



Progetti reali con ARDUINO

Introduzione alla scheda Arduino (parte 1^a)

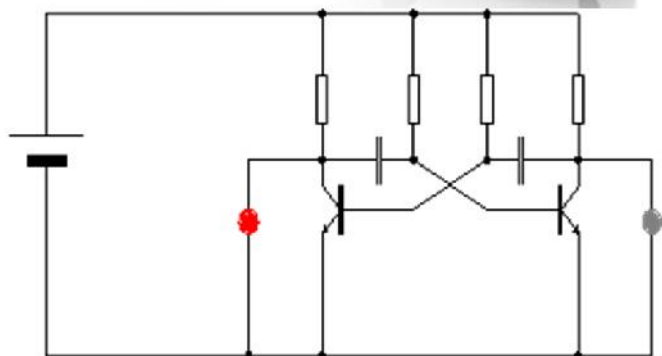
gennaio 2015 – Giorgio Carpignano

I.I.S. PRIMO LEVI

C.so Unione Sovietica 490 (TO)

Materiale didattico:

www.istitutoprimelevi.gov.it





Una parola sulla sicurezza

I componenti elettronici sono **tossici**: alcuni contengono Piombo e altri metalli altamente inquinanti (non disperdere!)

Non trascinate i vostri piedi sulla moquette: alcuni componenti elettronici sono molto sensibili ai campi elettrostatici

Effettuare e/o modificare i collegamenti solo quando la scheda Arduino è scollegata dall'alimentazione

Cos'è Arduino?

È un progetto **Open Source** (sei libero di utilizzare e modificare anche il software e le librerie)

È una piccola scheda con un suo ambiente di programmazione e una sua filosofia di sviluppo

È possibile programmare con un piccolo computer

È completamente “**stand-alone**” (funziona anche in modo autonomo dopo la programmazione)

Può comunicare con una moltitudine di altri dispositivi sia di “**input**” che di “**output**”

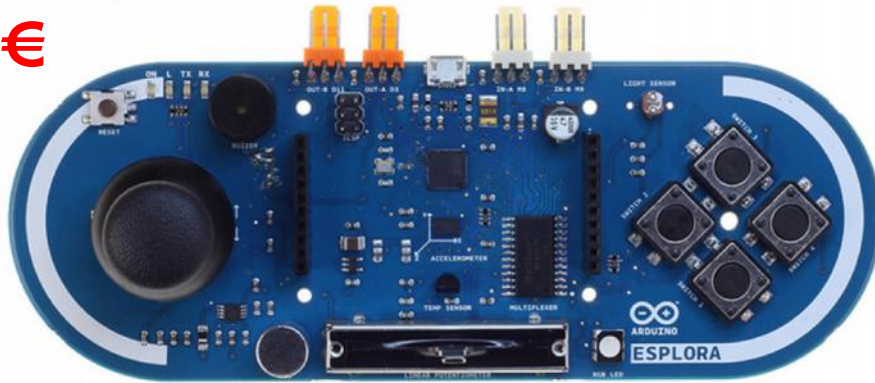
Può lavorare con una piccola batteria da **9V**

Può parlare (comunicare!) con altri computer, telefoni cellulari, ecc.

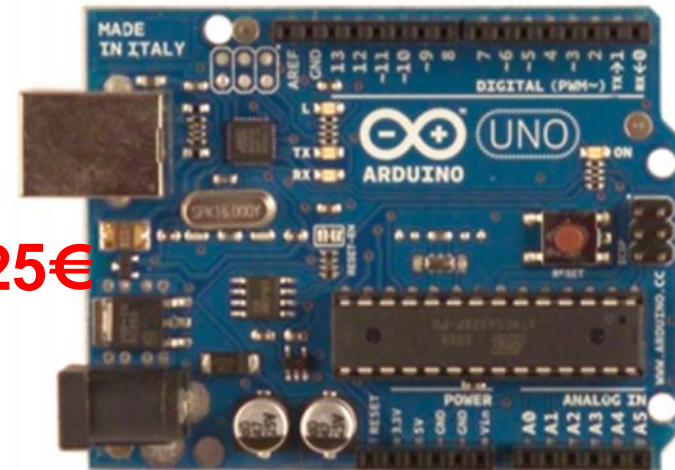
Cos'è Arduino?

- Elaborazione (tutti i progetti sono open source)
- Economico, più veloce e aperto (software a costo zero)
- Utilizza l'**ATmega328** come chip del microcontrollore (altri micro: "PIC", "8031", etc.)

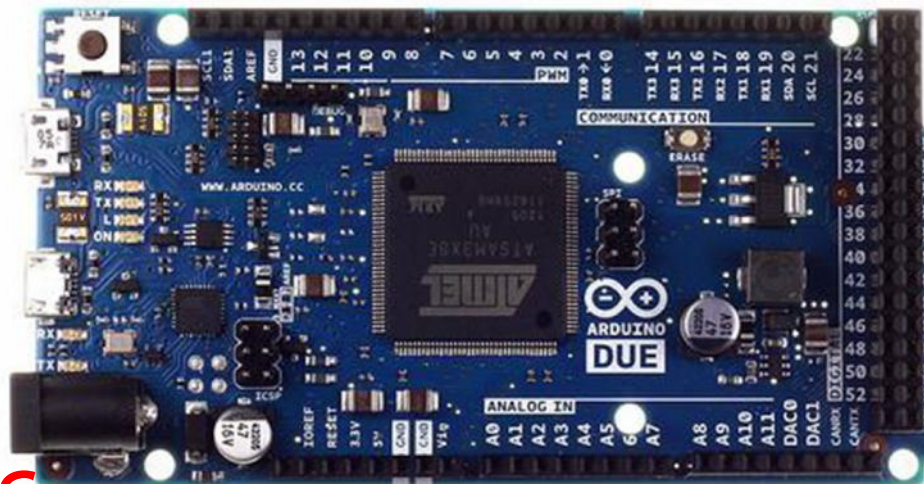
49€



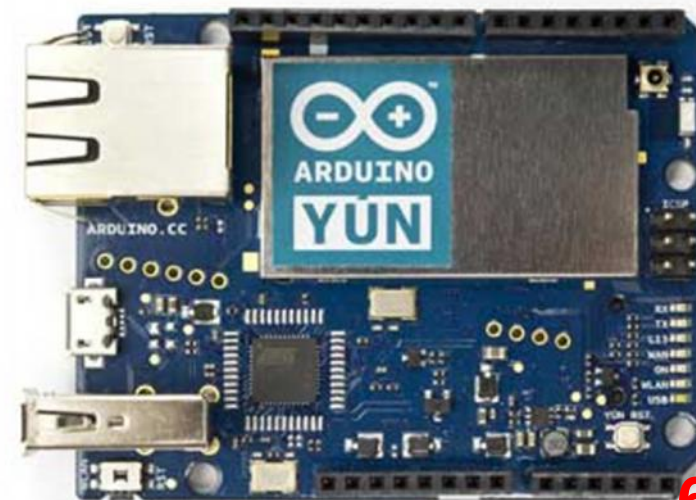
25€



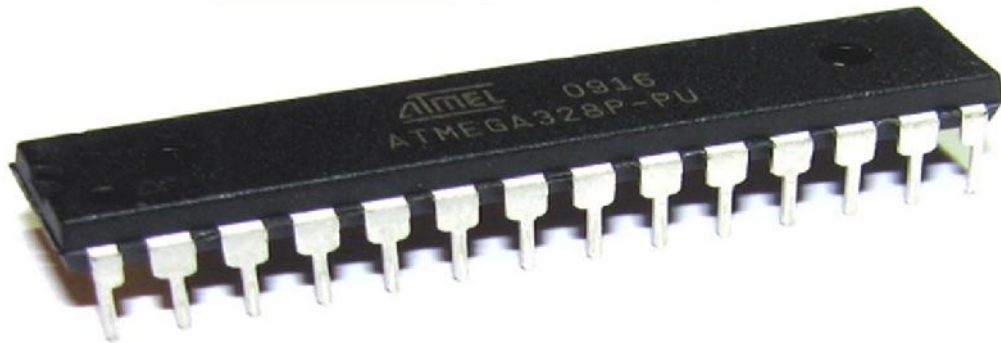
44€



64€



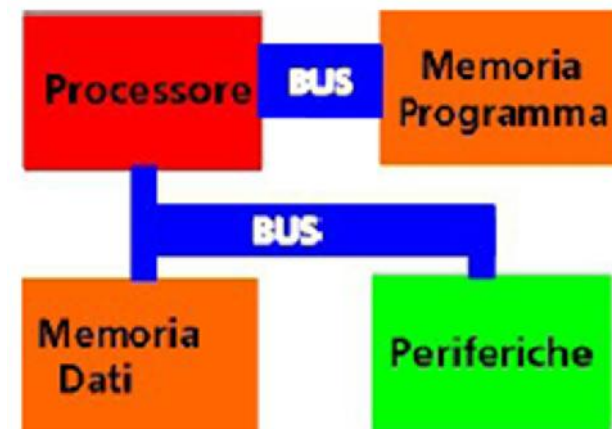
Cos'è Arduino?



Perché non basta usare solo l'integrato ATmega328?

Nel processore è memorizzato un piccolo programma di **boot** che serve a caricare nella memoria flash del microcontrollore lo **sketch** (programma) creato con l'ambiente di sviluppo su Pc, successivamente **il software appena caricato sul microcontrollore viene eseguito dopo i primi 5 secondi.**

- Il processore è di tipo **RISC** e una istruzione macchina viene eseguita mediamente in un solo ciclo di clock.
- L'architettura è di tipo **Harvard** con canali di comunicazione dati e istruzioni separati.



