

# Controlli automatici

## Tecnologie informatiche per il controllo

**Prof. Paolo Rocco ([paolo.rocco@polimi.it](mailto:paolo.rocco@polimi.it))**

**Politecnico di Milano**

**Dipartimento di Elettronica, Informazione e Bioingegneria**

---



# Introduzione

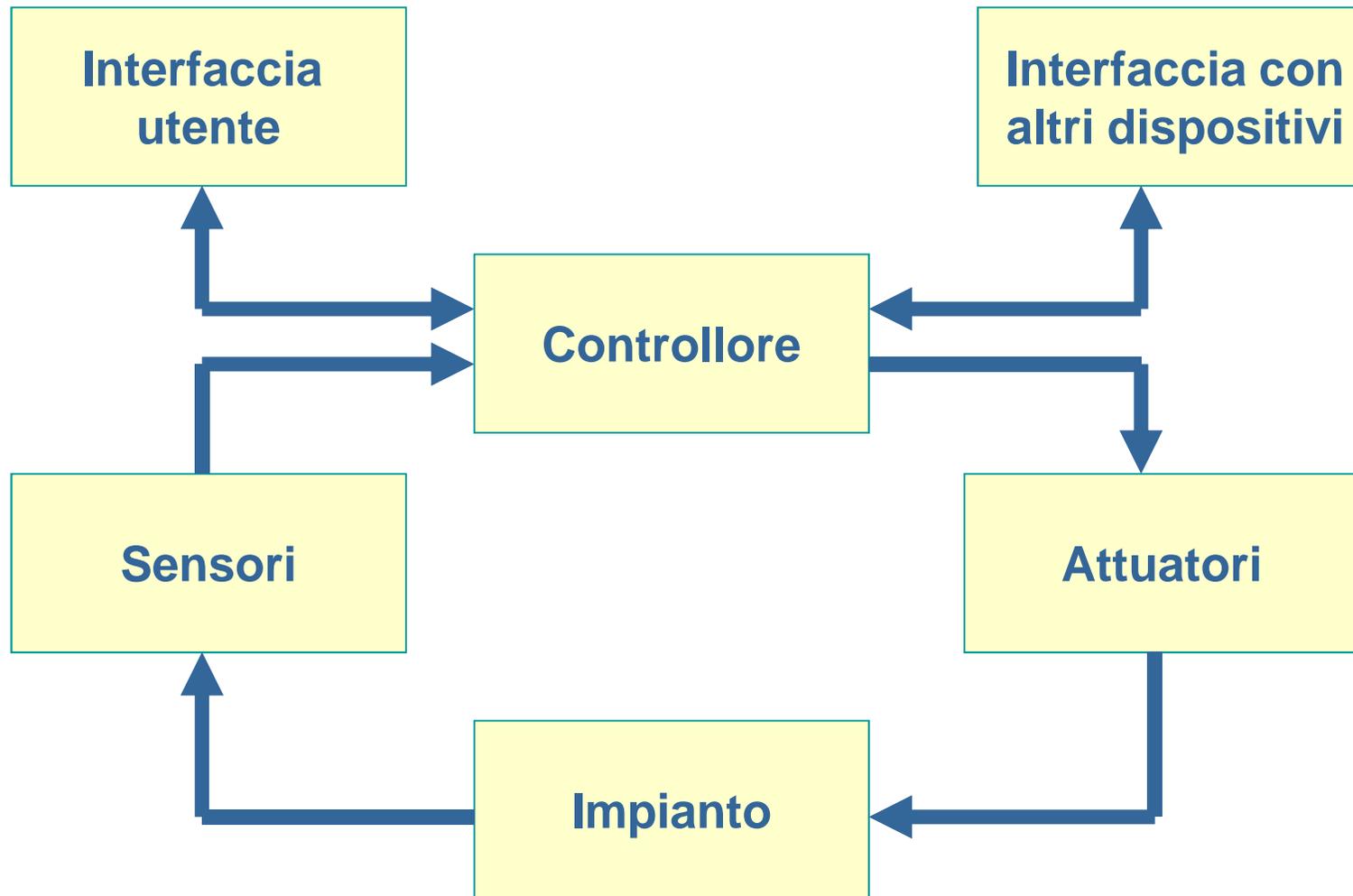
---



- In queste slide si fornirà un quadro, necessariamente sommario, del ruolo dei **sistemi informatici** in un sistema di automazione industriale, con riferimento in particolare alle architetture di controllo, alle reti di comunicazione, ai PLC e ai sistemi in tempo reale ed embedded.
- È importante essere a conoscenza almeno di un quadro di insieme di queste problematiche, peraltro in rapida e continua evoluzione.
- Studiare l'architettura (hardware e software) dei sistemi di controllo è un compito che va al di là, per sua stessa natura, dell'ambito dei sistemi meccatronici. Si farà quindi nel seguito riferimento a sistemi di controllo ed automazione non limitati strettamente al campo meccatronico.

- *Le presenti slide costituiscono prevalentemente una rielaborazione di materiale didattico elaborato dai proff. Cugola, Leva, Magnani e San Pietro del DEI (Politecnico di Milano).*
- *Diverse immagini riportate nelle presenti slide sono tratte dal testo: "Tecnologie dei sistemi di controllo, 2a ed.", G. Magnani, G. Ferretti, P. Rocco, McGraw-Hill Italia, 2007*

# Architettura generale di un sistema di controllo



# Controllo modulante e controllo logico

---



Il sistema di controllo esercita delle azioni che non si esauriscono in quelle note dei controllori analogici o digitali.

Si opera infatti una distinzione:

- Controllo **modulante**
  - controllo eseguito (in tecnologia analogica o digitale) da dispositivi che ad ogni istante si pongono come obiettivo che le variabili controllate inseguano i rispettivi riferimenti.
- Controllo **logico**
  - controllo eseguito da dispositivi che devono assicurare lo svolgimento di una o più sequenze di attività, la cui evoluzione è dettata dal verificarsi di eventi (come la conclusione di un'attività, l'insorgenza di anomalie, l'interazione con l'operatore).

Spesso il controllo logico opera a livelli più alti in un sistema di controllo, il controllo modulante a livelli più bassi.

# Controllo logico

---



Alcuni esempi di problemi di controllo logico:

- controllo di un ascensore
- distributore automatico di bibite
- sistema di illuminazione
- sistema semaforico

In tutti questi esempi sono previste delle **azioni** che devono essere eseguite al verificarsi di determinati **eventi**.

La sequenza delle azioni è gestita dal controllo logico, l'esecuzione delle singole azioni dal controllo modulante.

Un sistema costituito da un robot che preleva pezzi da un nastro trasportatore e li manipola comporta per esempio problematiche sia di controllo modulante sia di controllo logico.

# Controllo logico

---



Le specifiche di un sistema di controllo logico sono costituite da **sequenze di azioni**, descritte generalmente in linguaggio naturale.

Si pongono due problemi:

- formalizzare queste specifiche
- determinare metodologie di sintesi di un controllore che ne garantisca il soddisfacimento

Per quanto riguarda il primo aspetto, esistono formalismi di progressiva diffusione.

Per quanto riguarda il secondo aspetto, si fa generalmente ricorso alla **teoria dei sistemi ad eventi discreti**.

# Sistemi a eventi discreti

---



Un sistema ad eventi discreti è un **sistema dinamico** (quindi dotato di stato) in cui:

- Lo stato assume valori in un insieme discreto
- L'evoluzione dello stato è dettata esclusivamente dall'occorrenza di eventi asincroni (ossia la cui temporizzazione non rispetta caratteristiche di regolarità).

Un esempio classico è costituito dalla coda di persone ad uno sportello, in cui lo stato è costituito dal numero di persone in coda e gli eventi sono l'arrivo di nuove persone in coda o l'uscita di altre.

Tra gli strumenti matematici di più largo utilizzo per la modellazione dei sistemi ad eventi discreti vi sono gli **automi a stati finiti** e le **reti di Petri**.

Pur sottolineandone l'importanza ai fini della progettazione e della comprensione di un sistema di automazione industriale, queste problematiche non saranno trattate nelle presenti lezioni.

# Architettura di un sistema di controllo

---



Il sistema di controllo elabora informazione (acquisendola dai sensori e trasferendola agli attuatori) e come tale il suo progetto, nei casi più complessi, va condotto con le tecniche proprie dell'ICT (Information and Communication Technology).

Vi sono vari livelli in cui tale progetto si articola (si usa la terminologia standard IEEE):

- **Preliminary Design**
  - si analizzano alternative di progetto e si definiscono preliminarmente l'architettura, i componenti e le interfacce
- **Functional Design**
  - si definiscono le funzionalità e le interfacce tra i componenti del sistema
- **Detailed Design**
  - si rifinisce ed espande il progetto preliminare del sistema e dei componenti a livello tale che il progetto è sufficientemente completo per cominciarne l'implementazione
- **Architectural Design**
  - si definiscono i componenti hardware e software del sistema e le loro interfacce

# Progetto preliminare

---

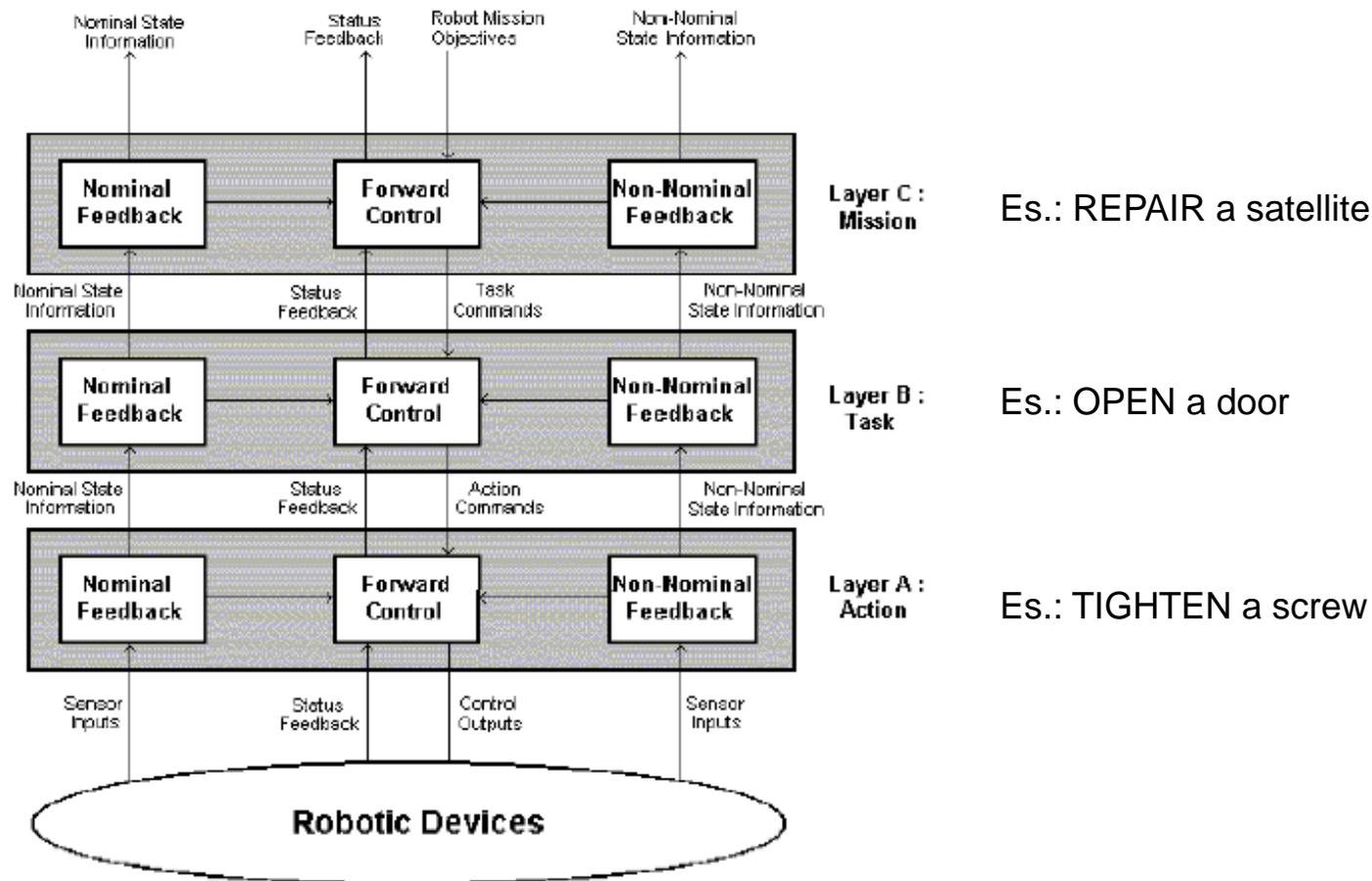


- Una prima fase progettuale consiste nella raccolta e nell'analisi dei **requisiti d'uso** del sistema
- Occorre specificare sia il comportamento atteso del sistema (che cosa deve fare) sia il comportamento escluso (che cosa è bene che il sistema non faccia)
- Successivamente si passa alla definizione di massima delle funzionalità del sistema
- In generale si può dire che le funzionalità del sistema di controllo sono scomposte in funzionalità più semplici (divide et impera)
- Si ottiene una **gerarchia di funzionalità** da implementare con gli opportuni componenti
- Si trasferiscono metodologie di progetto tipiche del mondo informatico all'ambito controllistico o di automazione più in generale



# Un esempio: CDM dell'ESA

Un esempio di scomposizione gerarchica di funzionalità è la CDM (Control Development Methodology) dell'ESA (European Space Agency), articolata nei livelli Mission, Task e Action

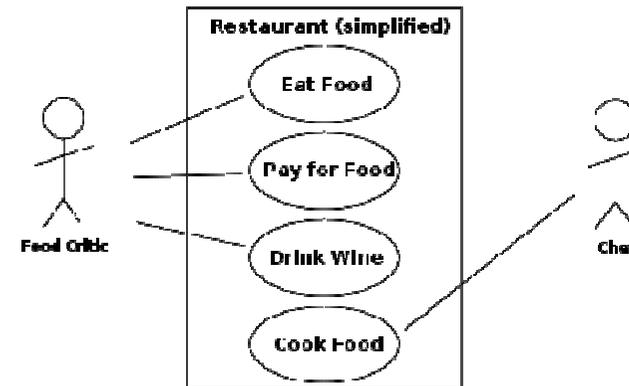


# Diagrammi UML

Nella definizione delle funzionalità di un sistema informatico sono molto usati formalismi grafici.

