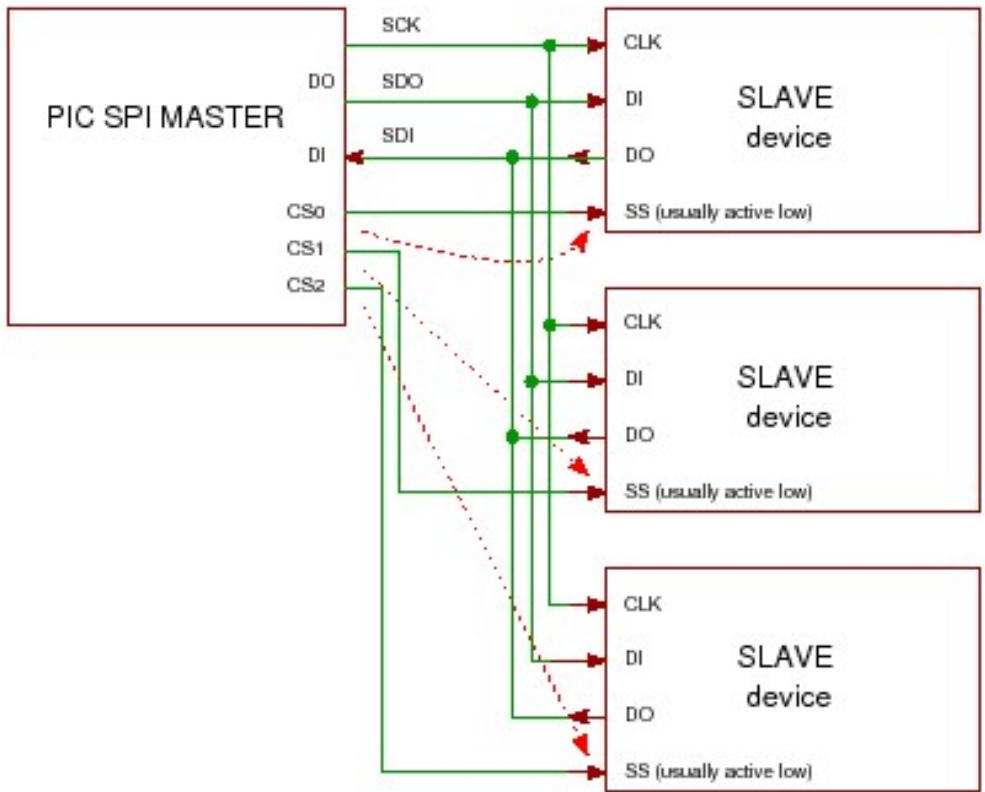
La comunicazione mediante porta SPI necessita di tre collegamenti:

- SDO/MOSI (Serial Data Output) dal quale i dati vengono inviati;
- SDI/MISO (Serial Data Input) dal quale i dati vengono ricevuti;
- SCK/SCLK (Serial Data Clock) su cui viene trasmesso il segnale di clock.

I dati trasmessi e ricevuti passano su due connessioni differenti e, per questa ragione, la comunicazione è detta di tipo *full-duplex*. Più periferiche possono essere collegate allo stesso bus di comunicazione, ma, in questo caso, solo una di esse è di tipo "master", mentre tutte le altre sono di tipo "slave".







I2C

Richiede unicamente due linee di comunicazione:

- SDA (Serial Data Line) per la trasmissione/ricezione dei dati ;
- SCL (Serial Clock Line) per la trasmissione del clock di sincronizzazione.

