

# Introduzione ai sistemi embedded

Braione Pietro, revisione Domenico G. Sorrenti

Sistemi Embedded

Anno accademico 2019/20

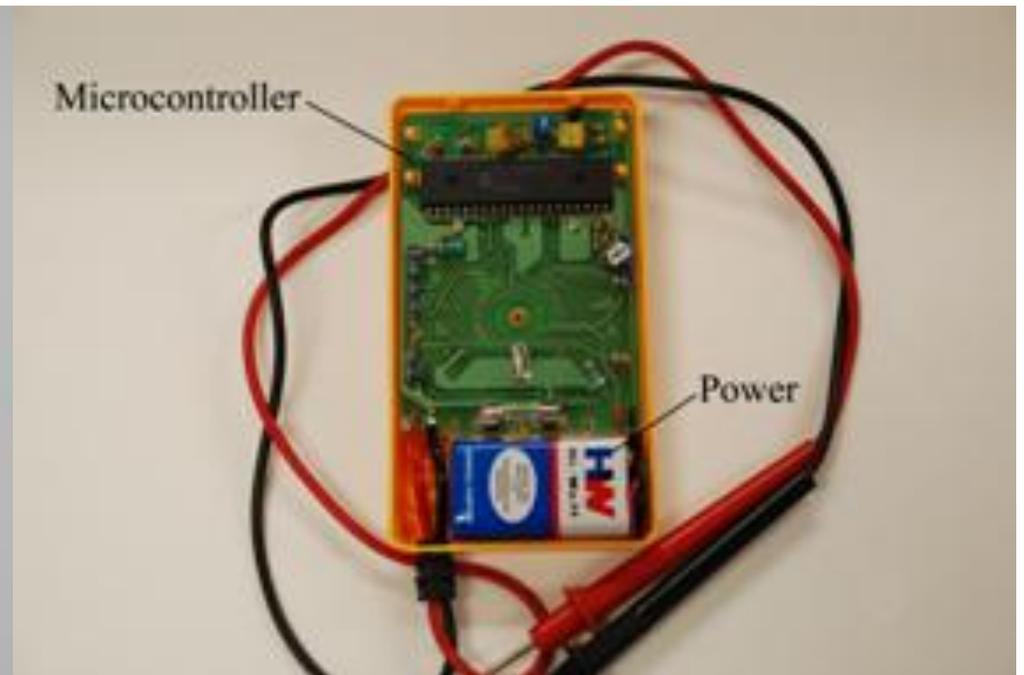
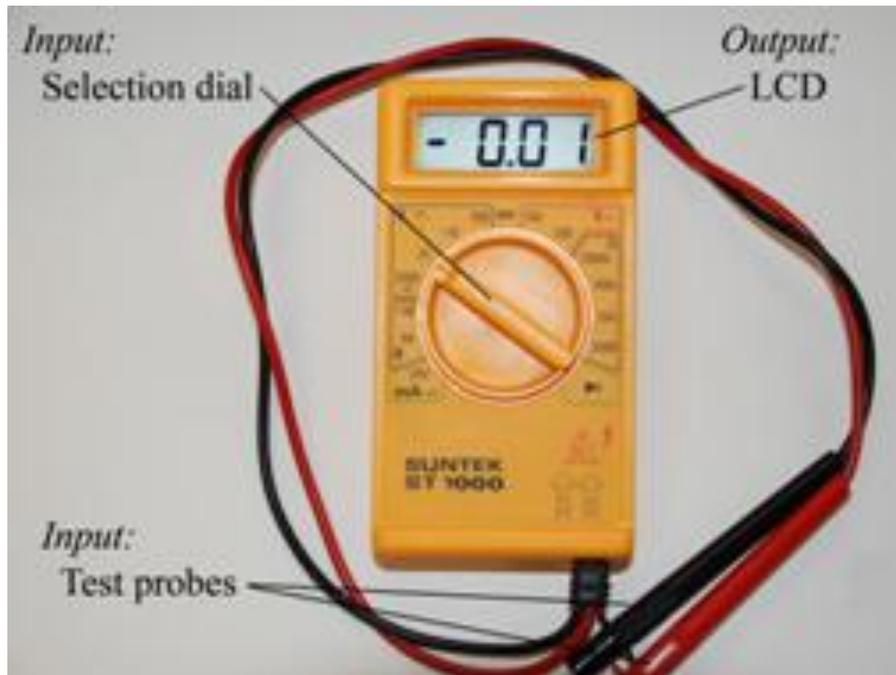
# Sistemi embedded

- Cosa sono?
- Perché meritano di essere studiati?

