

# Verso Arduino... E oltre!

Introduzione alla scheda elettronica Arduino  
come strumento didattico e di prototipazione.

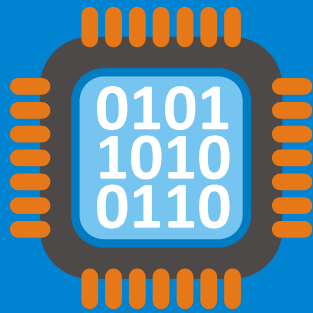
**Elettronica libera**

ARM e mondo embedded

ViGLug – 21.03.2015

A cura di Andrea "Tsumi" Boero ([mail@tsumi.it](mailto:mail@tsumi.it)) - CC BY-SA 4.0

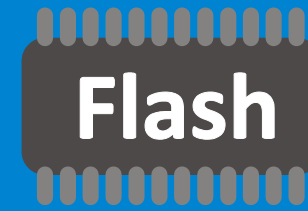
# Cos'è un microcontrollore? (1)



Processore  
(CPU)

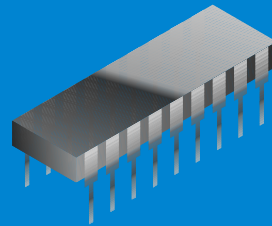


Memoria di lavoro  
(RAM)

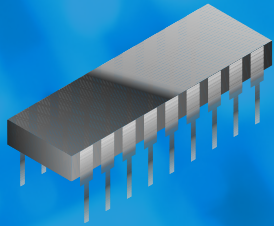


Memoria non volatile  
(ROM e EEPROM)

Su un unico Chip!



# Cos'è un microcontrollore? (2)



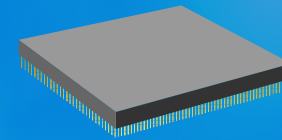
## MICROCONTROLLORE

Scarsa potenza di calcolo

Basso consumo energetico

Bassa produzione di calore

Basso costo



## MICROPROCESSORE

Elevata potenza di calcolo

Alto consumo energetico

Alta produzione di calore

Alto costo

# Dove li usiamo?

Vengono utilizzati in tutti quegli ambiti dove sono richiesti:

- Ridotta potenza di calcolo
- Bassi consumi
- Costo contenuto

Quasi tutti i dispositivi elettrici ne montano almeno uno:

- Piccoli oggetti elettronici di uso comune: carte di credito, orologi digitali, telecomandi, ecc...
- Elettrodomestici: lavatrici, forni a microonde, frigoriferi, ecc...
- Componenti degli stessi PC: mouse, tastiere, chiavette USB, ecc...

# Cos'è Arduino?

Arduino è una scheda elettronica che integra un microcontrollore ed alcuni componenti di contorno necessari al suo funzionamento.



→ **È Open Hardware:** gli schemi, il software sono liberamente accessibili e modificabili.