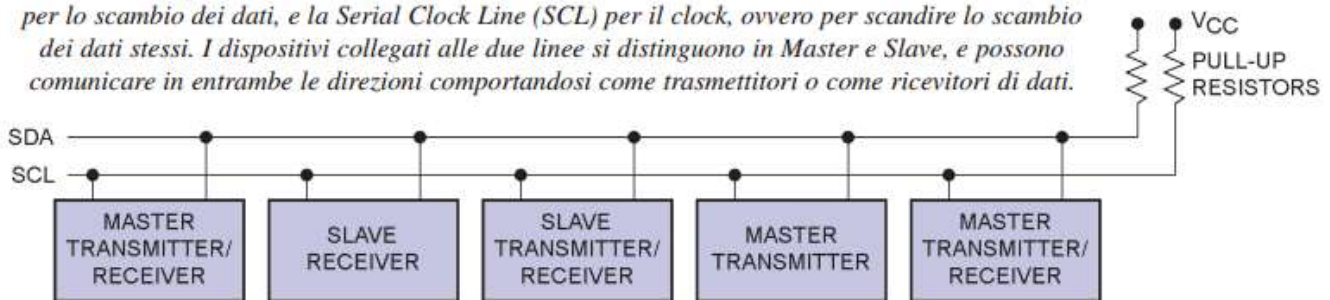
Sullo stesso bus possono essere collegati fino a 112 dispositivi differenti.

A differenza dell' SPI, sia i dati in trasmissione che quelli in ricezione viaggiano sulla stessa connessione fisica.

Il bus è di tipo half-duplex perché solo un componente alla volta può trasmettere i dati; inoltre, la comunicazione può essere iniziata solo dal master, mediante una stringa che contiene un bit di start, l'indirizzo dello slave con il quale vuole comunicare e un bit che indica se è richiesta una risposta.

Successivamente vengono inviati, in sequenza, i dati veri e propri, iniziando sempre dal bit più significativo.

L'I²C-Bus è un protocollo di comunicazione che utilizza solo due linee: la Serial Data Line (SDA) per lo scambio dei dati, e la Serial Clock Line (SCL) per il clock, ovvero per scandire lo scambio dei dati stessi. I dispositivi collegati alle due linee si distinguono in Master e Slave, e possono comunicare in entrambe le direzioni comportandosi come trasmettitori o come ricevitori di dati.



I segnali che transitano sulla linea hanno valore '1' o '0' e le tensioni che li rappresentano sono quelle d'alimentazione e di massa rispettivamente.

Le due linee non sono lasciate ad un valore indefinito, ma vengono collegate all'alimentazione attraverso una resistenza di pull-up. In questo modo le due linee permangono ad un valore "alto-debole", facilmente modificabile da un dispositivo.

Grazie all'utilizzo delle resistenze di pull-up, per ottenere un valore '1' sulla linea sarà sufficiente mettere il segnale d'uscita in alta impedenza. In questo modo se un altro dispositivo impone un valore '1' ed un altro un valore '0', quest'ultimo segnale sarà prevalente sul precedente. Questo consente di adoperare un meccanismo d'arbitraggio, che può risolvere problematiche di collegamento.

Ogni dispositivo collegato a queste linee è dotato di un indirizzo univoco, di 7 o 10 bit e può agire sia da master che da slave, secondo le funzioni previste al suo interno.

Il master si occupa di iniziare la trasmissione e di generare la tempistica del trasferimento, mentre lo slave è quello che riceve una richiesta.

Entrambe le due categorie appena descritte possono assumere il ruolo di trasmittente o ricevente.

Le modalità di trasferimento dati possono essere riassunte come segue:

I) A trasmette dati a B

1. A(master) spedisce l'indirizzo di B(slave) sul bus
2. A (master-transmitter) trasmette i dati a B (slave-receiver)
3. A termina il trasferimento

II) A vuole ricevere dati da B

1. A (master) spedisce l'indirizzo di B(slave) sul bus
2. B (slave-transmitter) spedisce i dati ad A (master-receiver)
3. A termina il trasferimento

La trasmissione inizia con la generazione di un segnale di start, immesso sulla linea dal master, dopo un controllo sull'occupazione del bus.

La condizione di start consiste nel lasciare la linea "scl" allo stato "alto", mentre la linea "sda", subisce una transizione dallo stato '1' allo stato '0'.

Dopo la generazione del segnale di start inizia la trasmissione dei dati vera e propria.

