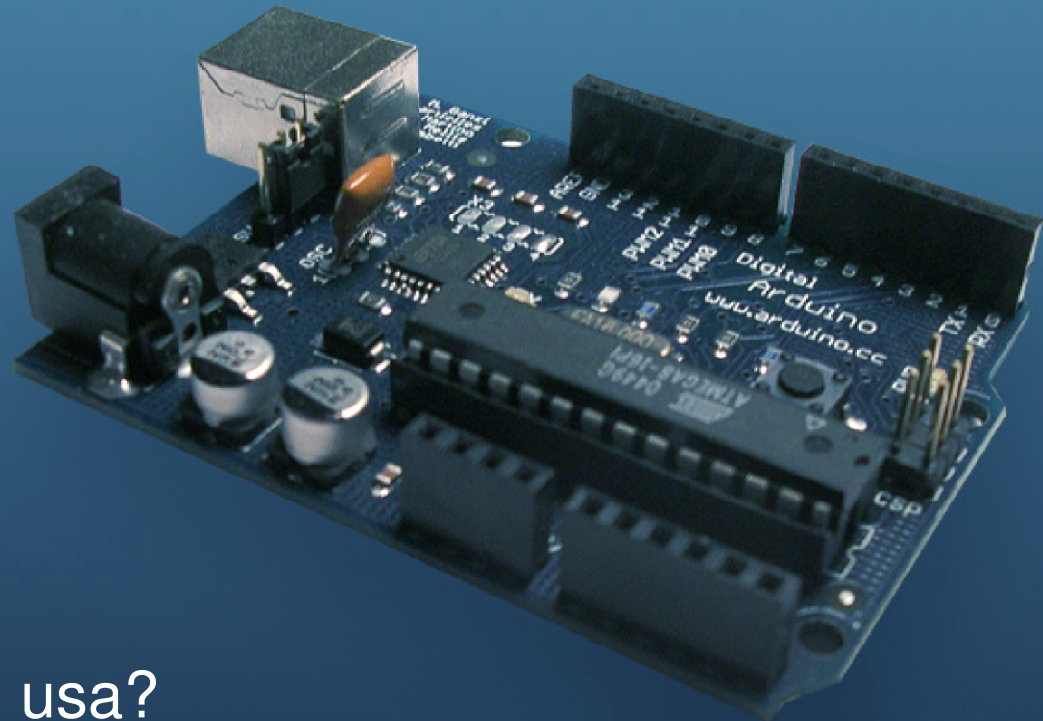


# Arduino

Physical Computing I/O board

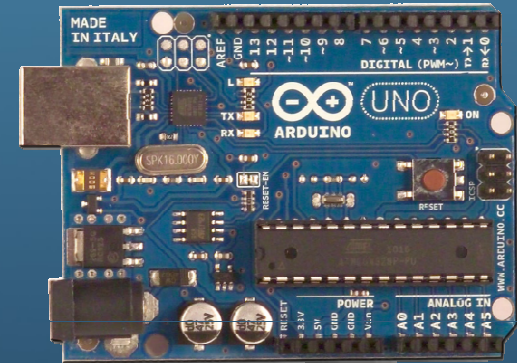


- Cosa è?
- Come lo si usa?
- Come iniziare ?



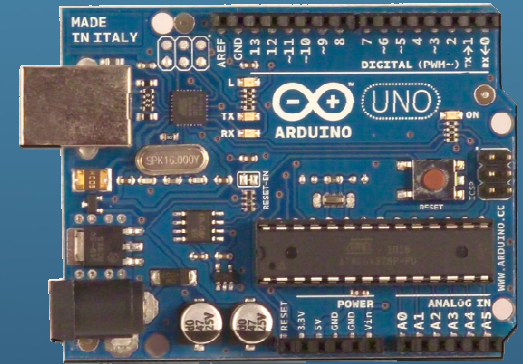
# Cosa è Arduino?

*“Arduino è una piattaforma di sviluppo open-source basata su una semplice scheda I / O con un  $\mu$ C e un ambiente di sviluppo che implementa hardware e software facile da usare.”*



*Arduino può essere usato per sviluppare oggetti interattivi stand-alone (indipendenti) o può essere interfacciato a un PC per la sua programmazione tramite un normalissimo cavetto USB e il microcontrollore sulla scheda è programmato utilizzando un linguaggio adatto anche ai principianti grazie all'ambiente di sviluppo integrato scritto in JAVA. ( [www.arduino.cc](http://www.arduino.cc), 2006 )*

# Cosa è Arduino?



- Arduino è sviluppato da un gruppo internazionale di tecnici e programmatori. E' una idea italiana!
- Lo sviluppo è aiutato da una enorme community online con esempi e consigli.
- Economico
- L'home page del progetto è <http://www.arduino.cc>.