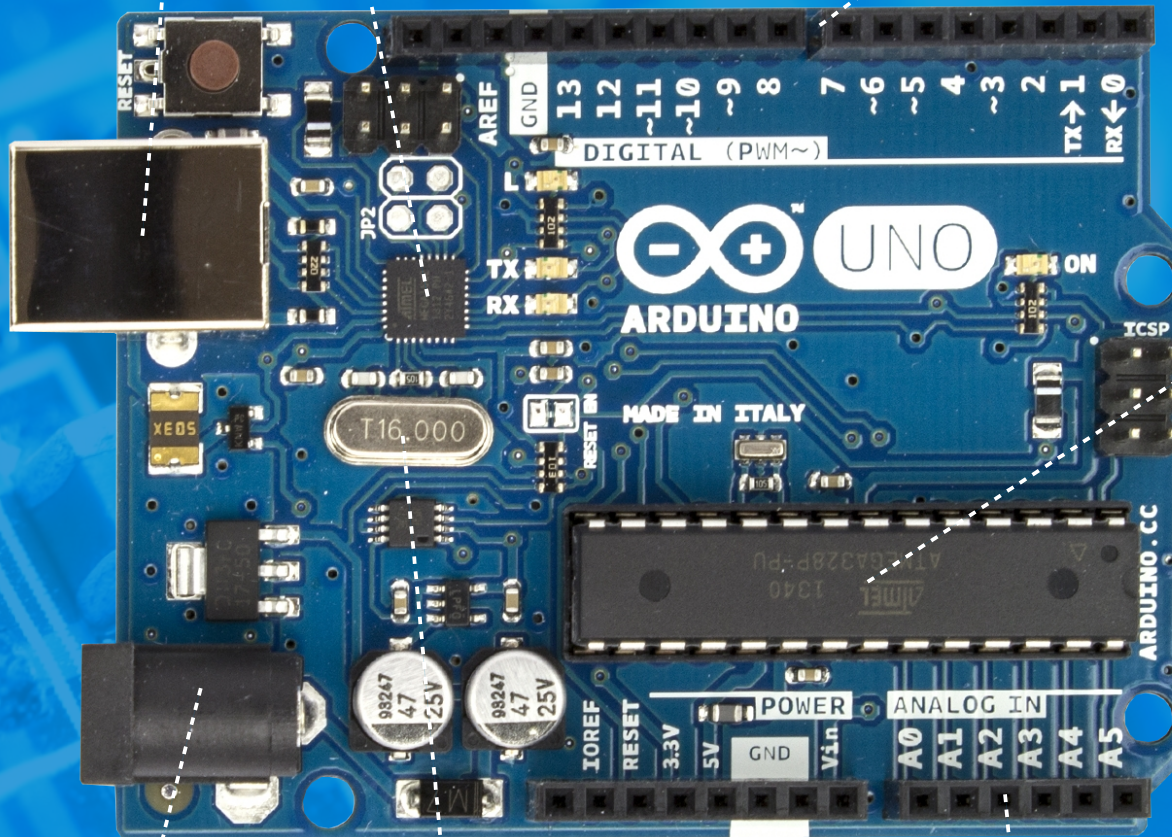
- **È orientato alla didattica:** è disponibile una grande quantità di documentazione.
- **È semplice:** la scheda contiene tutto l'hardware necessario al funzionamento del microcontrollore ed il software comprende un ambiente di sviluppo completo.