# Un multimetro



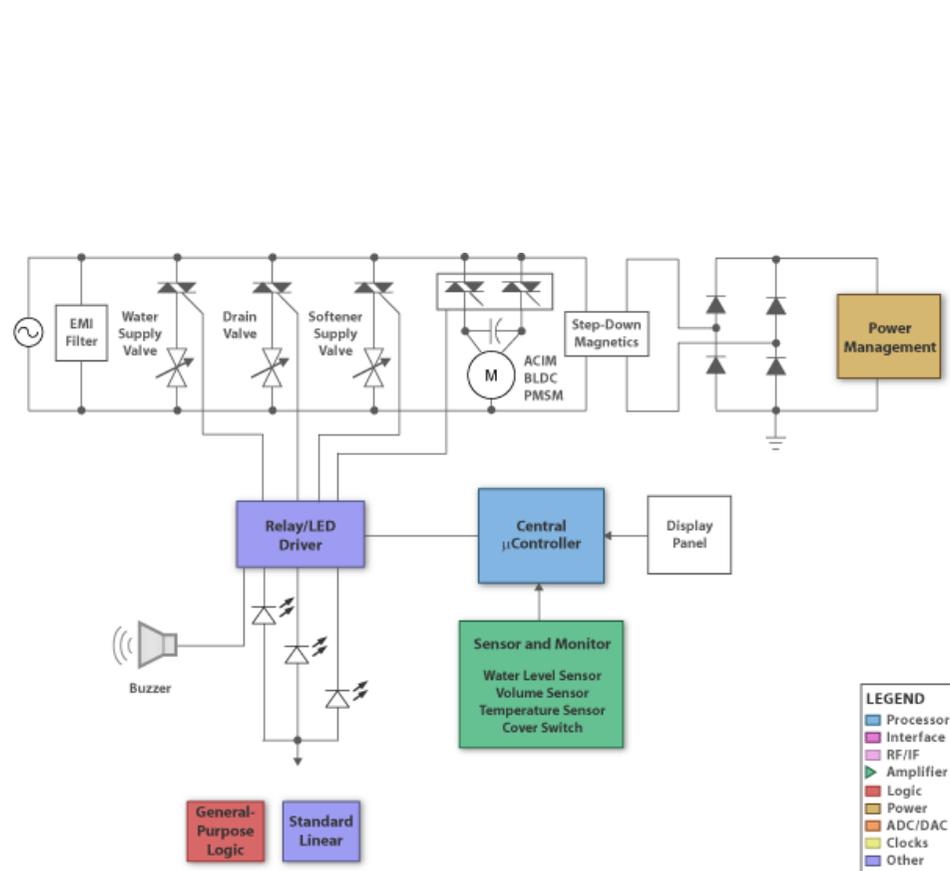
# Un multimetro



# Una lavatrice



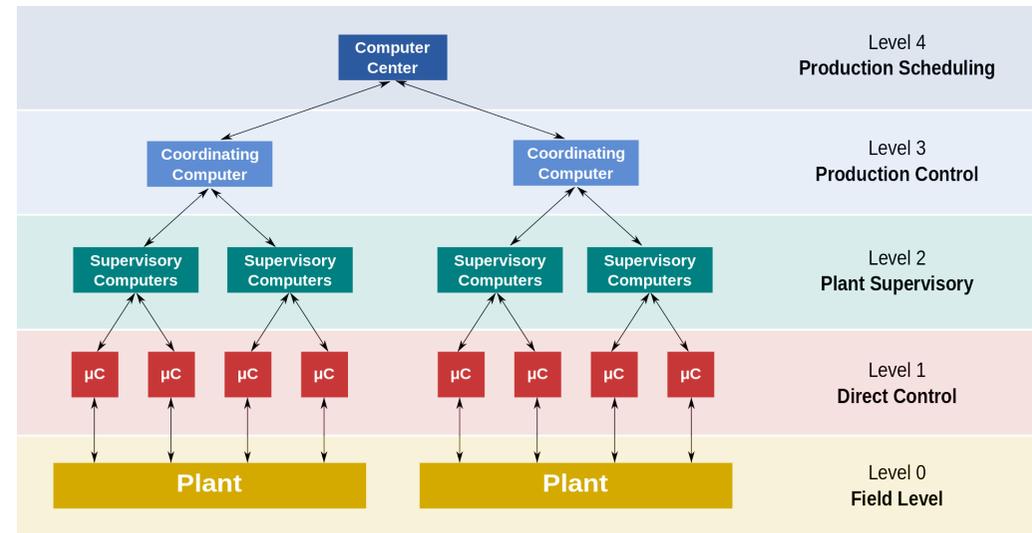
# Una lavatrice



# Un impianto industriale



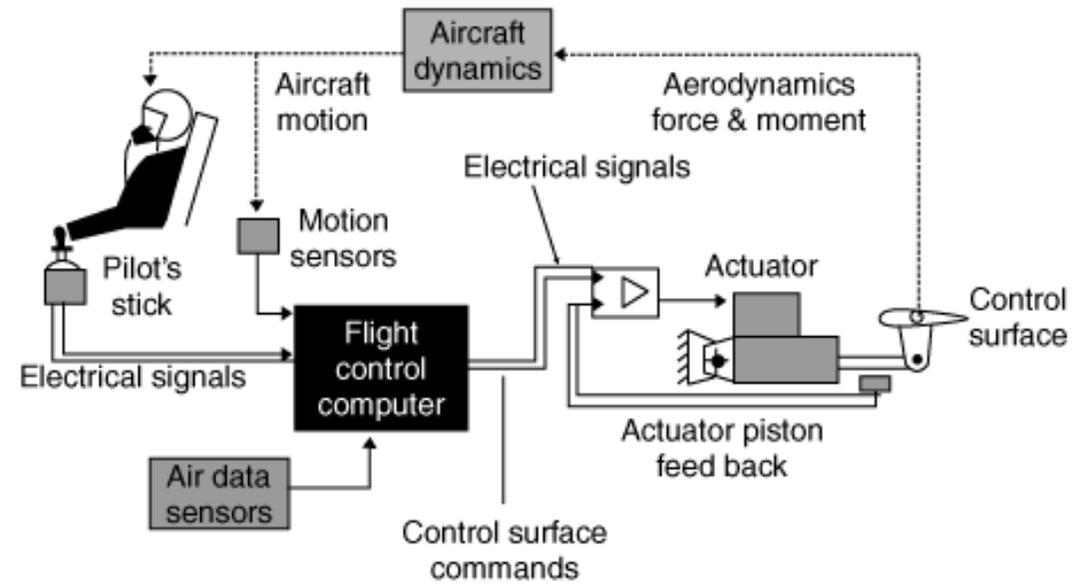
# Un impianto industriale



Un aereo passeggeri



# Un aereo passeggeri



# Cosa hanno in comune?

- Sono sistemi con una rilevante componente fisica