# Il microcontrollore

- Un microcontrollore ( $\mu\text{C}$ ) è un dispositivo di elaborazione dati simile al microprocessore ( $\mu\text{P}$ ).
- Sul  $\mu\text{P}$  sono basati tutti i calcolatori, dai personal computer ai mainframe.
- Sul  $\mu\text{C}$  sono basati dispositivi per scopi industriali o per applicazioni particolari (automazione elettronica, domotica).

# Il Microprocessore contiene:

- Una **ALU** (Arithmetic Logic Unit) dove avvengono i calcoli.
- Diversi **registri** per la memorizzazione temporanea dei dati e la gestione delle istruzioni.
- Un **bus interno** ad alta velocità.
- Circuiti di controllo e di **temporizzazione** per coordinare tutte le attività.
- Tre bus (**Data Bus, Address Bus, Control Bus**) per comunicare col mondo esterno (dispositivi di memoria, dispositivi di ingresso-uscita).

# Il Microprocessore ha bisogno per funzionare:

- Di un'ampia area di memoria esterna sia di lavoro (**RAM**) che di massa (hard disk, ecc...), dato che tratta grandi quantità di dati.
- Di **interfacce** verso dispositivi esterni di **Input** (tastiera, mouse, scanner, hard disk, floppy, microfoni, ecc...).
- Di **interfacce** verso dispositivi esterni di **Output** (monitor, stampante, altoparlanti, hard disk, floppy, modem, ecc...).
- Di potenza di **alimentazione** anche di decine di watt.

# Il Microcontrollore invece possiede su un unico chip:

Una **CPU** RISC (Reduced Instruction Set Computer).

Una piccola memoria di programma (**EPROM-EEPROM**).

Una piccola memoria di lavoro **RAM** (alcuni kB).

Porte di **ingresso/uscita**.

**Contatori, timer, convertitore A/D**.

**UART** (ricevitore-trasmettitore asincrono universale), interfacce di comunicazione di vari tipi.

# Il Microcontrollore...

Contiene sia quello che possiede un  $\mu$ P (anche se in quantità molto ridotta), sia parti che il  $\mu$ P non possiede.

Ha ingombro minimo e richiede poca potenza di alimentazione.

**RISC** significa Reduced Instruction Set Computer, ovvero Elaboratore con insieme di istruzioni ridotto.

Sono inoltre semplici istruzioni che richiedono circa il medesimo tempo di elaborazione.

Le istruzioni perciò:

- sono poche decine
- sono eseguite molto velocemente
- non serve un clock molto elevato per un efficiente funzionamento (dai 4 agli 8 MHz per i tipi più semplici, fino a 33-50 MHz per i tipi più evoluti).



# Microcontrollori

Esiste un grande numero di modelli di microcontrollori; tra i principali si ricordano:

- ATMEL AVR
- Dallas / Intel 8051
- Microchip PIC
- Motorola 6811

# Struttura semplificata di un Microcontrollore

Flash Program memory

Program Counter

Instruction Register

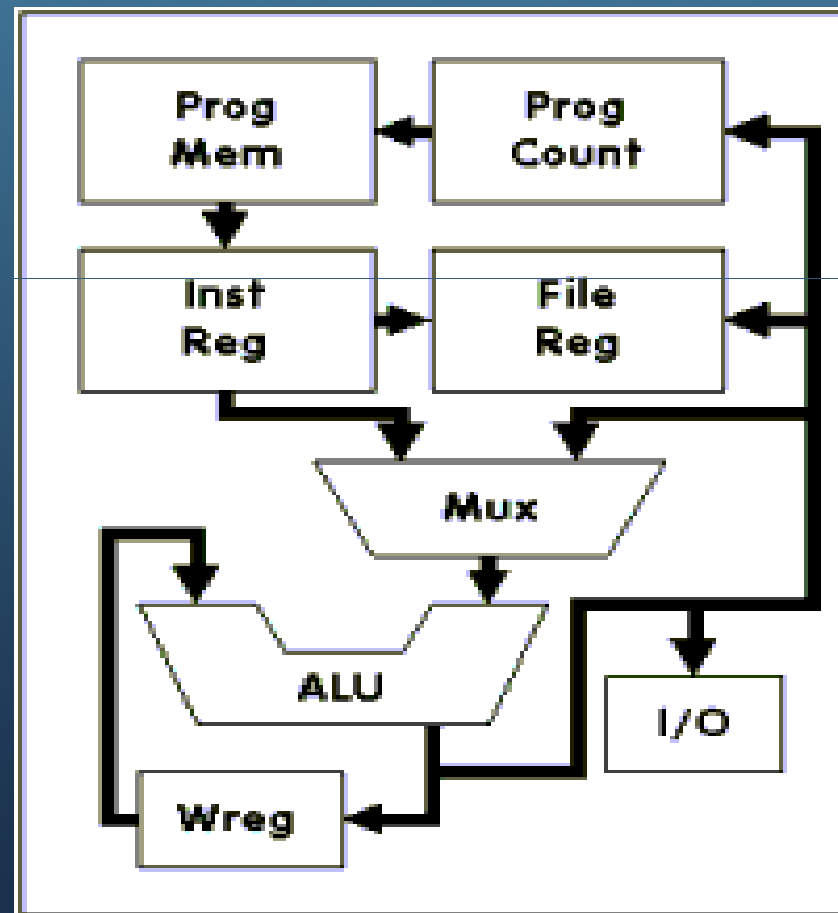
RAM File Registers

Address Mux

ALU

W register (accumulator)