Caratteristiche di Arduino

ARDUINO UNO:

8-bit microcontroller
ATMEGA328P

Alimentazione: 3,3V accetta
anche 5V

32 Kbyte di FLASH

2 Kbyte di RAM

1 Kbyte di EEPROM

Frequenza clock = 16 MHz

- 14 pin configurabili come ingressi / uscite digitali (0V oppure 5V)
- 6 pin di ingressi analogici ADC (0V÷5V) Risoluzione: 10 bit (da 0 a 1023 valori)

ARDUINO DUE:

32-bit microcontroller
AT91SAM3X8E

Alimentazione: 3,3V

512 Kbyte di FLASH

96 Kbyte di RAM

Frequenza clock = 84 MHz

- 54 pin configurabili come ingressi / uscite digitali (0V oppure 3,3V)
- 12 pin di ingressi analogici ADC (0V÷3,3V) Risoluzione: 12 bit (da 0 a 4095 valori)
- 2 pin di uscite analogiche DAC (0V÷3,3V) Risoluzione: 12 bit

Caratteristiche di Arduino

ARDUINO UNO:

- 6 pin di uscite analogiche simulate con tecnica PWM (0÷5V) Risoluzione: 8 bit (da 0 a 255 valori)
- 1 porta USB
- **source digital output:** 3mA oppure 40mA (max.)
- **sink digital output:** 6mA oppure 40mA (max.)

ARDUINO DUE:

- 11 pin di uscite analogiche simulate con tecnica PWM (0÷3,3V) Risoluzione: 8/10/12 bit
- 2 porte USB
- **source digital output:** 3mA oppure 15mA (max.)
- **sink digital output:** 6mA oppure 9mA (max.)

Gli I/O (Input/Output) digitali sono in grado di controllare interruttori, pulsanti, led, motori e altro.

Gli ingressi analogici sono in grado di leggere la posizione di manopole o altri sensori diversi.

Le uscite analogiche vengono “simulate” con la tecnica digitale denominata **Pulse With Modulation (PWM)**.

La neonata scheda «Arduino Yún»

Arduino YUN si distingue dagli altri modelli perché può comunicare con la distribuzione **Linux** basata su OpenWRT chiamata «**Linino**», in grado di assicurare tempi di startup rapidi, installata sul processore dell'Atheros AR9331.

Offre così un potente computer connesso con il microcontrollore Arduino in grado di fornire:

Collegamento ad una rete wireless e rete Lan ethernet

Creare un web server

Creare un data logger

Gestione dispositivi USB



Caratteristiche della scheda «Arduino Yún»

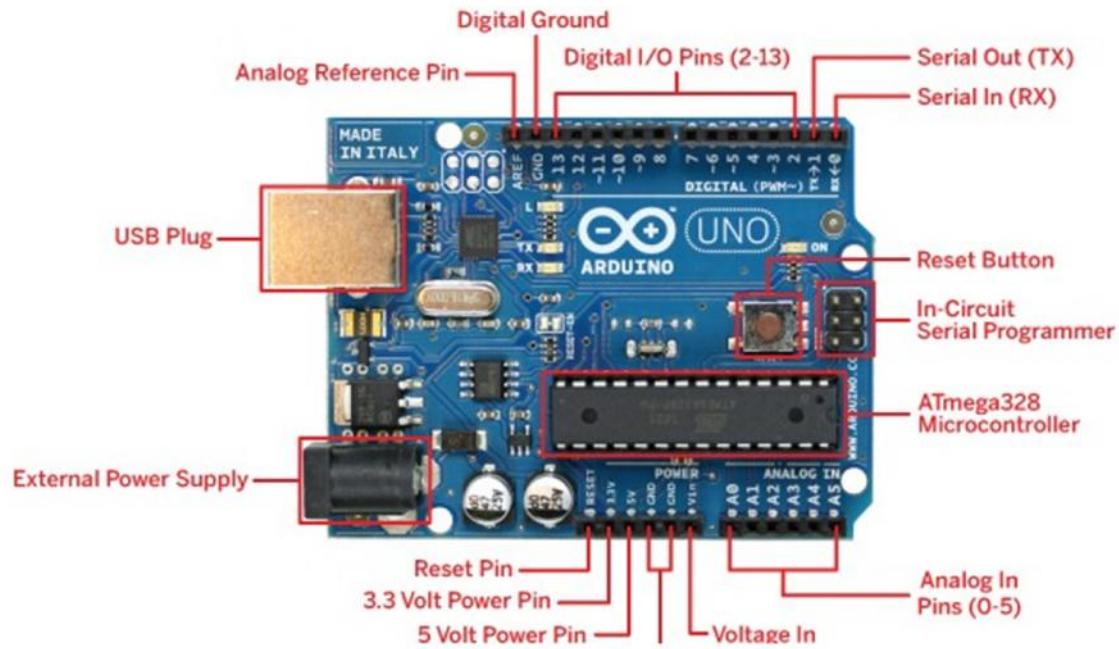
AVR Arduino microcontroller		Linux microprocessor	
Microcontroller	ATmega32u4	Processor	Atheros AR9331
Operating Voltage	5V	Architecture	MIPS @400MHz
Input Voltage	5V	Operating Voltage	3.3V
Digital I/O Pins	20	Ethernet	IEEE 802.3 10/100Mbit/s
PWM Channels	7	WiFi	IEEE 802.11b/g/n
DC Analog Input Channels	12	USB Type-A	2.0 Host/Device
Current per I/O Pin	40 mA	Card Reader	Micro-SD only
DC Current for 3.3V Pin	50 mA	RAM	64 MB DDR2
Flash Memory	32 KB	Flash Memory	16 MB
SRAM	2.5 KB	PoE compatible	802.3af card support
EEPROM	1 KB		
Clock Speed	16 MHz		

Come funziona Arduino?

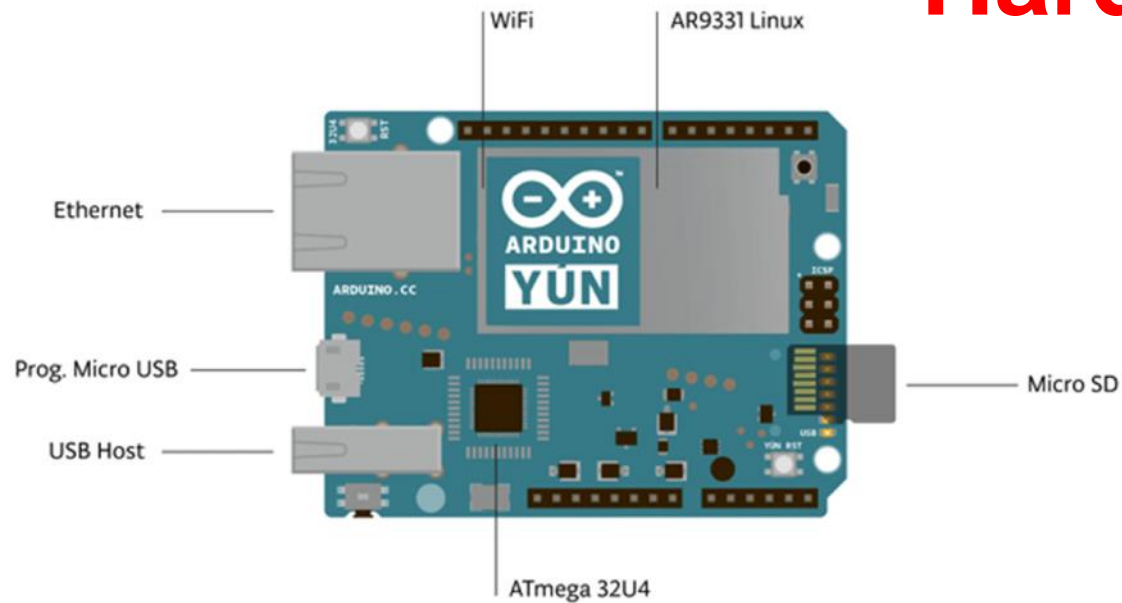
Ma come si programma la scheda?

