**Arduino lezione 02 facciamo lampeggiare un led**

<http://www.maffucci.it/2010/11/17/arduino-lezione-02-facciamo-lampeggiare-un-led/>

Pubblicato il [novembre 17, 2010](http://www.maffucci.it/2010/11/17/arduino-lezione-02-facciamo-lampeggiare-un-led/) da [admin](http://www.maffucci.it/author/admin/" \o "Visualizza tutti gli articoli di admin)

A distanza di qualche mese sono riuscito a sistemare gli appunti che ho realizzato in bozza per i miei studenti per continuare il corso su Arduino ed in questa lezione vi propongo il canonico esercizio “blink”, che trovate anche su Arduino.cc.  
Farò alcune variazioni ed integrazioni all’esempio in modo che possiate entrare nel dettaglio del linguaggio di programmazione.  
Userò in questa lezione la scheda Arduino Duemilanove e la versione dell’IDE n. 21 in ambiente Mac, ovviamente la programmazione è identica per tutti i sistemi operativi.

Il programma, sketch in Arduino, che vi consente di far lampeggiare un led è secondo me il primo programma che dovete realizzare per familiarizzare con l’ambiente di sviluppo ma soprattutto per verificare il funzionamento della scheda.

In questa lezione viene dato per scontato che lo studente conosca i concetti di:

* corrente
* tensione
* resistenza

inoltre sappia cosa sono e come funzionano i componenti:

* diodi led
* resistori

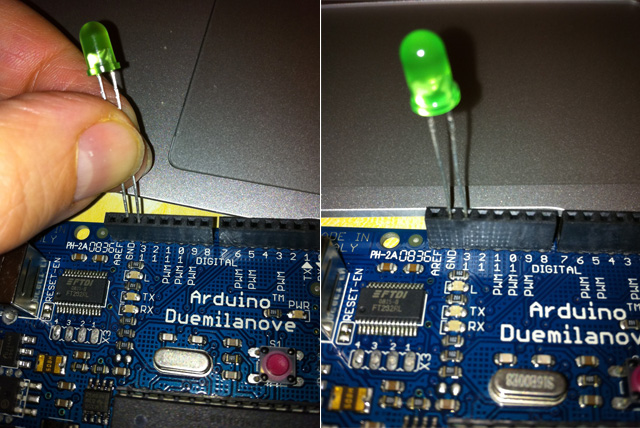
Se non avete queste nozioni di base nessun problema! [Scrivetemi](mailto:michele@maffucci.it) sarò ben lieto di spiegarvi in modo pratico e sintetico i concetti di elettronica necessari che vi permetteranno di proseguire agevolmente nel corso.

**Partiamo!**

Come potete notare sulla scheda è presente un piccolo diodo led indicato da una “L” che potete utilizzare come trasduttore per la realizzazione del vostro primo programma.



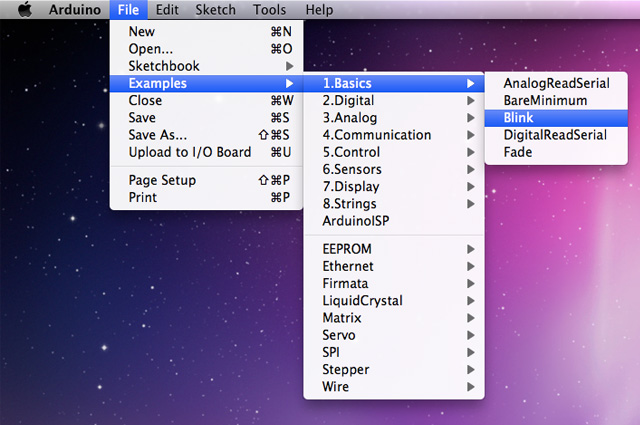
Se però volete incominciare con la pratica potete sfruttare le uscite digitali per pilotare un led più grande. Come mostrato nell’immagine che segue inserite il CATODO del diodo nel foro indicato con GND e l’anodo nel foro indicato con 13.

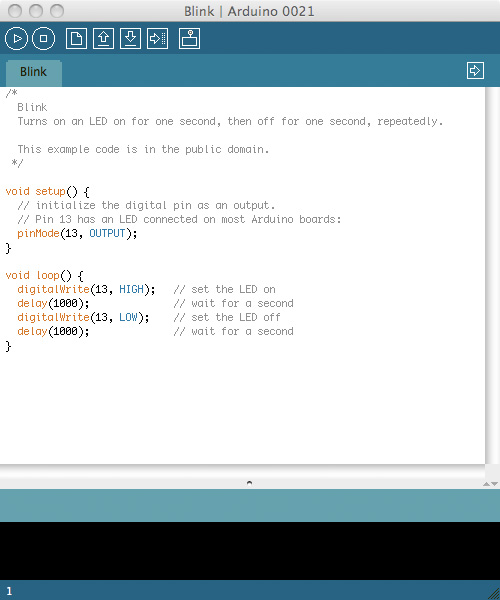


Aprite il programma Arduino se è la prima volta che avviate il programma verrà immediatamente aperto uno sketch vuoto, altrimenti dal menù **File > New** aprite un nuovo sketch e salvatelo con il nome che desiderate nella cartella in cui volete collezionare i vostri programmi. In ambiente Mac in fase di installazione viene creata in modo automatico**una cartella di nome Arduino** nella cartella “Documents” in questo modo se salvate senza indicare la cartella di destinazione ritroverete lo sketch creato in Documents > Arduino

