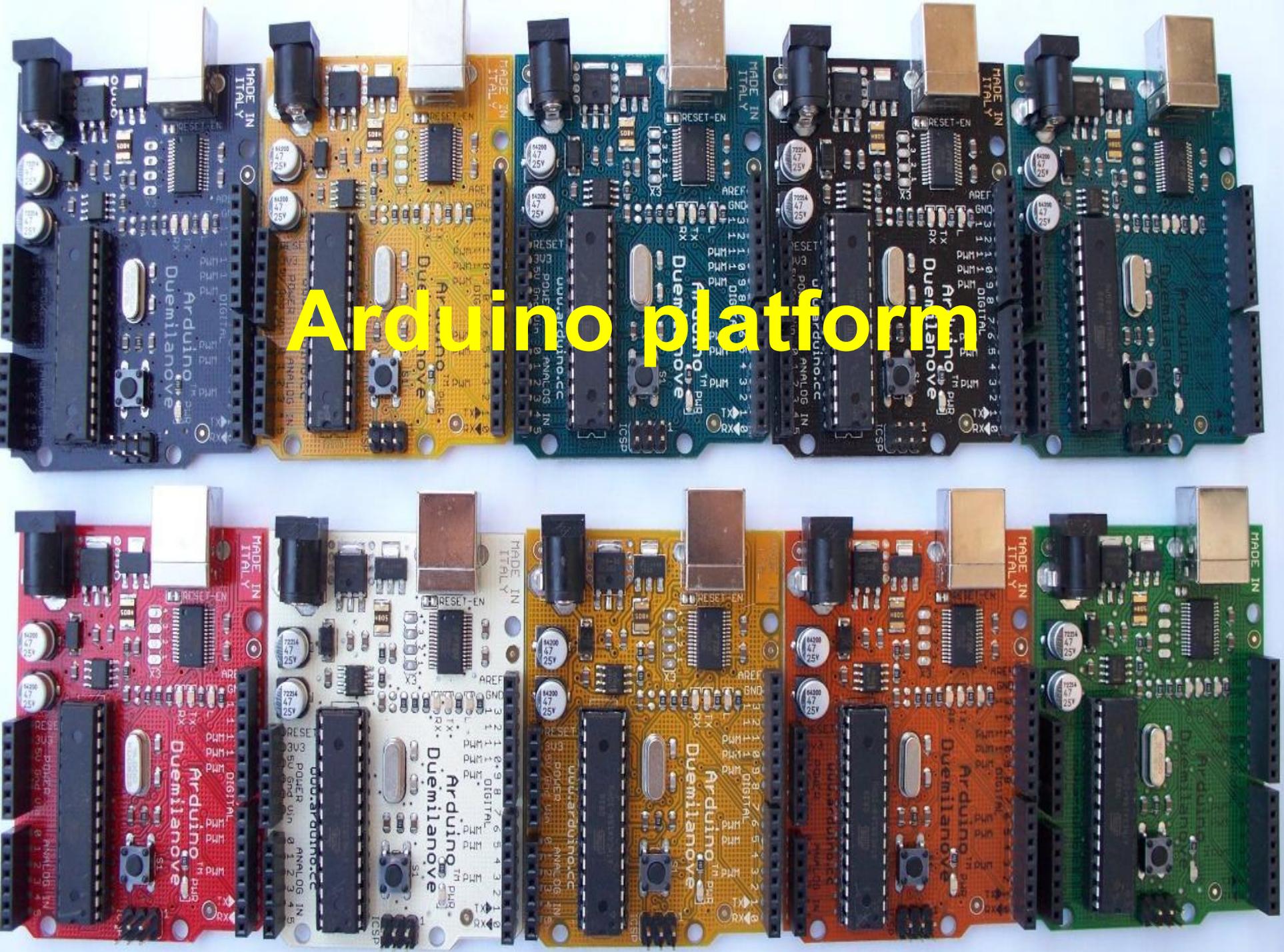


# Arduino platform



# Che cos'è Arduino?

- E' una piattaforma *open-source* basata su una semplice scheda a microcontrollore e un ambiente di sviluppo per scrivere il software.
- Può essere usata per sviluppare oggetti interattivi , per acquisire input da switch e sensori, per controllare luci, motori e altri output fisici.
- La scheda può essere assemblata a mano oppure la si può comprare preassemblata;
- l'ambiente di sviluppo integrato *open-source IDE* può essere scaricato gratuitamente .