- **È economico:** la maggior parte dei modelli oscilla tra i 15 ed i 40€.

# Com'è strutturato?

Porta USB

Programmatore/  
Interfaccia Seriale

Ingressi Digitali



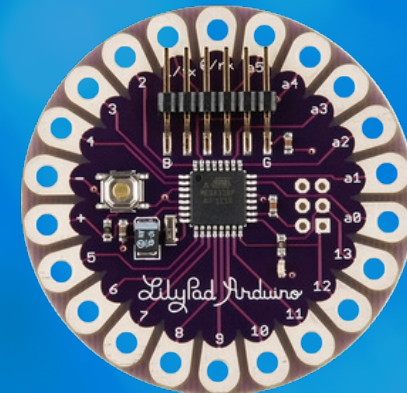
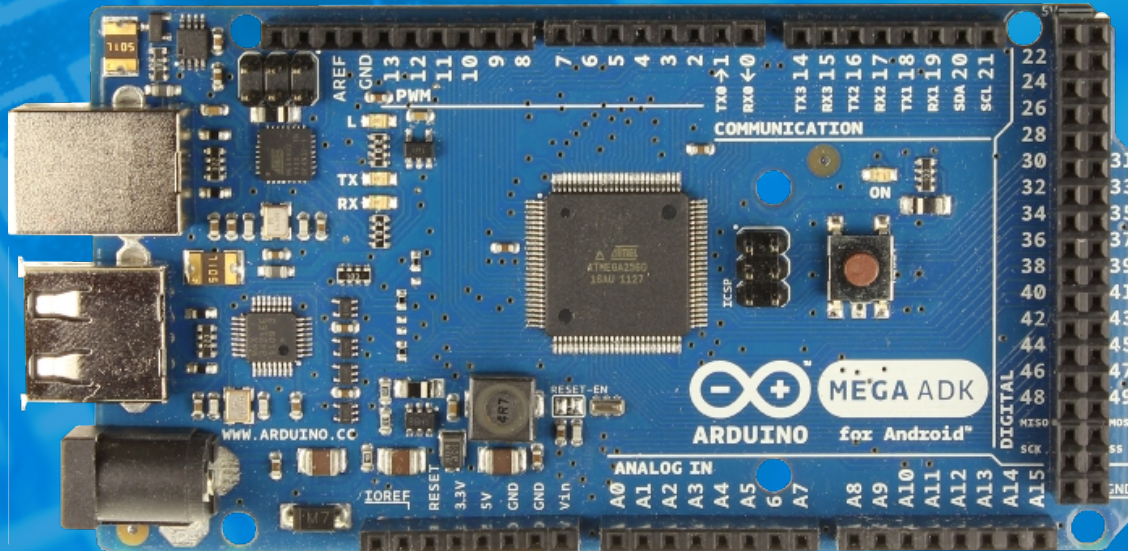
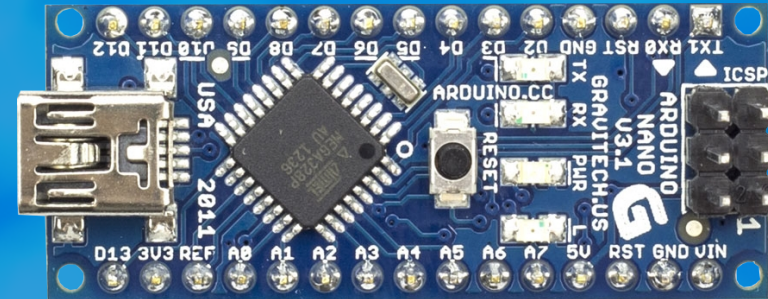
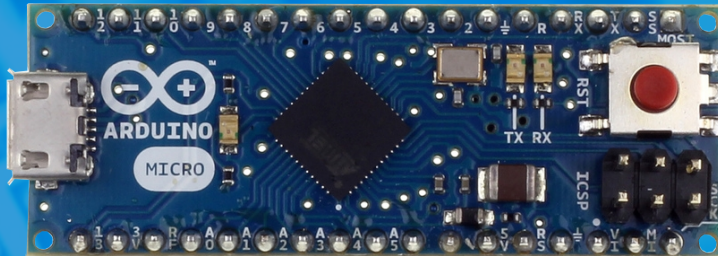
Microcontrollore

