

# SISTEMI DIGITALI INTEGRATI

MASSIMO RUO ROCH

3/10/2016

## PROGETTAZIONE SISTEMI EMBEDDED

I **sistemi embedded** contengono, al loro interno, sia hardware dedicato sia software che gira tipicamente su un microcontrollore (per il primo serve il VHDL, per il secondo il linguaggio C). Questo corso si chiama "*Sistemi Digitali Integrati*" ed è un corso di sistemi, cioè è un corso dove prendiamo componenti probabilmente fatti da altri e li dobbiamo selezionare e collegare tra loro per fargli fare delle funzioni, sfruttando appieno le potenzialità. Un minimo di analogica entra sempre.

Noi ci occupiamo di sistemi che sono elettrici e informatici. Sono a tutti gli effetti dei computer ma non ci occupiamo di personal computer. Noi ci occupiamo di sistemi embedded.

"*To Embed*" in inglese vuol dire incastonare. Un sistema embedded è un sistema in cui c'è un'unità di calcolo, un microprocessore, una memoria etc. quindi un sistema di elaborazione. Tuttavia, visto da fuori non lo sembra.

Esempi: telefoni, alzacristalli delle auto, telecomando, controllo del riscaldamento, antifurto, etc.

Quando si fanno i sistemi embedded è **fondamentale il costo**, perché questi sistemi devono costare poco per avere successo (dobbiamo avere un certo *target price*). Inoltre, il sistema deve sì essere bello, ma **deve fare solamente ciò che viene chiesto al minor costo possibile**: non deve essere il più potente del mondo, perché altrimenti sarebbe uno spreco. Inoltre è importante che si abbiano consumi bassi.

*Il sistema deve funzionare bene ma deve essere minimale.*

Dentro un sistema embedded abbiamo un **sistema di elaborazione** (quindi microprocessore, memoria, canale di comunicazione fra i due come bus) e sicuramente delle periferiche che facciano da interfaccia verso il mondo (come ad esempio un led, un pulsante, un bottone touchscreen di tipo capacitivo).

L'interfaccia I/O può essere quindi variegata e non è solo una tastiera e uno schermo. Quindi, ciò ha un'altra conseguenza: il modo tradizionale di scrivere i programmi per un sistema embedded non funziona, perché il debug potrebbe non esistere (esecuzione e test passo passo): non possiamo fermare uno stream di dati da un canale di un sensore per eseguire un test.

Inoltre, caricare il codice può portare fino a 10 minuti. **Il trial and error non funziona o perlomeno porta via un sacco di tempo.**