Uno dei più noti è **UML** (Unified Modelling Language)

- UML fornisce una serie di strumenti grafici per creare modelli visuali di sistemi software orientati agli oggetti
- È oggi uno standard nel progetto di questi sistemi



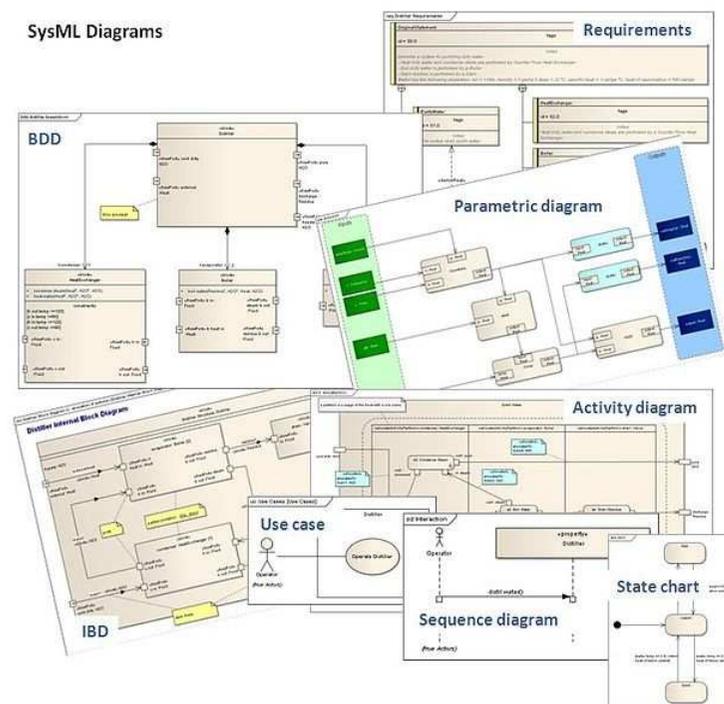
Esempio di "use case diagram" in UML

# Progetto funzionale

L'**architettura funzionale** del sistema di controllo:

- definisce i singoli componenti per mezzo delle loro **interfacce**
- definisce le interazioni tra i componenti ed in particolare i canali di comunicazione ed i dati
- definisce per ciascun componente le funzioni, i requisiti in termini di capacità di memoria e le loro responsabilità

- Anche lo sviluppo dell'architettura funzionale si può condurre con strumenti grafici di ausilio alla progettazione.
- Ad esempio si usano diagrammi **SysML** (evoluzione del linguaggio UML).



# Progetto hardware/software

---



L'implementazione finale del sistema informatico, dopo il progetto di dettaglio, passa dalla realizzazione dell'architettura hardware e software:

## Architettura hardware

- Componenti hardware del sistema di controllo
  - Sensori
  - Attuatori
  - Organi di controllo
  - Canali di comunicazione (cavi...)

## Architettura software

- Componenti software del sistema di controllo
- Connessioni logiche tra i componenti
- Dati che fluiscono tra i componenti

# Alcune problematiche architetture

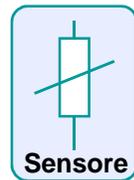
---



Lo sviluppo dell'architettura hardware/software di un sistema di controllo (e in particolare di uno per applicazioni meccatroniche) deve tener conto di alcune **peculiarità** che lo differenziano dallo sviluppo di un generico sistema informatico:

- Grandi quantità di dati e segnali da gestire con diverse scale temporali, d'importanza e di criticità
- Determinismo temporale (particolarmente critico in meccatronica).
- Semplicità e comodità nella costruzione e nella (ri)configurazione del sistema (mettere le mani su un sistema di controllo dentro un impianto non è come farlo su una rete di PC in un ufficio).
- Tolleranza ai guasti ed ai malfunzionamenti (sia del software sia dell'hardware).
- Interfaccia operatore (che coinvolge anche l'hardware) adatta a personale con grado di cultura ed addestramento molto variabile.

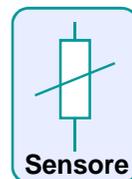
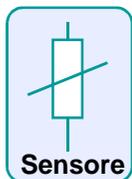
# Connessioni di componenti di campo



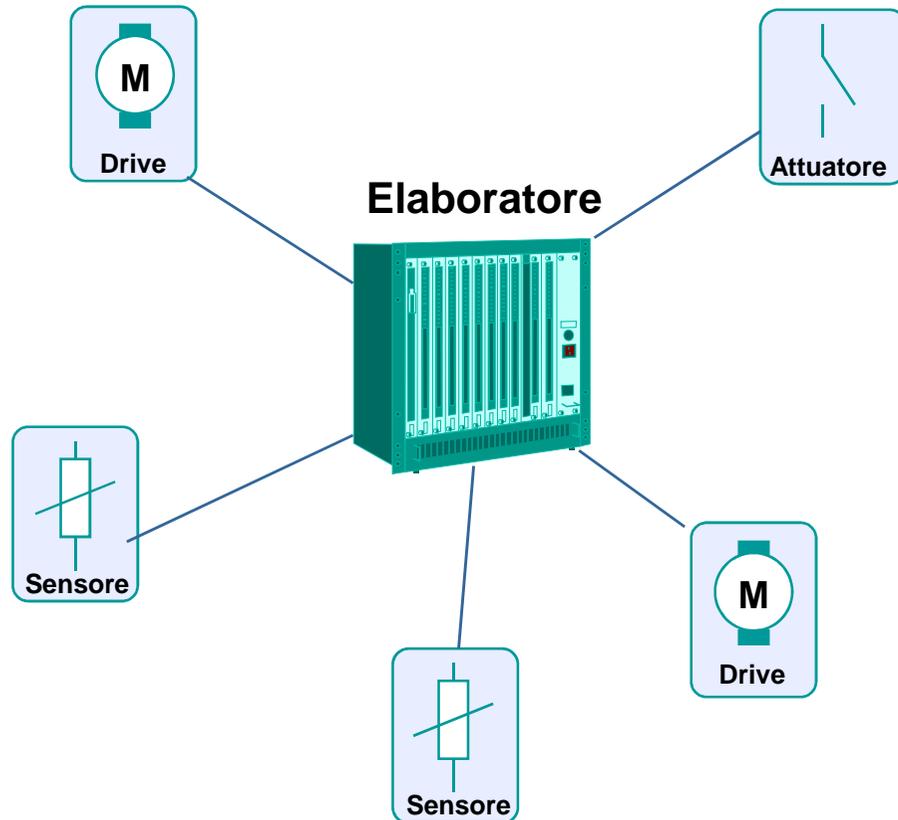
Nel caso più semplice di architettura hardware di sistema di controllo, due componenti “di campo”, un trasduttore e un attuatore sono collegati al controllore, con collegamenti analogici (in corrente o tensione).



Come sono organizzate le connessioni se vi sono numerosi componenti di campo?



# Architettura tradizionale



Architettura **centralizzata** con collegamenti di tipo **analogico** punto – punto

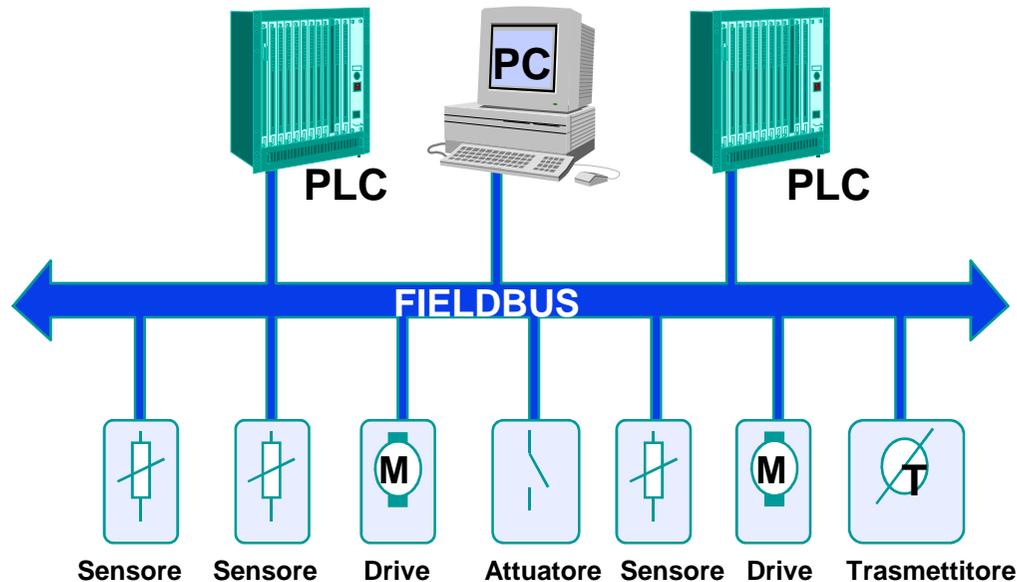
## Vantaggi

- Affidabilità, sistema collaudato

## Svantaggi

- Elevato numero collegamenti
- Costo dei cablaggi
- Elevata sensibilità ai disturbi (necessità di schermatura)
- Scarsa flessibilità e scalabilità

# Architettura a bus



Architettura a bus con trasmissione digitale dei segnali

## Vantaggi

- Risparmio sui costi di cablaggio
- Facilità di aggiunta e rimozione di dispositivi
- Condivisione delle risorse
- Flessibilità
- Decentramento di funzioni
- Intelligenza distribuita (funzionalità diagnostiche locali)

## Svantaggi

- Maggior costo dei dispositivi
- Diverse metodologie di progettazione
- Problemi di interoperabilità (far funzionare insieme dispositivi di produttori diversi)

# Un esempio di architettura hw: CNC

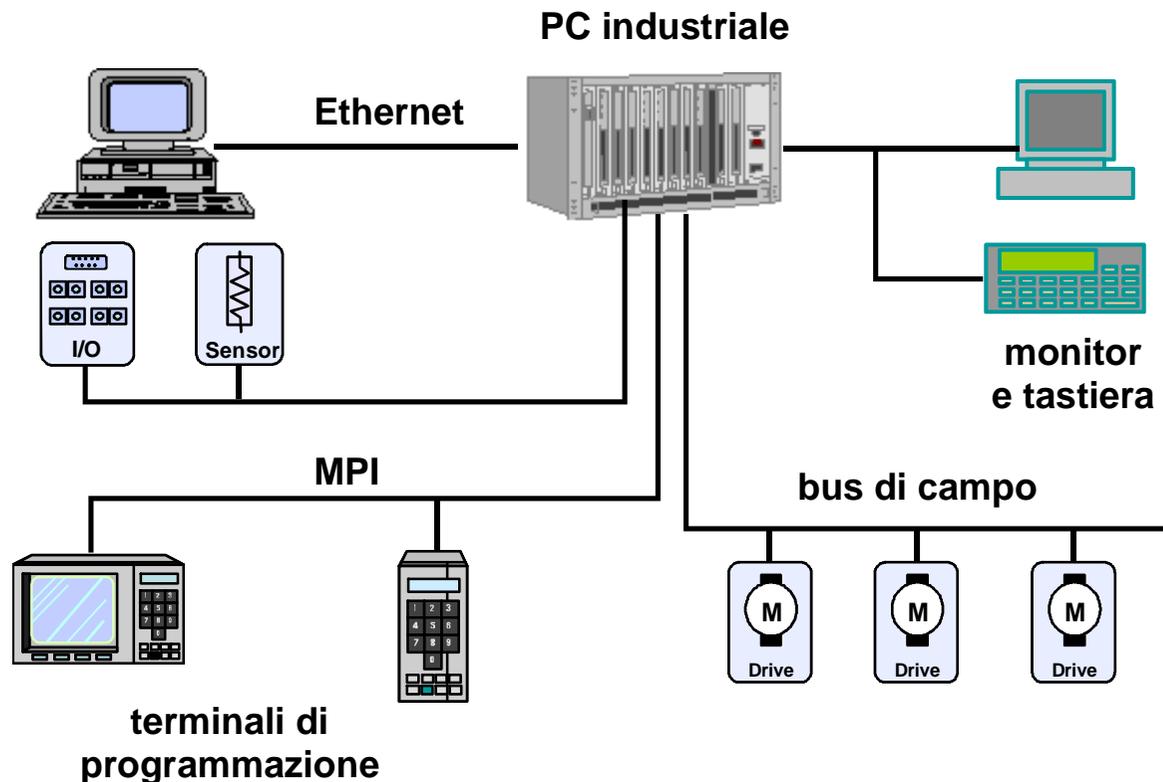
---



- I CNC (Computer Numerical Control) sono dispositivi utilizzati nel controllo di macchine utensili e altre macchine di produzione.
- Il compito primario del CNC è quello della pianificazione o generazione delle traiettorie dell'utensile e conseguentemente degli assi di macchina.
- Il valore di un CNC è per la maggior parte costituito dal software estremamente sofisticato necessario per la pianificazione e l'esecuzione dei programmi di lavoro (cosiddetti *part program*)
- L'architettura software è quindi estremamente complessa

Nel seguito si accennerà all'[architettura hardware](#) del Simumerik 840Di di Siemens, come esempio di struttura complessa, articolata in più componenti connessi da opportune comunicazioni digitali.