- Che contengono (uno o più) calcolatori
- I quali elaborano informazione
- Che è prodotta/consumata dal processo fisico

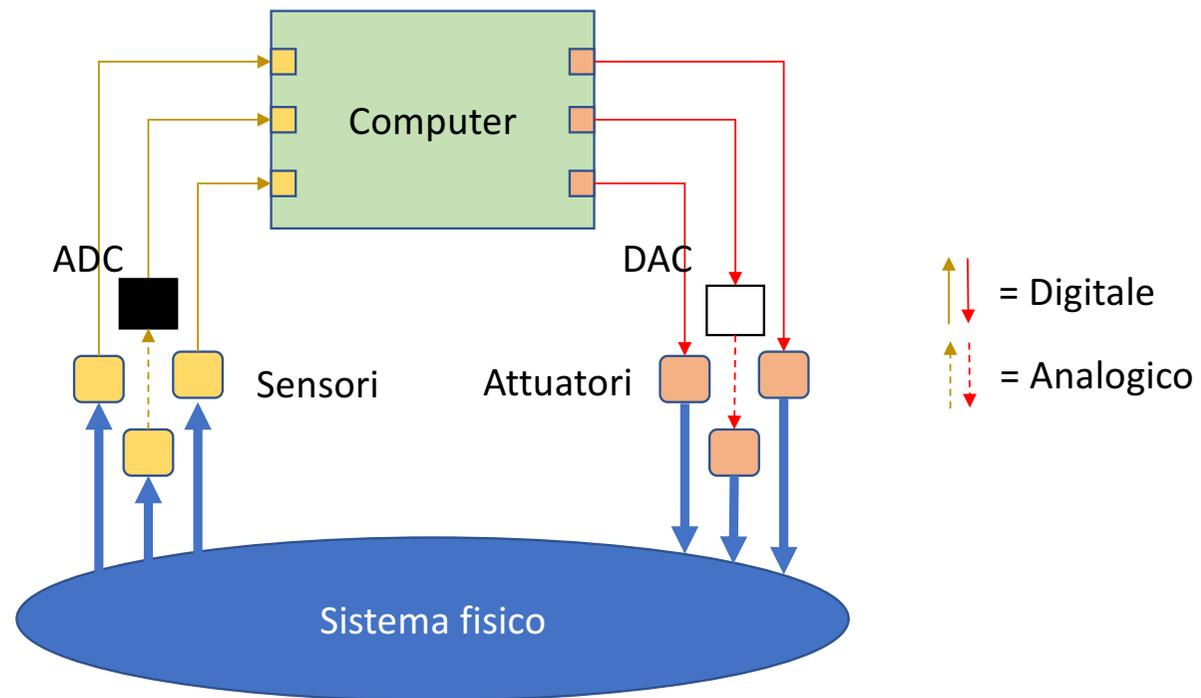
Questi sistemi vengono detti  
**sistemi embedded o sistemi cyber-fisici**

*“electronic programmable sub-systems that are generally an integral part of a larger heterogeneous system” [ART2005].*

