



<http://www.marrazzoantonio.altervista.org>

## Facciamo lampeggiare un led

Passiamo a fare qualche esercizio, il più semplice per iniziare a programmare con Arduino è il canonico esercizio “blink”, che trovate anche negli esempi del programma. Si tratta di far lampeggiare un diodo led all’infinito con tempi prestabiliti.

Prima di affrontare l’argomento o le programmazione è opportuno che lo studente ripeti l’argomento diodo led e resistenza di caduta.

Per prima cosa dobbiamo collegare il diodo led, volendo potremmo anche utilizzare il famoso diodo led integrato sulla scheda e indicato con L, ma a noi non basta, così andremo a collegare all’uscita 13 un diodo led collegando in serie una opportuna rd (A verso il pin 13 e K verso massa), in alcuni testi troverete che la rd non viene utilizzata, ciò dipende dal diodo che utilizziamo e dall’intensità (luminosa) che vogliamo ottenere.

Passiamo alla programmazione

Apriamo il programma Arduino, se è la prima volta che avviamo il programma verrà immediatamente aperto uno sketch vuoto, altrimenti dal menù File > New possiamo aprire un nuovo sketch e salvarlo con il nome che desideriamo nella cartella in cui archiviamo i vostri programmi.

Il programma che ci accingiamo a realizzare è detto blink è possibile reperirlo sul sito Arduino.cc oppure come già accennato, direttamente sugli esempi del nostro programma:

File > Examples > 1.Basics > Blink

Prima di iniziare è opportuno ricordare che il programma (vedi lezione precedente) oltre ai blocchi commento è composto da due sezioni:

- setup()
- loop().

realizziamo il programma e analizziamolo anche con l’aiuto dei commenti.

```
// Esempio 01: far lampeggiare un led

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

void setup()
{
  pinMode(LED, OUTPUT); // imposta il pin digitale come output
}

void loop()
{
  digitalWrite(LED, HIGH); // accende il LED
  delay(1000);             // aspetta un secondo
  digitalWrite(LED, LOW); // spegne il LED
  delay(1000);            // aspetta un secondo
}
```



<http://www.marrazzoantonio.altervista.org>

Di seguito l'immagine della schermata del nostro primo sketch

```
// Esempio 01: far lampeggiare un led

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

void setup() {
  pinMode(LED, OUTPUT); // imposta il pin digitale come output
}

void loop() {
  digitalWrite(LED, HIGH); // accende il LED
  delay(1000);             // aspetta un secondo
  digitalWrite(LED, LOW);  // spegne il LED
  delay(1000);             // aspetta un secondo
}
```

### Analizziamo passo passo il codice

```
// Esempio 01: far lampeggiare un led
```

Con 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: 16.01.2017
```

```
// Autore: Antonio
```

È importante notare che ogni linea di commento deve essere preceduta da “//”

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

Ad esempio:

```
/* Esempio 01: far lampeggiare un led
```

```
Data ultima modifica: 16.01.2017
```

```
Autore: Antonio */
```

Successivamente abbiamo:

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



<http://www.marrazzoantonio.altervista.org>

**#define** consente di sostituire, in fase di compilazione, all'interno del nostro programma il valore 13 ogni volta che compare la parola LED.

( la compilazione avviene quando fate click su "Verify" oppure quando effettuate l'upload sulla scheda.) in alternativa troviamo in alcuni programmi al posto di "define LED 13" " int led = 13" Col primo si crea un'etichetta "LED" che verrà sostituita prima di compilare il programma col numero 13. Nel secondo caso si crea una variabile di nome "led" che vale 13. Di solito si usa il #define quando si deve indicare dei dati fissi, ovvero che non verranno cambiati durante l'esecuzione del programma.

Poi troviamo :

```
void setup()
```

**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** indica dove incomincia il codice.

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

**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** 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** possiede due argomenti, il primo definisce il pin, il secondo indica lo stato.



<http://www.marrazzoantonio.altervista.org>

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.