# Un esempio di architettura hw: CNC



- PC industriale: PC con caratteristiche costruttive adatte ad uso in ambito industriale
- Scheda motion control sul PC industriale con funzioni PLC e interfaccia a bus di campo
- MPI: Multi Point Interface: interfaccia veloce verso terminali di comando

Nel CNC si chiude l'anello di controllo di posizione, mentre quelli di velocità e corrente si chiudono sugli azionamenti. La comunicazione verso gli azionamenti avviene tramite "bus di campo". Nel Sinumerik si usa il bus PROFIBUS DP.

# Reti di comunicazione digitale

---



- Si è visto che dispositivi di campo (trasduttori, attuatori, valvole, drive, ecc..) possono essere collegati tra loro e con uno o più sistemi di elaborazione mediante **reti di comunicazione digitale**
- Ovviamente il campo di applicazione di tali reti è enormemente più vasto e riguarda qualsiasi dispositivo (dal PC allo smartphone) “connesso in rete”
- È allora opportuno avere qualche elemento di base di conoscenza delle reti digitali, rimandando per i dettagli a testi specifici

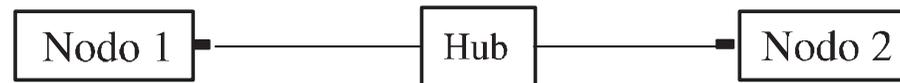
# Rete di calcolatori



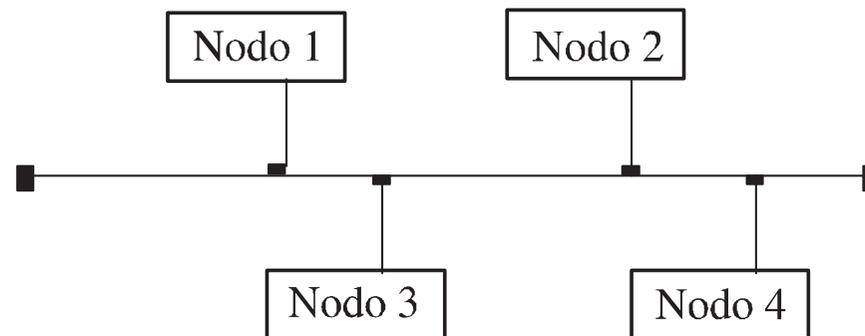
Una rete di calcolatori è un sistema informatico costituito da uno o più calcolatori collegati da un sistema di comunicazione.

Esistono varie **topologie** (ovvero disposizioni fisiche dei componenti) di rete:

Collegamento punto a punto (a stella)



Collegamento a bus lineare (multidrop)



Collegamento daisy-chain



# Protocollo di comunicazione

---



Per protocollo di comunicazione si intende l'insieme delle regole seguendo le quali due interlocutori possono comprendersi.

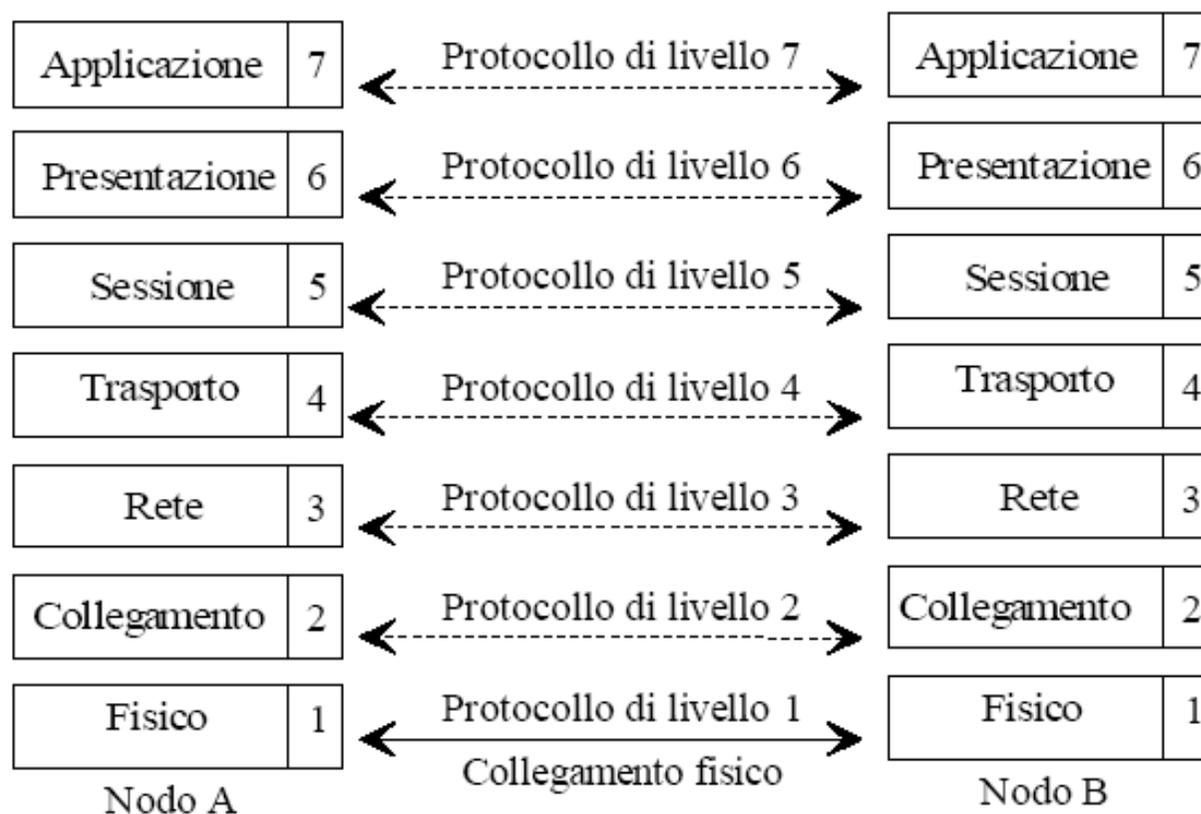
I protocolli utilizzati dai calcolatori sono organizzati secondo una **gerarchia**.

Ogni protocollo si appoggia ai protocolli di più basso livello per fornire il servizio (per esempio un protocollo con correzione d'errore può essere costruito su un protocollo di puro trasporto).



# Modello ISO/OSI

Il modello di riferimento **OSI** (Open System Interconnection) dell'ISO (International Standard Organization) dà una descrizione concettuale in cui le funzioni realizzate dalle reti sono organizzate come una serie di livelli.



# Modello ISO/OSI

---



- Livello 7:** interfacce e servizi ai programmi applicativi (trasferimento file, collegamento remoto)
- Livello 6:** codifica informazioni e loro conversione (dati binari rappresentati come testo, figure,...)
- Livello 5:** fornisce servizi per la gestione efficiente della comunicazione (apertura, gestione, chiusura)
- Livello 4:** realizza funzioni di trasporto indipendenti dalla struttura della rete. Interfaccia tra i livelli sopra e sotto.
- Livello 3:** gestisce l'instradamento dei messaggi. Stabilisce un sentiero logico anche tra nodi non direttamente connessi.
- Livello 2:** corretto trasferimento di stringhe di bit organizzate in frame. Metodo di accesso al mezzo.
- Livello 1:** gestione meccanica ed elettrica della comunicazione.

Non è necessario che in una rete siano implementati tutti e sette i livelli: i livelli fisico, collegamento e applicazione sono i più utilizzati nei sistemi di controllo.

# Modello ISO/OSI e internet

---



Come si mappano i livelli del modello ISO/OSI su internet?

**Livelli 7, 6 e 5:** Internet protocol suite (Telnet, FTP, SMTP, HTTP)

**Livello 4:** TCP (Transmission Control Protocol) o UDP (User Datagram Protocol)

**Livello 3:** IP (Internet Protocol)

**Livelli 2 e 1:** non specificati (di solito Ethernet)

# Livello fisico

---



Si occupa del trasferimento di bit su un canale di comunicazione, nei suoi aspetti meccanici ed elettrici.

La trasmissione è di norma di tipo seriale.

## Alcune problematiche

- tipo di codifica elettrica del segnale logico;
- tipo e funzione di ogni segnale di controllo (per sincronizzare trasmissione e ricezione);
- livelli di tensione e corrente;
- numero di conduttori;
- tolleranze massime e minime dei tempi di salita, commutazione, ecc. dei segnali elettrici;
- tipo di collegamento (single ended o differenziale) e isolamento;
- caratteristiche meccaniche dei connettori, dei cavi, e in generale degli apparati.

# Mezzi trasmissivi



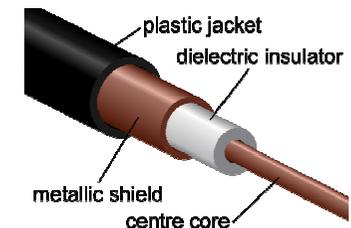
- **Doppino intrecciato** (*twisted pair*)

- Costituito da una coppia di fili di rame avvolti in una guaina e ritorti
- Può essere con o senza schermo
- Tipicamente usato in telefonia, consente velocità di comunicazione medio-alte (100 Mbps su rete locale, meno su rete telefonica)



- **Cavo coassiale**

- Filo centrale in rame ricoperto da guaina e maglia esterna in rame
- Consente velocità di comunicazione medio-alte (100 Mbps)



- **Fibra ottica**

- Fibra di vetro in grado di trasportare segnali luminosi
- Consente massima insensibilità ai rumori ma ha costo elevato



- **Wireless**

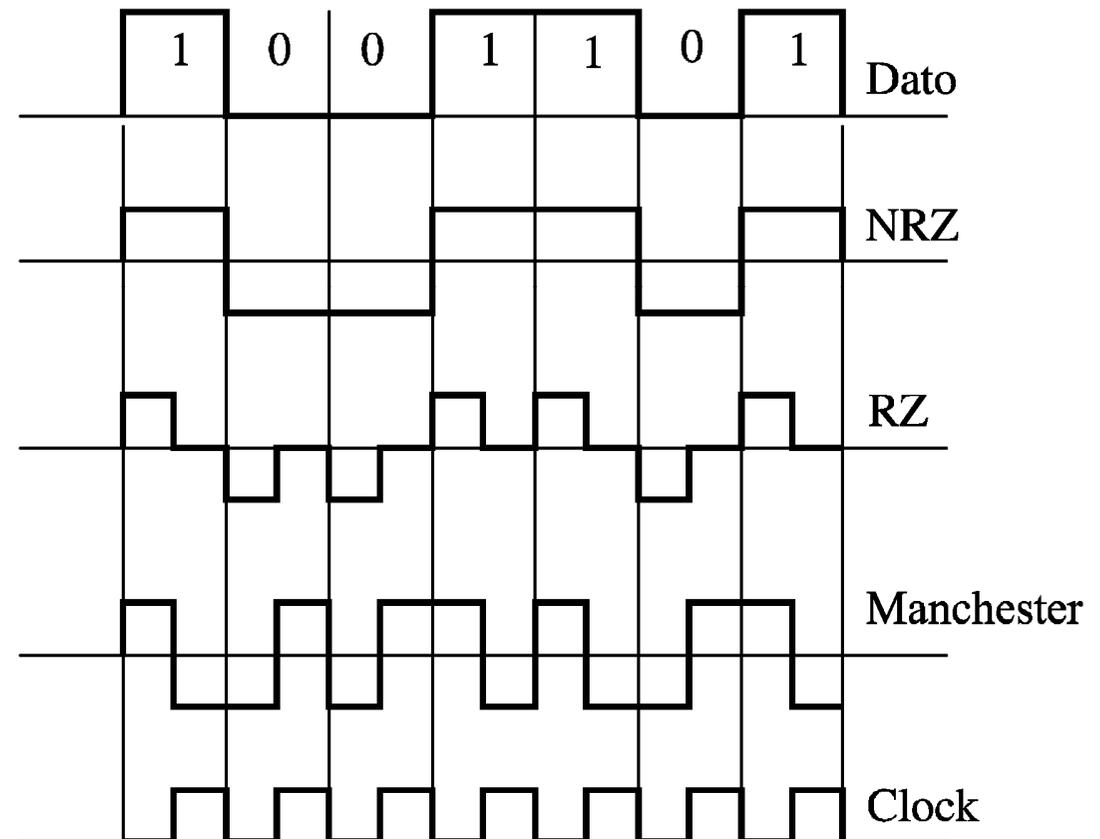
- In crescita anche in ambito industriale