# Perchè Arduino?

1. **Economica** - La piattaforma arduino è economica comparata alle altre piattaforme a microcontrollore.
2. **Cross-platform** - il sw gira su Windows, Macintosh OSX, and Linux operating systems. Molti sistemi a microcontrollore sono limitati a funzionare su Windows.
3. **Ambiente di programmazione semplice** - lo possono usare i principianti , è abbastanza flessibile per utenti più esperti ;
4. **Open source e extensible software**- il Sw di Arduino è gratuitamente scaricabile [www.arduino.cc](http://www.arduino.cc) , è uno strumento creato da programmatori con esperienza. Il linguaggio di programmazione è una espansione del C++, chi vuole capire i dettagli tecnici può fare un salto da Arduino al linguaggio di programmazione del microcontrollore AVR sul quale è basato. Si può infatti aggiungere in Arduino il codice AVR.
5. **Open source e extensible hardware** - Arduino è basato sul microcontrollore Atmel ATMEGA8 e ATMEGA168. Tutti i materiali sono pubblicati con la licenza delle Creative Commons license.

# Arduino : l'ambiente di sviluppo

L'ambiente di sviluppo contiene:

un *text editor* per scrivere il codice ,

una *message area*,

una *text console*,

un *toolbar* con pulsanti e una serie di menu.

Arduino è connesso al PC mediante porta seriale per caricare i programmi sulla scheda hardware .

## Writing Sketches

I programmi sw scritti che si usano con Arduino si chiamano **sketch**.

Gli sketch sono salvati con l'estensione **.ino**.

Supportano le caratteristiche **cutting/pasting** e **searching/replacing text**.

L' *message area* mostra il *feedback* mentre si sta salvando e si sta esportando lo sketch mostrando sul display gli errori .

Nell'angolo in basso a destra della finestra vengono indicati il numero di porta seriale e la scheda hw corrente. I *toolbar buttons* permettono di verificare e caricare i programmi , creare, aprire, e salvare gli sketch , e aprire il *serial monitor*.

verifica gli errori presenti nel codice

compila e carica il codice sulla scheda

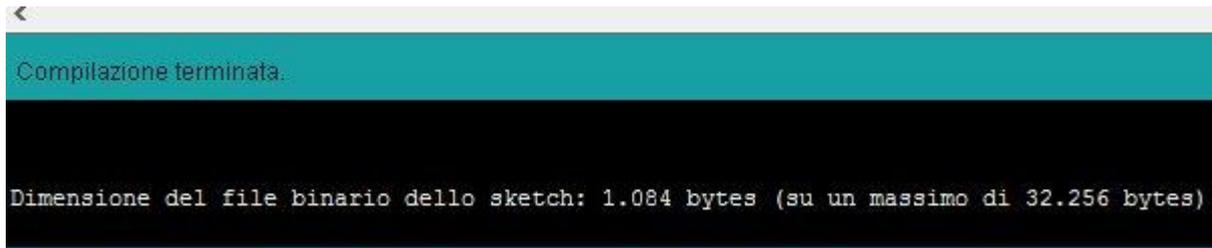
crea un nuovo sketch



Aprire un menu di tutti gli sketch realizzati

salva lo sketch

apre il serial monitor



message AREA



indica il tipo di scheda e il numero della porta seriale collegata

# alcune tipologie di *board*

<http://arduino.cc/en/Main/Products>

## Arduino Uno

Microcontroller **ATmega328**

Operating Voltage **5V**

Input Voltage (recommended) **7-12V**

Input Voltage (limits) **6-20V**

Digital I/O Pins **14** (of which 6 provide PWM output)

Analog Input Pins **6**

DC Current per I/O Pin **40 mA**

DC Current for 3.3V Pin **50 mA**

Flash Memory **32 KB** (ATmega328) of which 0.5 KB used by bootloader

SRAM **2 KB** (ATmega328)

EEPROM **1 KB** (ATmega328)