Il primo byte trasmesso è quello composto dall'indirizzo dello slave con l'aggiunta di un ulteriore valore che indica al ricevente quale è l'operazione a lui richiesta. In base allo standard previsto dal protocollo questo bit può assumere due valori con i seguenti significati:

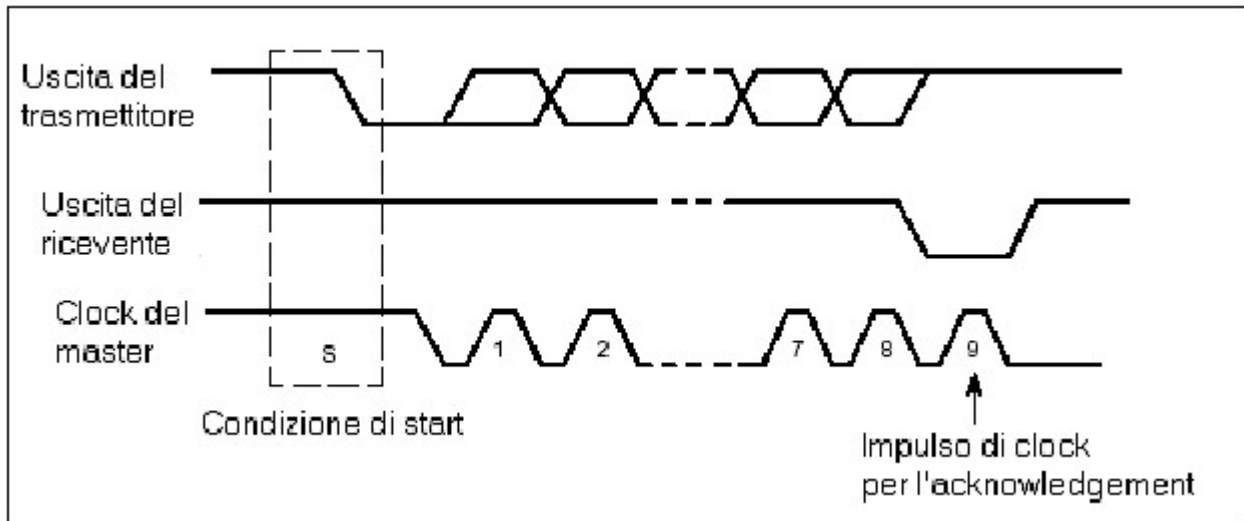
- '0' : ricezione dati;
- '1' : trasmissione dati;

L'ordine di trasmissione dei bit è quello dal più significativo al meno significativo.

Le successive comunicazioni seguono sempre quest'ultimo criterio.

Dopo la trasmissione di ogni byte chi trasmette ha l'obbligo di lasciare la linea "sda" allo stato "alto", in modo da permettere a chi riceve dei dati, di darne conferma tramite il meccanismo dell'acknowledgement: esso consiste nell'abbassare la linea "sda" in corrispondenza del nono impulso presente sulla linea "scl".

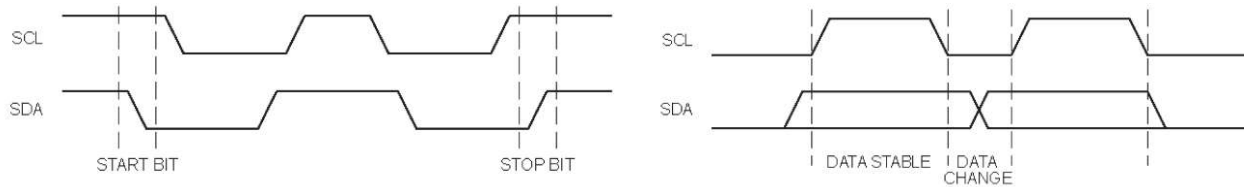
Un esempio del processo appena esposto lo si può vedere nella Fig.



Se l'ack non venisse generato, chi ha iniziato la trasmissione può interrompere la comunicazione, utilizzando la condizione di stop.

Un dispositivo può non generare il segnale di acknowledgement ad esempio, quando è inabilitato a ricevere, perché sta eseguendo delle funzioni in real-time, oppure quando non può più immagazzinare altri dati.

Il segnale di stop è sempre generato dal master, come pure quello dello start. Esso consiste nel far variare la linea sda dallo stato "basso" a quello "alto" in corrispondenza del periodo "alto" della linea "scl".