# Codifica del segnale logico



I dati binari scambiati tra due calcolatori possono essere trasmessi direttamente sul canale (segnalazione in **banda base**)

- Codifica **Non Return to Zero**: non impone la transizione a un livello di riferimento (zero)
- Codifica **Return to Zero**: impone la transizione a zero
- Codifica **Manchester**: codifica i valori con i segni delle transizioni (in pratica si trasmette il clock o il clock invertito)



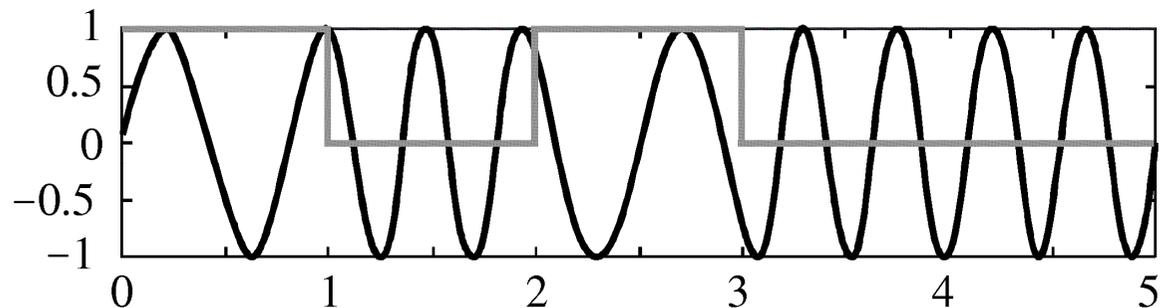
# Codifica del segnale logico



I dati binari possono essere anche trasmessi per modulazione di un segnale portante (segnalazione in **modulazione**)

Protocollo **HART**:

L'informazione binaria è codificata inviando sinusoidi di frequenza diversa (1220 Hz e 2200 Hz)



# Standard elettrici e meccanici

---



Alcuni standard:

- EIA **RS-232C**
  - uno degli standard più vecchi, ma tuttora in uso
  - distanza massima: 15 m, velocità massima: 20 kbit/s
  - usa 3 fili: GND (ritorno comune), TXD (trasmissione), RXD (ricezione)
  
- EIA **RS-422**
  - velocità massima: 115 kbit/s per distanze fino a 1200 m, 10 Mbit/s fino a 12 m
  - usa collegamenti differenziali anziché single-ended
  - più adatto all'uso in ambito industriale
  - si possono collegare fino a 10 ricevitori
  
- EIA **RS-485**
  - molto usato in ambito industriale
  - velocità e distanze come RS-422
  - si possono collegare fino a 32 trasmettitori e 32 ricevitori
  - i trasmettitori possono porsi nello stato di alta impedenza (e scollegarsi)

# Direzione di trasmissione

---



La trasmissione può avvenire con le seguenti modalità di direzione:

- **Simplex** (il senso della trasmissione è unico)
- **Half duplex** (la trasmissione è possibile, *alternativamente*, nei due sensi)
- **Full duplex** (la trasmissione è possibile, *contemporaneamente*, nei due sensi)

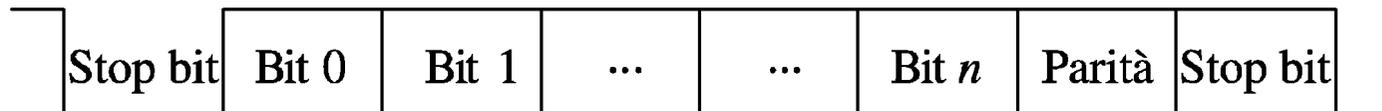
Quasi tutte le apparecchiature attuali sono, a livello fisico, predisposte per la trasmissione full duplex.

Tuttavia i protocolli possono limitare la trasmissione alla modalità half duplex.

# Trasmissione asincrona



- Nella **trasmissione asincrona** il trasmettitore prende l'iniziativa di inviare il dato
- La trasmissione avviene per carattere (da 5 a 8 bit di informazione)
- Il formato della trasmissione è il seguente:



# Trasmissione sincrona



- Nella **trasmissione sincrona** si richiede la sincronizzazione di trasmettitore e ricevitore (mediante clock o segnale Manchester)
- L'informazione è trasmessa come un blocco di dati preceduto e seguito da sequenze di bit di sincronizzazione e controllo.
- L'insieme è denominato frame:

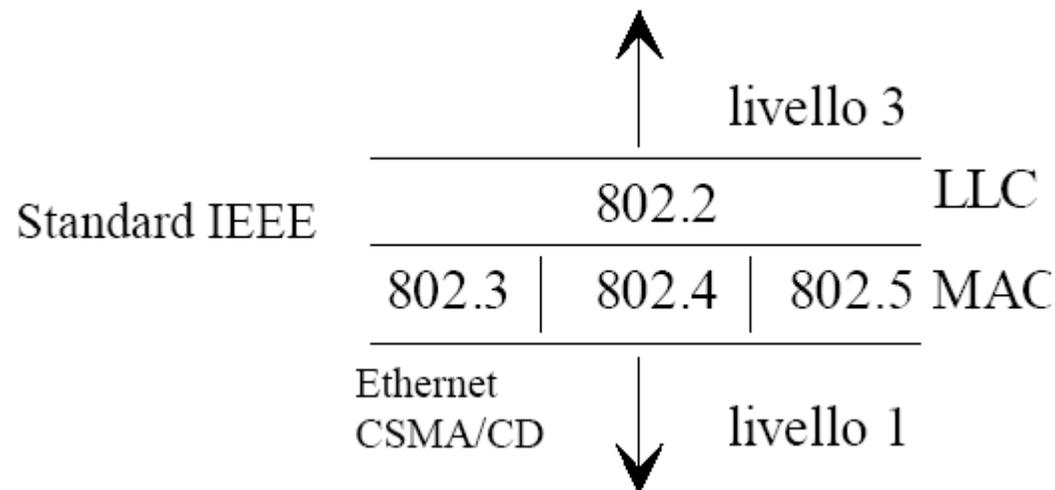
Start frame	Indirizzo destinatario	Indirizzo mittente	Dati	Checksum	End frame
-------------	------------------------	--------------------	------	----------	-----------

- L'efficienza di trasmissione è più alta della trasmissione asincrona
- La gestione della linea sincrona è però più onerosa
- Circuiti integrati denominati **USART** (Universal Synchronous Asynchronous Receiver Transmitter) gestiscono le modalità di trasmissione/ricezione (sia sincrona sia asincrona), scaricando l'elaboratore da queste attività

# Livello collegamento (data link)



- Si occupa di realizzare connessioni prive di errori fra coppie di nodi della rete.
- Suddivide i dati che riceve dal livello rete e li organizza in frame, introducendovi informazioni di controllo, come indirizzo del mittente e del destinatario e bit aggiuntivi.
- Trasmette i frame e verifica la corretta ricezione sulla base dei segnali di acknowledge del destinatario, ripetendo se del caso la trasmissione.



- **LLC** Logical Link Control - controllo del collegamento logico
- **MAC** Medium Access Control - controllo di accesso al mezzo fisico

# Controllo di accesso al mezzo fisico

---



Per il MAC sono stati definiti i seguenti standard alternativi:

- CSMA/CD (IEEE 802.3)
- Token-Bus (IEEE 802.4)
- Token-Ring (IEEE 802.5)

## CSMA/CD (Carrier Sensing Multiple Access / Collision Detection)

- Protocollo di tipo *spontaneo*: ogni nodo può decidere di trasmettere in qualsiasi momento. Trasmissioni contemporanee sono gestite da un meccanismo di *collision detection* che prevede una logica di “ritentativi”.
- Non è in grado di garantire un limite superiore certo al ritardo di risposta: non è quindi ritenuto adatto per le applicazioni in cui questo deve essere garantito (es. applicazioni in tempo reale)
- È utilizzato in Ethernet (la cui velocità è molto elevata, per cui il problema è poco critico)

# Controllo di accesso al mezzo fisico

---



## Token-Bus / Token-Ring

- In ogni istante ha diritto a trasmettere solo un nodo, quello che possiede il *token* (gettone), ovvero un codice particolare.
- I nodi si passano in sequenza “circolare”(round-robin) il token
- Il token dà diritto a trasmettere per un tempo massimo stabilito (THT: Token Holding Time), dopo di che va lasciato.
- Possono esserci stazioni attive (*master*) e passive (*slave*): le slave possono trasmettere solo se interrogate da un master.
- Il ritardo di risposta di ogni unità ha un limite superiore (*worst-case*)
- Si ha però un tempo d’attesa inutile

## Accesso mediante arbitro

Nelle reti di campo spesso l’allocazione del mezzo fisico è realizzata mediante un arbitro del bus (master): è un dispositivo collegato al bus con la funzione di deciderne l’allocazione tra i diversi nodi della rete.

# Ethernet

---



- Sviluppata da Xerox nel 1976, è il protocollo più usato nelle applicazioni per ufficio ed è anche di largo utilizzo per applicazioni industriali.
- Il protocollo Ethernet copre i livelli 1 e 2 della pila OSI
- Il **mezzo fisico** ha subito varie evoluzioni:
  - cavo coassiale spesso (*Thick Ethernet*): 10 Mbit/s fino a 500 m
  - cavo coassiale sottile (*Thin Ethernet*): 10 Mbit/s fino a 200 m
  - doppino ritorto (*Twisted Pair Ethernet*), per reti con concentratore (hub): 10 Mbit/s fino a 100 m o 500 m
  - doppino ritorto (*Fast Ethernet*), per reti con concentratore (hub): 100 Mbit/s fino a 100 m
  - *fibra ottica*: 10 Mbit/s fino a 2 Km
  - *Gigabit Ethernet*: 1Gbit/s
- La segnalazione è in banda base con codifica Manchester
- La topologia dei collegamenti è varia (a bus lineare, a stella con hub, ...)

# Ethernet: livello collegamento



- Sottolivello MAC: si utilizza il protocollo CSMA/CD
- Sottolivello LLC: si utilizza il frame Ethernet II (802.2)

Preambolo	Indirizzo destinatario	Indirizzo mittente	Tipo	dati	Frame check CRC
7 byte + SFD SFD = 01010111	6 byte	6 byte	2 byte	46-1500 byte	4 byte

- Preambolo (7 byte): consente la sincronizzazione
- SFD (Start Frame Delimiter): sequenza 10101011, segnala l'inizio della parte utile
- Indirizzo fissato nell'hardware (scheda d'interfaccia) dal costruttore (se si cambia la scheda cambia l'indirizzo)
- Tipo: specifica il protocollo incapsulato nel frame
- Dati: da 46 a 1500 byte
- Frame check CRC: verifica di correttezza della trasmissione

# Ethernet: livello collegamento

---



- L'indirizzo della scheda Ethernet è denominato **MAC address**
- È composto da 48 bit: i primi 24 identificano l'organizzazione che assegna l'indirizzo ("Organization Unique Identifiers" - OUI), gli altri 24 sono assegnati dall'organizzazione con il solo vincolo dell'univocità
- Computer collegati ad Ethernet possono inviarsi dati utili utilizzando protocolli di alto livello (come il protocollo **TCP/IP** utilizzato in Internet).
- L'interfaccia Ethernet ignora i pacchetti con indirizzo diverso dal proprio (non carica la CPU con questo test)

# Bus di campo (fieldbus)

---



- Un bus di campo è una **rete digitale** concepita per collegare tra loro, ed eventualmente a elaboratori o ad altra rete aziendale, i **dispositivi di campo** (sensori, attuatori, regolatori), eliminando i collegamenti analogici (linee 4÷20 mA o  $\pm 10$  V)
- L'idea di "bus di campo" è che tutti i componenti di un sistema di controllo, dal PC per l'interfaccia operatore al sensore o all'attuatore, sono pensati come dispositivi in rete.
- In questo modo, concettualmente, un solo "cavo" (o bus) percorre tutto l'impianto e connette tutti i dispositivi.
- Si hanno conseguentemente vantaggi in termini di risparmio nei costi di cablaggio e di qualità delle trasmissioni (digitali anziché analogiche)
- In generale i bus di campo consentono la trasmissione ottima di volumi di dati piccoli, con criticità temporale.