Clock Speed **16 MHz**

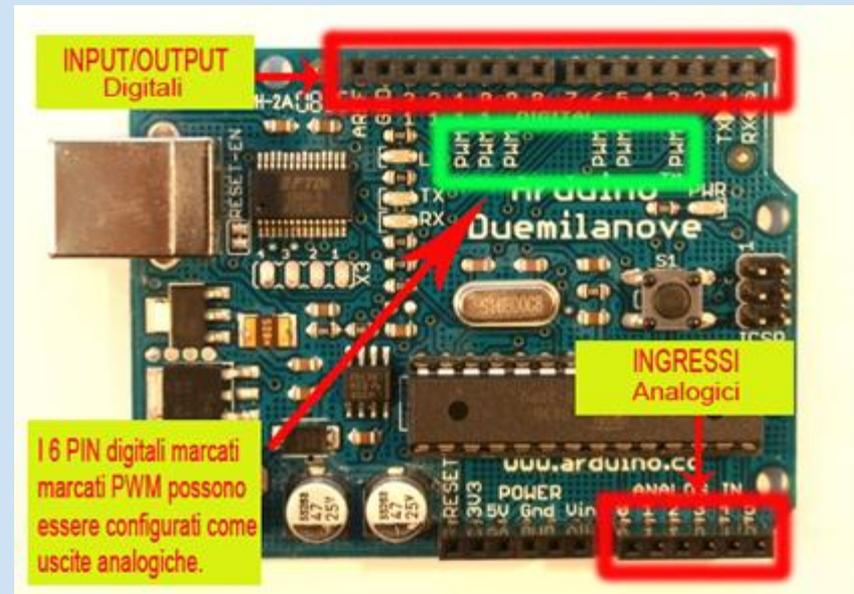


*prima dell'arduino uno*

## Arduino Duemilanove o Diecimila

ATmega168

16 MHz



*curiosità: 10000 pari al numero di schede realizzate nel primo stock di produzione*

# Arduino BT: la versione Bluetooth

**ATmega328**

**16 MHz auto-reset versione Bluetooth**

**modulo Bluegiga WT11**

**con questo modulo aggiuntivo la scheda**

**può essere programmata in modalità wireless e può inviare e rx dati**

**Alimentazione esterna da 1,2 a 5,5 V**



# Arduino Ethernet: per la connessione alla rete

**ATmega328**

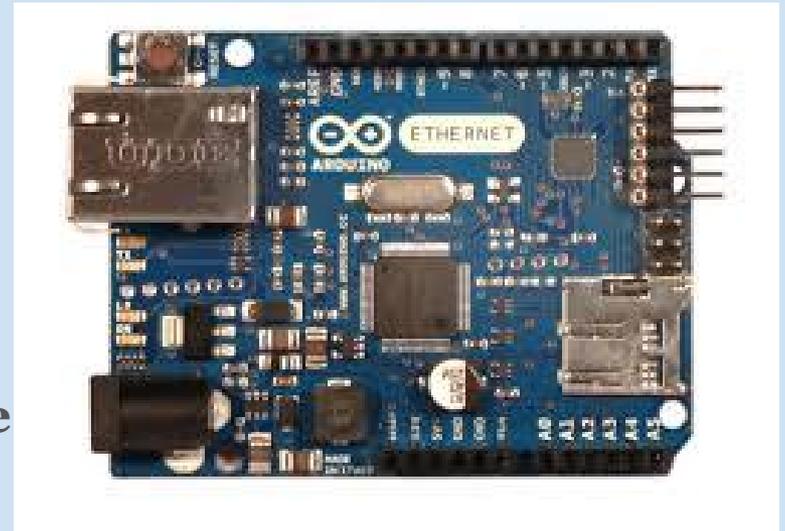
**16 MHz auto-reset**

**come Arduino uno ma i pin da 10 a 13 sono riservati per la comunicazione Ethernet**

**no USB**

**lettore di schede MicroSD per memorizzare dati**

**In alternativa Arduino 1 + shield Ethernet**



# Arduino Mega 2560: aumentano i pin

ATmega2560

16 MHz auto-reset

54 pin **digitali** di Input/Output

( di cui 14 PWM, 16 pin di input analogico)



# Arduino Mega ADK: ANDROID

E' pensata per lo sviluppo di accessori per i cellulari Android

ATmega2560

16 MHz auto-reset

54 pin **digitali** di Input/Output

( di cui 14 PWM, 16 pin di input **analogico** )



è integrato hw necessario all'interfacciamento con cellulari

# Arduino LilyPaD: indossa l'elettronica

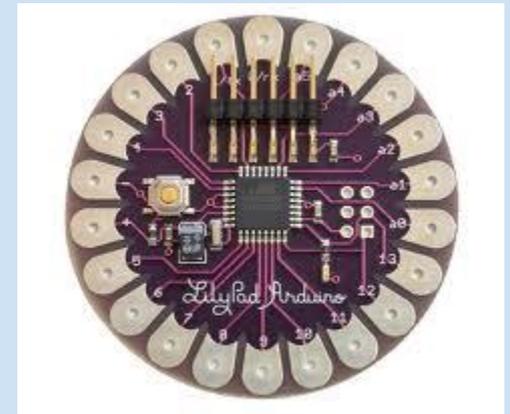
E' pensata per essere cucita sui tessuti, basta filo conduttivo per i collegamenti per capi intelligenti

ATmega 328

8 MHz

14 pin **digitali** di Input/Output

( di cui 6 pin PWM e 6 di IN **analogico** )



# Arduino nano: le dimensioni contano

43mm x 18mm

connettore mini USB

i componenti sono distribuiti su entrambe le facce dell