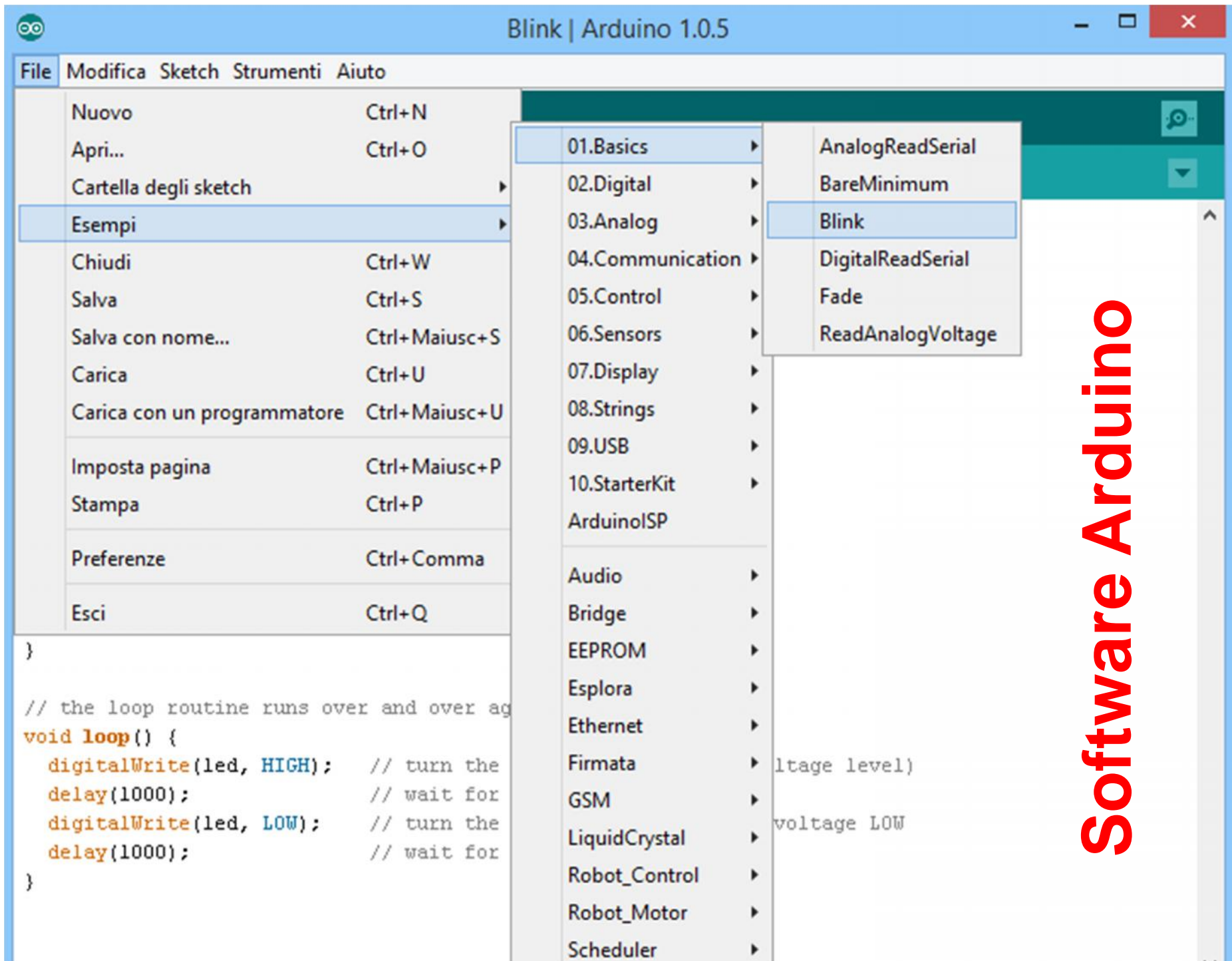
È senza tastiera, mouse e schermo

- Occorre scrivere e compilare il programma sul PC con un apposito software **IDE** (**I**ntegrated **D**evelopment **E**nvironment)
- Scaricarlo sulla memoria **flash** della scheda Arduino e eseguirlo
- La scheda Arduino può essere usata senza il PC.



Hardware Arduino

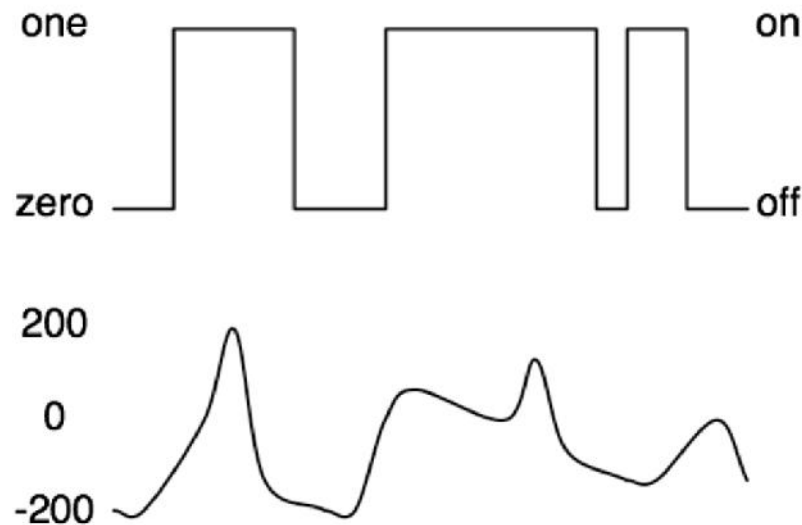




Software Arduino

Digitale o Analogico?

- **Digitale** possiede solo due valori: **0** e **1** (**Basso** o **Alto**)
- **Analogico** - ha molti (infiniti) valori



I computer in realtà non lavorano con tensioni analogiche. Nella conversione da analogico a digitale (ADC) si perdono delle informazioni.

Più elevato è il numero dei bit utilizzato nella conversione minore sarà l'errore commesso.

Arduino Uno Software

```
Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink $
Turns on an LED on for one second,
then off for one second, repeatedly.
This example code is in the public domain.
*/
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

Done uploading.
Binary sketch size: 1026 bytes (of a 30720 byte maximum)

6 Arduino Duemilanove w/ ATmega328 on COM4
```

Questo è il codice completo per un LED lampeggiante.

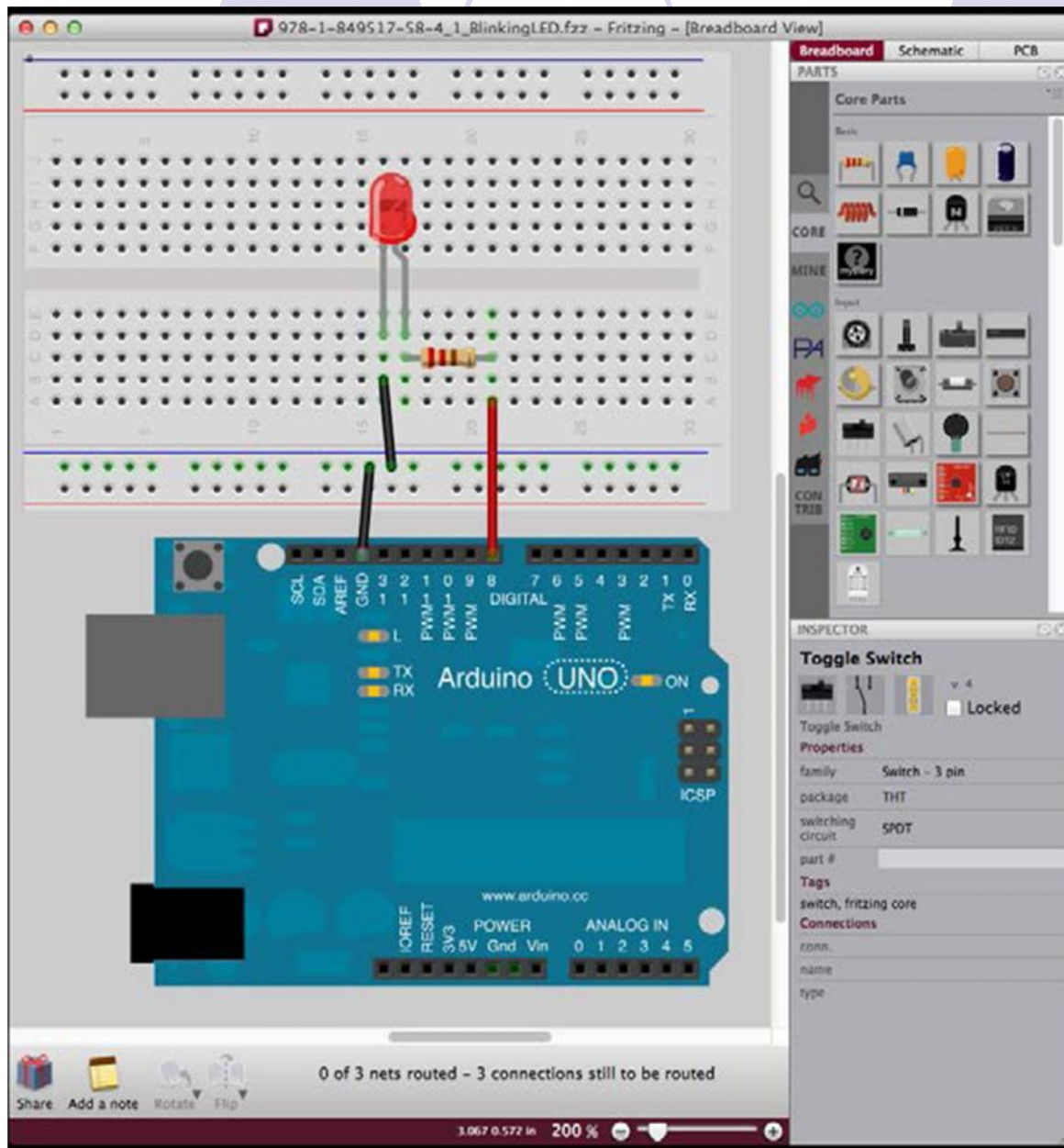
Arduino definisce le varie funzioni utili come `digitalWrite()` e `delay()`. Se ne riparla più avanti.



Arduino & Processing

- Visita il sito di riferimento: <http://processing.org/>
- Permette di costruire un applet che gira sul Personal Computer quindi non serve per la scheda Arduino
- **Processing** possiede la stessa interfaccia grafica di elaborazione di Arduino ma utilizza un set di istruzioni differenti da quelle necessarie per la scheda Arduino
- Permette di controllare con il mouse e/o la tastiera uno o più dispositivi di input/output presenti sulla scheda Arduino e viceversa.

Cos'è «Fritzing»?



Fritzing può essere definito come un progetto o iniziativa "open source" in grado di supportare i progettisti hardware, artisti, ricercatori e hobbysti a realizzare disegni elettronici creativi e di facile utilizzo.