# Bus di campo e modello OSI



I bus di campo non realizzano tutti i livelli della **pila OSI**.

- Sono realizzati i livelli 1 (fisico), 2 (collegamento) e 7 (applicazione).
- Non sono realizzati i livelli da 3 a 6 (rete, trasporto, sessione, presentazione).
- Inoltre viene aggiunto un livello 8 (livello utente) che fornisce molte funzioni importanti, quali blocchi funzionali, servizi di descrizione dei dispositivi, gestione della rete.

Utente	8
Applicazione	7

Collegamento	2
Fisico	1

Mezzo di collegamento

# Bus di campo e standardizzazione

---



- Da tempo vari produttori e utilizzatori di componenti cercano di definire un **protocollo standard** per le reti di dispositivo (che definisca aspetti hardware, software, tipo e formato di informazioni)
- Uno standard di questo tipo garantirebbe intercambiabilità ed interoperabilità di dispositivi di campo di produttori diversi
- Tuttavia ad oggi non è stato possibile definire tale standard universale
- Un'importante iniziativa di standardizzazione è costituita dal **FOUNDATION Fieldbus**
  - è una rete nata per rimpiazzare le linee analogiche 4÷20 mA
  - nella versione H1 ha una velocità di 31.25 kbit/s
  - è usata principalmente nell'industria di processo
  - sito web: <http://www.fieldbus.org>

# Standard riconosciuti

---



La normativa [IEC 61158](#) (del 1999) riconosce i seguenti standard:

- Type 1: Foundation Fieldbus H1
- Type 2: ControlNet
- Type 3: PROFIBUS
- Type 4: P-Net
- Type 5: FOUNDATION fieldbus HSE (High Speed Ethernet)
- Type 6: SwiftNet (un protocollo sviluppato da Boeing, oggi ritirato)
- Type 7: WorldFIP
- Type 8: Interbus

Il mercato nel [controllo di processo](#) è dominato da:

- [FOUNDATION Fieldbus](#)
- [PROFIBUS PA](#)

Le due tecnologie usano lo stesso layer fisico (collegamenti bifilari, codifica Manchester, modulazione in corrente, velocità 31.25 kbit/s) ma non sono intercambiabili.

# Fieldbus per l'automazione di fabbrica

---



Nell'automazione di fabbrica si usano generalmente bus di campo diversi da quelli usati nel controllo di processo.

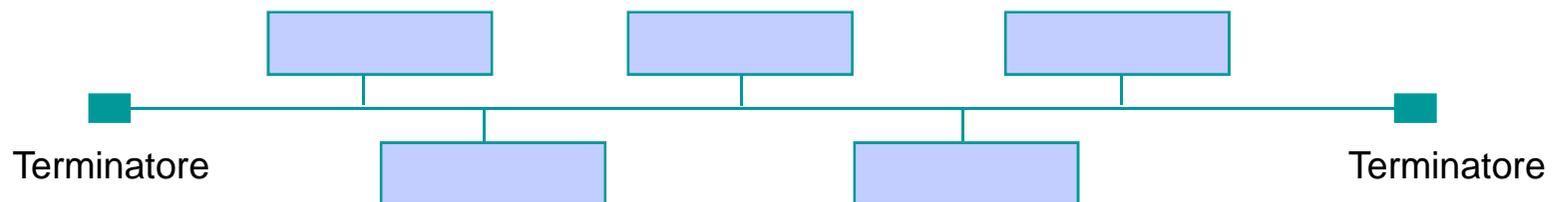
Di particolare rilievo sono i seguenti standard:

- **PROFIBUS DP**  
di grande rilevanza industriale e ampiamente supportato
- **CAN**  
concepito diversi anni fa, ma tuttora di grande importanza soprattutto nel settore veicolare
- **Ethernet industriale**  
in rapida evoluzione e diffusione

# PROFIBUS



- PROFIBUS (Process Field Bus) è un bus di campo di larga diffusione
- È disponibile in varie versioni, di cui le più diffuse sono:
  - **PROFIBUS PA** (Process Automation), usato nel controllo di processo
  - **PROFIBUS DP** (Decentralized Peripherals), usato nell'automazione di fabbrica
- Usa una topologia a bus con terminazione ai due lati
- Si possono connettere fino a 32 stazioni su un singolo elemento
- Sito web: <http://www.profibus.com>



# PROFIBUS-DP

---



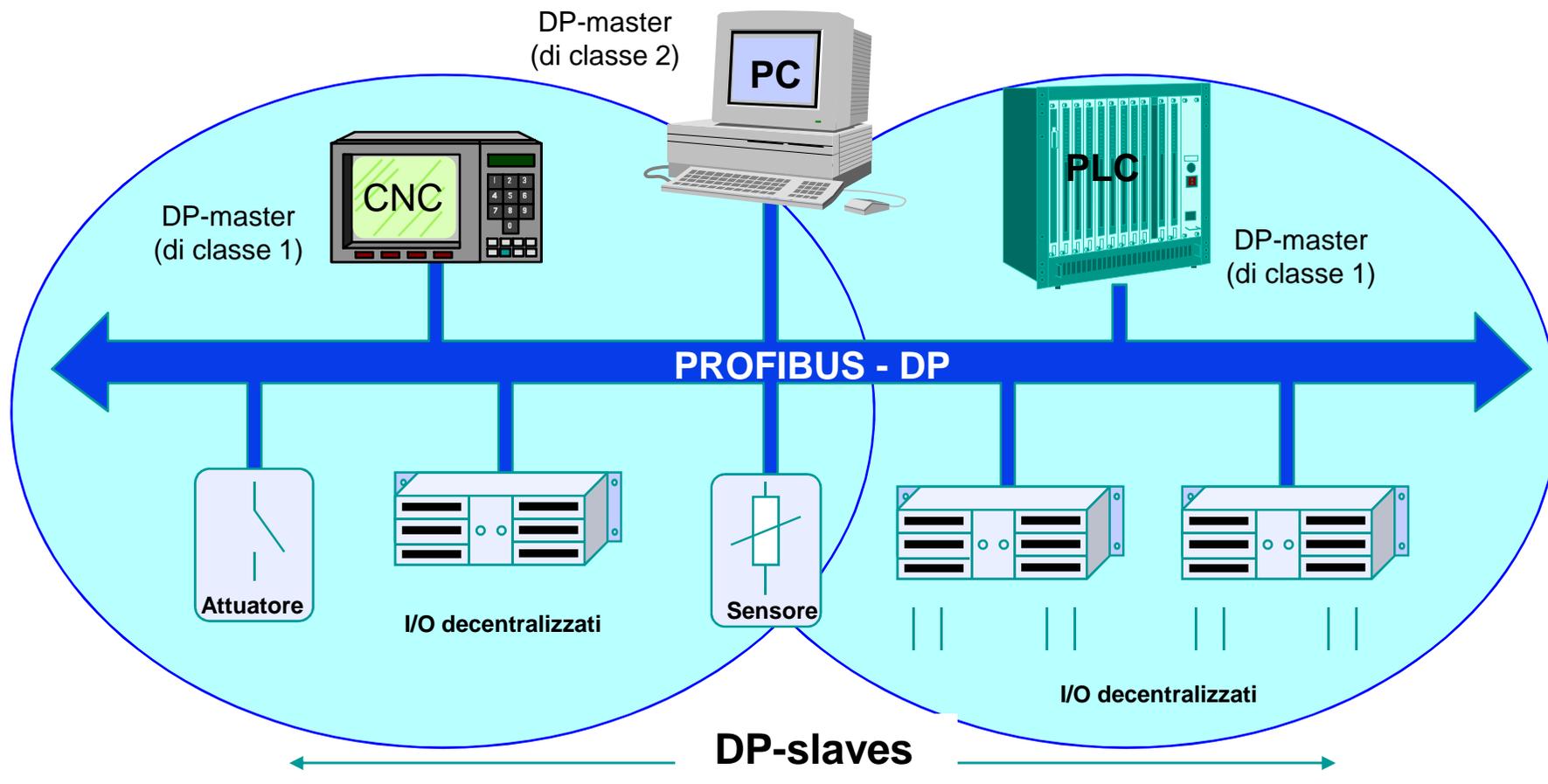
- È orientato all'automazione industriale
- Realizza i livelli 1 e 2 (fisico e di collegamento)
- Per il livello fisico (livello 1 della pila OSI) si usano due tipologie:
  - RS 485 (velocità da 9.6 Kbit/s a 12 Mbit/s)
  - fibra ottica
- Sopra questi livelli definisce dei **profili** (livello utente) tipici dell'automazione industriale. I profili sono specifiche definite dai costruttori o dagli utenti finali che riguardano proprietà, caratteristiche e comportamento dei dispositivi connessi.
- Vi è una vasta gamma di prodotti disponibili: PLC, PC, I/O, Azionamenti, Valvole, Encoder,...

# PROFIBUS-DP: sistema multi-master

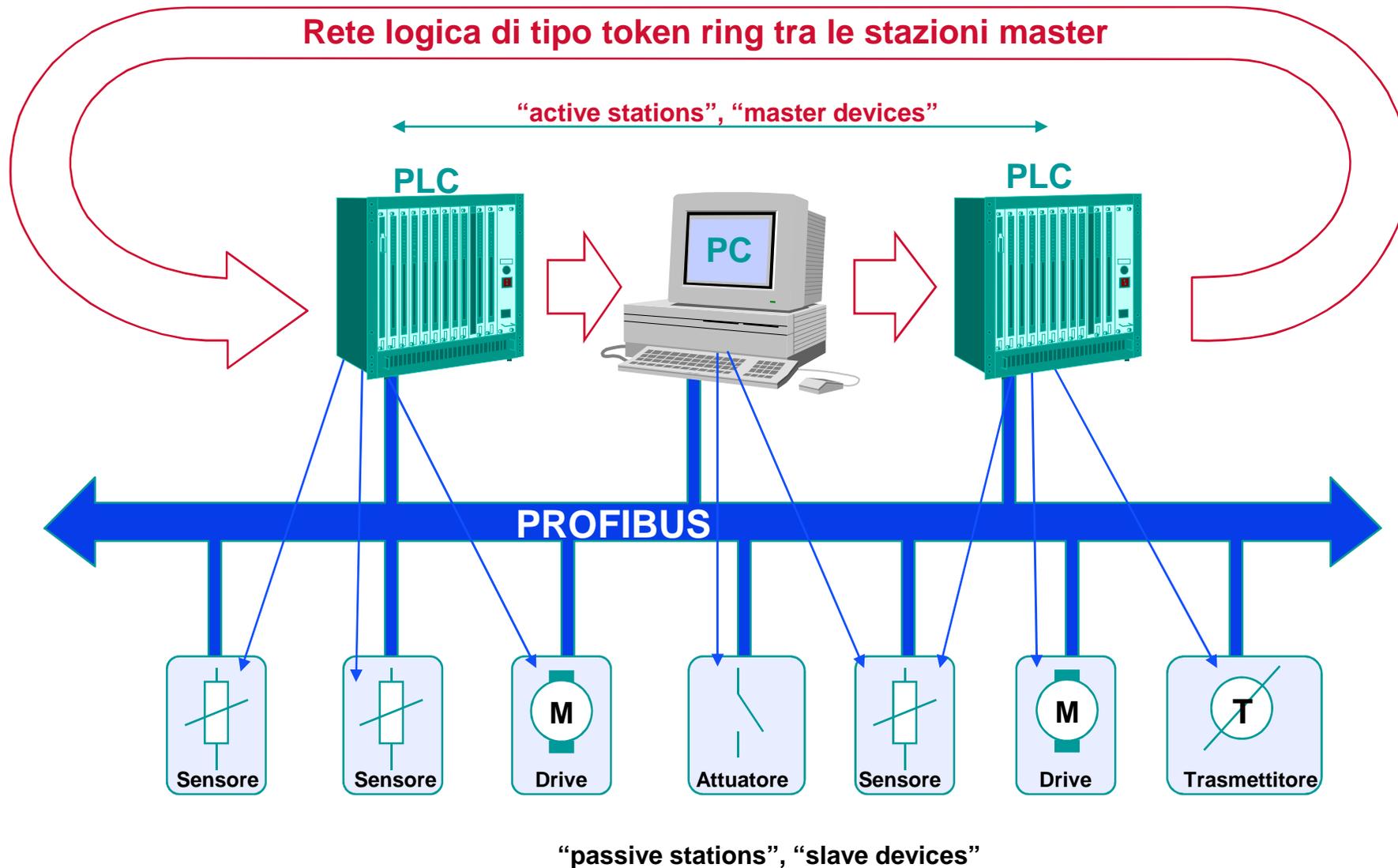


Si usa un sistema di comunicazione di tipo **master-slave**:

- Master di classe 1: permanentemente connesso
- Master di classe 2: connesso occasionalmente per configurazione