Il programma [blink](http://arduino.cc/en/Tutorial/Blink) potete trovarlo sul sito Arduino.cc oppure direttamente dal vostro programma:

**File > Examples > 1.Basics > Blink**



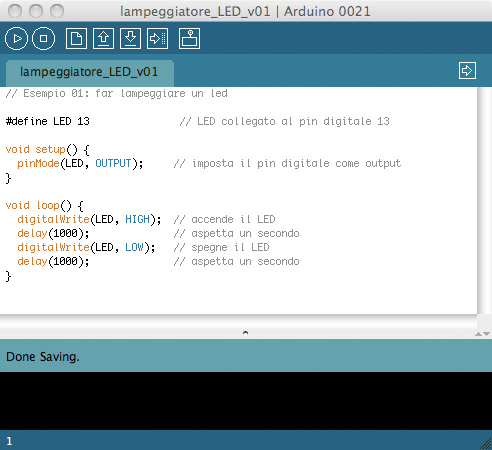


di seguito vi riporto la mia versione che ho rinominato “lampeggiatore led v01″ in cui ho inserito i commenti in italiano, se desiderate potete utilizzare la mia versione.

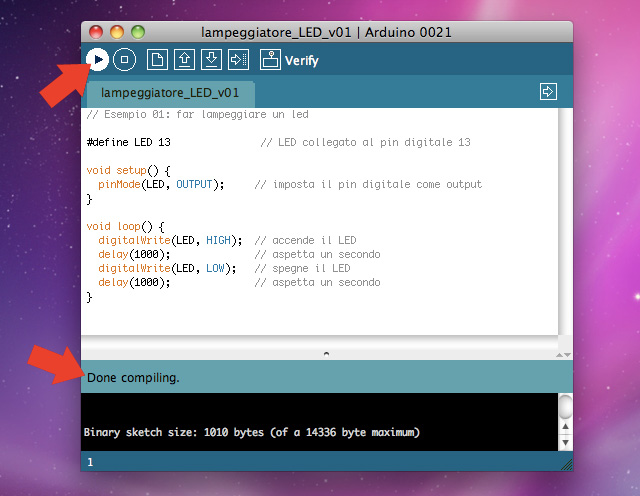
[view plaincopy to clipboardprint?](http://www.maffucci.it/2010/11/17/arduino-lezione-02-facciamo-lampeggiare-un-led/)

1. // Esempio 01: far lampeggiare un led
3. #define LED 13            // LED collegato al pin digitale 13
5. **void** setup() {
6. pinMode(LED, OUTPUT);     // imposta il pin digitale come output
7. }
9. **void** loop() {
10. digitalWrite(LED, HIGH);  // accende il LED
11. delay(1000);              // aspetta un secondo
12. digitalWrite(LED, LOW);   // spegne il LED
13. delay(1000);              // aspetta un secondo
14. }

Di seguito l’immagine della schermata del vostro primo sketch



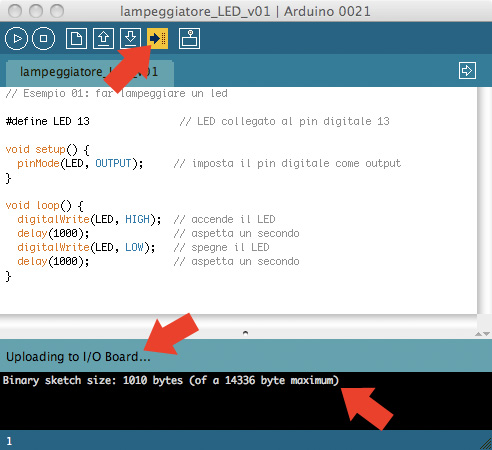
Ora dovete verificare il vostro programma facendo click su “**Verify**” e se tutto è corretto in basso all’IDE compare il messaggio “**Done compiling**” ciò indica che l’IDE Arduino ha tradotto il vostro sketch in un programma eseguibile sulla scheda Arduino.

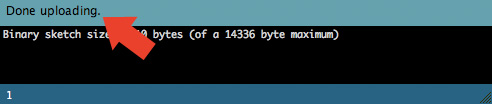


Ora dovete caricare sulla scheda il vostro programma facendo click su “**Upload to I/O Board**“, in questo modo la scheda interrompe l’esecuzione di ciò che stava eseguendo, carica il nuovo programma in memoria e lo esegue.

Durante la fase di trasferimento accadono due cose:

**01.** appare il messaggio: **Uploading to I/O Board…** e la dimensione in byte del vostro sketch e alla fine compare il messaggio: “**Done uploading**” che indica che il trasferimento del programma alla scheda ha avuto termine;





**02.** sulla scheda i led identificati con RX e TX lampeggiano, ogni volta che viene inviato un byte e durante la fase di uploading del programma lampeggiano continuamente.



**Analizziamo passo passo il codice**

Non abbiate paura l’analisi riga per riga vi porterà alla comprensione del linguaggio di programmazione anche se non siete programmatori.

// Esempio 01: far lampeggiare un led

Cone le due barre “//” indichiamo l’inizio di un commento e il testo che lo segue, in questo caso, identifica il titolo del nostro programma.  
Molto spesso è importante indicare subito dopo il titolo, la data dell’ultima variazione e il nome dell’autore, in modo che a distanza di tempo possiamo identificare le versioni di sviluppo ed il nome dell’ultimo programmatore che ha eseguito modifiche:

Ad esempio:

// Esempio 01: far lampeggiare un led

// Data ultima modifica: 16.11.2010

// Autore: Michele Maffucci

notate che ogni linea di commento deve essere preceduta da “//”

Se volete evitare di scrivere per ogni linea “//” potete far precedere il testo commento dai caratteri “/\*” e farlo concludere con i caratteri “\*/”

Ad esempio:

/\* Esempio 01: far lampeggiare un led

Data ultima modifica: 16.11.2010

Autore: Michele Maffucci \*/

#define LED 13 // LED collegato al pin digitale 13

[#define](http://arduino.cc/en/Reference/Define) consente di sostituire, in fase di compilazione, all’interno del vostro programma il valore 13 ogni volta che compare la parola LED.  
Vi ricordo che la compilazione avviene quando fate click su “Verify” oppure quando effettuate l’upload sulla scheda.

void setup()

[void](http://arduino.cc/en/Reference/Void) indica ad Arduino che stiamo dichiarando una funzione di nome setup(), cioè un porzione di codice che esegue un’operazione specifica

{

La [parentesi graffa aperta](http://arduino.cc/en/Reference/Braces) indica dove incomincia il codice.

pinMode(LED, OUTPUT); // imposta il pin digitale come output

[pinMode](http://arduino.cc/en/Reference/PinMode) è un’istruzione che dice ad Arduino come usare un determinato pin.  
Tra parentesi tonde vengono specificati gli argomenti che possono essere numeri e lettere.  
I pin digitali possono essere utilizzati sia come INPUT che come OUTPUT. Nel nostro caso poiché vogliamo far lampeggiare il diodo LED dobbiamo definire il pin di OUTPUT.  
Le parole INPUT e OUTPUT sono costanti definite, che non variano mai nel linguaggio di Arduino.

}

La [parentesi graffa chiusa](http://arduino.cc/en/Reference/Braces) indica la fine della funzione setup().

void loop()

Indica la sezione di codice principale, il nucleo del vostro programma, che verrà ripetuto all’infinito fino a quando non si spegne la scheda.

digitalWrite(LED, HIGH); // accende il LED

L’istruzione [digitalWrite](http://arduino.cc/en/Reference/DigitalWrite) possiede due argomenti, il primo definisce il pin, il secondo indica lo stato.  
digitalWrite è un’istruzione in grado di impostare un pin definito come OUTPUT ad un valore HIGH o ad un valore LOW, in modo più semplice permette di accendere o spegnere un led connesso al pin specificato nel primo argomento, nel nostro caso LED.

***Tenete conto che dire che su un determinato pin vi è uno stato HIGHT, vuol dire che su di esso viene applicata una tensione di +5 V, mentre se lo stato e LOW vuol dire che sul pin è applicata una tensione di 0V.***

delay(1000); // aspetta un secondo

[delay()](http://arduino.cc/en/Reference/Delay) è un’istruzione molto semplice che non fa altro che interrompere per un determinato tempo l’esecuzione del programma.  
L’istruzione ha un solo argomento numerico che indica il numero di millisecondi di attesa.  
Nel nostro caso con “delay(1000)” diciamo che il programma deve bloccarsi per **1000 millisecondi ovvero 1 secondo**.

Quindi guardando insieme le due istruzioni:

digitalWrite(LED, HIGH); // accende il LED

delay(1000); // aspetta un secondo

Diciamo ad Arduino di accendere il diodo led sul pin 13 per 1 secondo.

digitalWrite(LED, LOW); // spegne il LED

Ordiniamo al led sul pin 13 di spegnersi

delay(1000); // aspetta un secondo

tiene spento il diodo connesso al pin 13 per 1 secondo.

Ora per concludere questa seconda lezione vi invito ad eseguire i seguenti esercizi:

1. variate il tempo di accensione e spegnimento, con tempi uguali di accensione e spegnimento
2. variate il tempo di accensione e spegnimento, con tempi diversi di accensione e spegnimento
3. provate ad eseguire il programma con tempi inferiori ai 500 millisecondi, cosa succede?