



[Schema della lezione](#)

[Applicazioni](#)

[Caratteristiche](#)

[Sistemi real-time](#)

SERT'13

E1.1

Lezione E1

Introduzione ai sistemi embedded

Sistemi embedded e real-time

3 ottobre 2012

Marco Cesati

Dipartimento di Ingegneria Civile e Ingegneria Informatica
Università degli Studi di Roma Tor Vergata

Di cosa parliamo in questa lezione?

In questa lezione parliamo in modo informale dei sistemi embedded:

- 1 Applicazioni
- 2 Funzioni tipiche
- 3 Caratteristiche commerciali
- 4 Hardware e software dei sistemi embedded
- 5 Rapporto con i sistemi real-time



[Schema della lezione](#)

[Applicazioni](#)

[Caratteristiche](#)

[Sistemi real-time](#)

SERT'13

E1.2

Cosa è un sistema embedded

Qual è la definizione di sistema embedded?

- Un sistema programmabile che non è pensato per essere riprogrammato dall'utente ⇒ applicazione specializzata
- Interagisce fortemente con l'ambiente circostante ⇒ sistema **immerso** o **integrato**
- **Interfaccia utente** invisibile o comunque non conforme a quella di un normale calcolatore
- Altissima diffusione: sistemi **pervasivi** (*invisible computing* o *disappearing computing*)

Un *sistema embedded* (od anche *sistema immerso* o *integrato*) è la struttura di supporto al funzionamento di una applicazione:

- con un significativo livello di interazione con il mondo esterno
- in cui comandi o modalità operative non sono necessariamente selezionati direttamente da un operatore umano

Il mercato dei sistemi embedded

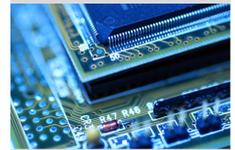
Quanti dispositivi programmabili sono presenti in quest'aula?

Ragionevole stima:

- Meno di una decina di processori per computer ad uso generale
- Più di 100 microprocessori per sistemi embedded (telefoni cellulari, orologi, chiavette memoria flash, proiettore, microcontrollori entro i computer, impianto di amplificazione, sensori di fumo, sistema d'allarme, ...)

Statistiche del World Trade Office:

- Attualmente il 98% dei dispositivi programmabili sono **embedded**
- Nel 2000 il numero di sistemi embedded: circa 8 miliardi
- Nel 2010 il numero di sistemi embedded: circa 16 miliardi (3 per ogni essere umano)
- Nel 2020 il numero di sistemi embedded: circa 40 miliardi (proiezione prudente)



Diffusione dei sistemi embedded

- **Automotive**: maggiore settore d'impiego dei sistemi embedded
- **Avionica**: si stima che circa la metà del costo di un aereo sia dovuta ai sistemi embedded; di tali costi, l'85% è dovuto allo sviluppo del software
- **Automazione industriale**: grande diffusione di sistemi embedded che utilizzano tecniche di controllo remoto, sistemi distribuiti, tecnologie wireless, RFID, ...
- **Telecomunicazioni**: sistemi embedded impiegati come calcolatori elettronici ottimizzati per lo svolgimento di particolari funzioni: apparati di rete, telefoni cellulari evoluti, ...
- **Elettronica di consumo e domotica**: apparati di videoregistrazione, televisori, grandi e piccoli elettrodomestici, giocattoli, ...
- **Salute e apparati medicali**: apparati elettro-medicali, pacemaker, sistemi di monitoraggio, ...