# PROFIBUS-DP: passaggio del token



# CAN



- CAN (Controller Area Network) è un bus di comunicazione seriale, progettato per applicazioni real-time e nato in **ambito veicolistico**, anche se oggi è usato (in versioni evolute nel tempo) in applicazioni diverse.
- Il bus CAN originale è stato introdotto da Bosch nel 1986
- Il bus CAN consente la comunicazione tra controllori, sensori ed attuatori con velocità fino a 1Mbit/sec, e ha diversi punti di forza:
  - bassi costi di progettazione e implementazione
  - operatività in condizioni critiche (veicolari, ad esempio, ma anche industriali, quali ambienti con forti vibrazioni e disturbi di tipo elettromagnetico)
  - facilità di configurazione e modifica (soprattutto le sue evoluzioni)
  - rilevamento automatico degli errori e autodiagnostica
- Nell'automazione industriale si usa prevalentemente in due versioni:
  - **CANOPEN** (<http://www.can-cia.org>)
  - **DeviceNet** (<http://www.odva.org>)

# CAN – livello fisico

---

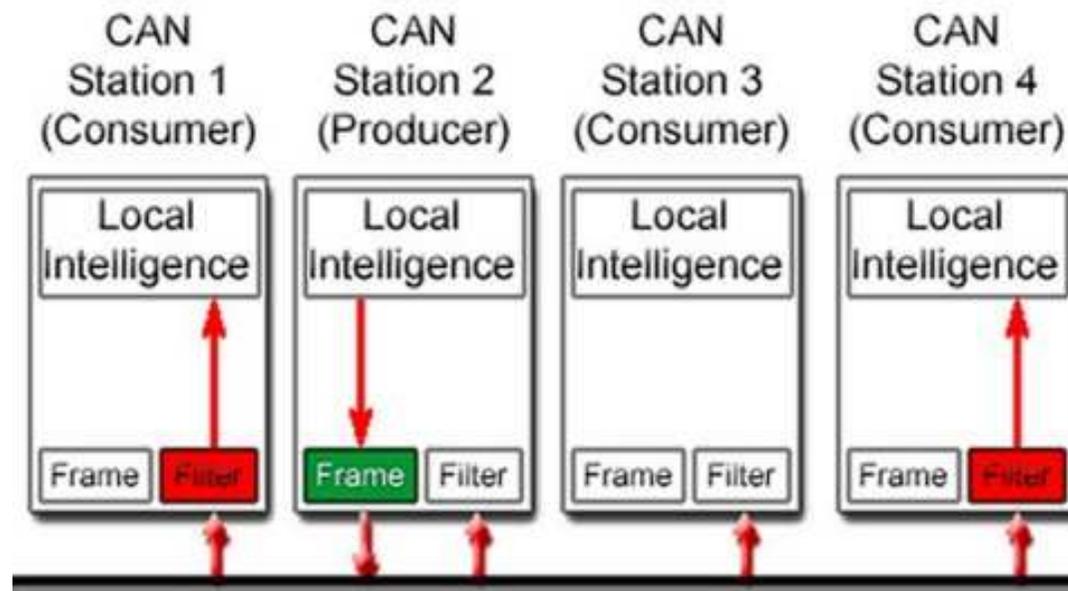


- Si usa un doppino differenziale i cui due conduttori sono detti CAN L (Can Low) e CAN H (Can High).
- Esistono due possibili tipi di trasmissione:
  - Low Speed: 125 kb/s, max 40 m, da 2 a 20 nodi
  - High Speed: da 125 kb/s a 1 Mb/s, max 40 m, da 2 a 30 nodi.
- Per la correzione degli errori, ogni messaggio viene ritrasmesso fino a che tutti i riceventi (e quindi il bus) non segnalano alcun errore.

# CAN – livello collegamento



- La comunicazione è sempre di tipo broadcast.
- I pacchetti non contengono indirizzi ma bensì un identificatore, che definisce anche la priorità del messaggio.
- Ogni nodo “ascolta” tutto il traffico e filtra, elaborandoli, solo i messaggi di suo interesse.



# CAN – accesso al bus

---

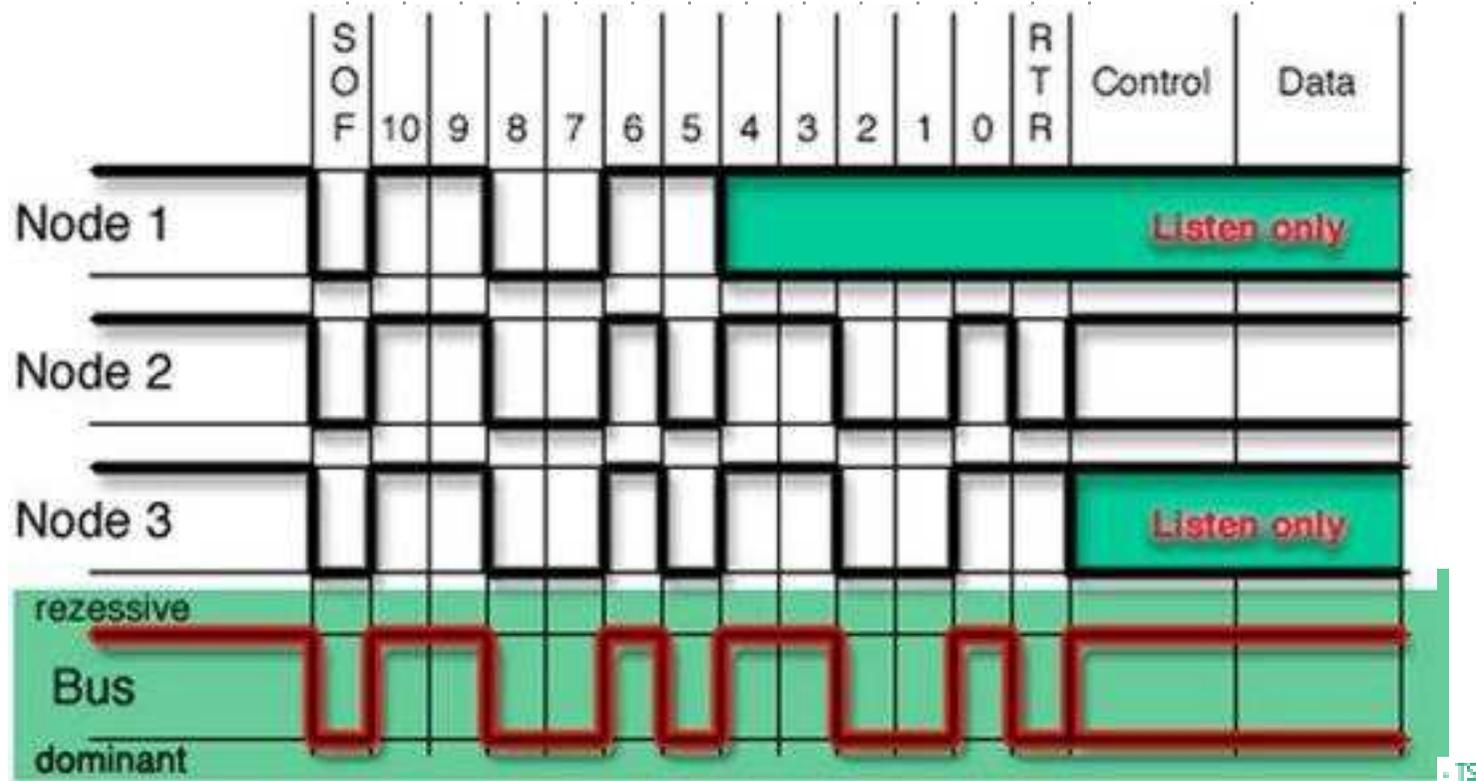


- L'approccio usato dal CAN genera dei conflitti per l'accesso al bus. Questi vengono risolti tramite un sistema di arbitraggio detto "bit-wise", che si basa sul fatto che ai due valori possibili dei bit posti sul bus dai dispositivi si dà l'interpretazione di "dominante" e "recessivo".
- Quando un bit recessivo è inviato sul bus e qualche altro nodo vi ha posto il bit dominante, il nodo che ha emesso il bit recessivo deve ritirarsi. I nodi che non vincono l'arbitraggio diventano automaticamente stazioni riceventi e non ritentano trasmissioni fino a che il bus non è di nuovo libero.
- Questo tipo di arbitraggio è detto **CSMA/CA** (Carrier Sens Multiple Access / Collision Avoidance), ed è del tipo di quello usato da Ethernet.

# CAN – accesso al bus



Esempio di arbitraggio:



# Ethernet industriale

---



- Un'importante tendenza in atto consiste nell'utilizzo di **Ethernet Industriale**, ovvero del protocollo Ethernet in ambiente industriale (automazione e controllo di processo).
- L'enorme diffusione delle schede Ethernet riduce i costi di realizzazione della rete e favorisce l'interoperabilità
- Esistono vari protocolli in cui si adatta Ethernet all'uso in una rete di campo (che richiede prestazioni di tempo reale):
  - **EtherCAT** (<http://www.ethercat.org/>)
  - **EtherNet/IP** ([www.odva.org/default.aspx?tabid=67](http://www.odva.org/default.aspx?tabid=67))
  - **Powerlink** (<http://www.ethernet-powerlink.org/>)
  - **PROFINET** (<http://www.profibus.com/index.php?id=9>)
  - **SERCOS III** (<http://www.sercos.com/>)

# Integrazione con sistemi informativi

---



- I sistemi di controllo sono sempre più spesso visti come una parte del **sistema informativo** (hardware/software) della fabbrica o dell'impianto.
- Il sistema di controllo in senso stretto deve sempre più spesso integrarsi con
  - il monitoraggio dell'impianto
  - la gestione della produzione
  - la gestione degli approvvigionamenti
  - la logistica
- Un modello di riferimento per rappresentare tutte le funzioni dell'azienda, dal marketing alla produzione alla distribuzione, in una visione gerarchica è il **CIM** (Computer Integrated Manufacturing)

# Piramide CIM

Nel modello gerarchico CIM ogni livello manipola e trasferisce informazione ed elabora strategie impartendo comandi ai livelli inferiori.



# Piramide CIM

---



I livelli specificati nella piramide CIM sono i seguenti:

- **Azienda**
  - formata da più stabilimenti
  - sistemi di supporto alle decisioni, pianificazione dei flussi fisici (materiali) e finanziari (investimenti).
- **Stabilimento**
  - vengono integrati i vari comparti: produzione, logistica, amministrazione, manutenzione.
  - è gestito dal sistema informativo aziendale.
- **Reparto**
  - è gestita la base dati della produzione e si coordinano le varie celle per realizzare l'intero processo produttivo.
- **Cella**
  - in una cella viene eseguito un sottoprocesso produttivo completo con varie macchine e sistemi di controllo
- **Sistemi di controllo**
  - funzioni di controllo di macchine e processi
- **Officina (campo)**
  - funzioni di misura e comando sui processi produttivi (sensori, attuatori)

# Componenti di un sistema CIM

---



A livello più basso:

- Attuatori e sensori
- Controllori modulanti (**PID**)
- Controllori logici (**PLC**: Programmable Logic Controller)
  - gestiscono gli assi, le sequenze di operazioni, le protezioni, le interazioni con l'operatore.
- Sistemi di controllo distribuito (**DCS**: Distributed Control Systems)
  - sistemi di controllo che integrano funzioni di calcolo del controllo a livello di loop, controllo logico, monitoraggio, gestione degli allarmi.
- **CNC** (Controllo Numerico Computerizzato)
  - gestiscono macchine multiasse con movimenti interpolati degli assi
- **IPC** (Industrial Personal Computer)
  - PC strutturati per poter operare in ambito industriale, tipicamente con sistemi operativi real time