Alimentazione  
esterna

Quarzo  
oscillatore

Ingressi Analogici

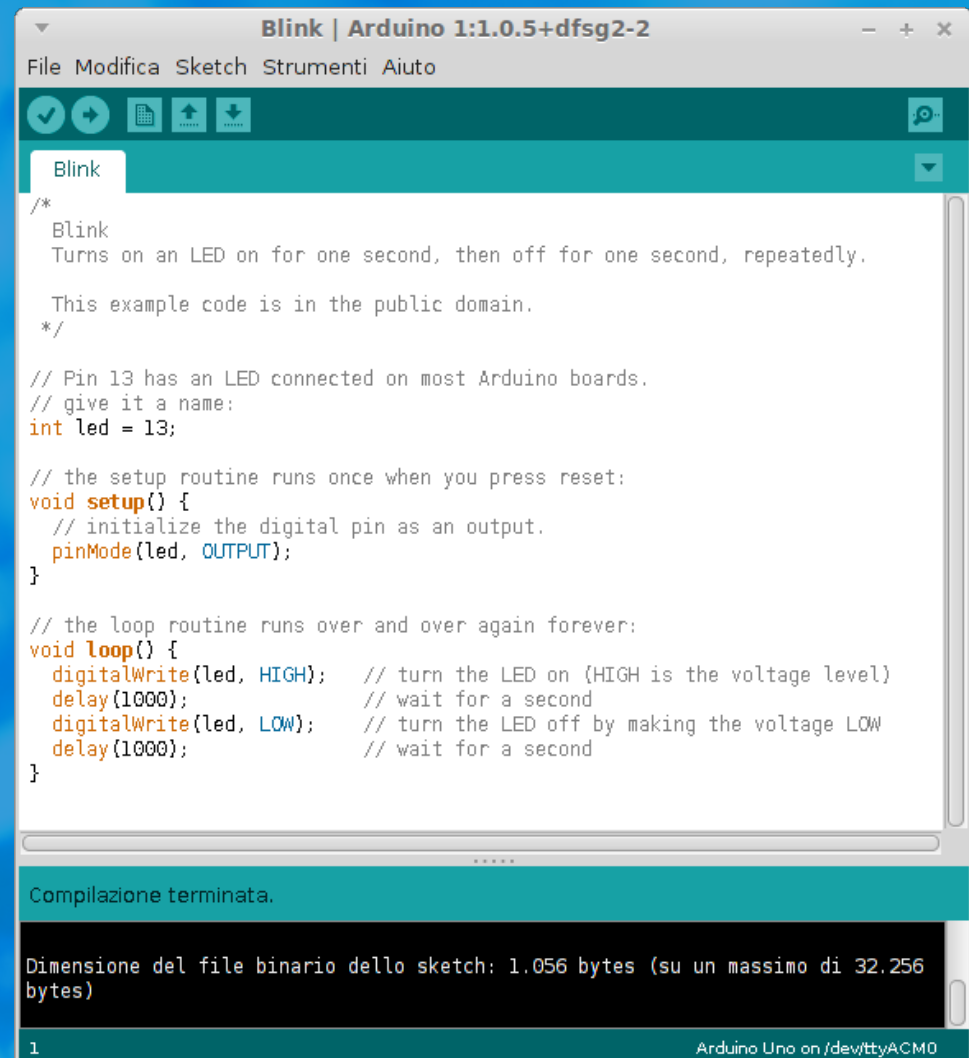
# Alcuni altri modelli



# Il software

Il software Arduino è un ambiente completo per programmare il microcontrollore della scheda e comprende:

- Un editor per il codice
- Un compilatore
- Il software necessario a caricare il programma sul microcontrollore
- Una console seriale



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 1:1.0.5+dfsg2-2". The menu bar includes "File", "Modifica", "Sketch", "Strumenti", and "Aiuto". The toolbar contains icons for saving, undo, redo, and uploading. The main editor window displays the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Below the code editor, a green status bar indicates "Compilazione terminata." (Compilation finished). The console window shows the message: "Dimensione del file binario dello sketch: 1.056 bytes (su un massimo di 32.256 bytes)". The status bar at the bottom indicates "1" and "Arduino Uno on /dev/ttyACM0".



# Cosa è in grado di fare?

Due tipi di pin:

- **Digitali:** possono rilevare o produrre (Input/Output) un segnale alto o basso (0 o 5 volt), alcuni sono dotati di PWN (Pulse-Width Modulation)
- **Analogici:** rilevano il voltaggio in ingresso (da 0 a 5 volt) e lo mappano su 10 bit (valori da 0 a 1023)

Caratteristiche del microcontrollore (ATmega328P, Arduino UNO):

- **Frequenza di Clock:** 16Mhz
- **ROM:** 32 Kbyte
- **RAM:** 2 Kbyte
- **EEPROM:** 1 Kbyte
- Risorse hardware **estremamente limitate**, ma sufficienti a gestire anche una decina di dispositivi collegati

# Utilizzi pratici

- Come strumento didattico nel campo dell'elettronica (pura e programmabile)
- Come base per la creazione di prototipi

Per svariati motivi:

- ✓ **Semplicità di utilizzo** → Permette una partenza “veloce” anche ad un neofita
- ✓ **Ambiente completo** → La scheda contiene tutti i componenti minimi necessari per lavorare.
- ✓ **Hardware aperto** → Possibilità di studiarlo a fondo o modificarlo in base alle proprie esigenze.
- ✓ **Comunità** → Molta documentazione disponibile creata dalla community su cui basarsi.

# Andare oltre

→ Arduino è sicuramente un'ottima scheda ma, nonostante tutte le sue versioni, potrebbe non essere adatta ad un progetto più avanzato

Ci possono essere vari motivi per costruire una scheda derivata:

- ✓ Ridurre gli spazi → La scheda Arduino è comoda da usare ma mediamente ingombrante rispetto al solo microcontrollore.
  - ✓ Ridurre i costi → La scheda Arduino è decisamente costosa rispetto al prezzo del solo microcontrollore
  - ✓ Eliminare o aggiungere hardware → potremmo voler eliminare la parte USB, perché inutilizzata ed aggiungere componenti necessari ai sensori che andranno collegati.
- Possibilità di utilizzare l'Arduino come base per il prototipo e poi rimpiazzarlo nel prodotto finito.
- Possibilità di utilizzare l'Arduino come programmatore per molti microcontrollori, anche di modelli diversi.

Grazie per l'attenzione :-)

Domande?

Background slide: <http://pixelperfectdigital.com>

Clip Arts: <https://openclipart.org>

Boards Arduino: <http://arduino.cc>