Il trasferimento dei dati da un dispositivo Master a uno Slave, viene preceduto e terminato da due condizioni particolari: il bit di start e il bit di stop, generati tramite la combinazione dei segnali SCL e SDA. Durante il trasferimento dei dati, ogni singolo bit viene accompagnato da un impulso positivo di clock che per tutta la durata del dato deve rimanere a livello costante.

Durante la trasmissione avvengono contemporaneamente due processi:

1. la sincronizzazione dei clock
2. l'arbitraggio

Il primo processo permette a due dispositivi con velocità di funzionamento differenti di comunicare senza incorrere nella perdita di dati.

L'arbitraggio consente di utilizzare il bus come un multi-master, nel senso che è possibile collegare fra loro più elementi con la facoltà di iniziare trasferimenti di dati, senza che avvengano perdite d'informazione.

Il processo consiste nel paragonare ciò che si trasmette con quello che effettivamente si trova sulla linea "sda". Quando due dispositivi trasmettono due valori differenti, quello prevalente risulta essere il valore "basso".

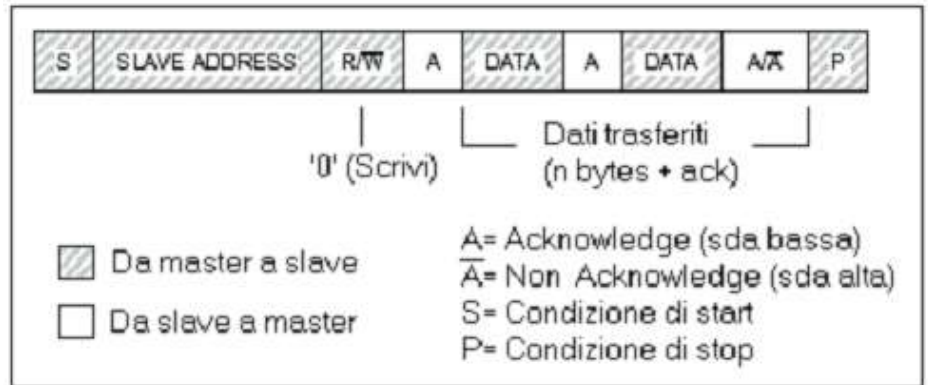
Nel caso in cui un master trasmetta un livello '0' e un altro trasmetta un livello '1', quest'ultimo dovrà disabilitare il suo stato d'uscita, poiché sulla linea vedrà un valore diverso da quello che gli voleva trasmettere.

Nel caso in cui il dispositivo abbia capacità sia di master che di slave, è possibile che l'altro elemento lo stia contattando, quindi dovrà passare all'istante, dallo stato master allo stato slave.

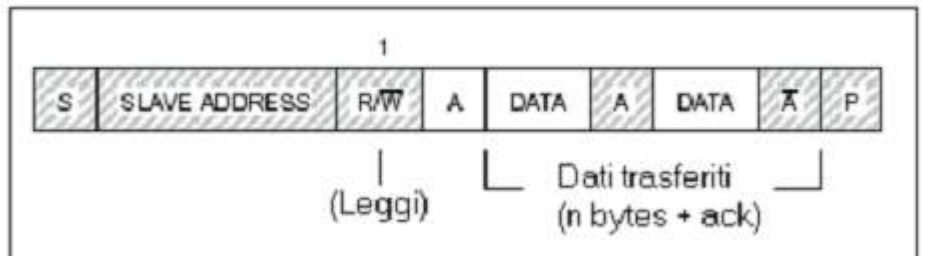
La procedura d'arbitraggio permette a due elementi di iniziare entrambi la trasmissione e di continuarla fino a che il processo va a buon fine. In questo modo non si perdono dati.