# Componenti di un sistema CIM

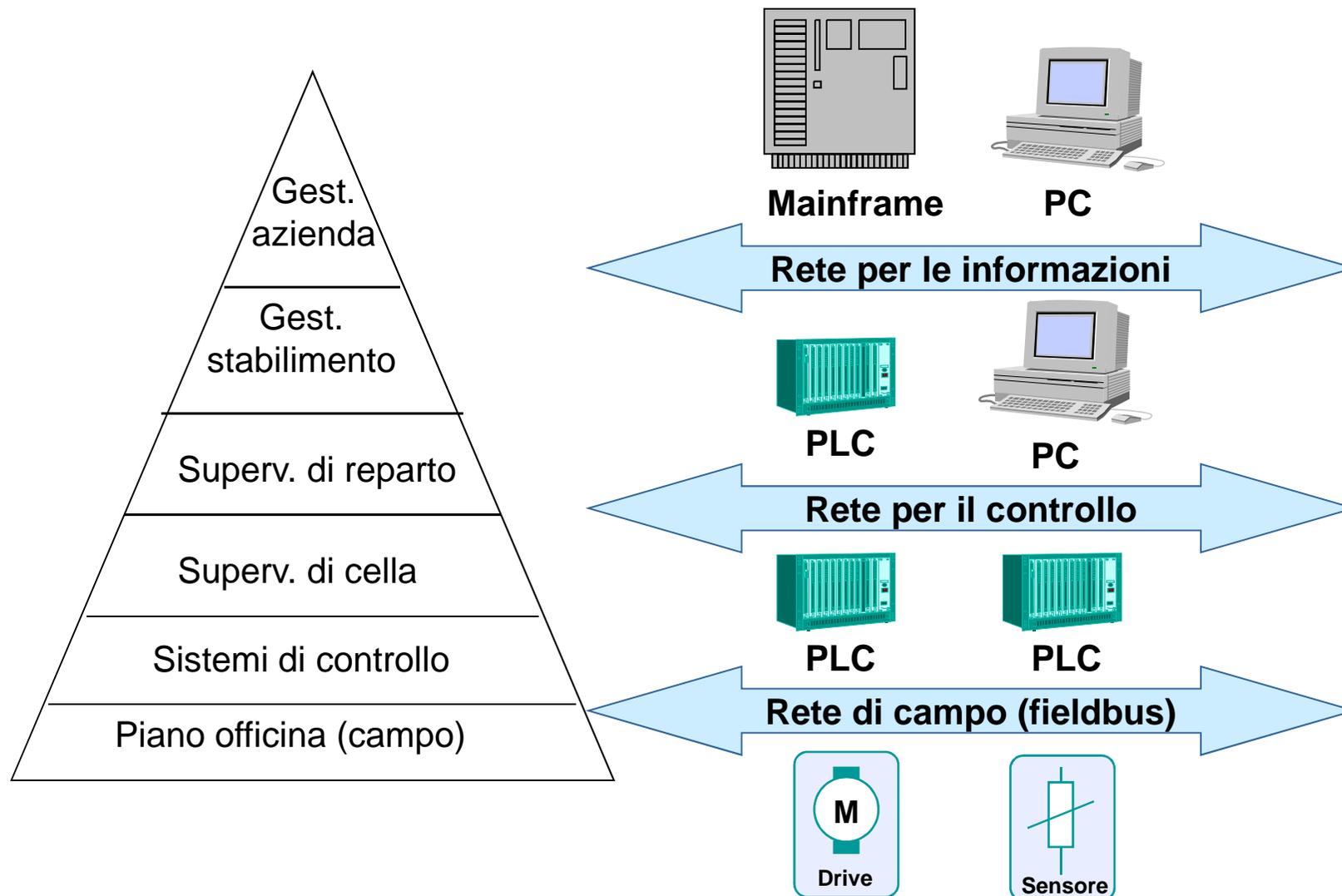
---



A livello più alto:

- **SCADA** (Supervisory Control And Data Acquisition)
  - sistemi che supportano le funzioni di acquisizione ed archiviazione dati, interfaccia uomo-macchina (con sinottici statici e dinamici), grafici, monitoraggio e gestione allarmi, scambio dati con i pacchetti software commerciali più diffusi sui PC, come i fogli elettronici.
- Sistemi **CAD** (Computer Aided Design)
  - software che permette la progettazione assistita al calcolatore
- Sistemi **CAE** (Computer Aided Engineering)
  - sistemi software che supportano la progettazione anche nelle fasi di analisi e simulazione
- Sistema **ERP** (Enterprise Resource Planning)
  - sistema applicativo gestionale che riunifica la gestione di tutte le attività di un'impresa (amministrazione, finanza, reporting, pianificazione, produzione, manutenzione impianti, logistica, controllo qualità...)

# Sistema CIM e reti



# Sistema CIM e reti

---



## ■ Reti per le informazioni

- collegano i sistemi informativi di alto livello con altri elementi informativi di azienda (livelli 4, 5 e 6 della piramide CIM)
- non vi sono specifiche di tempo reale
- le informazioni sono di tipo complesso (file, ecc..)

## ■ Reti per il controllo

- collegano i dispositivi dedicati al controllo con quelli di supervisione (livelli CIM 2, 3 e 4)
- vi sono specifiche di correttezza e vincoli temporali
- le informazioni sono di tipo non molto complesso
- si tratta in genere di reti proprietarie

## ■ Reti di campo (fieldbus)

- collegano i controllori (modulanti e logici) con sensori ed attuatori dotati di interfaccia digitale
- specifiche stringenti di tempo reale
- le informazioni sono di tipo semplice

# Che cos'è un PLC?

---

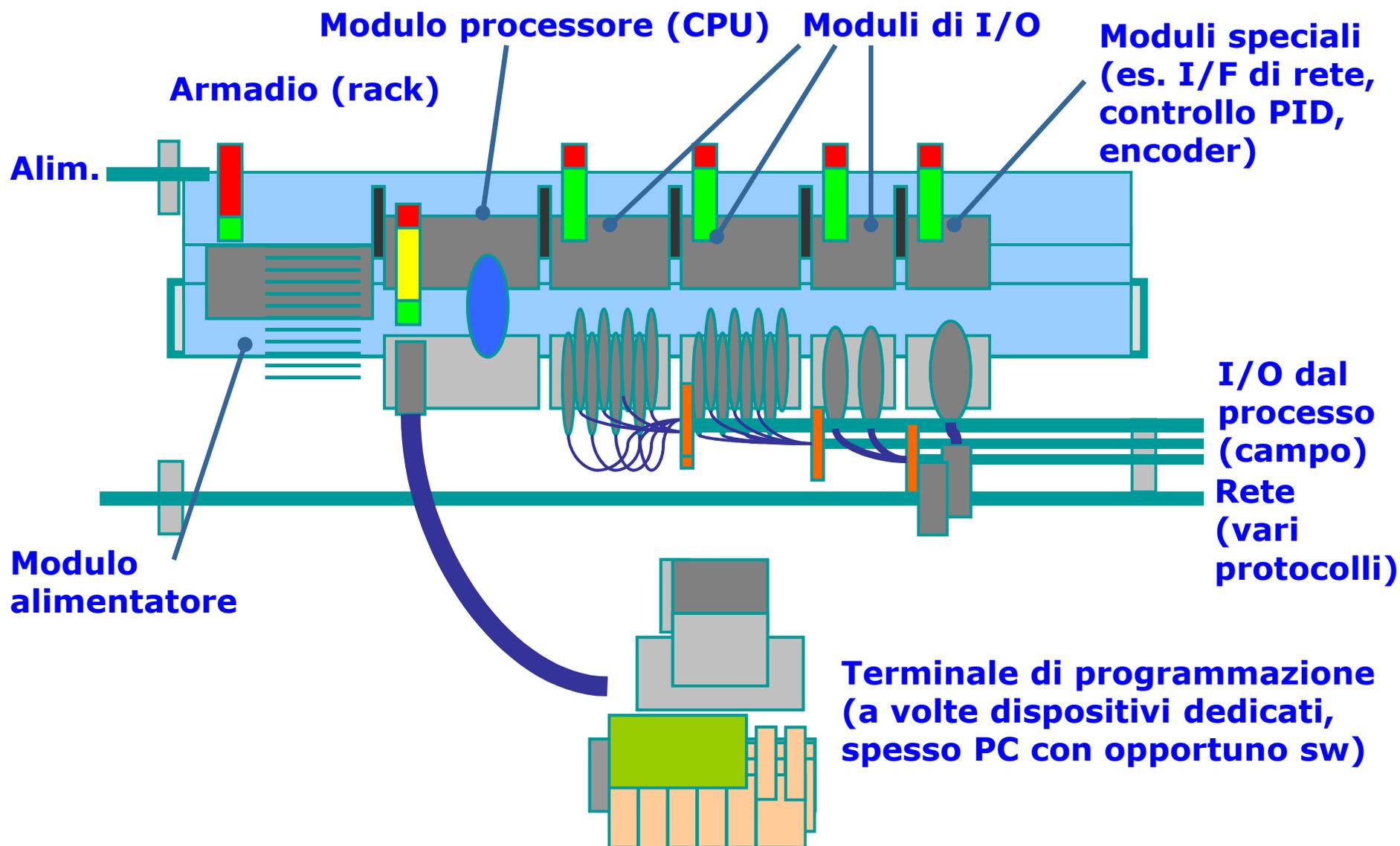
Un **PLC** (Programmable Logic Controller) è un dispositivo che fu introdotto nell'automazione per sostituire i circuiti sequenziali a relay, sostituendoli con un dispositivo in grado di acquisire ingressi ed eseguire su di essi operazioni logiche previa programmazione.

La diffusione dei PLC in automazione è enorme: essi infatti realizzano il “**controllo logico**” che insieme al controllo modulante svolge funzioni di basso livello essenziali per il processo produttivo.





# Componenti fondamentali



# Ciclo di copia massiva

---



- Il PLC esegue periodicamente il **ciclo di copia massiva degli ingressi e delle uscite**, articolato nei seguenti passi:
  - Lettura degli ingressi fisici e aggiornamento coi valori così ottenuti di un'area specifica della memoria (registro immagine degli ingressi)
  - Esecuzione sequenziale del programma utente, che pone i risultati in un'altra area di memoria (registro immagine delle uscite)
  - Esecuzione dei programmi di gestione del sistema (ad esempio di diagnostica)
  - Scrittura sulle uscite fisiche dei valori corrispondenti conservati nell'area di memoria
- L'esecuzione è quindi ciclica: il tempo che il PLC impiega per una singola elaborazione viene chiamato **tempo di ciclo** (in genere da 10 a 100 ms).
- Il PLC non ha modo di accorgersi di variazioni degli ingressi durante il ciclo.
- Inoltre un ingresso non viene acquisito con un intervallo di campionamento costante.

# Linguaggi di programmazione

---



La normativa IEC 1131-3 definisce cinque linguaggi di programmazione per i PLC, di cui tre grafici e due testuali.

## Linguaggi grafici:

diagramma funzionale sequenziale (**SFC**, **S**quential **F**unctional **C**hart),

linguaggio a contatti (**LD**, **L**adder **D**iagram),

diagramma a blocchi funzionali (**FBD**, **F**unction **B**lock **D**iagram).

## Linguaggi testuali:

lista di istruzioni (**IL**, **I**nstruction **L**ist),

testo strutturato (**ST**, **S**tructured **T**ext).

Vi sono oggi diversi ambienti di sviluppo che supportano più di un linguaggio di programmazione.

# Ladder diagram

---

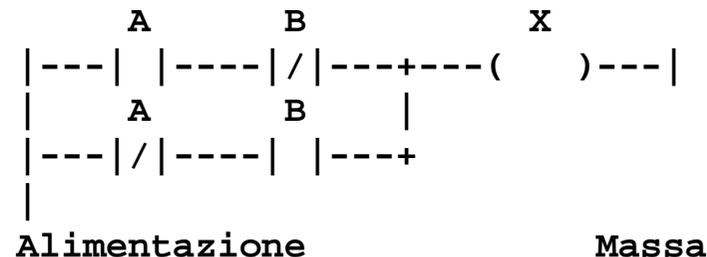


- Il Linguaggio a Contatti o Diagramma a Scala ([Ladder Diagram](#), LD) è derivato dai disegni dei sistemi di controllo realizzati con relé elettromeccanici
- Si basa sui concetti di contatto e bobina ed è stato inizialmente pensato per funzioni di logica binaria; poi è stato esteso per trattare anche numeri interi e/o reali.
- È un linguaggio di basso livello e poco strutturato, non molto adatto a sistemi complessi. Tuttavia è importante perché è stato il primo linguaggio grafico per PLC, è presente in tutti i PLC industriali ed è uno standard di fatto del mercato americano.

# Ladder diagram: concetti fondamentali



- Un ladder diagram si compone di:
  - due linee verticali dette “**montanti**”: il montante di sinistra si intende connesso all'alimentazione, quello di destra a massa
  - alcuni collegamenti orizzontali tra i montanti, detti “**pioli**” o “**rung**” che contengono a sinistra dei **contatti** e a destra delle **bobine**
  - la “corrente” può fluire solo da sinistra verso destra
  - a ciascun contatto e a ciascuna bobina sono associate delle variabili logiche
  - i pioli vengono esplorati dal primo in alto fino all'ultimo in basso
  - se c'è continuità elettrica tra il montante di sinistra e la bobina, la variabile logica associata alla bobina risulta 1 (true), altrimenti è 0 (false)

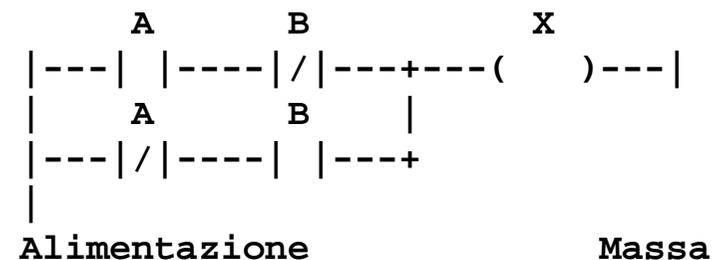


# Ladder diagram: elementi fondamentali



- | |--: **contatto normalmente aperto** (se il bit associato è 1 il contatto è chiuso)
- | / |-- : **contatto normalmente chiuso** (se il bit associato è 1 il contatto è aperto)
- ( )--: **bobina normale** (se alimentata, assegna al bit associato il valore 1)
- (L)--: **bobina latch** (se alimentata, assegna al bit associato il valore 1, che rimane tale finché non si alimenta una bobina unlatch associata allo stesso bit)
- (U)--: **bobina unlatch** (se alimentata, assegna al bit associato il valore 0, che rimane tale finché non si alimenta una bobina latch associata allo stesso bit)
- |P|--: contatto a **riconoscimento di fronte positivo** (se il bit associato passa da 0 a 1 il contatto si chiude per un ciclo)
- |N|--: contatto a **riconoscimento di fronte negativo** (se il bit associato passa da 1 a 0 il contatto si chiude per un ciclo)

Esempio di codice LD (funzione  $X = A \text{ xor } B$ )





# Ladder diagram: altri elementi

## Temporizzatore (normale):



- Se al temporizzatore giunge corrente (da sinistra) esso conta il tempo fino a raggiungere la durata impostata
- A questo punto la variabile NomeTemp va a 1 e così rimane fino al reset del temporizzatore, che si ha quando cessa la continuità elettrica verso di esso.
- In ogni altro caso, NomeTemp vale 0.