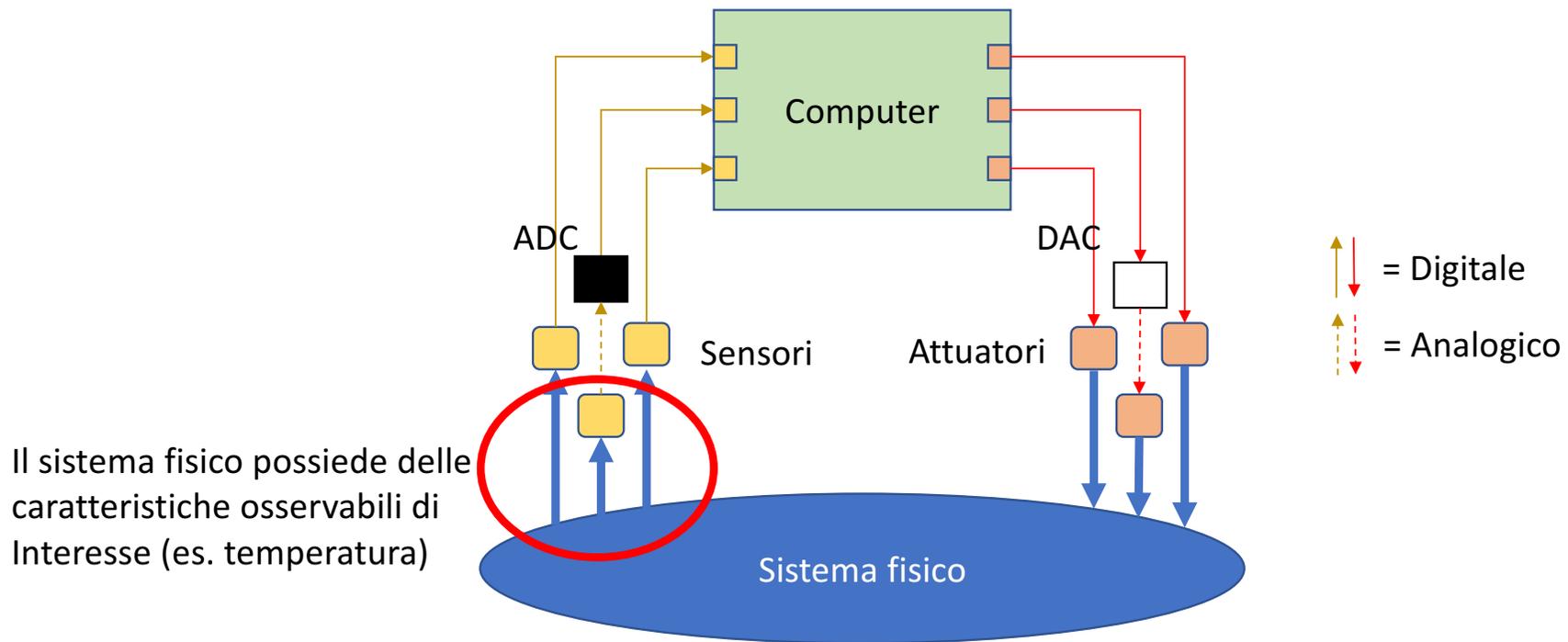
# Caratteristiche salienti

- Il processo fisico influenza la computazione e viceversa (feedback)
- La computazione deve avvenire
  - In maniera **reattiva**: computazione in reazione a input dal processo fisico
  - In maniera **concorrente**: diversi fenomeni del processo fisico che scatenano la computazione reattiva possono avvenire contemporaneamente
  - In **tempo reale**: l'output deve essere prodotto in un tempo compatibile con la dinamica del processo fisico
    - dinamica del processo / costante di tempo,
    - tempo compatibile non è necessariamente un tempo molto breve.