Il formato delle trasmissioni potrà assumere quindi i raggruppamenti di bytes presentati nelle figure seguenti. In particolare esse rappresentano i tipi di comunicazione :

- Master-transmitter Slave-receiver



- Master-receiver Slave-transmitter



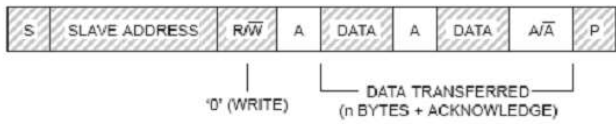
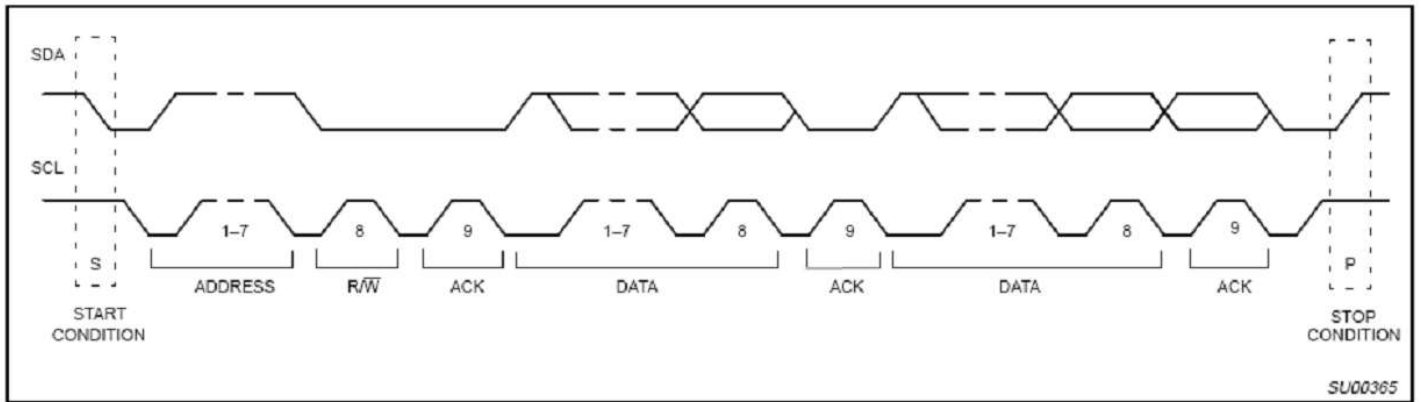
- modalità combinata



PCF8574

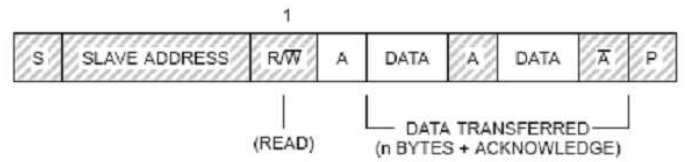
PCF8591

MCP23017



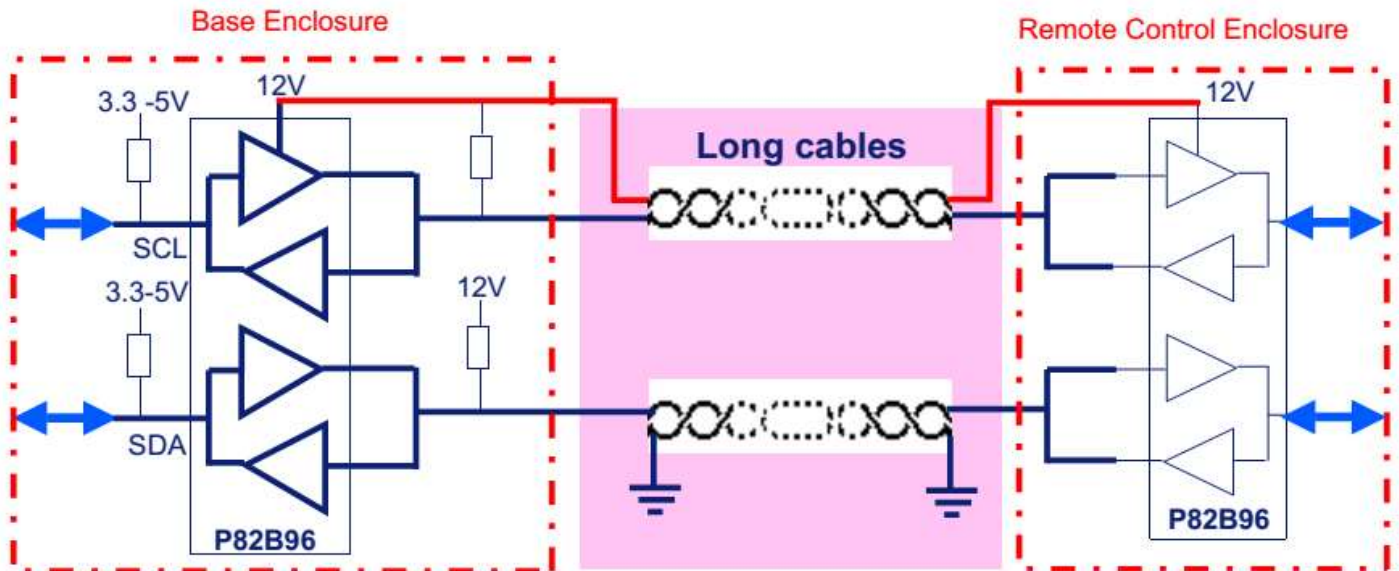
FROM MASTER TO SLAVE
 FROM SLAVE TO MASTER
 A = ACKNOWLEDGE (SDA LOW)
 \bar{A} = NOT ACKNOWLEDGE (SDA HIGH)
 S = START CONDITION
 P = STOP CONDITION

