

# Lampeggio di una uscita digitale

## LAMPEGGIO DI UNA USCITA DIGITALE

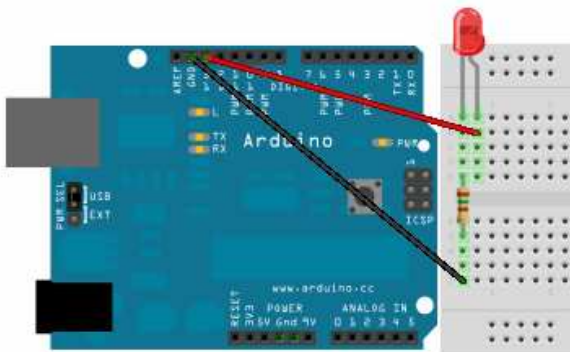
Questo esempio mostra la cosa più semplice che si possa fare con Arduino per vedere un effetto di un programma sull'uscita: un LED che lampeggia.

### REQUISITI HARDWARE

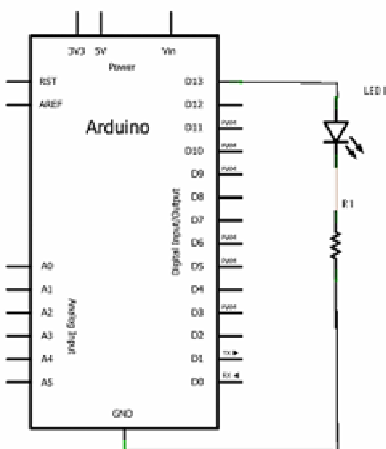
- Scheda Arduino Uno
- Breadboard
- LED
- Resistenza da 220 Ohm
- fili di protipizzazione
- Cavo USB

### CIRCUITO

Per costruire il circuito, montare sulla breadboard il LED e la resistenza. Il catodo del LED (polo negativo = piedino più corto) va direttamente collegato ad un lato della resistenza. L'anodo del LED (polo positivo = piedino più lungo) va collegato attraverso un filo di protipizzazione al pin 13 della morsettieria DIGITAL di Arduino. L'altro lato della resistenza va collegato al pin GND.



Lo schema elettrico del circuito è il seguente:



## SVILUPPO DEL PROGRAMMA

Collegare la scheda Arduino Uno al computer mediante il cavo USB ed avviare l'ambiente di sviluppo Arduino.

In uno Sketch vuoto inserire il seguente programma:

```
/*
 * Lampeggio
 *
 * Questo è l'esempio base di Arduino. Si accende un LED per un secondo
 * poi viene spento per un secondo, e così via ...
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

void setup() {
  // Inizializza il pin 13 come uscita
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // emetti 1 sul pin 13 (LED on)
  delay(1000);           // attendi per un secondo
  digitalWrite(13, LOW); // emetti 0 sul pin 13 (LED off)
  delay(1000);           // attendi per un secondo
}
```

Un programma per Arduino deve sempre contenere almeno le due funzioni `setup()` e `loop()`. La funzione `setup()` viene chiamata una volta sola al reset del microcontroller. La funzione `loop()` viene chiamata ripetutamente alla massima velocità possibile che dipende dalla durata della funzione `loop()` stessa.

## FUNZIONE SETUP

I piedini digitali sono connessi a porte di I/O configurabili sia come ingressi che come uscite in modo da potere sviluppare a partire dallo stesso hardware di base una grande varietà di applicazioni.

La configurazione della direzione di un piedino va fatta una volta per tutte al reset del programma e non va più cambiata.

La funzione `pinMode()` consente di impostare un piedino come ingresso oppure uscita.

La funzione richiede due parametri:

- numero del piedino: può assumere valori da 0 a 13
- tipo di piedino: può essere INPUT oppure OUTPUT

Quindi la chiamata:

```
pinMode (13, OUTPUT);
```

imposta il piedino 13 come uscita digitale che può assumere i due valori LOW = 0 Volt oppure HIGH = 5 Volt.

## FUNZIONE LOOP

La funzione loop() viene ripetutamente eseguita fino allo spegnimento o al reset del microcontroller.

Nella funzione loop() è presente una sequenza di quattro azioni:

```
digitalWrite(13, HIGH); // emetti 1 sul pin 13 (LED on)
```

Emette sul pin 13 una tensione a livello alto (5 V); il LED si accende perchè circola una corrente di circa 20 mA attraverso il LED e la resistenza verso massa.

```
delay(1000); // attendi per un secondo
```

La funzione delay vuole come parametro un tempo espresso in millisecondi ed introduce, mediante un loop interno, una attesa attiva di una durata circa uguale al parametro richiesto. In questo caso si determina una attesa di circa 1 secondo.

```
digitalWrite(13, LOW); // emetti 0 sul pin 13 (LED off)
```

Emette sul pin 13 una tensione a livello basso (0 V); il LED si spegne perché non è più alimentato.

```
delay(1000); // attendi per un secondo
```

La seconda chiamata alla funzione delay introduce una ulteriore attesa di circa 1 secondo.

Ogni esecuzione della funzione loop() dura quindi circa 2 secondi; nel primo secondo il LED è acceso mentre nel successivo il LED è spento.

Poiché la funzione loop() si ripete indefinitamente il LED lampeggia accendendosi e spegnendosi ripetutamente.

La frequenza di lampeggio è:

$$F = \frac{1}{T} = \frac{1}{2} = 0,5 \text{ Hertz}$$

dove T è il periodo di lampeggiamento che è di 2 secondi.

## COMPILAZIONE

Compilare il programma facendo click sul bottone Verify della barra dei bottoni a scelta rapida oppure sulla voce di menu Sketch > Verify/Compile.

Nella barra dei messaggi compare il messaggio "Compiling ...".

Se non ci sono errori al termine della compilazione compare il messaggio "Done compiling" e nella console compare la dimensione in byte del programma compilato.

Se ci sono errori nella barra dei messaggi compare il messaggio "Error compiling" e nella console compare la diagnostica degli errori.

## UPLOAD

Prima di effettuare l'upload di un programma verificare la corretta configurazione della scheda e del collegamento alla scheda.

La selezione della scheda si effettua dal menu Tools > Board > Arduino Uno

La selezione del canale di configurazione si effettua dal menu Tools > Serial Port > Comx.

Se nella lista delle porte seriali disponibili sono presenti più porte verificare qual è la porta di Arduino accedendo nel menu di avvio del sistema operativo ad impostazioni > Pannello di controllo > Sistema > Hardware > Gestione periferiche > Porte (COM) ed individuare il numero di porta associato ad Arduino.

Caricare il programma sulla scheda Arduino facendo click sul bottone "Upload" della barra dei bottoni a scelta rapida oppure sulla voce File > Upload to I/O Board.

Nella barra dei messaggi compare il messaggio "Uploading to I/O Board...".

Se non ci sono errori al termine della compilazione compare il messaggio "Done uploading" e nella console compare la dimensione in byte del programma caricato.

Se ci sono errori nella barra dei messaggi compare il messaggio "Error uploading" e nella console compare la diagnostica degli errori.

Durante il caricamento i LED TX ed RX sulla scheda Arduino lampeggiano indicando una comunicazione tra l'ambiente di sviluppo e la scheda.

In questa fase sulla scheda è in esecuzione il bootloader precaricato nella memoria FLASH.

Il bootloader riceve il programma applicativo, lo carica nella memoria FLASH e al termine del caricamento passa il controllo al programma caricato che entra in esecuzione.