# Struttura

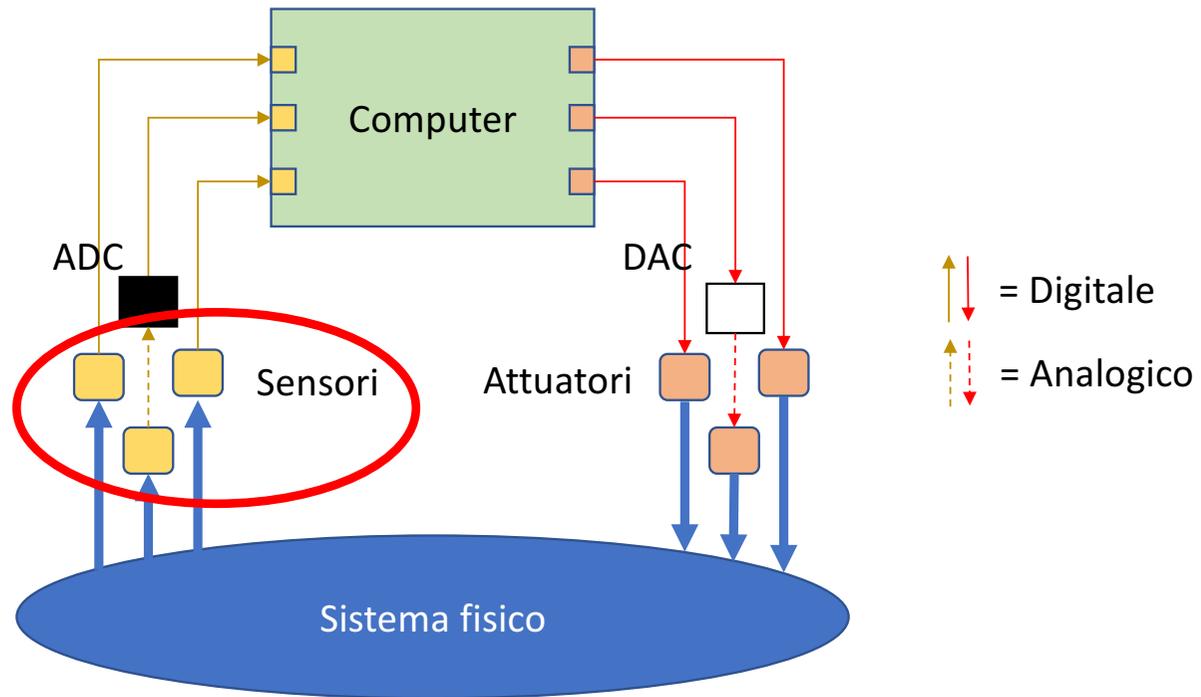


# Struttura



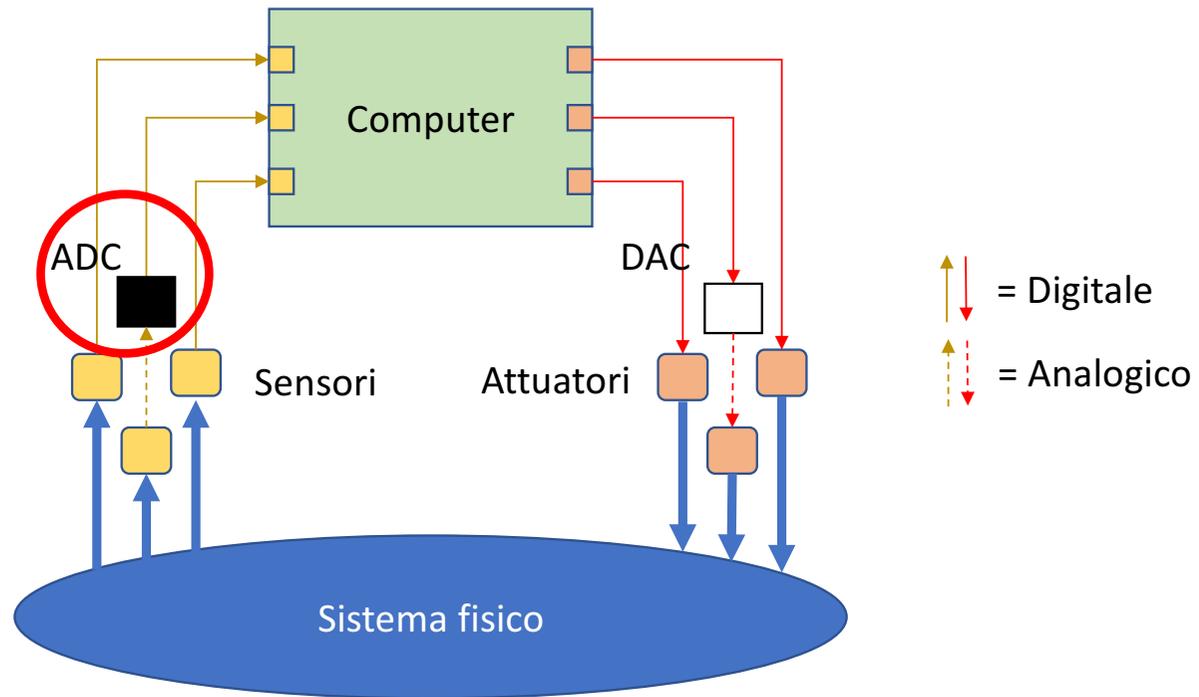
# Struttura

I **sensori** misurano tali caratteristiche e le convertono in segnali digitali o analogici