Trasferimento M→S



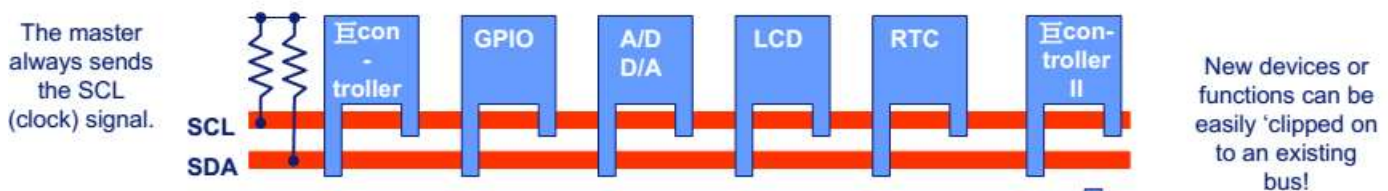
Trasferimento S→M

Driving I²C bus signals long distances



- Normal I²C logic levels (3.3 or 5 V)
- I²C currents (3mA)
- Conventional CMOS logic levels (2-15V)
- Higher current option, up to 30mA static sink
- Normal I²C logic levels (3.3 or 5 V)
- I²C currents (3mA)

I²C Bus Basics - Address and Data



Each device is addressed individually by software with a unique address that can be modified by hardware pins.

The open drain/collector outputs provide for a "wired-AND" connection that allows devices to be added or removed without impact and always require a pull-up resistor.



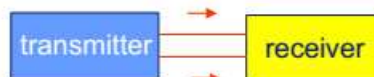
Write data



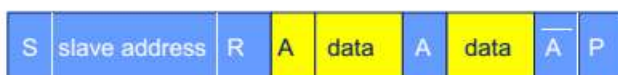
< n data bytes >

Master

Slave



Read data



< n data bytes > last data byte

S = Start condition
A = Acknowledge
P = Stop condition

R/\bar{W} = read / write not
 \bar{A} = Not Acknowledge

OneWire

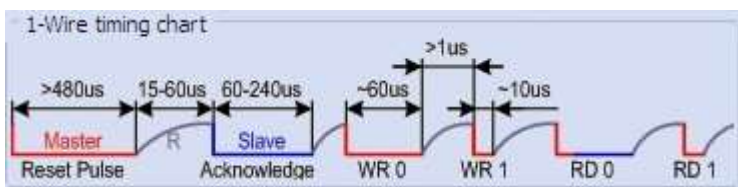
Continuando nel nostro percorso di descrizione di bus di comunicazione, passiamo al bus con il minor numero di connessioni possibili: una connessione unica per la trasmissione/ricezione di dati e segnale di sincronizzazione: il bus OneWire.