I/O Ports



# Struttura semplificata di un Microcontrollore

**Program Counter:** contatore che punta alla successiva istruzione da eseguire tra quelle contenute nel Program Memory.

**Instruction Register:** registro che contiene le istruzioni eseguibili; le confronta con il Program Memory e le dà per l'esecuzione alla ALU.

**ALU:** esegue calcoli matematici e logici sui dati in ingresso e deposita i risultati di calcolo nell'Accumulatore.

# Piattaforme di sviluppo (boards)

Ci sono un numero crescente di piccole compagnie che producono piattaforme di sviluppo per l'utilizzo dei microcontrollori:

- Arduino (AVR)
- Basic Stamp (PIC)
- HandyBoard (6811)
- BX-24 (AVR)

Di norma queste "piattaforme" sono di tipo commerciale "chiuso" e non possono essere riutilizzate in altri progetti senza autorizzazione del produttore primario.

Fa eccezione la piattaforma Arduino, realizzata in modalità Open Source e distribuita sotto licenza Creative Commons.

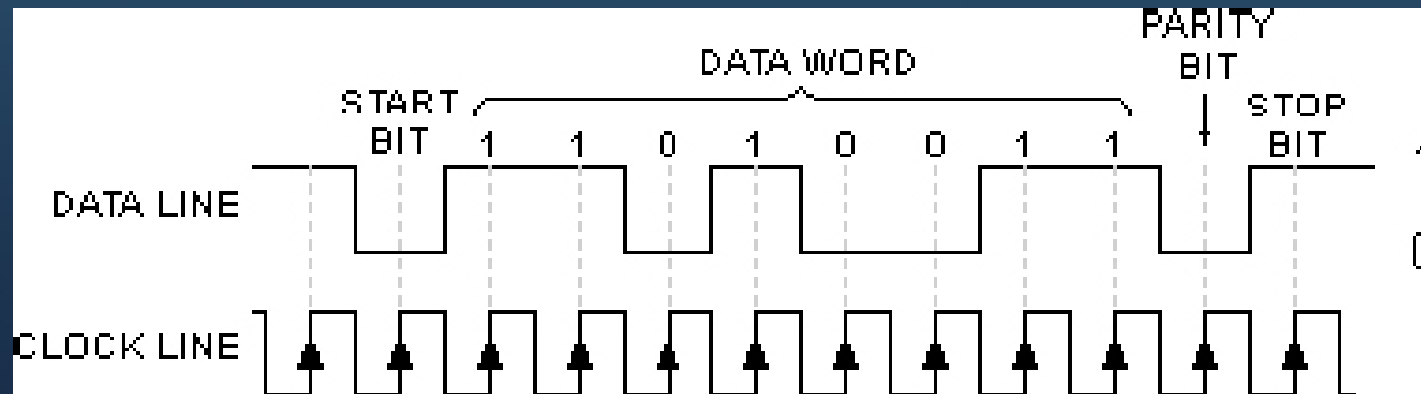
# Comunicazione seriale

La maggior parte dei microcontrollori operano in stand-alone, ovvero non necessitano di lavorare connessi ad altri computer.

Per caricare il programma nel  $\mu\text{C}$  occorre solitamente collegarlo alla porta seriale di un computer.

Nelle comunicazioni seriali i dati vengono spediti un bit alla volta.

I cavi seriali utilizzati presentano un conduttore di trasmissione (TX) e uno di ricezione (RX). Questo consente di ottenere una comunicazione bidirezionale nella quale i dati vengono inviati e ricevuti contemporaneamente.



# Comunicazione seriale

Per realizzare la comunicazione seriale occorre definire:

## Baud Rate

Velocità di trasmissione tra computer e microcontrollore. Valore tipico 9600 bits per second (baud).

## Bit di start

Serve ad "avvisare" il ricevitore che la trasmissione dei dati sta per iniziare.

## Data bits

Il dato che deve essere inviato.

Ad esempio la lettera H in binario è 01001000 (ASCII 72).

## Bits di stop

Servono a indicare al ricevitore che la trasmissione è finita.

# Comunicazione seriale

Per realizzare la comunicazione seriale occorre definire:

## Bit di parità

Un semplice **controllo di errore** che indica se il numero di bit a 1 inviati è pari o dispari.

Spesso non è utilizzato.