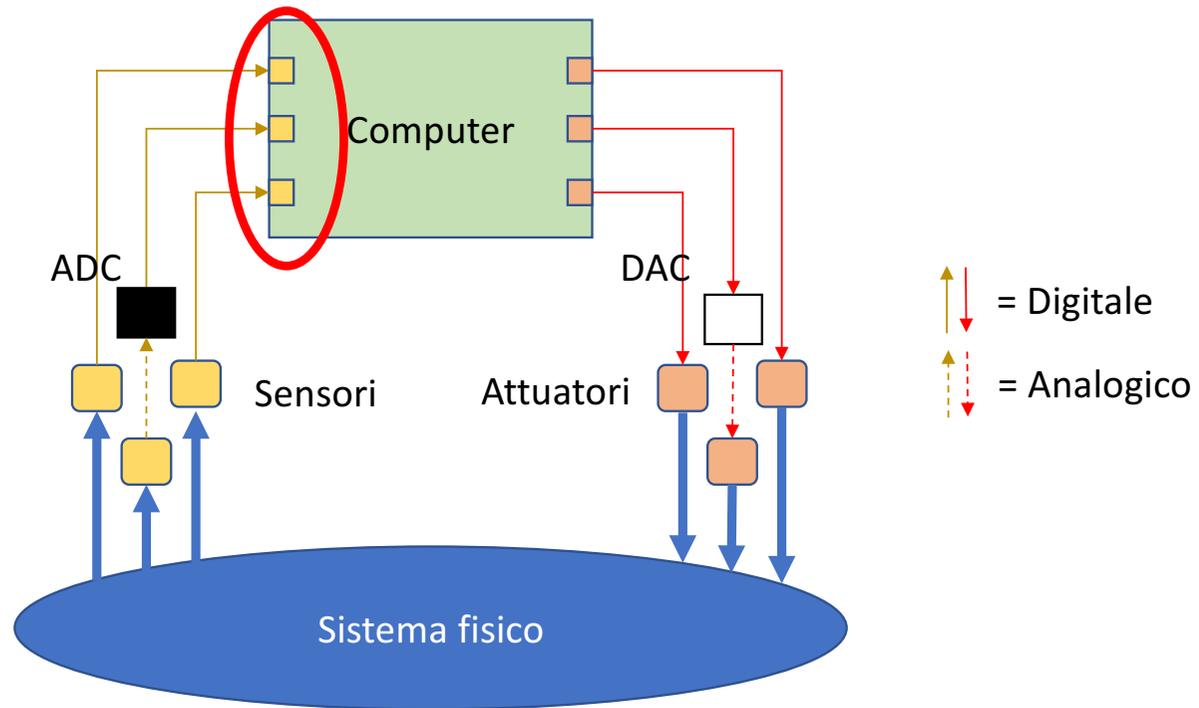
# Struttura

I segnali analogici vengono convertiti in segnali digitali

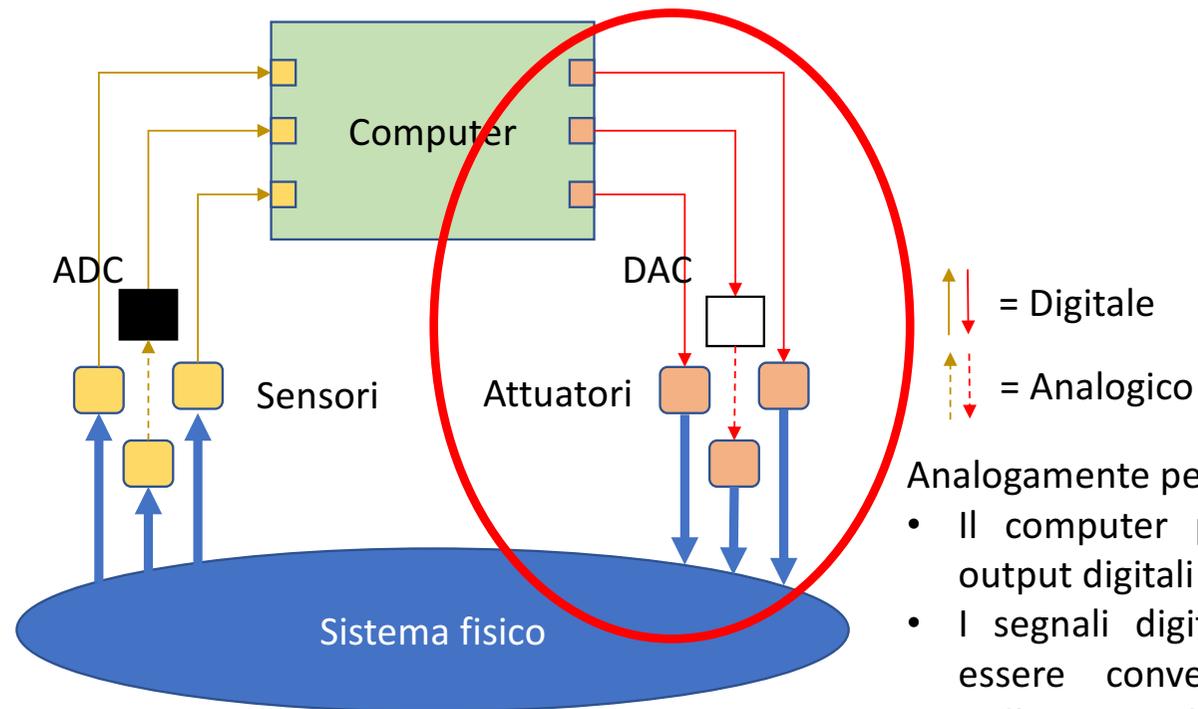


# Struttura

Il computer accetta in input i segnali digitali prodotti



# Struttura



- Analogamente per gli output:
- Il computer produce segnali di output digitali
  - I segnali digitali possono dover essere convertiti in analogici, se l'attuatore lo richiede
  - Gli **attuatori** convertono i segnali in grandezze fisiche (es. coppia), comunque analogiche

# Un mondo di sistemi embedded (1)

- Automotive
  - Controllo trazione e trasmissione
  - Assistenza alla guida
  - Entertainment e comunicazione
- Avionica (circa 50% costo di un aereo)
  - Controllo superfici di volo
  - Navigazione
  - Comunicazione
  - Manutenzione
  - Applicazioni militari (sistemi d'arma, sistemi tattici)

# Un mondo di sistemi embedded (2)

- Automazione industriale
  - Controllo impianti
  - Logistica
- Telecomunicazioni
  - Smartphones
  - Apparat di comunicazione digitale
- Consumer electronics
  - Forni a microonde, lavatrici, lavastoviglie, frigoriferi, televisori...
  - Domotica
- Medicina
  - Pacemakers
  - Radiodiagnostica e radioterapia
  - ...

# Internet of Things e Industria 4.0

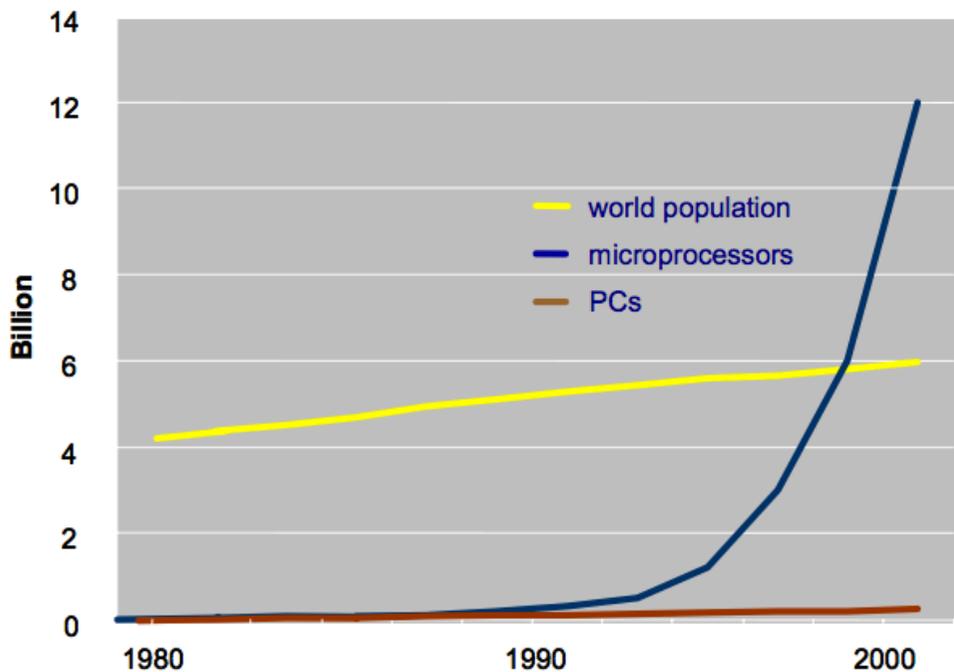
- Convergenza con tecnologie di comunicazione
- Internet of Things (IoT):
  - Sistemi embedded + tecnologie Internet
  - Accesso a sensori ed attuatori via PAN → LAN → MAN → WAN
  - Possibilità per i sistemi embedded di interagire tra di loro (M2M)
  - E con servizi esterni
- Industria 4.0:
  - IoT per i sistemi industriali
  - Integrazione con data analytics e business intelligence