Il bus OneWire è stato realizzato da Dallas Semiconductor per consentire comunicazioni tra due o più dispositivi attraverso un unico filo di interconnessione. Non essendoci un filo dedicato per la trasmissione del segnale di sincronizzazione, quest'ultimo deve essere trasmesso assieme ai dati.

Per evitare problemi di cortocircuiti sulla linea è presente un pull-up, cioè una resistenza collegata tra la linea e l'alimentazione, e tutti i dispositivi possono imporre uno zero sulla linea attivando il transistor/MOSFET di pilotaggio della porta connessa. La velocità di comunicazione non è elevatissima (è prefissata a 16 kbps), tuttavia permette comunicazioni tra dispositivi in modo davvero semplice. Si tratta di un bus di tipo Master/Slave, dove ogni dispositivo ha un indirizzo univoco e questo deve essere utilizzato dal master per iniziare la comunicazione con lo slave.

La codifica dei simboli differisce notevolmente dai casi esaminati in precedenza, perché sono codificati in base alla lunghezza dell'impulso, come indicato in tabella

Reset	Linea a 0 per almeno $480\mu s$
Valore logico 1	1 - $15\mu s$ linea 0
Valore logico 0	$60\mu s$ linea a 0



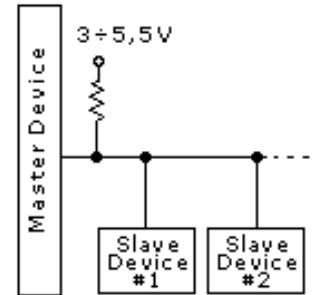
Dopo il reset la comunicazione su rete 1-Wire avviene in porzioni di tempo di lunghezza prestabilita, tra 60 e $120\mu s$, chiamati "time slots". Per esempio, per scrivere un uno logico sul bus, un dispositivo deve porre a 0 il bus, rilasciarlo dopo circa 10-15 μs e aspettare la fine dello slot, cioè almeno 45 μs . Viceversa per scrivere uno zero logico sul bus il dispositivo deve porre a 0 il bus, rilasciarlo dopo circa 60 μs e aspettare la fine dello slot.

Nel caso di ricezione, cioè durante la comunicazione di dati dallo slave verso il master, lo slave mantiene la linea al valore 0 per 60 μs qualora voglia inviare un valore logico zero mentre non fa nulla se intende inviare il valore logico uno a seguito di ogni impulso generato dal

master.

Per accertarsi che il pacchetto di dati sia stato correttamente ricevuto/inviato, è presente il controllo mediante CRC a 8.

La rete OneWire è dunque un bus a basso costo composto da un'architettura multi-punto, in cui il master (un PC o un uC) ha una configurazione di uscita a open-drain o tri-state, con un resistore di pull-up a Vcc, mentre gli slaves hanno un'uscita open-drain con cui possono portare a 0 il segnale sul bus. Il sistema permette un rigido controllo della comunicazione, perché nessun nodo è autorizzato a parlare senza esplicita richiesta del master, e nessuna comunicazione è permessa tra gli slaves se non tramite il master. Sia il master che gli slaves sono configurati come transceiver, permettendo un flusso di dati in entrambe le direzioni, iniziando sempre dal bit meno significativo.



UN'INDIRIZZO PERSONALE PER OGNI DISPOSITIVO

All'interno di ogni dispositivo 1-wire è presente una memoria ROM contenente un numero di serie personale di 64 bit, che rappresenta l'indirizzo del dispositivo, e che, quindi, lo identifica univocamente nel bus 1-Wire. Questo indirizzo personale è composto da 8 byte e diviso in 3 parti. La prima parte, composta da un byte, contiene il codice identificativo della famiglia a cui appartiene il dispositivo, che ne indica quindi il tipo (sensore di temperatura, memoria eeprom, ADC programmabili, etc...)

La seconda parte è composta da 6 byte contenenti un numero personale, rappresentano l'indirizzo univoco del dispositivo.

L'ultima parte contiene un byte di CRC (Cyclic Redundancy Check), il cui valore dipende dai dati contenuti nei primi 7 byte. Con 2^{48} codici identificativi, conflitti o indirizzi duplicati sulla rete sono praticamente impossibili.