(<http://fritzing.org>)

Installa il software per Arduino Uno

Effettua il download del software da:

www.istitutoprimelevi.gov.it → “**Studenti**” →
“**Progetti e Lavori**” oppure da www.arduino.cc/

È disponibile per:

- Windows XP / Vista / Windows 7 e 8
- Mac OS X PPC

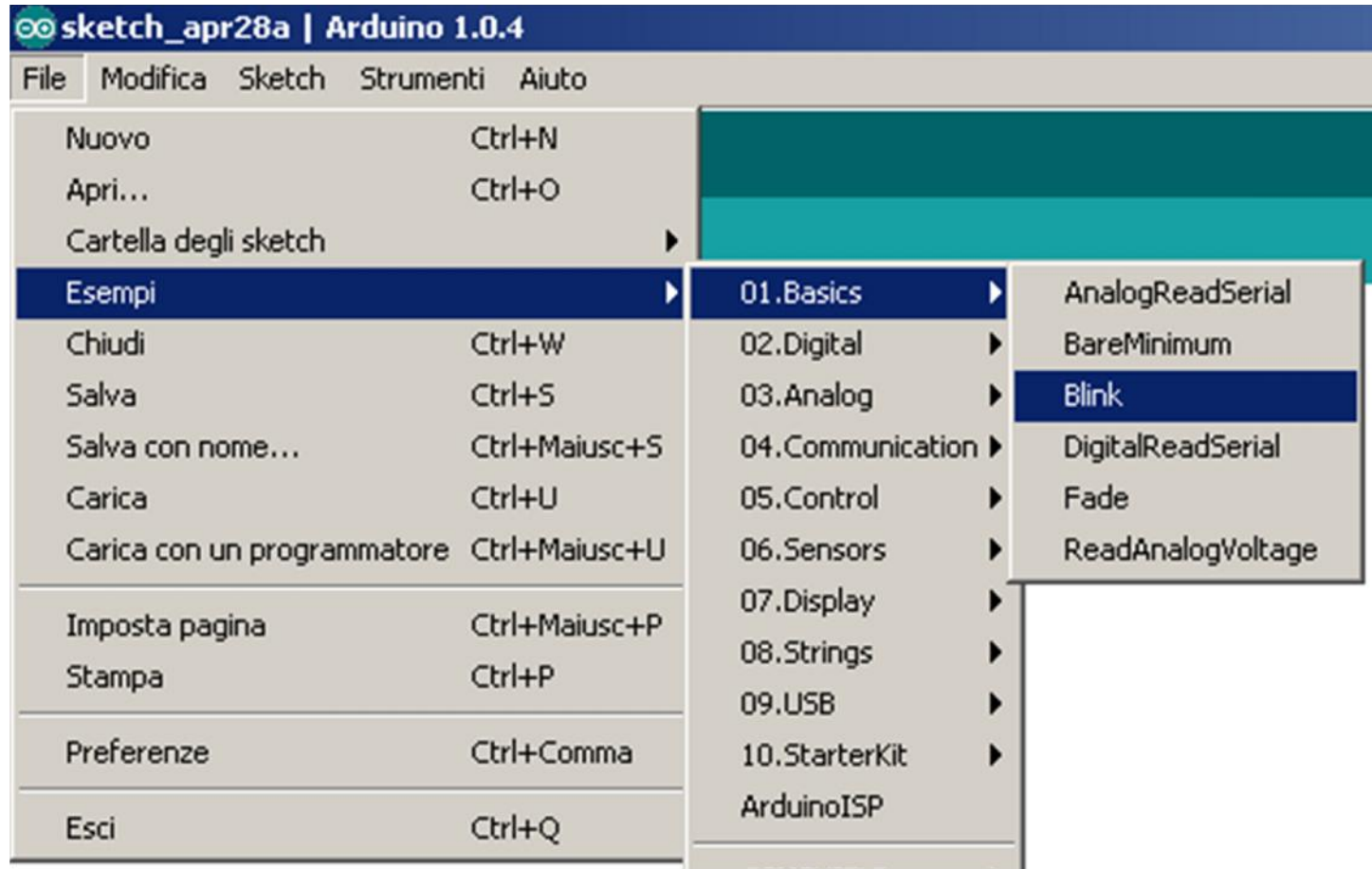
Come installare i driver

- Nella cartella "drivers", scegliere quella appropriata
- Windows: decomprimere il driver, collegare la scheda e selezionare la cartella del driver.

Utilizzo di esempi “on line” con Arduino Uno

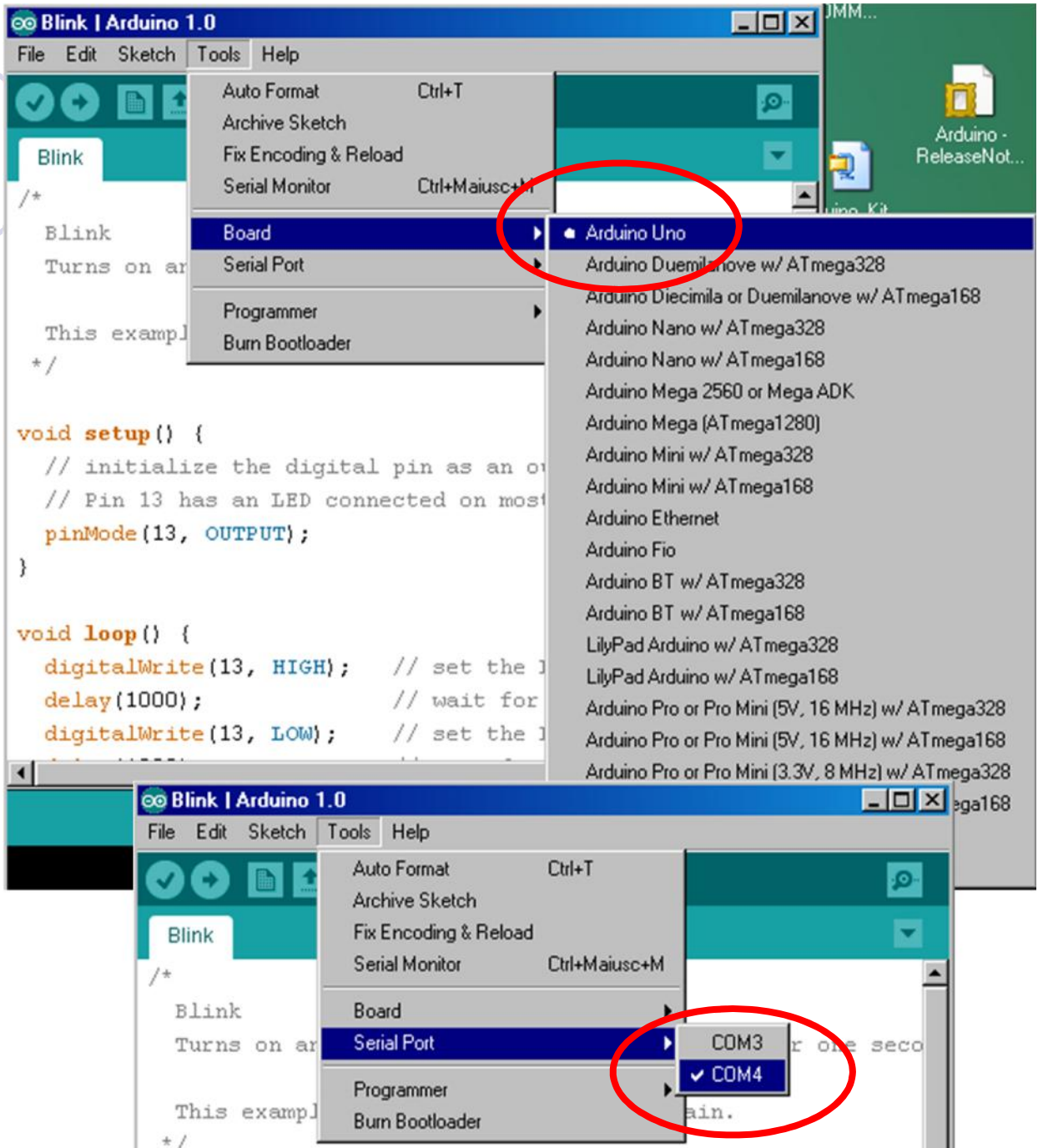
I programmi sono chiamati “**sketch**” (**schizzo**)

Come visualizzare uno **sketch** di esempio denominato
“**Blink**”

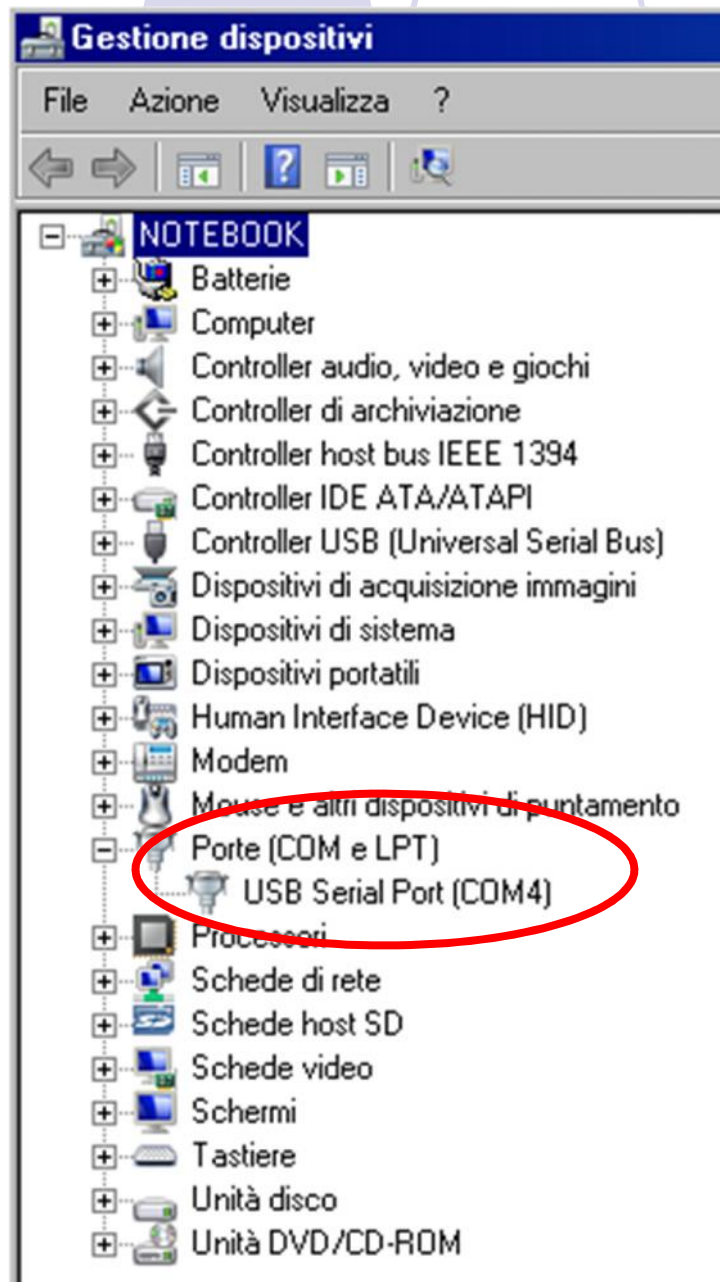


Errori

È necessario selezionare la porta seriale e il tipo di scheda utilizzata se la scheda Arduino **“non risponde”**



Qual è la mia porta seriale?



Da

→ «**Pannello di Controllo**»

→ «**Sistema**»

→ Utilizzare «**Gestione dispositivi**»

per conoscere la porta **COM**
utilizzata dalla scheda
Arduino (nella figura viene
utilizzata la porta COM4)

Utilizzare la scheda Arduino Uno

Scrivere il programma utilizzando

“**Blocco Note**” oppure **IDE di Arduino**

(l'ambiente di sviluppo integrato IDE di Arduino è un'applicazione multiplatforma scritta in **Java**)

Salvarlo con estensione “**.INO**” (nella vecchia versione si chiamava “**.PDE**”)

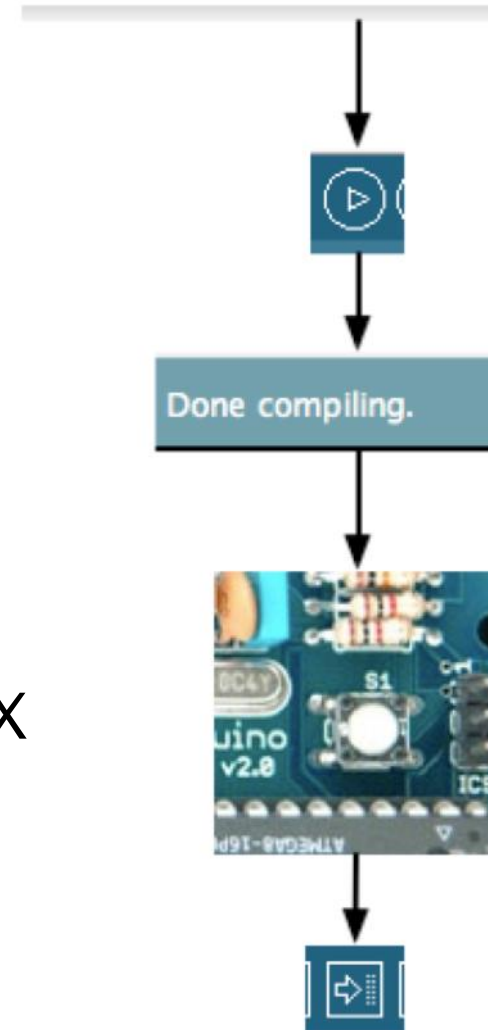
Compilarlo ovvero verificare la presenza di errori in modo da poterlo eseguire con un linguaggio macchina specifico del microcontrollore utilizzato (ATmega328)

Scaricare sulla scheda Arduino il software compilato.

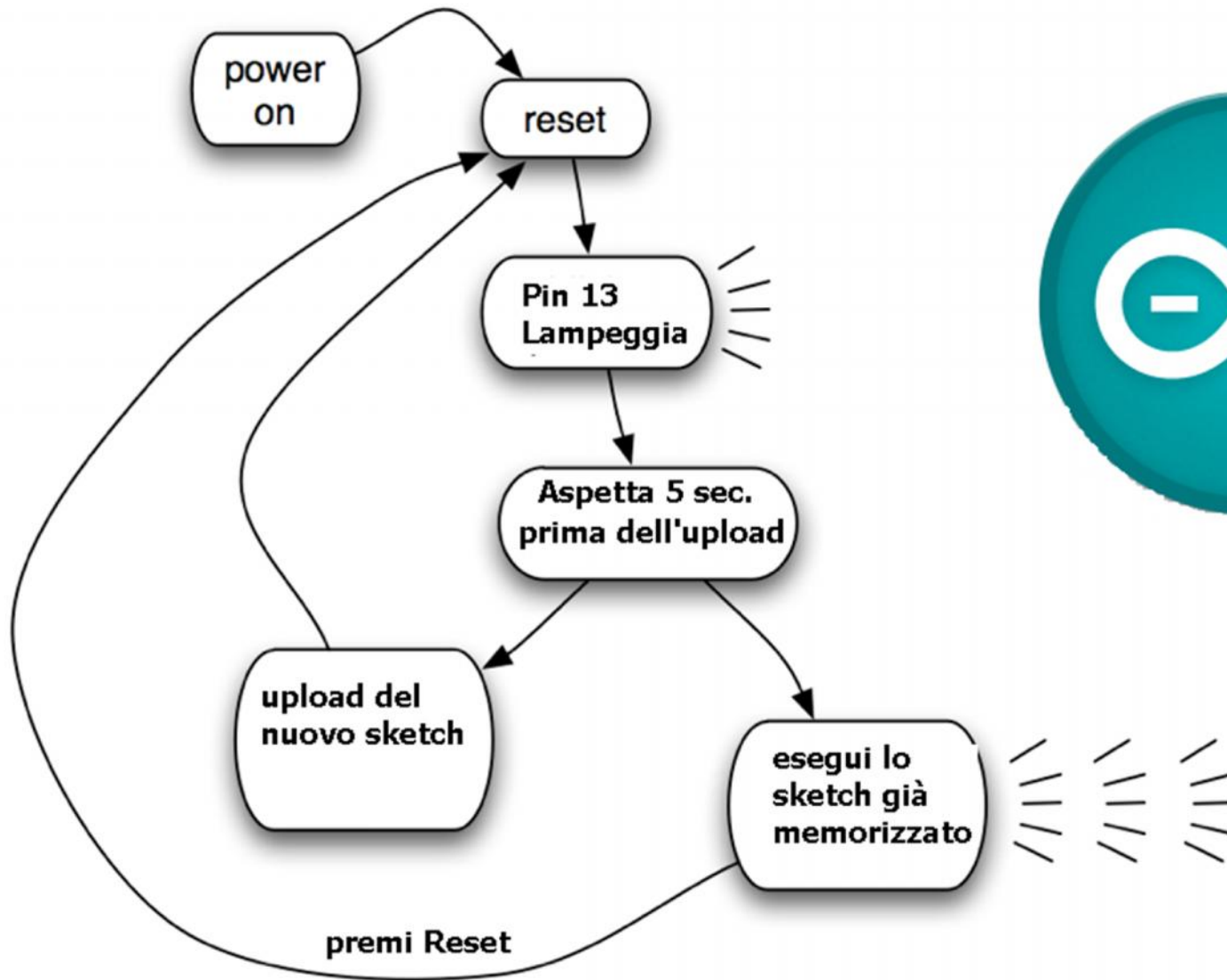
Durante il caricamento, i led denominati TX / RX lampeggiano per indicare che i dati sono trasferiti

Infine, il programma verrà eseguito (dopo circa 5 sec.)

```
void setup() {  
  pinMode(ledPin, OUTPUT);  
}  
void loop() {  
  digitalWrite(ledPin, HIGH);  
  delay(1000);  
  digitalWrite(ledPin, LOW);  
  delay(1000);  
}
```



Scheda Arduino ciclo di funzionamento

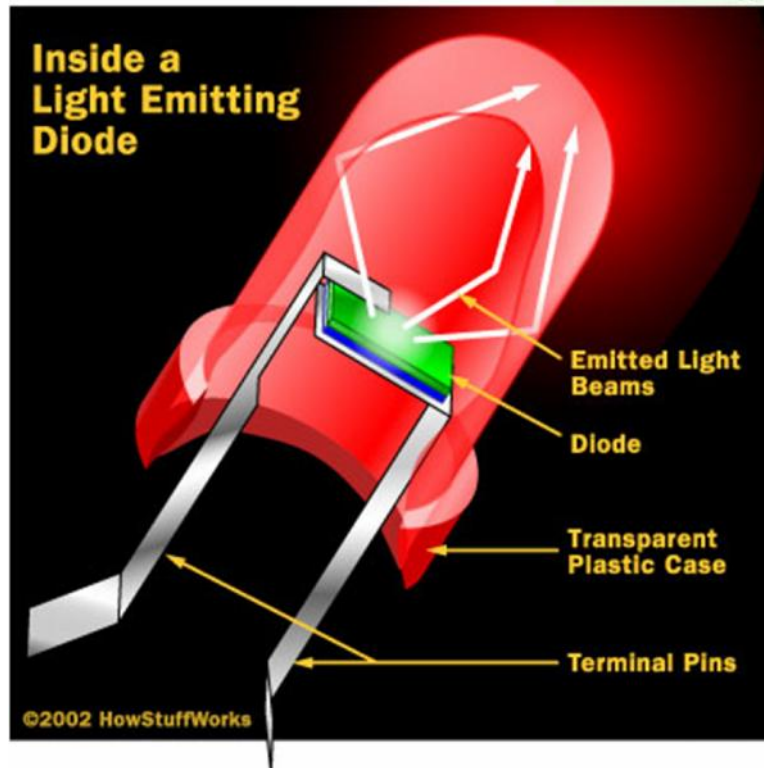
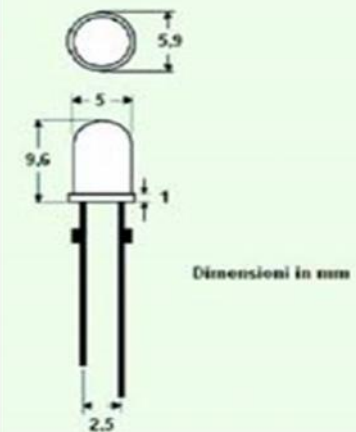
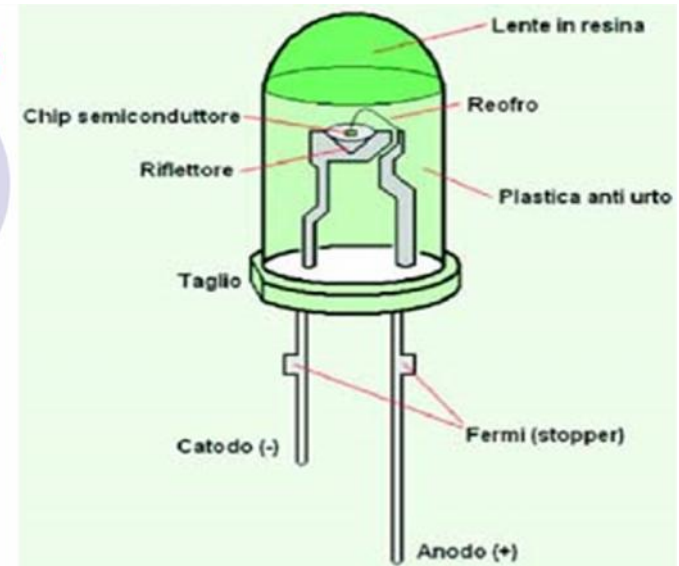


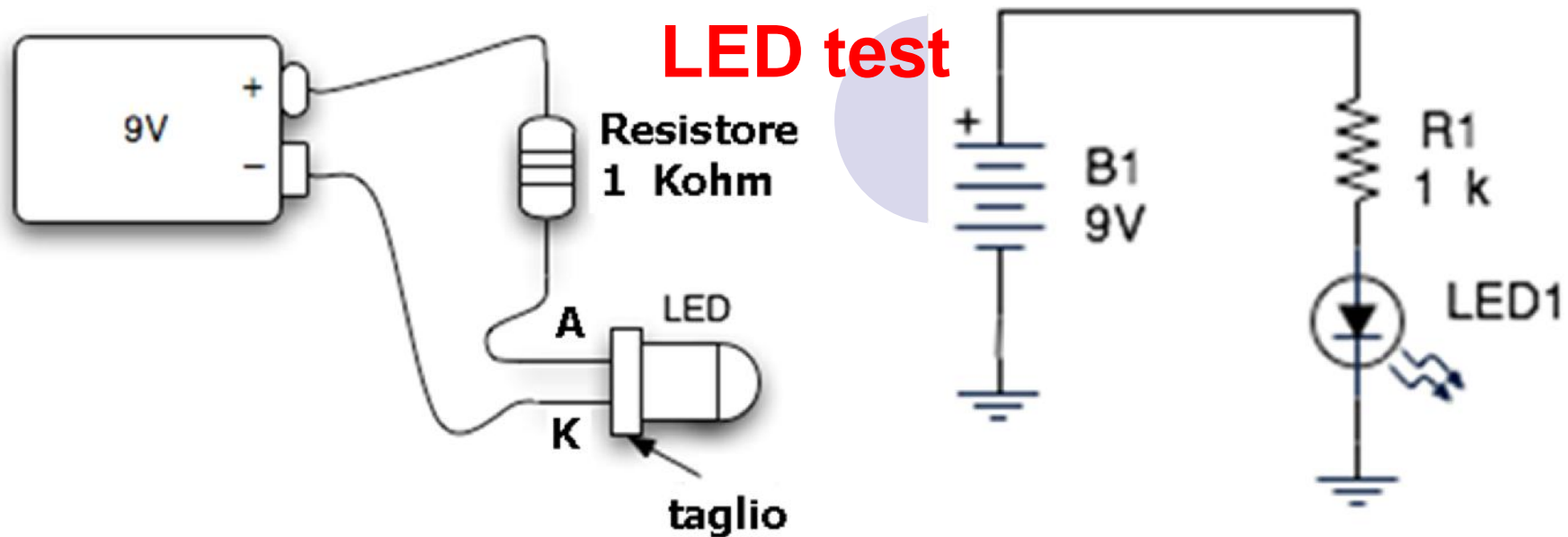
LED = Light-Emitting Diode

La corrente può scorrere solo in un verso nel diodo led (la tensione presente quando è illuminato è di circa 1,8V tra Anodo e il Catodo di un led colore rosso)

Occorre un resistore per limitare la corrente (valore max = 20 mA) da collegare in serie

Alcuni LED proiettano la loro luce con un fascio molto ristretto (15°), altri proiettano con un fascio molto ampio ($>60^\circ$)

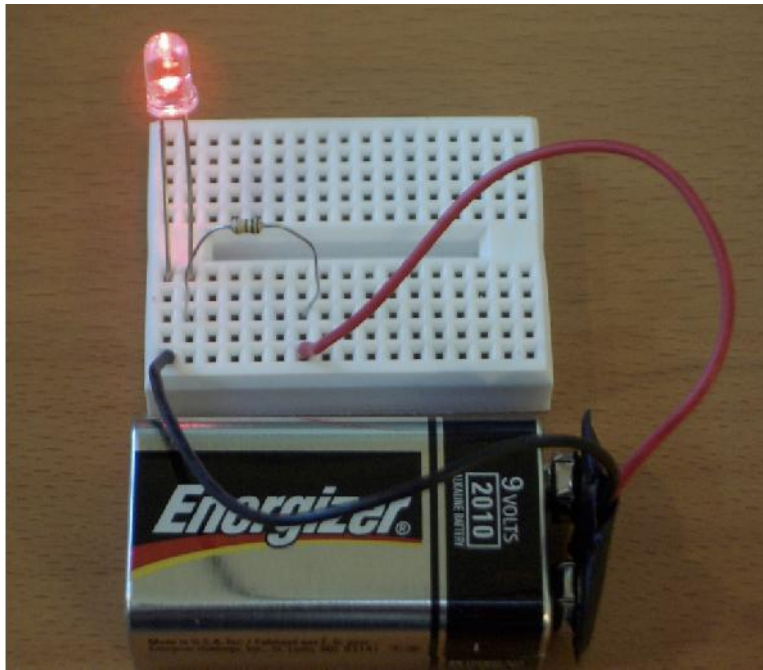




La parte piatta del LED (catodo) va collegata al polo negativo della batteria

Maggiore è la resistenza ($R1 = 1K = 1000 =$ marrone, nero, rosso, oro) minore sarà la corrente e la luminosità del LED

La particolarità del LED è quella di non possedere nessuna temporizzazione (**Ton** = led acceso e **Toff** = led spento), in pratica rimane acceso fino a quando non si interrompe il collegamento alla batteria da 9V



LED test

Montaggio e cablaggio del circuito precedente sulla breadboard senza saldature

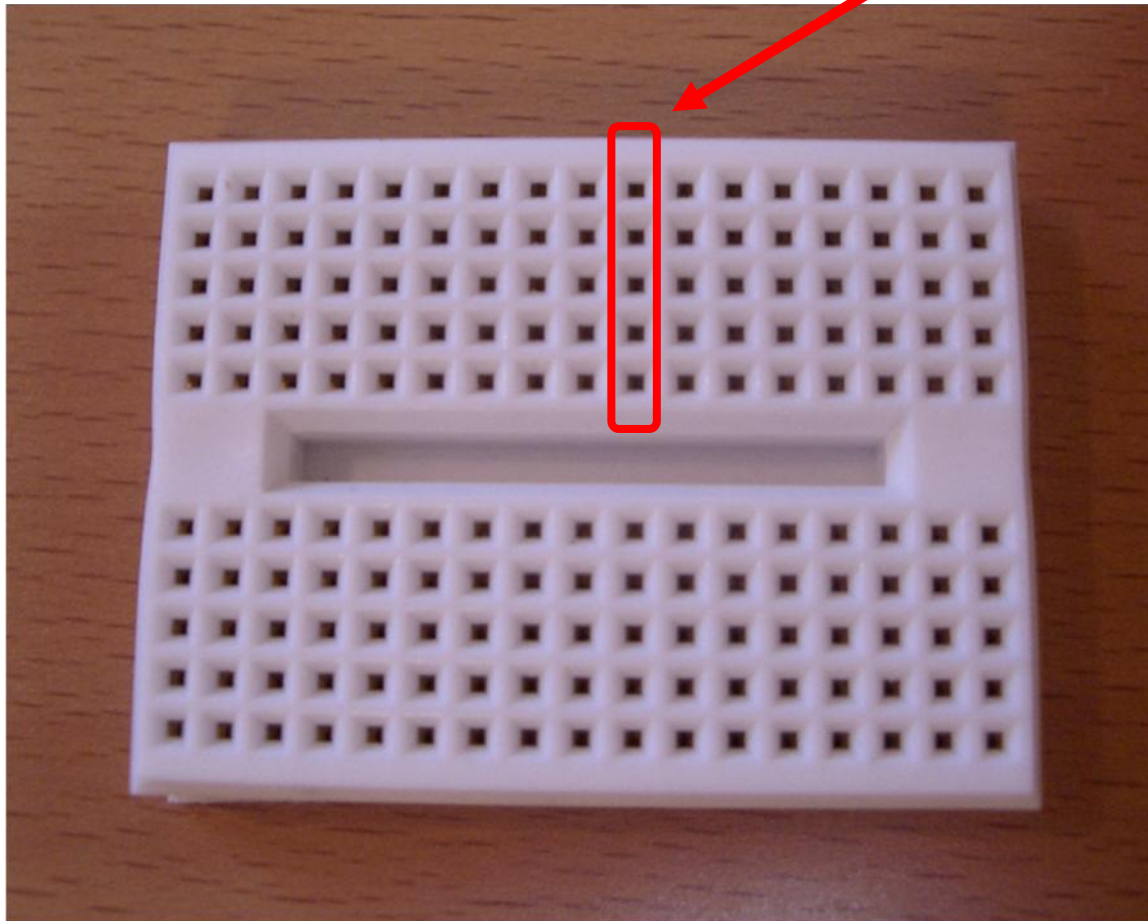
RESISTOR COLOR CODE

Examples:
 green-blue-brown - 560 ohms
 red-red-red - 2 200 ohms (2.2k)
 brown-black-orange - 10 000 ohms (10k)

0 - Black	5 - Green	20% - none
1 - Brown	6 - Blue	10% - silver
2 - Red	7 - Purple	5% - gold
3 - Orange	8 - Grey	
4 - Yellow	9 - White	

Codice colore per le resistenze

La Breadbord senza saldatura



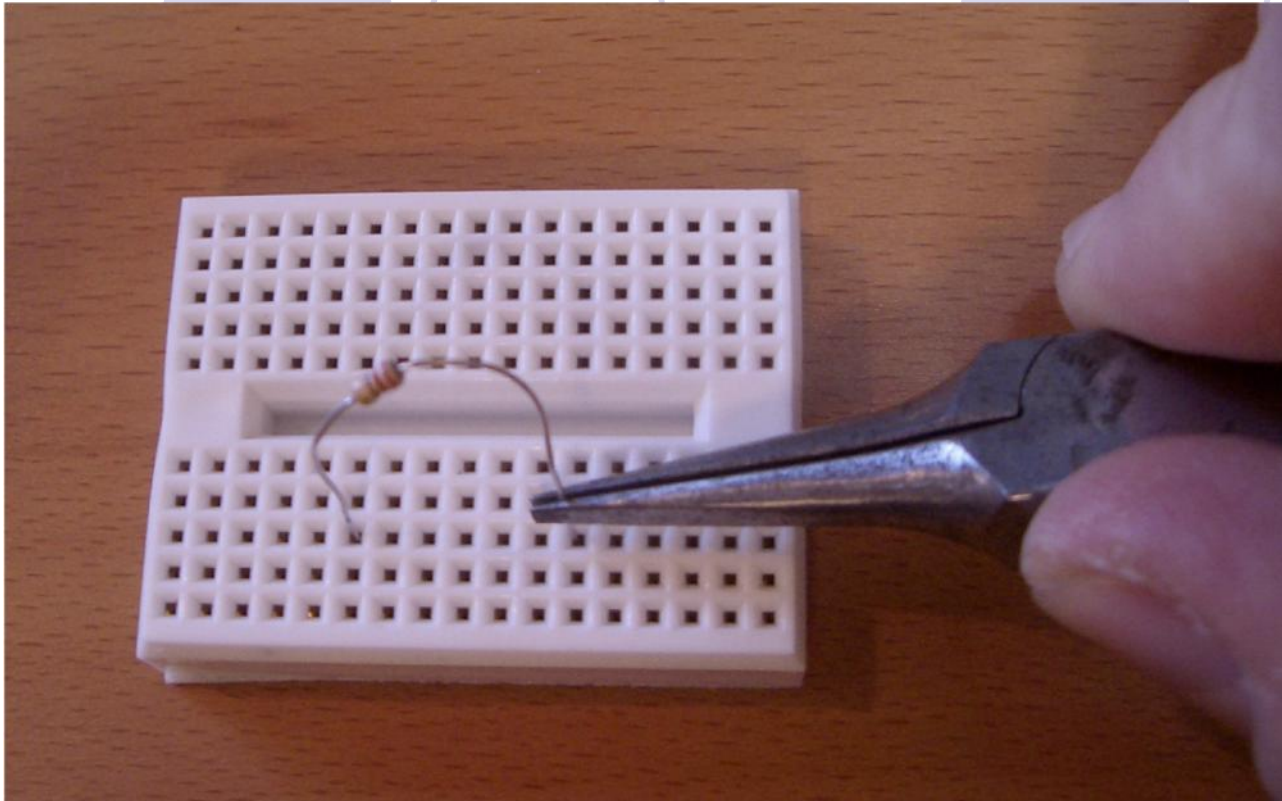
I gruppi di 5 contatti sono collegati tra loro nel senso verticale.

Inserire il cavo precedentemente spelato (max 1 cm) nei fori per effettuare una connessione.

È molto facile e risulta più veloce di una saldatura.

Purtroppo, le breadboard si usurano e sono costose (>10€).

Collegamenti senza saldatura sulla breadboard



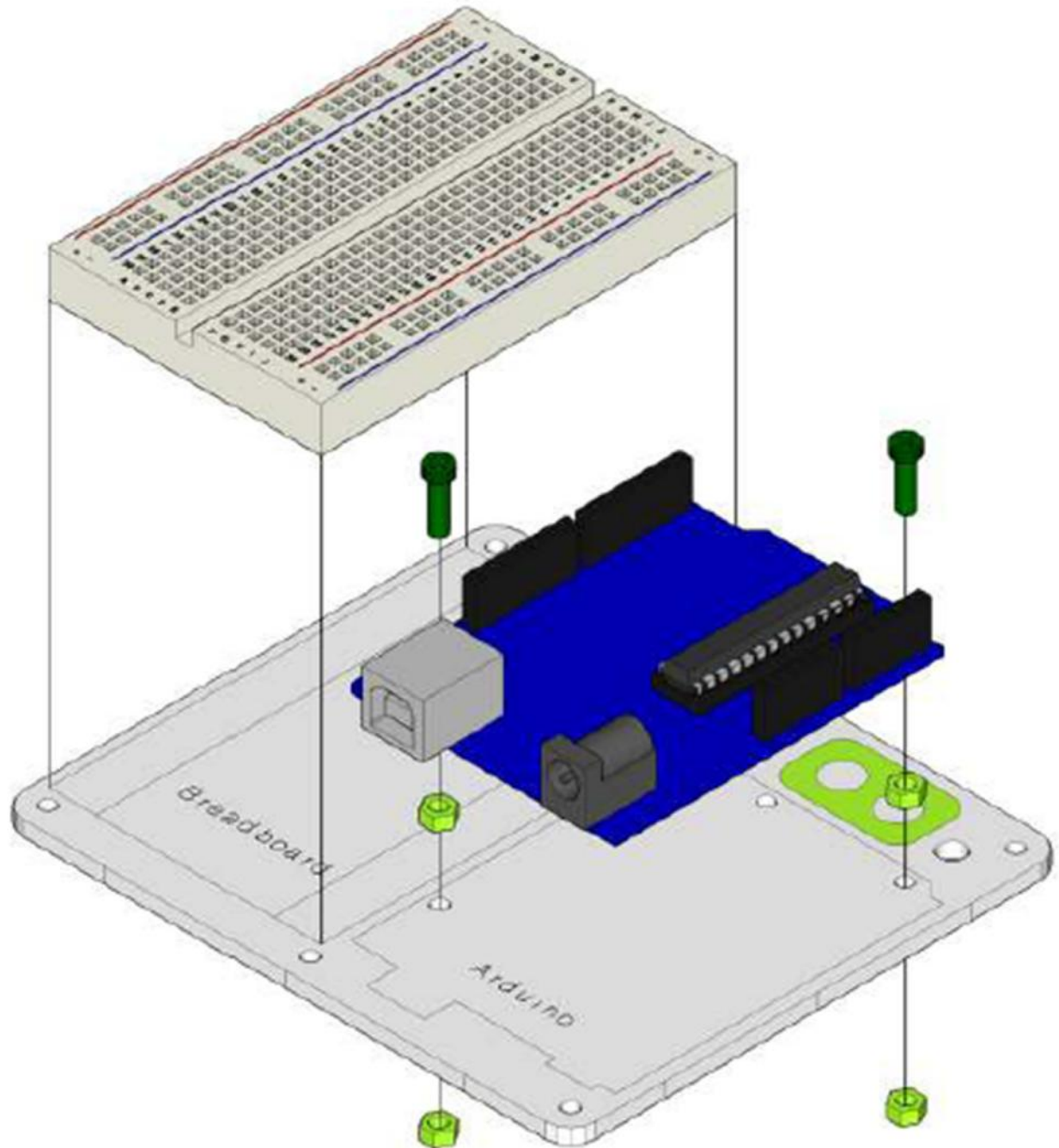
- Utilizzando pinze con becco sottile a punta ci si può aiutare nell'inserimento del contatto.

Inserito il filo, spingere il terminale verso il fondo del contatto interno.

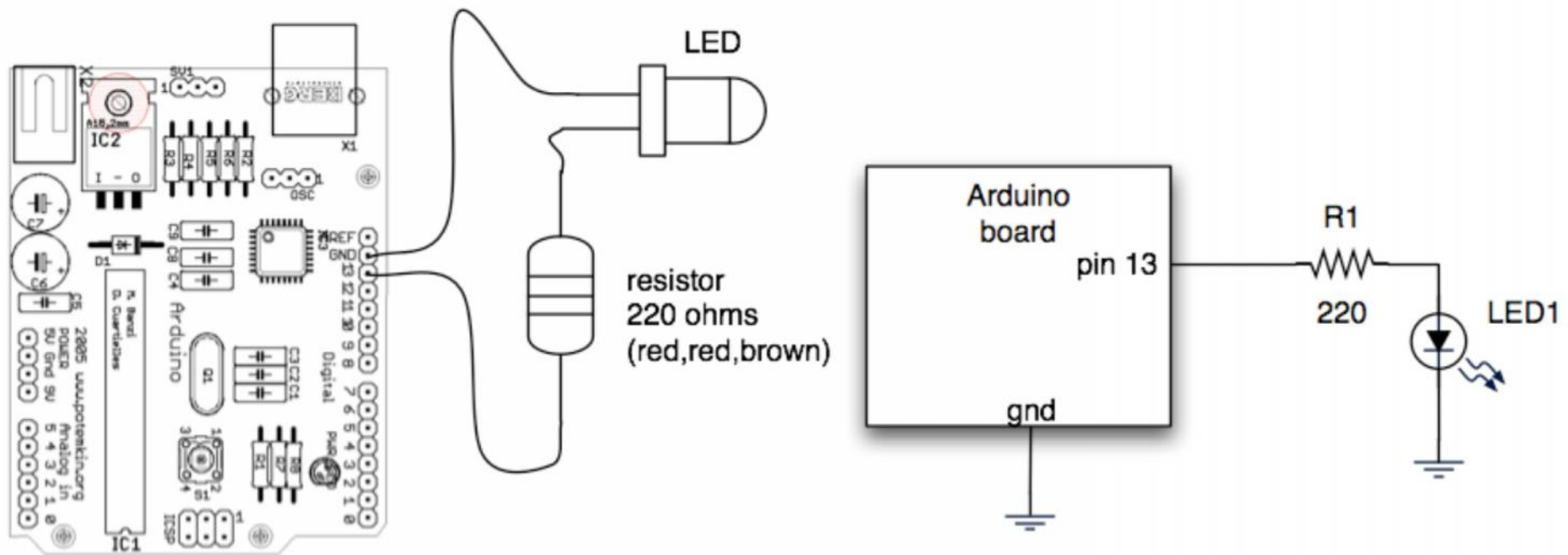
Consigli per un corretto utilizzo della scheda Arduino

Preparare una base di plastica, legno o metallo (se in metallo ricoprire con un foglio di plastica adesiva non conduttrice. Le dimensioni variano a seconda della breadboard)

- Montare e fissare la scheda Arduino
- Montare e fissare la bassetta breadboard



Circuito con LED lampeggiante



Negli schemi elettrici le tensioni più elevate (ad esempio: **Vcc = 5V**) vengono disegnate nella parte superiore dello schema.

I nodi comuni, come “**GND**” (**Ground**) sono collegati tutti allo stesso potenziale.

Utilizza il pin digitale 13 per collegare il led e la resistenza da 220 ohm (rosso-rosso-marrone-oro) in serie.

Software LED lampeggiante

Software per effettuare l'accensione ad intermittenza di un diodo LED

Denominazione del file «**Blink_1.ino**»
da scaricare dal sito:
www.istitutoprimorelevi.gov.it/node/324

```
/*  I.I.S. Primo LEVI - Torino
   Progetto: Blink_1
   Autore: Questo e' un esempio di pubblico dominio
   Descrizione: Accendi il LED per 0,2 secondi, in seguito spegnilo
   per 0,2 secondi e ripeti il ciclo all'infinito.
   Data: 03/12/2010 */
void setup() // funzione di configurazione dei Input/Output
{ // inizializza il pin 13 come output, perche' e' collegato un LED
  pinMode(13, OUTPUT);
}
void loop() // programma principale (main) --> ciclo infinito (loop)
{
  digitalWrite(13, HIGH); // accendi il LED forzando un livello ALTO sul pin 13
  delay(200);             // aspetta 0,2 secondi
  digitalWrite(13, LOW); // spegni il LED forzando un livello BASSO sul pin 13
  delay(200);            // aspetta 0,2 secondi
}
```



Arduino Sketch

Nel linguaggio C occorre:

- Dichiarare le variabili da utilizzare
- Inizializzare la scheda definendo gli input (ingressi) digitali / analogici e gli output (uscite).
- Funzione: **setup()** – viene eseguito una sola volta all’inizio per inizializzare le periferiche.
- Funzione: **loop()** – vengono eseguite ripetutamente tutte le istruzioni comprese tra le parentesi graffe.

“Linguaggio C” per Arduino Uno

Il linguaggio è “**C standard**” (più facile rispetto al C++)
possiede moltissime funzioni utili già implementate:

pinMode() - impostare un pin come ingresso o uscita

digitalWrite() - impostare un pin output digitale a livello alto / basso

digitalRead() - leggi lo stato di un pin definito come input digitale

analogRead() - legge e converte la tensione di un pin analogico in un valore numerico (10-bit)

analogWrite() - scrive un valore "analogico" con PWM (8-bit)

delay() - aspetta un periodo di tempo (espresso in millisecondi)

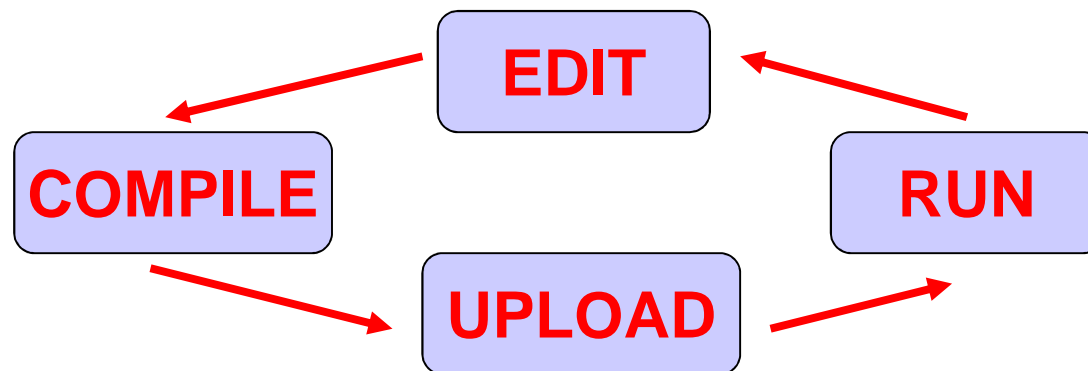
millis() – si ottiene il tempo da quando la scheda è stata accesa.

E molte altre funzioni, comprese le “**librerie**” (raccolta di funzioni necessarie per colloquiare con i dispositivi di input / output).

Ad esempio: libreria per i display LCD, servo, trasmissione / ricezione di dati seriali, ecc.

Ciclo di sviluppo del software

- Effettuare i cambiamenti che si desidera (**edit**)
- In seguito si effettua la compilazione del software (**compile**)
- Dopo si provvede a caricare sulla memoria flash della scheda Arduino (**upload**)
- Con un ritardo di 5 secondi circa il software viene messo in esecuzione sul microcontrollore ATmega328 (**run**)



Aggiungi hardware e software per controllare 2 LED

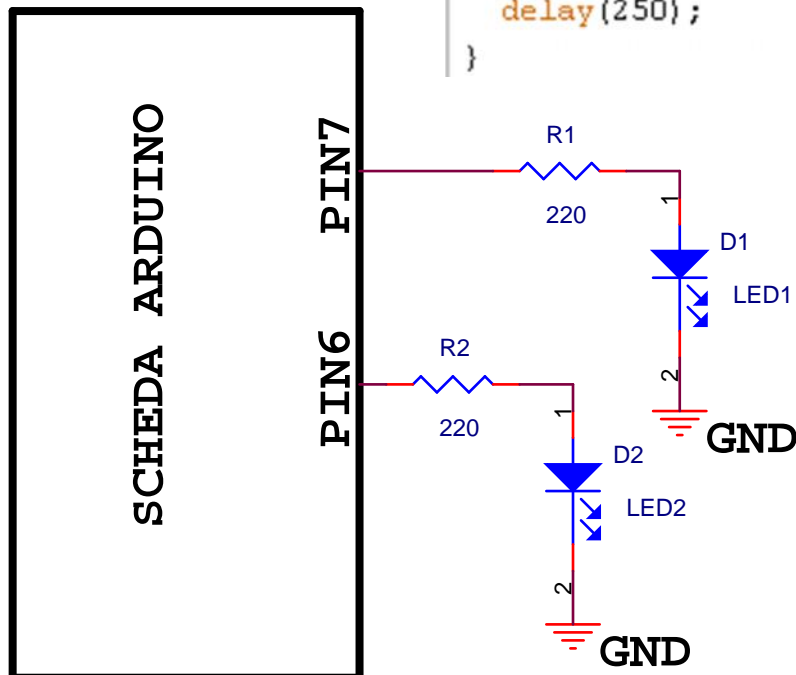
Esercizio N. 1 Data: 03/12/2010

Progetto: Blink_2

Autore: Questo e' un esempio di pubblico dominio

Descrizione: Accendere 2 LED alternativamente per 250 msec. */

```
void setup() // funzione di configurazione dei Input/Output
{ // inizializza il pin 7 e 6 come output, perche' sono collegati ai LED
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
}
void loop() // programma principale (main) --> ciclo infinito (loop)
{
  digitalWrite(7, HIGH); // accendi il LED1 forzando un livello ALTO sul pin 7
  digitalWrite(6, LOW); // spegni il LED2 forzando un livello BASSO sul pin 6
  delay(250); // aspetta 0,25 secondi
  digitalWrite(7, LOW); // spegni il LED1 forzando un livello BASSO sul pin 7
  digitalWrite(6, HIGH); // accendi il LED2 forzando un livello ALTO sul pin 6
  delay(250); // aspetta 0,25 secondi
}
```



Blink_2.ino
da scaricare dal sito:
www.istitutoprimolevi.gov.it/node/324