# Il valore dei sistemi embedded

Industry sector	Annual global value	ES value (%)	ES value	EC growth (%)	ES growth (%)
Automotive	800 b€	40%	320 b€	10%	10%
Avionics/Aerospace	750 b€	50%	370 b€	5%	14%
Industrial automation	200 b€	55%	110 b€	5%	7%
Telecommunications	1000 b€	55%	550 b€	9%	15%
Consumer electronics and intelligent homes	300 b€	60%	180 b€	8%	15%
Health and medical systems	130 b€	40%	50 b€	?	18%
<b>Total</b>	<b>3180 b€</b>		<b>1580 b€</b>		

From <https://msdn.microsoft.com/en-us/library/dd129911.aspx>: Estimated total value, through 2006, and growth, through 2010, of major industry sectors using embedded systems  
 (ES = embedded systems, EC = electronic components)

# La pervasività dei sistemi embedded



The worldwide number of microprocessors [Bies2005] as an indicator for the penetration of embedded systems

- 2004: circa 2 processori per persona sulla terra, previsti circa 3 per il 2010 [BuildAR2004]
- 2005: Il 98% dei processori prodotti sono usati in sistemi embedded [EmbC2005].

# Progettare sistemi embedded

- Ormai riconosciuta come una disciplina a sè
  - Non è solo “programmare un computer più piccolo”
  - Ma pone delle sfide ingegneristiche peculiari e complesse
- La complessità è dovuta alla varietà dei domini applicativi
  - Nota: I sistemi embedded sono di norma special-purpose
  - (ma spesso costituiti da componenti general-purpose)
- Ogni tipo di applicazione pone requisiti diversi
- Questi dettano le scelte progettuali

# Tipologie di requisiti

- Efficienza
- Flessibilità
- Affidabilità
  - **Business-critical**: mette a rischio la riuscita del business
  - **Safety-critical**: mette a rischio vite umane
- Vincoli real-time
- Grado di interazione con l'uomo
- Mercato

# Efficienza

- L'**efficienza** è la caratteristica di un sistema che porta a termine il proprio funzionamento con un quantità limitata di risorse
- Alcuni attributi:
  - Peso
  - Dimensioni fisiche
  - Consumo energetico
  - Occupazione di memoria del software
  - Grado di riuso
  - Costo

# Flessibilità

- Molti sistemi embedded real-time sono soggetti ad un insieme di requisiti che evolve nel tempo, dovuti a modifiche nel sistema stesso, nelle richieste del mercato, nell'ambiente in cui il sistema deve operare, etc.
- La capacità del sistema di accomodare soluzioni a nuovi requisiti, diversi da quelli per cui è stato progettato, è chiamata flessibilità.
- Un importante aspetto non-funzionale di un sistema embedded real-time è che il progetto dovrebbe essere abbastanza flessibile da consentire di adattarsi ad evoluzione dei requisiti con un costo minimo.

# Affidabilità

- L'**affidabilità** è la caratteristica di un sistema il cui funzionamento produce gli effetti voluti o scostamenti accettabili da questi
- I sistemi safety-critical richiedono un'affidabilità molto elevata:
  - Controllo centrali nucleari
  - Controllo superfici di volo aerei
  - Impianti di frenata automobili, ABS, airbag...
  - Pacemakers
- Questo perché l'impatto del loro funzionamento sull'ambiente (persone, cose...) è spesso immediato e profondo

# Attributi di affidabilità

- **Correttezza:** il sistema rispetta i requisiti
- **Robustezza:** il sistema si comporta accettabilmente anche in situazioni non specificate nei requisiti
- **Sicurezza (security):** il sistema impedisce usi non autorizzati
- **Innocuità (safety):** il sistema non ha comportamenti pericolosi
- Occorre valutare e dare priorità ai diversi attributi in fase di progettazione e orientare la progettazione in maniera da rispettare tali valutazioni

# Vincoli real-time

- Nei sistemi non embedded il tempo di computazione è essenzialmente una delle dimensioni dell'efficienza
- In certi domini applicativi embedded vi possono essere vincoli real-time
- I vincoli real-time determinano l'affidabilità del sistema
  - Una risposta esatta arrivata in ritardo è una risposta sbagliata
  - Se ABS di un autoveicolo interviene in ritardo compromette la safety