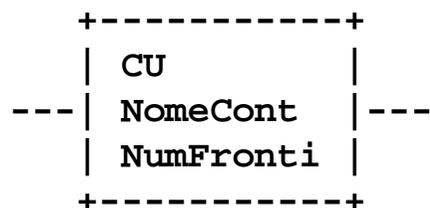
## Temporizzatore (a ritenuta):



- Analogo al precedente
- Quando cessa la continuità elettrica verso di esso non si resetta, bensì ferma il conteggio del tempo al valore raggiunto in quel momento.
- Quando la continuità ritorna, il conteggio riprende quindi dal valore raggiunto in precedenza
- Per resettarlo, si usa un apposito comando

NomeTemp  
--- (RES) ---

## Contatore:



- Analogo al temporizzatore a ritenuta
- Vengono contati i fronti di salita, da 0 a 1, dell'ingresso, fino al numero NumFronti
- Per resettarlo, si usa un apposito comando:

NomeCont  
--- (RES) ---

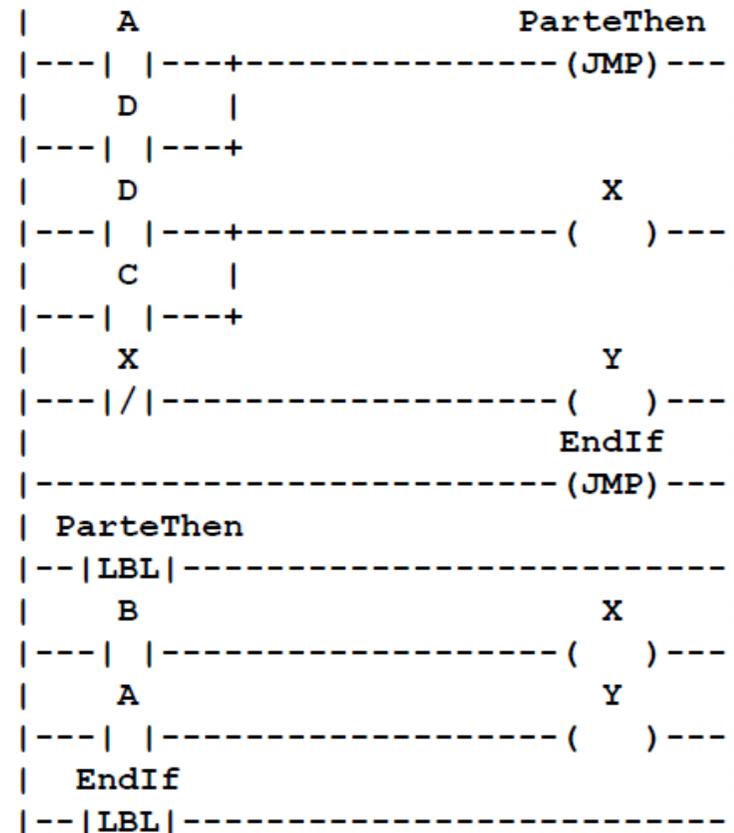


# Ladder diagram: salti

- (JMP)--- Se la bobina **JMP** è alimentata, il PLC salta al rung successivo a quello che contiene il solo elemento **LBL**
- |LBL|---

Esempio di codice LD:

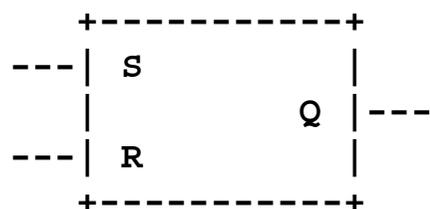
```
if (A or D) {  
    X = B; Y = A;}  
else {  
    X = D or C; Y = not X;}
```





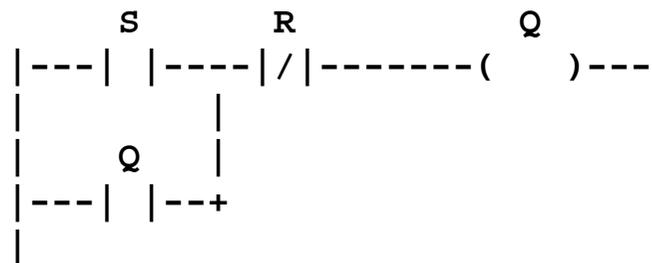
# Ladder diagram: esempio

## Flip-Flop SET RESET



L'uscita Q viene posta a 1 se l'ingresso S (Set) passa a 1 e tale rimane finché non viene posto a 1 l'ingresso R (Reset) che porta a 0 l'uscita (l'ingresso di RESET è prioritario)

$$Q = (\text{NOT } R) \text{ AND } (Q \text{ OR } S)$$



Per esempio questo programma potrebbe costituire la [logica di abilitazione di un motore](#):

- S: interruttore di abilitazione
- R: interruttore di disabilitazione
- Q: motore in moto

# Sistemi real time

---



- Un sistema di elaborazione **real time** (ovvero in tempo reale) deve essere in grado di eseguire tutti i suoi compiti senza violare vincoli temporali specificati.
- Deve quindi rispondere in un modo certo ed entro tempi fissati a eventi esterni, prevedibili o no.
- La correttezza complessiva di un'operazione dipende quindi non solo dalla correttezza logica ma anche dal momento in cui l'operazione è eseguita

## Esempio

I sistemi di controllo devono reagire entro tempi rigidamente vincolati a certi segnali di ingresso, pena danneggiamenti anche gravi dell'impianto controllato o dell'ambiente in cui l'impianto opera.

# Sistemi real time

---



- Tempo reale non vuol dire necessariamente veloce: il fallimento nel calcolare una risposta entro un certo tempo può essere altrettanto scorretto, e avere effetti altrettanto gravi, che calcolarla prima del tempo dovuto.
- Si può quindi dire che tempo reale non vuol dire essere veloci, ma essere sempre **puntuali**.
- A volte si definiscono “in tempo reale” sistemi che in realtà non lo sono (come i sistemi interattivi in tempo reale).

# Hard real time e soft real time

---



I sistemi real time sono classificati in base alla conseguenza del mancato soddisfacimento dei vincoli temporali:

- **Hard real time:**
  - La classe dei sistemi real time per cui il fallimento nel rispettare i vincoli temporali (spesso molto stringenti) può portare a gravi danni o addirittura disastri
  - Esempi: sistemi di controllo, sistemi medicali
  
- **Soft real time:**
  - La classe dei sistemi real time per cui il fallimento nel rispettare i vincoli temporali (più o meno stringenti) non porta a danni immediati, ma al più a una diminuzione dei benefici, che sono decrescenti col tempo
  - Esempi: trasmissioni audio/video, aggiornamento di display

# Requisiti hardware di un sistema real time

---



Il requisito di tempo reale richiede, dal punto di vista hardware:

- L'utilizzo di uno o più processori di adeguata velocità di elaborazione
- Che il tempo di esecuzione delle istruzioni sia noto (almeno nei valori massimi)
- Che l'accesso alla memoria ed ai dispositivi di I/O sia veloce, affidabile e deterministico
- La garanzia di una tempificazione di riferimento certa
- La presenza di funzioni di autodiagnostica
- La presenza di ridondanze strutturali, in modo da poter continuare ad operare anche in presenza di malfunzionamenti

# Sistemi operativi real time

---



- Se l'applicazione di controllo è molto semplice, il modo in cui i vari compiti (*task*) vengono eseguiti in parallelo e in cui sono gestite le risorse è totalmente determinato dal programma utente
- In tutti gli altri casi il dispositivo di controllo dovrà prevedere un (minimo di) **sistema operativo** che si occupi almeno della pianificazione dell'esecuzione dei processi (**scheduling**) e della gestione della comunicazione tra i processi
- Il requisito di tempo reale si traduce quindi in requisiti sul sistema operativo

# Requisiti di un sistema operativo real time

---



Un sistema operativo real time deve avere le seguenti caratteristiche:

- Avere una pianificazione dell'esecuzione che preveda un meccanismo di assegnazione di priorità ai processi
- Essere **multitasking pre-emptive**, ossia essere in grado di interrompere in qualsiasi istante un processo per trasferire le risorse ad un processo più prioritario
- Supportare il cambio di contesto e l'allocazione della memoria
- Evitare situazioni di stallo (**deadlock**)
- Realizzare un meccanismo di sincronizzazione e comunicazione tra processi
- Gestire le interruzioni (interrupt)

# Sistemi operativi real time

---



I sistemi operativi real time presentano difficoltà di progettazione e verifica a posteriori.

- Il progetto è complicato: operazioni ritenute sicure nella pratica di progetto possono non essere più corrette per i sistemi operativi real time
- Il sistema implementato non può spesso essere provato sul campo (si pensi al controllore di una centrale nucleare): serve un simulatore dell'ambiente;
- È molto importante che il loro sviluppo sia fin dall'inizio rigoroso, basato su tecniche precise e assistito da strumenti opportuni.
- Occorre fare uso di metodologie consolidate di ingegneria del software.

# Sistemi operativi real time

---



Alcuni sistemi operativi real time di interesse sono i seguenti:

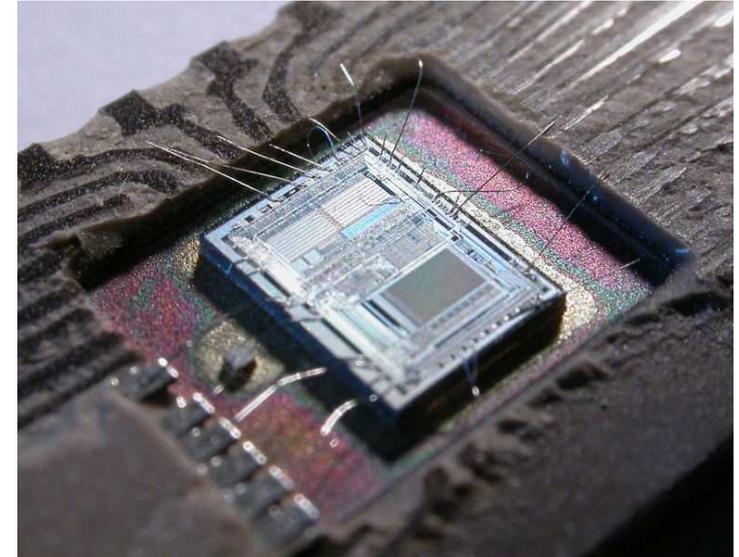
- **Proprietari:**
  - QNX
  - VxWorks
  - Windows CE
  
- **Open source:**
  - RTAI Linux
  - Xenomai

# Sistemi embedded



Un **sistema embedded** :

- È inserito in altri prodotti (es. in automobile)
- È “single purpose”: fa una cosa sola in modo efficiente ed economico (contro i PC che sono “general purpose”)
- Avendo dei compiti noti già durante lo sviluppo può essere progettato con hardware/software specificamente studiati per la tale applicazione.
- L’hardware può essere ridotto ai minimi termini per ridurre lo spazio occupato, riducendo così anche i consumi, i tempi di elaborazione (maggiore efficienza) e il costo di fabbricazione.
- Inoltre l'esecuzione del software è spesso in real-time per permettere un controllo deterministico dei tempi di esecuzione
- Ha un’interfaccia utente primitiva o inesistente



Sono sistemi embedded quelli installati nei telefonini, nelle automobili, negli elettrodomestici, nei ricevitori GPS, nei satelliti, nei giocattoli, ecc.

# Sistemi embedded

---



## Alcune caratteristiche dei sistemi embedded

- Dal punto di vista software sono dotati di sistema operativo real time
- Hanno piccola occupazione di memoria (tipico 1÷2 KB di memoria RAM/ROM), e devono essere eseguiti da CPU di piccola potenza e consumo
- Devono funzionare sempre senza interventi manuali
- È bene che non abbiano parti in movimento, che possono rompersi, consumano molta energia, sono lente, richiedono driver complessi, occupano spazio
- Devono gestire e recuperare, eventualmente con funzionalità ridotte, situazioni di errore
- Devono ripartire (reboot) automaticamente in caso di arresto, eventualmente in configurazioni di sicurezza e “istantaneamente”
- Devono fare autotest di funzionamento (watchdog, ecc)