```
delay(1000); // aspetta un secondo
```

**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:

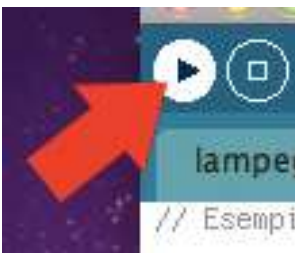
```
digitalWrite(LED, HIGH); // accende il LED  
delay(1000); // aspetta un secondo
```

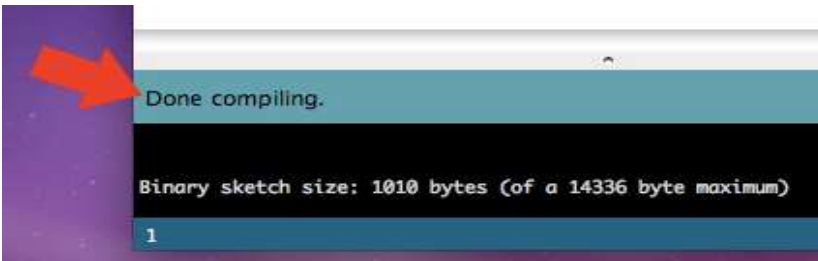
Diciamo ad Arduino di accendere il diodo led sul pin 13 per 1 secondo.

```
digitalWrite(LED, LOW); // spegne il LED  
delay(1000); // aspetta un secondo
```

Ordiniamo al led sul pin 13 di spegnersi per 1 secondo

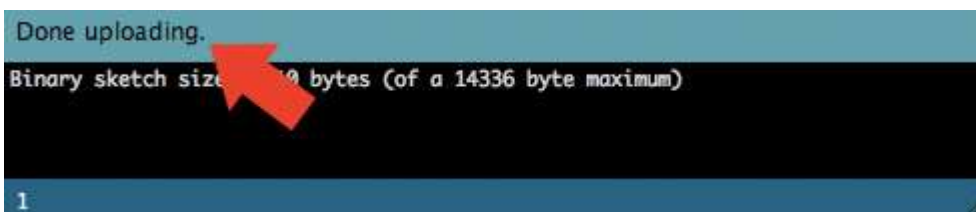
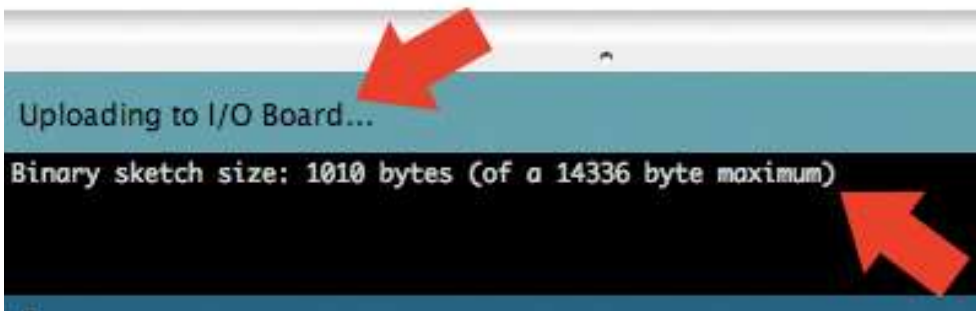
Ora non ci resta che verificare il nostro programma facendo click su "Verify" e se tutto è corretto in basso all'IDE compare il messaggio "Done compiling" ciò indica che l'IDE Arduino ha tradotto il nostro sketch in un programma eseguibile sulla scheda Arduino. Il passo successivo è quello di caricare sulla scheda il nostro 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:

- 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;
- 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.







<http://www.marrazzoantonio.altervista.org>

Dopo aver verificato il funzionamento non ci resta che esercitarci a:

- variare il tempo di accensione e spegnimento del diodo led,
- collegare un secondo diodo led,
- verificare il funzionamento del programma usando “ int led”

```
/* Esempio 02: far lampeggiare un led
Data ultima modifica: 16.01.2017
Autore: Antonio */
int ledPin = 13; // LED connesso al pin digitale D13
void setup()
{
// inizializza il pin del LED come uscita digitale:
pinMode(ledPin, OUTPUT);
}
// Viene dichiarato un loop senza uscita
// Le istruzioni interne al loop vengono continuamente eseguite
void loop()
{
digitalWrite(ledPin, HIGH); // Accende il LED
delay(1000); // aspetta un secondo
digitalWrite(ledPin, LOW); // spegne il LED
delay(1000); // aspetta un secondo

}
```

## SCHEMA DI COLLEGAMENTO:

