

# INTERNET OF THINGS

Dott. Antonio Giovanni Lezzi

# INTERNET OF THINGS

- **Domotica e Bus**
- Logica Cablata e Programmata
- Microprocessori e Microcontrollori
- Sistema Embedded
- Modello di Von Neumann e Harvard
- Raspberry Pi e Arduino

Dott. Antonio Giovanni Lezzi

# DOMOTICA

- L'etimologia della parola **domotica** è data dall'unione della parola domus, che in latino significa "casa", e del suffisso greco ticos, che indica le discipline di applicazione
- È la scienza interdisciplinare che si occupa dello studio delle tecnologie atte a **migliorare la qualità della vita** nella casa e più in generale **negli ambienti antropizzati**
- Questa area fortemente interdisciplinare richiede l'apporto di molte tecnologie e professionalità, tra le quali ingegneria edile, architettura, ingegneria energetica, automazione, elettrotecnica, elettronica, telecomunicazioni, informatica

# DOMOTICA

- La domotica consente di ottenere un notevole **incremento delle prestazioni** e delle possibilità offerte dai diversi impianti presenti nell'abitazione, **ottimizzando i consumi** e permettendo **l'integrazione di diverse funzioni** quali controllo, comfort, sicurezza, risparmio energetico e comunicazione.
- Un sistema domotico consente infatti di realizzare, sia in ambito residenziale che terziario, la gestione coordinata di tutte queste funzioni

# DOMOTICA PER LA CASA

- Ci troviamo così di fronte ad una **«casa intelligente»**, ovvero un ambiente che mette a disposizione di chi lo vive impianti che vanno ben oltre quello tradizionale
- Accensione delle luci, apertura di porte e tapparelle, gestione della climatizzazione, sistemi di sicurezza e sistemi di comunicazione dialogano all'interno o verso l'esterno, anche attraverso telefoni cellulari, smartphone e rete internet
- Una sinergia in grado di aumentare comfort, risparmio energetico, sicurezza e connettività con le reti di comunicazione
- Un impianto domotico infatti può essere realizzato inizialmente con funzioni minime per poi aggiungerne di nuove, in base alle sopravvenute necessità. Una volta predisposto il cavo al quale collegare i dispositivi, in ogni momento sarà infatti possibile aggiungerne di nuovi senza dover intervenire con ulteriori opere murarie.

# DOMOTICA PER IL TERZIARIO

- L'utilizzo della domotica, oltre che in ambito residenziale, può essere esteso anche al terziario, si parla di **«building automation»**
- Questo significa dar vita ad ambienti di lavoro dove, oltre agli impianti di illuminazione, riscaldamento e sicurezza, anche le reti informatiche e di comunicazione sono integrate in un unico sistema, che ne semplifica e ottimizza il controllo e la gestione
- Uffici  
Ospedali  
Banche  
Centri commerciali  
Alberghi

# TRE LIVELLI DI AUTOMAZIONE

- Possiamo prendere in considerazione tre diversi livelli, in ordine di complessità, che ci permettono di inquadrare meglio ciò che genericamente chiamiamo "domotica"
- **Dispositivi singoli**
- **Piccoli impianti**
- **Sistemi domotici**

# DISPOSITIVI SINGOLI

- Piccole apparecchiature che possono essere acquistate e installate direttamente dall'utente (telecomandi, motori elettrici, sensori)
- Questi dispositivi, dotati di alto contenuto tecnologico, portano all'interno della casa soluzioni innovative con funzioni di automazione o controllo che, anche se non sono inseriti in un sistema, possono essere considerati, in un certo senso, come un primo passo verso la domotica

# PICCOLI IMPIANTI

- Serie di dispositivi collegati tra loro e gestiti da un unico sistema di controllo  
(centralina o computer)
- È una sorta di piccolo impianto domotico presente solo in alcune parti della casa: può servire per gestire una specifica funzione (l'apertura delle porte) oppure per assicurare la gestione completa delle funzioni di un locale
- Esempio, il "controllo ambiente" di una camera per gestire luci, tapparelle, tv, ventilatore ecc.

# SISTEMI DOMOTICI

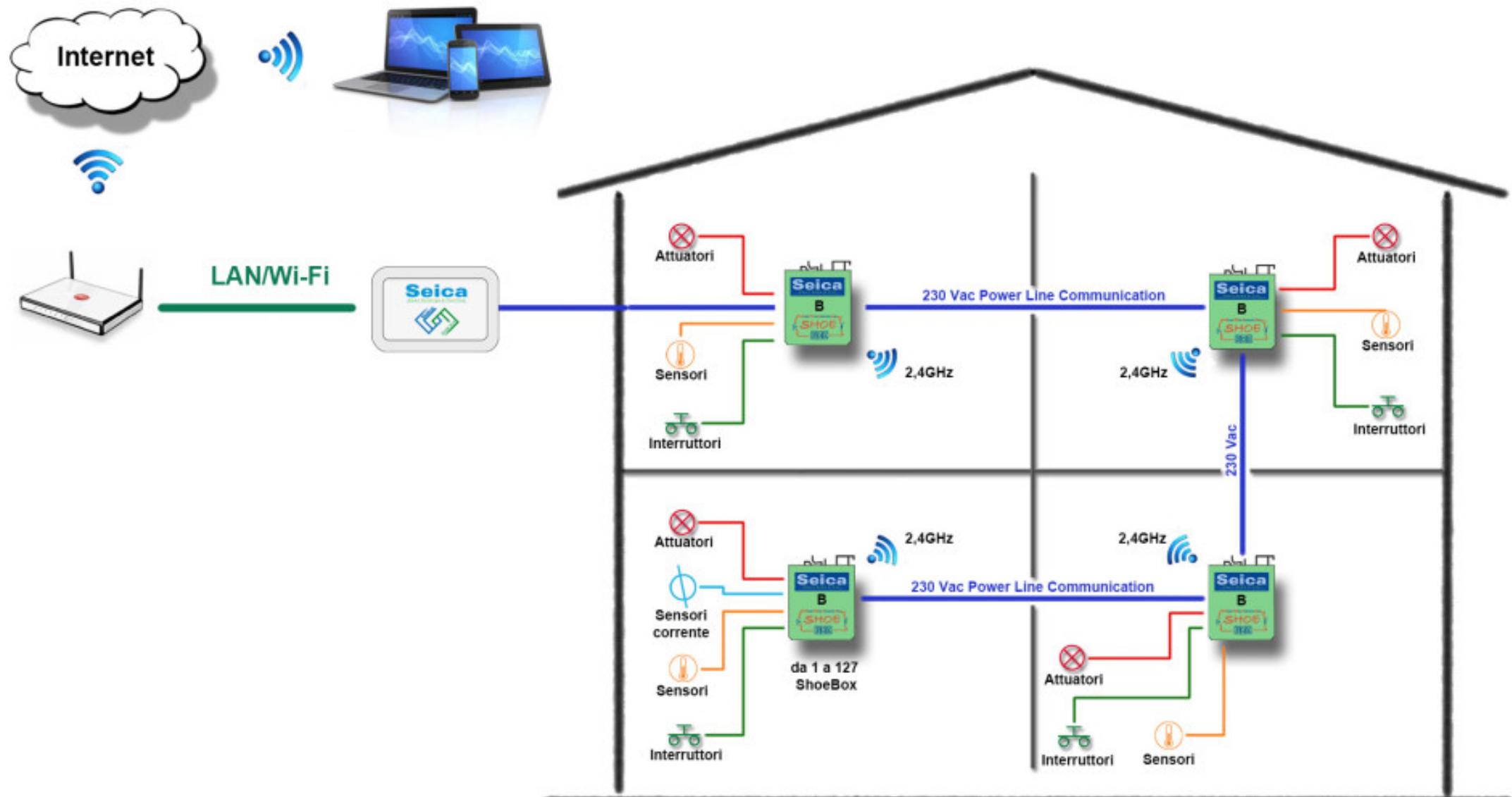
- A questo livello la domotica si esprime al massimo potenziale
- Il sistema domotico rappresenta la piena integrazione degli impianti (elettrico, idrico, riscaldamento, automazioni, sicurezza ecc.)
- La gestione dei diversi impianti viene affidata ad un sistema di controllo che ne ha la supervisione
- L'utente si "interfaccia" al sistema (usa tastiere, telecomandi, computer) per controllare il funzionamento dei dispositivi presenti all'interno della casa

# I SISTEMI BUS

- La tecnologia che oggi permette la realizzazione di un sistema domotico completo è costituita dai "sistemi bus"
- Per bus si intende una **linea dati che collega i diversi dispositivi** del sistema domotico e che trasmette tutte le informazioni di controllo
- Nei sistemi bus il comando di attivazione di un componente, ad esempio l'accensione della luce, non avviene in modo diretto, attraverso un filo elettrico, ma è controllato dal bus: il dispositivo di input (l'interruttore) invia un segnale nella rete, cioè nella linea dati del bus, che lo ritrasmette al dispositivo di output (la lampada).
- Solo a prima vista il risultato non cambia. La differenza fondamentale è legata al fatto che in questo modo il sistema domotico è informato di tutto quello che avviene nella casa e può comandare i diversi dispositivi
- L'informatica, ovvero una gestione di tipo software, si inserisce all'interno degli impianti e, come sappiamo, grazie alla versatilità e la flessibilità della programmazione è possibile potenziare le possibilità di controllo e di gestione di tutti i dispositivi che vengono installati nell'alloggio e che sono connessi al bus.

# I SISTEMI BUS

- Il bus permette di impartire comandi in modo automatico: se avviene un determinato evento, allora si aziona quel dispositivo. Oppure può effettuare delle programmazioni temporali o legate ad eventi
- Una caratteristica particolarmente interessante dei sistemi bus è la possibilità di poter immettere un comando (input) nel sistema attraverso dispositivi differenti. In questo modo l'utente, in funzione delle sue capacità, può scegliere il dispositivo di comando che meglio si adatta alle sue esigenze.
- Inoltre il sistema bus, proprio perché viene programmato via software, può esser facilmente modificato nel tempo
- Possibilità di connettere il "sistema domotico" della casa con altre reti (internet, rete GSM, cablaggi in fibra ottica ecc.)



# INTERNET OF THINGS

- Domotica e Bus
- **Logica Cablata e Programmata**
- Microprocessori e Microcontrollori
- Sistema Embedded
- Modello di Von Neumann e Harvard
- Raspberry Pi e Arduino

# INFORMATICA VS SISTEMI

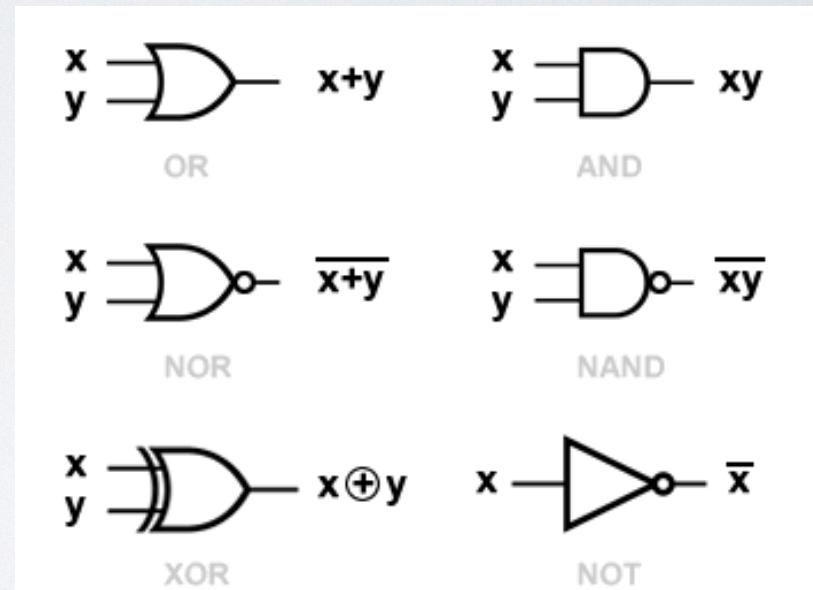
- Informatica e Sistemi, trattano computer in 2 modi diversi
- **Informatica**: programmazione comportamento e interfaccia utente
- **Sistemi**: architettura dei sistemi di elaborazione, detti anche sistemi a logica programmabile
- Cosa sono le architetture? Modo in cui i componenti HW e SW sono connessi

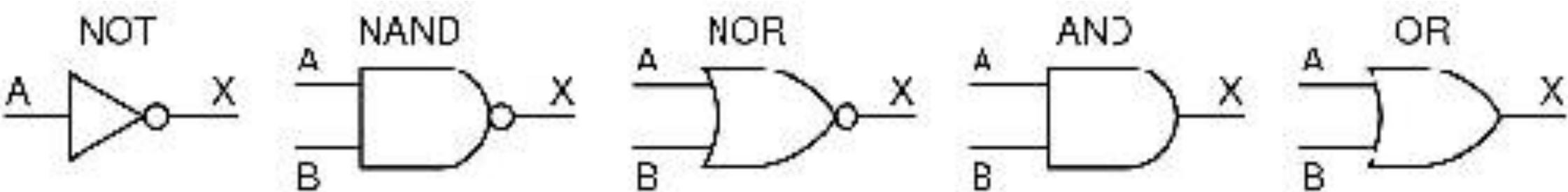
# LOGICA CABLATA E PROGRAMMATA

- **Logica:** ha tanti significati, qui indica il metodo (tecnologia) con cui è realizzata la funzione di un apparato elettronico. Si distingue:
- **Logica cablata:** dispositivi elettronici connessi tra loro in un certo modo per assolvere a una funzione
- **Logica programmabile:** l'apparato contiene un programma, una serie di istruzioni che indicano cosa fare
- Cambia programma-->cambia funzione

# LOGICA CABLATA

- Una porta logica cablata è una **porta logica** che implementa la **logica booleana** che utilizza solo componenti passivi come diodi e resistenze
- Una connessione logica cablata può svolgere la funzione di una porta AND o di una OR
- I principali limiti del dispositivo sono l'inabilità ad implementare la porta NOT e la mancanza di ripristino dei livelli





A	X
0	1
1	0

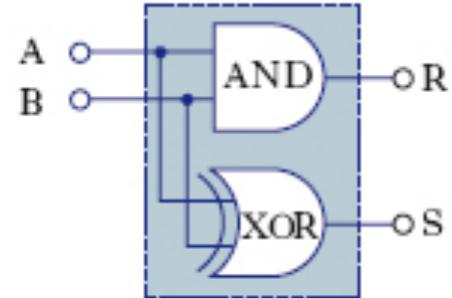
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

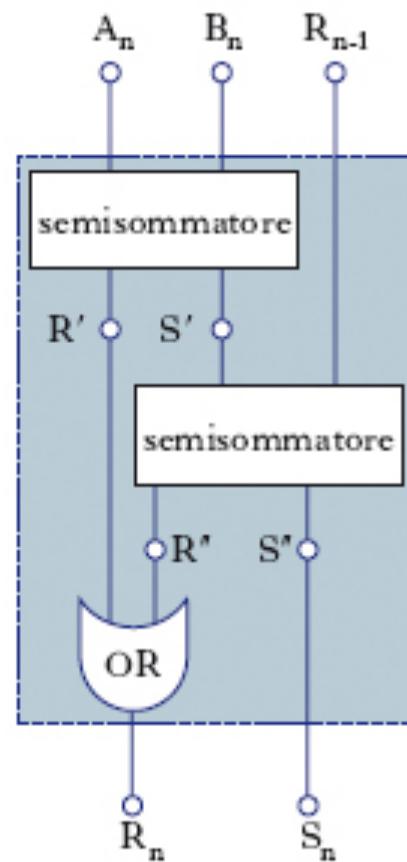
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

# SOMMATORE CON PORTE LOGICHE



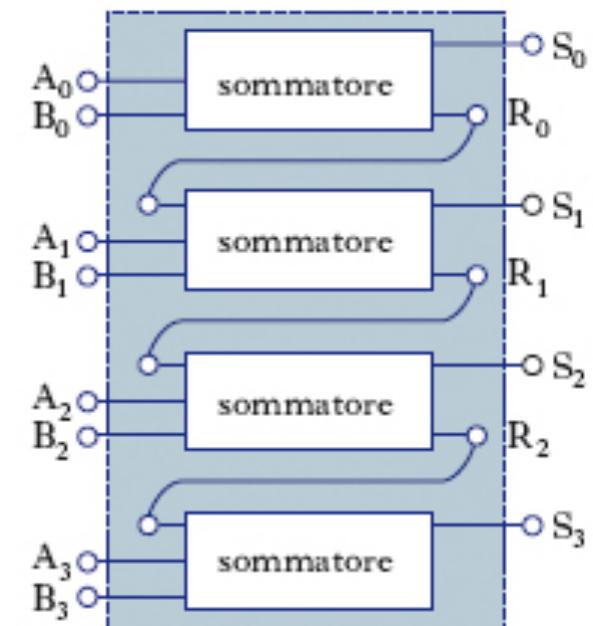
A	B	R	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

A



B

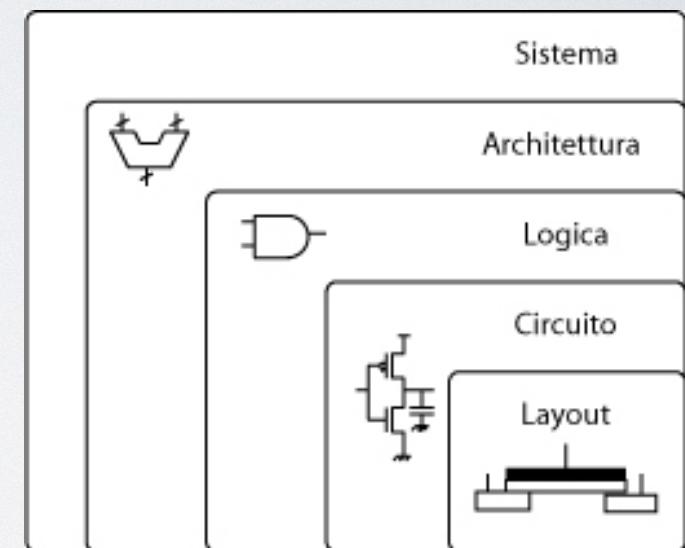
$A_n$	$B_n$	$R_{n-1}$	$S_n$	$R_n$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



C

# LIVELLI DI ASTRAZIONE

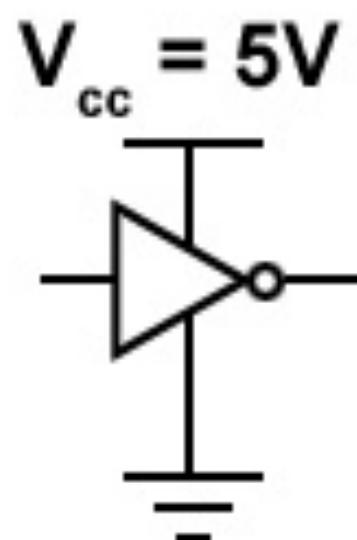
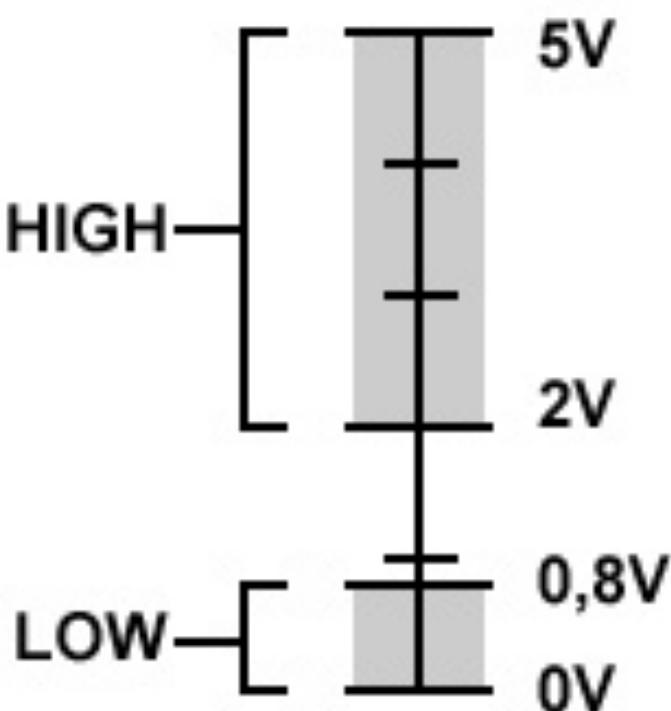
- A **livello di sistema** è noto ciò che il sistema deve fare ma non la sua struttura fisica. È composto da vari processing elements che interagiscono tra di loro utilizzando protocolli di comunicazione, si implementa usando il C/C++
- A **livello architetturale** il sistema elettronico è un insieme di componenti come registri, ALU, multiplexer connessi fra di loro. A questo livello si utilizzano linguaggi detti HDL (hardware description language) come il VHDL
- A **livello logico** ogni componente dell'architettura è una rete logica che svolge il compito previsto, è un insieme di connessioni di porte logiche (AND, OR, XOR NOT ecc).
- A **livello circuitale** ogni porta logica viene realizzata tramite un circuito (il cui elemento base è il transistor)
- Il livello più basso è il **livello di layout** in cui vengono specificate nel dettaglio le dimensioni di ogni singolo transistor, e di ogni connessione



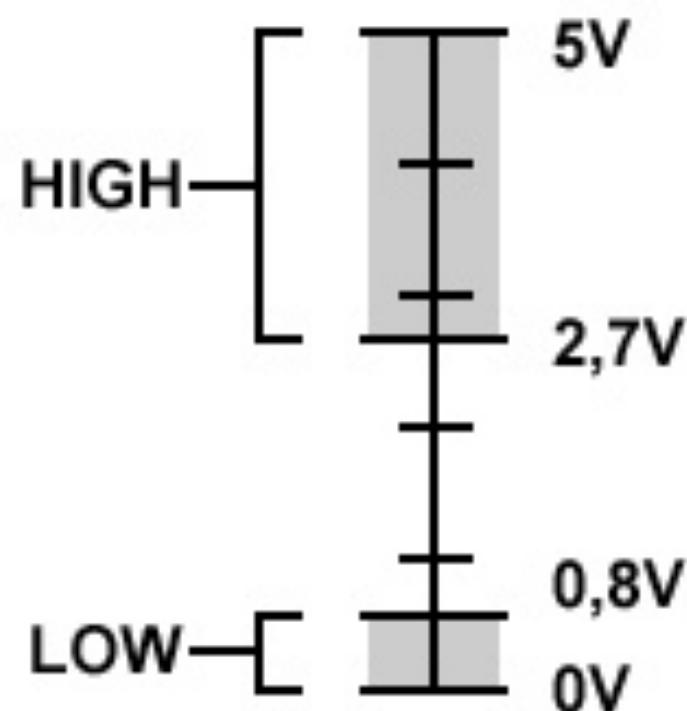
# LIVELLI LOGICI

- I due **livelli di tensione** usati nei circuiti digitali **rappresentano i numeri binari 0 e 1**, detti livelli logici.
- Generalmente **si associa il livello basso** (rappresentato solitamente con la **lettera L**, da low) **allo 0 e il livello alto** (rappresentato solitamente con la **lettera H**, da high) **all'1**, anche se è possibile utilizzare la rappresentazione opposta
- È usuale permettere una certa tolleranza nella tensione utilizzata; per esempio una tensione tra 0 e 2 volt può rappresentare lo 0 logico, mentre una tensione tra 3 e 5 volt può rappresentare l'1 logico
- Una tensione tra 2 e 3 volt sarebbe quindi non valida e potrebbe esistere soltanto in condizioni di errore oppure durante una transizione di livello logico che non avviene mai in modo istantaneo

**Livelli di tensione  
accettabili per i  
segnali di INGRESSO  
per i TTL**



**Livelli di tensione  
accettabili per i  
segnali di USCITA  
per i TTL**



# LOGICA PROGRAMMATA

- Il microcontrollore o MCU (MicroController Unit) è un **dispositivo elettronico integrato su singolo chip**, nato come evoluzione alternativa al Microprocessore ed utilizzato generalmente in sistemi embedded (applicazioni specifiche o special purpose) di controllo digitale
- È progettato per interagire direttamente con il mondo esterno tramite un **programma residente nella propria memoria** interna e mediante l'uso di pin specializzati o configurabili dal programmatore
- Sono disponibili in 3 fasce di capacità elaborativa (ampiezza del bus dati): 8 bit, 16 bit e 32 bit.
- L'ampia gamma di funzioni di comando e controllo disponibili, sia analogiche che digitali, permette l'impiego delle MCU in sostituzione di schede elettroniche cablate tradizionali ben più complesse e costose

# INTERNET OF THINGS

- Domotica e Bus
- Logica Cablata e Programmata
- **Microprocessori e Microcontrollori**
- Sistema Embedded
- Modello di Von Neumann e Harvard
- Raspberry Pi e Arduino

# MICROPROCESSORI

- I primi microprocessori sono apparsi negli anni '70 Intel 8080, oggi sono integrati praticamente in ogni apparecchio elettronico presente sul mercato: lavatrici, forni a microonde, telefonini, autovetture, ...
- Mediamente un microprocessore per essere definito embedded deve avere le seguenti caratteristiche:
- Deve essere dedicato al **controllo real-time** di uno specifico dispositivo o funzione
- Deve contenere il proprio **programma operativo** in qualche tipo di memoria non volatile
- Deve essere **trasparente all'utente** (deve funzionare come un hardware dedicato)

# MICROPROCESSORI E MICROCONTROLLORI

- Quali sono le differenze tra microprocessore e microcontrollore?
- Il primo potrebbe essere definito come il “cuore” di un computer fisso, portatile cellulari o scanner è un componente che ha bisogno di numerose integrazioni esterne aggiuntive per poter funzionare, tra le quali la memoria, un oscillatore di clock, periferiche di ingresso e uscita
- Un microcontrollore riunisce tutti gli elementi all'interno di un unico piccolo contenitore, e in teoria non ha bisogno di altri componenti esterni per poter funzionare.(comprende al suo interno la memoria per il programma, la memoria RAM, l'oscillatore di clock, il circuito di reset e le periferiche)
- Secondo la legge di Moore, il numero di transistor integrabili sullo stesso chip dovrebbe raddoppiare ogni 18 mesi e l'evoluzione del microprocessore ha seguito con buona approssimazione questa regola.

# USI DEL MICROCONTROLLORI

- Le capacità di calcolo di un microcontrollore sono estremamente ridotte
- La memoria RAM è formata da qualche centinaio di celle, e di solito non è espandibile
- Tra le tipiche applicazioni di un microcontrollore ci possono essere gli antifurti, gli strumenti di misurazione, quelli per la regolazione della luminosità, i carica batterie e i trasmettitori/ricevitori
- I microcontrollori sono progettati per eseguire un piccolo insieme di funzioni specifiche, come Digital Signal Processor, che svolge un piccolo insieme di funzioni di elaborazione del segnale ed è ampiamente utilizzato per regolare i freni su tutte e quattro le ruote, regolare l'aria condizionata in auto

# DIFFERENZE

<b>Microcontrollore</b>	<b>Microprocessore</b>
Il programma di gestione risiede al suo interno in un'apposita area di memoria non volatile (ROM o EPROM o FLASH)	Esegue il programma che risiede in un'area di memoria esterna e viene modificata in base ai comandi della CPU
È presente una memoria RAM per i registri e per i dati; in alcuni è presente una memoria EEPROM per i dati	Ha una memoria ridotta per i registri
All'interno sono implementate le decodifiche per le porte di input e output; i livelli logici sono dati dai segnali esterni	Le porte di input e output sono stabilite dai driver delle periferiche
Sono integrati i convertitori A/D, comparatori e generatori di segnali	Non sono integrate le funzioni
È presente sempre almeno un timer	È presente il clock della CPU
Sono dispositivi RISC (Reduced Instruction Set Computing)	Sono dispositivi CISC (Complex Instruction Set Computing)

# INTERNET OF THINGS

- Domotica e Bus
- Logica Cablata e Programmata
- Microprocessori e Microcontrollori
- **Sistema Embedded**
- Modello di Von Neumann e Harvard
- Raspberry Pi e Arduino

# SISTEMA EMBEDDED

- Un sistema embedded deve contenere solitamente le seguenti componenti:
  - Un microprocessore
  - Memoria RAM (random access memory)
  - Memoria non-volatile : ROM, EEPROM,, FLASH, ...
  - I/O (interfaccia con l'ambiente)
- In cosa un sistema embedded è diverso da un computer (PC)? Applicazione! Personal Computer devono poter svolgere una varietà virtualmente
- I sistemi embedded devono svolgere un limitato numero di task (as. controllo della temperatura di un forno, controllo del tempo di lavaggio di una lavatrice, ....)

# SISTEMA EMBEDDED

- Perche usare un sistema a microprocessore embedded?
- **Costo:** Microprocessore embedded è molto vantaggioso rispetto all'implementazione hardware in componenti discreti
- **Programmabilità:** La stressa piattaforma hardware permette di implementare differenti applicazioni
- **Flessibilità:** Le funzionalità possono essere semplicemente ri-programmate in firmware
- **Adattabilità:** Implementazione di sistemi intelligenti (“smart”) con capacità di adattarsi all’ambiente

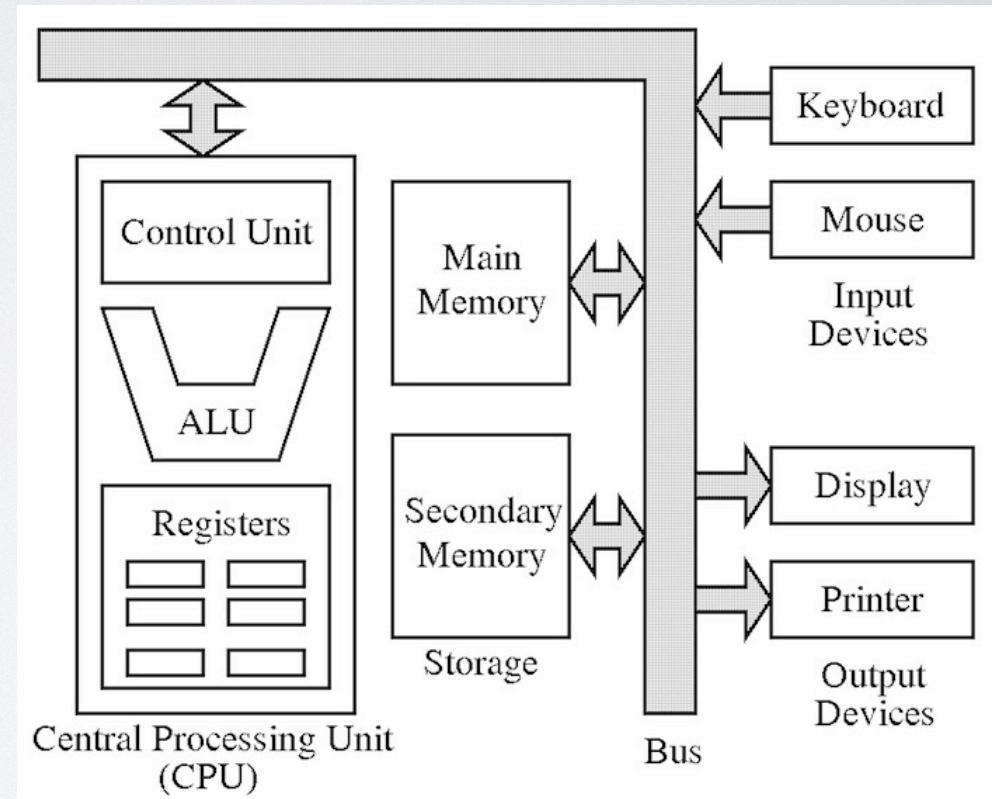
# INTERNET OF THINGS

- Domotica e Bus
- Logica Cablata e Programmata
- Microprocessori e Microcontrollori
- Sistema Embedded
- **Modello di Von Neumann e Harvard**
- Raspberry Pi e Arduino

Dott. Antonio Giovanni Lezzi

# MODELLO DI VON NEUMANN

- È una tipologia di architettura hardware la quale condivide i dati del programma e le istruzioni del programma nello stesso spazio di memoria
- Si contrappone all'architettura Harvard nella quale invece i dati del programma e le istruzioni del programma sono memorizzati in spazi di memoria distinti
- L'architettura di Von Neumann è l'architettura hardware su cui sono basati la maggior parte dei moderni computer programmabili



# MODELLO HARVARD

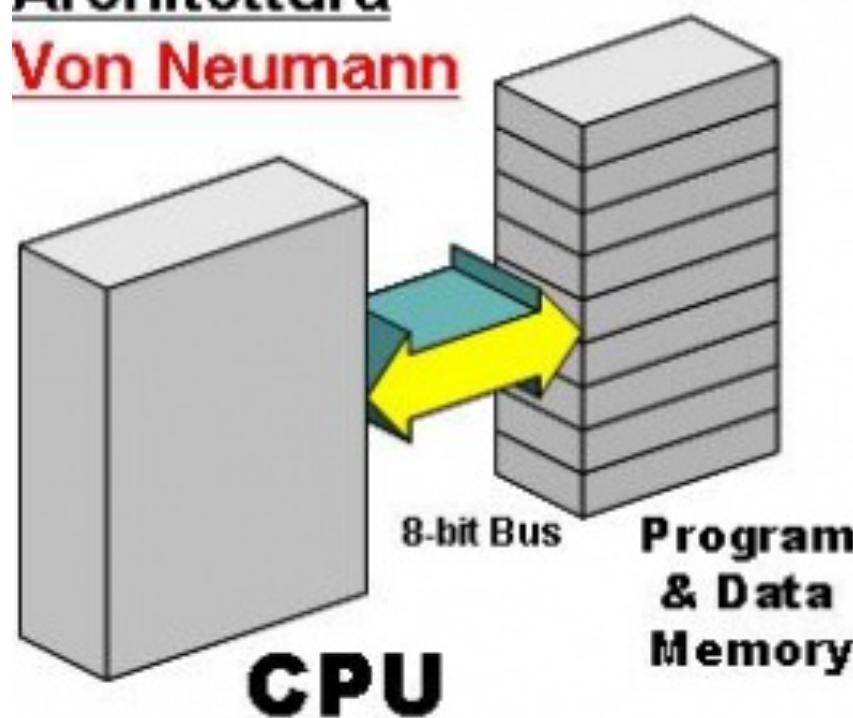
- È un tipo di architettura hardware per computer digitali in cui vi è separazione tra la memoria contenente i dati e quella contenente le istruzioni
- Il termine inizialmente indicava l'architettura del computer Harvard Mark I, un computer basato su relè che memorizzava le istruzioni su un nastro perforato mentre i dati venivano memorizzati in un contatore elettromeccanico a 23 cifre
- Questa macchina non era dotata di un'unità di immagazzinamento dei dati, questi erano interamente memorizzati dalla CPU e il loro caricamento e salvataggio era un processo eseguito in modo manuale agendo sui contatori

# DIFFERENZE TRA I MODELLI

- La differenza sostanziale che c'è tra le due architetture è la seguente:
- Nell'architettura di Von Neumann c'è un unico banco di memoria che racchiude Program memory e Data memory, di conseguenza la CPU è costretta a fare la fase di fetch-exec in modo sequenziale
- Nell'architettura Harvard possiamo invece distinguere i due banchi di memoria in modo tale che la fase di fetch può sovrapporsi a quella di exec e viceversa.
- Comunemente l'architettura Harvard possiamo trovarla in dispositivi nei quali sappiamo all'incirca qual è la grandezza del programma e qual è quella dei dati.
- Nelle CPU c'è una architettura Von Neumann, dato che, il nostro processore è general purpose e la dimensione del programma o dei dati che esso andrà ad utilizzare può variare

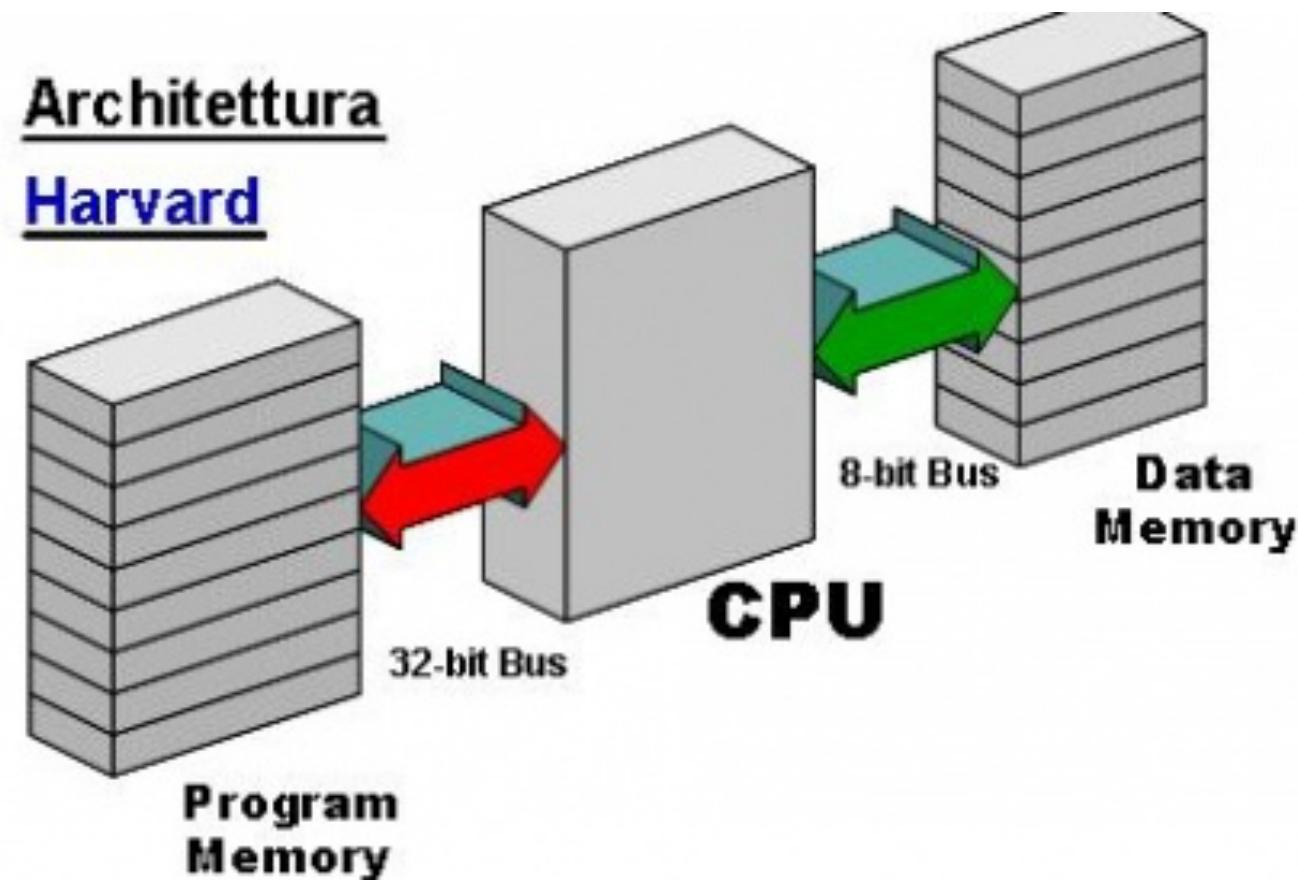
## Architettura

### Von Neumann



## Architettura

### Harvard



# INTERNET OF THINGS

- Domotica e Bus
- Logica Cablata e Programmata
- Microprocessori e Microcontrollori
- Sistema Embedded
- Modello di Von Neumann e Harvard
- **Raspberry Pi e Arduino**

Dott. Antonio Giovanni Lezzi

# ARDUINO

- È una piccola scheda elettronica con un microcontrollore (ATmega) e prende il nome da un Bar di Ivrea praticato da alcuni fondatori del progetto
- Il suo utilizzo è specialmente per fare **piccoli esperimenti di elettronica**, ma l'hanno utilizzato per **progetti molto interessanti**
- Questo microcontrollore non ha sistema operativo e viene programmato tramite un software IDE scrivendo i programmi attraverso il suo ambiente di sviluppo, con un linguaggio piuttosto semplice a partire da C e C++
- Grazie all'aggiunta di led, sensori, relay è possibile ad esempio realizzare un impianto di domotica
- Esempio: con un sensore di temperatura possiamo impostare Arduino in modo che se la temperatura scende sotto i 15 gradi accende il riscaldamento, se supera i 30 accende l'aria condizionata, collegando una Scheda GSM possiamo con un SMS accendere o spegnere qualcosa a distanza



# RASPBERRY PI

- Usa prevalentemente come **sistema operativo** delle distribuzioni linux specifiche
- Fu creato con l'obiettivo di insegnare ai ragazzi l'informatica e in particolare il linguaggio Python
- La **scheda SD è il suo 'hard disk'**, si collega il monitor alla porta **HDMI**, la tastiera e il mouse alle **porte USB** ecc
- Raspberry ha la **porta GPIO**, ovvero quei pin che possono essere programmati
- Rispetto ad Arduino risulta essere diverso: Raspberry Pi è una board con tutto quello che possiamo aspettarci da un computer: CPU, varie porte USB, Ethernet, HDMI, Jack audio



# ARDUINO VS RASPBERRY PI

- **Arduino** viene usato per la gestione di segnali e altro attraverso il microcontrollore e ci consente di realizzare dei progetti di vero e proprio hardware
- Su una **Raspberry Pi** dobbiamo montare un sistema operativo (il OS predefinito Raspbian OS è una distribuzione Linux derivata da Debian).
- Su Raspberry possiamo far girare i nostri software, ossia qualcosa di “meno elettronico” e può essere dedicata alla gestione casalinga (e non) del software.

# ARDUINO

- La scheda è dotata di una **memoria flash da 32 k bytes** (non tantissimi ma sufficienti per gestire programmi anche abbastanza complessi), **2 kbyte di sram ed 1 kbyte di eeprom**, i relativi **bus, il software di bootstrap** (il bootloader, il software di inizializzazione del microprocessore) ed il programma di gestione della connessione usb
- La **scheda è priva di sistema operativo** ed è in grado di gestire un solo programma per volta e quando si costruisce o si ricostruisce un prototipo bisogna quindi caricare, da pc, il relativo programma
- Una caratteristica fondamentale della memoria flash è la capacità di mantenere il suo stato anche in assenza di alimentazione per cui un programma, una volta caricato sulla scheda, resta memorizzato e disponibile sino a quando non ne viene caricato un altro