Scenario d'esempio: automotive

Il controllo degli autoveicoli è oggi una delle principali applicazioni dei sistemi embedded

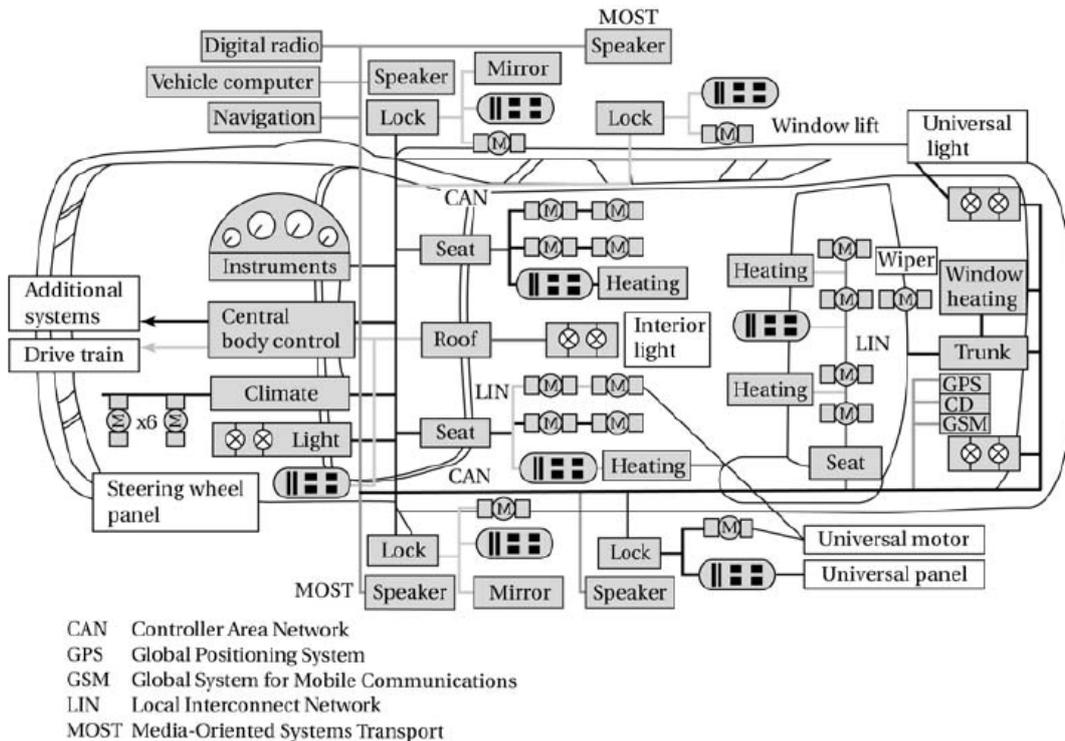
Progressivamente vengono eliminati i comandi diretti meccanici e idraulici e sostituiti da sistemi assistiti con collegamenti elettrici (**X-by-Wire**)

In una singola autovettura di fascia alta si possono trovare centinaia di sistemi embedded:

- chiamati **ECU** (**E**lectronic **C**ontrol **U**nit)
- fisicamente distribuiti nel veicolo
- in passato collegati da fasci di cavi (**harness**), pesanti decine di chilogrammi e difficili da mantenere
- oggi collegati da diverse reti interne (**intravehicle networks**) appositamente progettate
- nella maggioranza dei casi con stringenti requisiti di "qualità del servizio"



Scenario d'esempio: automotive (2)



Calcolatori elettronici e sistemi embedded

Un calcolatore elettronico è innanzi tutto *versatile*: può essere programmato per adattarsi ai più diversi ambiti applicativi

La chiave di volta per ottenere la versatilità è quella di dotare il calcolatore della maggiore quantità possibile di risorse hardware, compatibilmente con opportuni criteri di costo

Viceversa, il **sistema embedded** è una macchina dedicata allo svolgimento di un singolo compito, ovvero di una classe molto ristretta di compiti

Grazie alla specificità dell'applicazione eseguita, il **sistema embedded** può essere progettato in modo da ottimizzare particolari criteri di costo e prestazioni

I calcolatori elettronici ad uso generale seguono standard e architetture di riferimento; viceversa è difficile definire standard per i **sistemi embedded** in quanto ciascuna specifica applicazione porta a scelte progettuali radicalmente differenti