# Da applicazioni a requisiti (1)

- Esempio: forno a microonde
  - Mercato saturo, necessità di minimizzare i costi
  - Dimensioni sull'ordine della decina di centimetri, peso diversi chili
  - Nessun vincolo energetico
  - Limitata affidabilità
  - Interazione con uomo attraverso un'interfaccia semplice (pulsanti e visore a cristalli liquidi)
- Esempio: acceleratore di particelle per radioterapia
  - Costo dell'ordine di svariati milioni di Euro, mercato molto piccolo
  - Dimensioni: diversi metri, bunker + postazione esterna per tecnico/medico
  - Nessun vincolo energetico
  - Safety-critical: occorre evitare sovradosaggio

# Da applicazioni a requisiti (2)

- Esempio: smartphone
  - Costo: dipendente dal mercato di riferimento
  - Dimensioni: pochi centimetri, peso: tra  $10^2$  e  $10^3$  grammi
  - Stringenti vincoli energetici e termici
  - Integra diverse funzionalità (chiamata e piattaforma applicativa general-purpose)
  - L'affidabilità di alcune funzionalità (chiamata telefonica) è business-critical
- Esempio: monitoraggio e controllo grid distribuzione energia elettrica
  - Mercato molto piccolo, sistemi mediamente costosi
  - Dimensioni: intere regioni geografiche
  - Nessun vincolo energetico
  - Da business-critical a safety-critical

# Alcune dimensioni progettuali

- Piattaforma (HW + SW)
  - Microprocessore (MPU)
  - Microcontrollore (MCU)
  - Digital signal processor (DSP)
  - Logica programmabile (PLD, FPGA)
  - ASIC
- Scelta linguaggi, librerie o real-time OS
- Numero di componenti (centralizzazione vs. distribuzione)
- Quantità e tipo di memoria (SRAM, DRAM, ROM, FLASH)
- Velocità di calcolo (frequenza clock CPU)
- Batteria

# Da requisiti a scelte progettuali (1)

- Le piattaforme ASIC sono le più performanti, compatte ed efficienti dal punto di vista energetico, ma le meno flessibili e le più costose
  - Ad es., meno software e quindi meno memoria (spazio occupato, consumo)
  - Potenzialmente più affidabile di una soluzione software
- Costo ASIC:
  - Costo up-front per progetto: elevato, ma può essere ridotto con il riuso (IP)
  - Costo di produzione in serie: molto basso
- La soluzione ASIC può essere considerata in caso di mercati di massa, ma non è praticabile se si vuole poter aggiornare la funzionalità dopo il deployment

# Da requisiti a scelte progettuali (2)

- Linguaggio e librerie/OS:
  - La programmazione «a macchina nuda» (bare-bones) e l'uso del linguaggio macchina può permettere una maggiore efficienza (meno memoria, maggiore velocità)
  - Riduce però la scalabilità ed aumenta il rischio di avere un'affidabilità bassa
- Numero componenti:
  - Una soluzione centralizzata è potenzialmente meno costosa e più affidabile (nessuna sincronizzazione) di una distribuita
  - Ma se il sistema ha dimensioni elevate può essere meno affidabile (rumore sulle linee) e meno efficiente (peso) di una distribuita

# Da requisiti a scelte progettuali (3)

- Memoria:
  - Le memorie ROM e FLASH sono meno costose delle RAM, di solito sono utilizzate per il codice (che non cambia o cambia sporadicamente)
  - Le memorie DRAM sono meno costose e hanno una maggiore densità delle memorie SRAM, ma sono anche più lente e dissipano più energia
  - Maggior memoria permette di realizzare più funzionalità in maniera centralizzata, con possibile risparmio di costo
  - Allo stesso tempo significa maggiore occupazione di area e costo del componente
- Velocità di calcolo:
  - Permette più facilmente di soddisfare i vincoli real-time e di realizzare più funzionalità in maniera centralizzata, con un potenziale risparmio di costo
  - D'altra parte maggiore velocità di calcolo significa maggior costo del componente, minore efficienza, e potenzialmente più peso e volume (dissipazione calore)
- Batteria:
  - Deve fornire sufficiente energia per alimentare tutto il sistema
  - Allo stesso tempo più è capace, maggiore è il suo peso, volume e costo

# Tornando agli esempi... (1)

- Forno a microonde:
  - Principale driver: minimizzazione costi
  - Una soluzione basata su un microcontrollore a 8 bit è sufficiente
- Acceleratore per radioterapia:
  - Principale driver: safety
  - Soluzione distribuita basata su MCU a 32 bit per l'impianto con ridondanza hardware, più postazione standard PC per interfacciarsi a software di treatment planning

## Tornando agli esempi... (2)

- Smartphone:
  - Principali driver: costi, dimensione, peso, efficienza energetica
  - Diversi tipi di dispositivi hardware: ASIC + IP per codec radiofrequenza, DSP per audio/video processing, MPU per applicazioni; possibile SoC integra le diverse funzionalità
- Monitoraggio e controllo grid distribuzione energia elettrica:
  - Principali driver: costi, vincoli real-time, flessibilità, robustezza
  - Sistema fortemente distribuito, con unità locali basate su MPU + sistema operativo real-time e rete di comunicazione; centrali operative basate su standard PC