La tecnica di ridondanza di codice (CRC), molto applicata nei processi di trasmissione dati, permette al dispositivo MASTER di individuare se la lettura dei precedenti 56 bit è avvenuta correttamente, cioè non sono presenti "conflitti" nel processo di trasmissione

LA COMUNICAZIONE

I dati sulla rete OneWire usano i livelli logici TTL/CMOS standard e vengono trasferiti in "time slots", cioè porzioni di tempo di lunghezza prestabilita, tra 60 e 120us, in cui è possibile scambiare informazioni sul bus. Per esempio, per scrivere un uno logico sul bus, un dispositivo deve porre a 0 il bus, rilasciarlo dopo al più 15us e aspettare la fine dello slot, cioè almeno 45us.

Un clock di sistema non è richiesto, poiché ogni dispositivo OneWire contiene un oscillatore interno che si autosincronizza durante i fronti di discesa del clock del master.

Per iniziare la comunicazione il master deve inviare sulla rete un segnale di reset, portando

a 0 il bus per almeno 480us e, dopo averlo rilasciato, ascoltando l'impulso di presenza generato dagli slaves.

In seguito il master può individuare e selezionare un singolo slave presente sulla rete utilizzando i "ROM commands" previsti dal protocollo di comunicazione. Tra questi comandi almeno 4 sono comuni a tutti i dispositivi 1-Wire:

- 1) Search ROM [F0h]: mediante questo comando il master avvia un algoritmo di ricerca ciclico per conoscere i codici di ogni dispositivo presente sul bus. Ad ogni ciclo della ricerca il master individua un codice, lo memorizza e lo elimina dal ciclo successivo;
- 2) READ ROM [33h]: se un solo dispositivo è presente sul bus, il master può leggere direttamente il suo codice identificativo inviando il comando READ ROM, quindi, senza avviare un processo di ricerca;
- 3) MATCH ROM [55h]: una volta individuato il codice di ogni slave, il master può indirizzarli singolarmente inviando il comando MATCH ROM, seguito dai 64 bit di indirizzo;
- 4) SKIP ROM [CCh]: nel caso in cui sia presente un solo slave sul bus, il master può saltare la procedura di indirizzamento, inviando un comando di SKIP ROM, ed accedere così direttamente alle funzioni messe a disposizione dal dispositivo slave.

L'ALIMENTAZIONE DEI DISPOSITIVI

L'alimentazione dei dispositivi OneWire è ottenuta direttamente dal bus durante la comunicazione, quando la linea dati è a livello alto (3÷5,5V) [abilitando il "parasite power mode" collegando sul singolo device, il piedino VDD alla tensione di riferimento GND (0 V)].

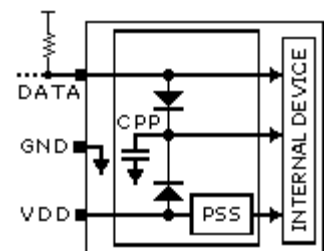
In questo caso un diodo (presente in ogni slave) viene polarizzato direttamente permettendo l'accumulo di energia nel condensatore interno al dispositivo (800pF).

Quando la tensione sul bus scende al di sotto della tensione presente sul condensatore, il diodo si trova polarizzato inversamente, isolando la carica presente sul condensatore.

La quantità di carica persa durante questo periodo viene ripristinata appena la linea dati ritorna alta.

DS1820 - Temp

DS1307 - RTC



RS 232

La RS232 è una linea di comunicazione asincrona, in quanto il segnale di temporizzazione non viene trasmesso, ma generato in modo autonomo da entrambe le periferiche collegate. La connessione è di tipo punto-punto, quindi permette di far comunicare tra loro solo due dispositivi.

I dati vengono trasmessi in modo seriale, in pacchetti da 8-9 bit racchiusi tra un bit di start ed uno di stop. La durata di ogni bit dipende dalla velocità di comunicazione; affinché tutto funzioni correttamente, quest'ultima deve essere uguale per entrambi i dispositivi.

Configurazione : impostare la velocità di comunicazione (baud-rate), il numero di bit del frame, la possibilità di utilizzare le connessioni per il controllo di flusso.