Funzioni tipiche dei sistemi embedded

- **Elaborazione**: capacità di elaborare i segnali provenienti dal mondo esterno
- **Comunicazione**: capacità di trasferire i segnali (“informazione”) da/verso il mondo esterno e all’interno del sistema embedded stesso
- **Memorizzazione**: capacità di preservare nel tempo l’informazione all’interno del sistema embedded

Ciascuna specifica applicazione realizzata da un sistema embedded ha diverse esigenze per le capacità di **elaborazione**, **comunicazione** e **memorizzazione**

Una stessa funzionalità (ad esempio, la capacità di acquisire immagini statiche tramite un sensore CCD) può essere oggetto di ottimizzazioni radicalmente differenti se applicata

- ad una macchina fotografica digitale
- ad un telefono cellulare
- ad una videocamera digitale
- ad una webcam per calcolatore elettronico

Caratteristiche “commerciali” dei sistemi embedded

- **Costo finale**: il costo finale del prodotto è un parametro molto importante per le scelte progettuali, ed incide su tutte le scelte di progetto e su tutte le altre caratteristiche
- **Time to market**: nella progettazione di un sistema embedded va sempre tenuto presente il momento in cui si vuol porre il prodotto sul mercato; impiegare un tempo troppo lungo per progettare un dispositivo significa spesso essere superati dalle evoluzioni del mercato
- **Tempo di vita**: altro fattore importante è il tempo di vita atteso per il prodotto; può variare da pochi giorni a diversi anni o decenni
- **Volume**: i volumi di produzione previsti per il sistema sono uno dei fattori più importanti da considerare in fase di progetto, in quanto sono strettamente legati ai costi di produzione ed alla possibilità di recuperare il denaro speso nella fase di sviluppo del sistema



Caratteristiche “hardware” dei sistemi embedded

- **Interfacce di comunicazione:** poiché tipicamente il costo di vendita di un sistema embedded è basso, la scelta delle interfacce di comunicazione è critica in quanto incide notevolmente sul prezzo finale del prodotto
- **Interfacce utente:** in molti sistemi embedded l'interfaccia utente è costituita da pochi pulsanti e/o LED; in altri, si utilizza l'interfaccia utente di un sistema ospitante (ad esempio, un sistema embedded all'interno di un autoveicolo)
- **Consumo energetico:** è un fattore cruciale da considerare per tutti i sistemi embedded che vengono alimentati tramite batterie
- **Dimensioni e peso:** in molti casi, le caratteristiche fisiche sono un altro fattore critico; spesso il sistema embedded deve essere molto piccolo, molto leggero oppure con una forma particolare (ad esempio, molto sottile)

Caratteristiche “software” dei sistemi embedded

- **Capacità di calcolo:** commisurate all'applicazione specifica svolta dal sistema embedded; eventuali capacità in eccesso da valutare in confronto al rapporto con il loro costo
- **Capacità di comunicazione:**
- **Capacità di memorizzazione:**
- **Dimensione del codice:** le capacità di memorizzazione dei sistemi embedded sono comunque limitate, quindi la dimensione del programma interno è un fattore importante
- **Qualità del servizio:** molte applicazioni su sistemi embedded hanno requisiti stringenti in termini di **QoS** (*Quality of Service*); come caso particolare, molte applicazioni richiedono l'erogazione di servizi **in tempo reale**, ossia con vincoli temporali stringenti
- **Aggiornamento del programma:** spesso è utile includere nei sistemi embedded la capacità di aggiornare i programmi in modo da correggere errori scoperti dopo la produzione ed introdurre nuove funzionalità



Dependability dei sistemi embedded

Oltre ai parametri legati al mercato, all'hardware ed al software, i sistemi embedded devono spesso essere *dependable*

In pratica, nella fase di progetto è spesso necessario considerare anche i seguenti aspetti:

- **Affidabilità**: valutazione realistica della probabilità che il sistema si guasti
- **Manutenibilità**: probabilità che il sistema possa essere riparato o sostituito entro un certo intervallo di tempo
- **Disponibilità**: probabilità che il sistema sia funzionante; dipende essenzialmente dall'**affidabilità** e dalla **manutenibilità**
- **Safety**: proprietà legata alla possibilità che in caso di guasto del sistema siano arrecati danni a persone o cose
- **Sicurezza**: capacità di resistenza del sistema contro gli utilizzi non autorizzati o non preventivati in fase di progetto

Sistemi real-time

Definizione informale di **sistema real-time**: è un sistema progettato per operare entro parametri temporali ben definiti

In pratica, un **sistema real-time** opera correttamente solo se per ogni configurazione degli ingressi (input) viene prodotto la giusta uscita **rispettando vincoli temporali ben determinati**

Perché in questo corso parliamo sia di sistemi embedded che di sistemi real-time?

- 1 La maggior parte dei sistemi embedded è anche real-time, e viceversa
- 2 Nella progettazione di un sistema embedded non si può trascurare quale sistema operativo e applicazione verrà eseguita
- 3 La tipologia di sistema operativo real-time è un fattore critico da considerare in tutte le applicazioni **safety-critical**
- 4 La **teoria della schedulazione real-time** studia cosa è possibile ottenere avendo a disposizione risorse hardware **limitate**



Sistemi embedded come sistemi real-time

- I sistemi embedded interagiscono in modo significativo con l'ambiente in cui sono "immersi"
- Nella maggior parte dei casi hanno il compito di reagire "rapidamente" a segnali provenienti dall'ambiente

Esempio: consideriamo il sistema embedded che controlla il rilascio degli *air-bag* di una automobile in conseguenza ad una collisione:

- Se il sistema embedded rilascia l'air-bag in ritardo, l'essere umano subisce l'urto della collisione, poi viene colpito con violenza dal cuscino in espansione
- Se il sistema embedded rilascia l'air-bag in anticipo, l'essere umano è ancora lontano dal cuscino ed il suo movimento non viene frenato; poiché il cuscino si affloscia subito, l'essere umano subisce l'urto della collisione

Il sistema embedded che controlla l'air-bag deve operare con vincoli temporali estremamente rigorosi, altrimenti l'air-bag è non solo inutile ma anche pericoloso



Organizzazione del software nei sistemi embedded

- La progettazione dei sistemi embedded deve tenere in considerazione aspetti quali la rapidità di sviluppo, l'economia di scala, la manutenibilità, ecc.
- Di conseguenza, non è possibile sviluppare l'hardware senza considerare anche il progetto del software
- Ad esempio, in un sistema embedded di fascia alta e con caratteristiche real-time si deve stabilire fin da subito quale sistema operativo real-time dovrà essere utilizzato
- Anche nei sistemi embedded in cui non è utile inserire un sistema operativo completo è necessario stabilire fin dall'inizio l'organizzazione del software di controllo
- Se il sistema embedded è **safety-critical**, la scelta dell'organizzazione del software svolge un ruolo cruciale nella possibilità di **certificare il sistema** per l'uso a cui è destinato



Teoria della schedulazione real-time

- Informalmente, molti risultati teorici della **teoria della schedulazione real-time** assicurano il rispetto dei vincoli temporali di un insieme di processi se l'utilizzo totale del dispositivo di calcolo è inferiore ad una certa soglia
- Dato un sistema real-time che deve assicurare l'esecuzione nei tempi previsti di un insieme di funzioni, aumentare le risorse di calcolo a disposizione in genere facilita l'effettivo rispetto dei vincoli temporali
- Lo sviluppo tecnologico delle piattaforme di calcolo ad uso generale continua ad accrescere la potenza di calcolo disponibile per il progettista del sistema
- Anche i sistemi embedded hanno un rapido sviluppo tecnologico, ma altri aspetti oltre alla potenza di calcolo possono predominare (consumo, costo, dimensione, ...)

Nei sistemi embedded la potenza di calcolo è generalmente ridotta, perciò il problema di cosa è possibile garantire rispettando predeterminati vincoli temporali è fondamentale

