

ARDWARE #1 I Led Arcobaleno

Obiettivo: Utilizzare Arduino come alimentatore per gestire 5 led arcobaleno (rosso, blu, arancione, giallo, verde)

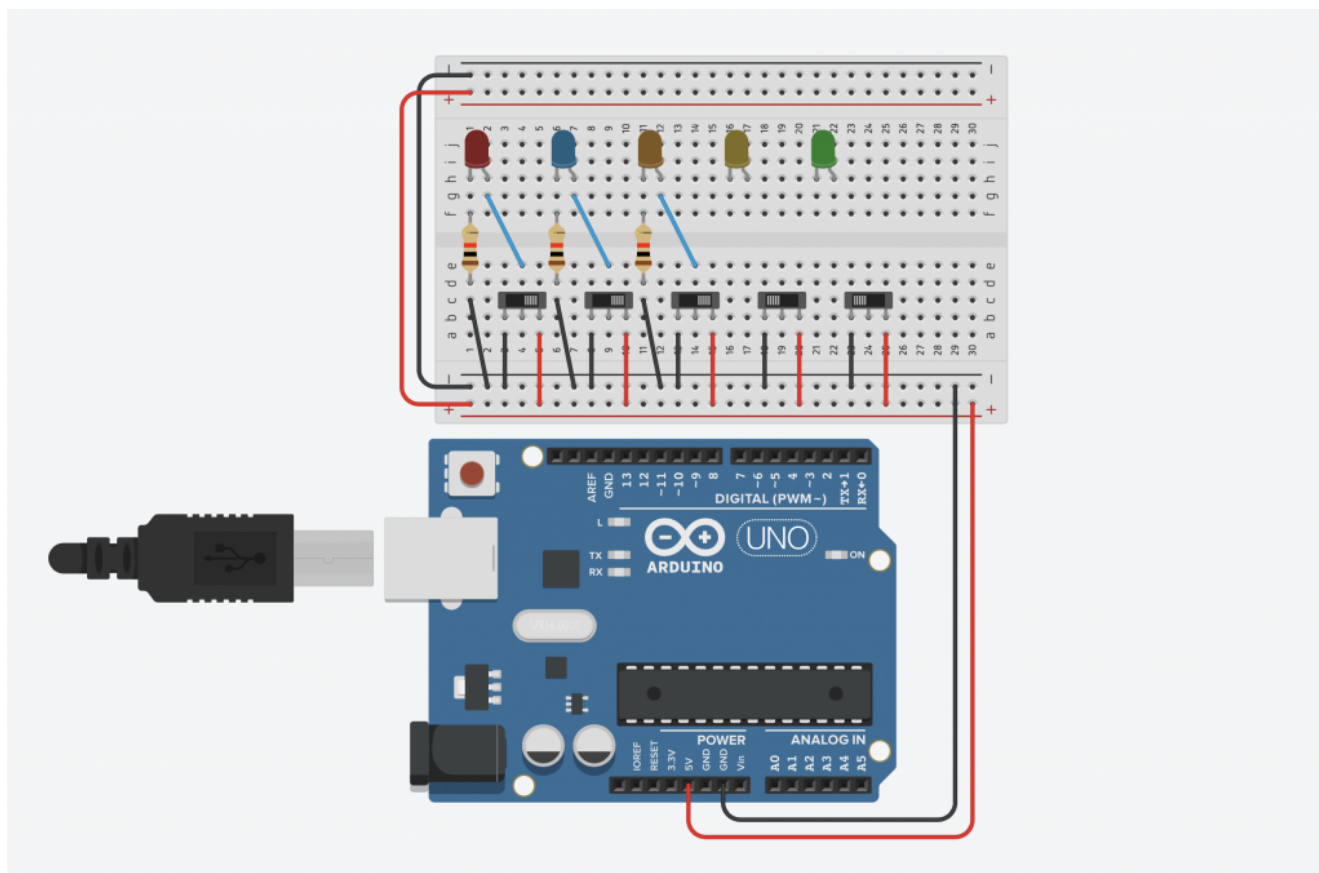
Componenti elettronici:

- Arduino
- 5 led
- 5 interruttori a scorrimento (slideswitch)
- 5 resistenze (100 Ohm)

Teoria: Obiettivo di questa esperienza è gestire 5 differenti led utilizzando Arduino come un semplice alimentatore. In questa esperienza, non c'è né codice né programmazione; i led sono gestiti nella vecchia maniera (old school) attraverso dei semplici interruttori che separano la lampada dalla rete di alimentazione.

Nell'esempio trattato in questo articolo, sono stati utilizzati degli **interruttori a scorrimento** i cui terminali sono collegati alla tensione di alimentazione (5 Volt) o alla massa (0 Volt). Modificando la posizione dell'interruttore, il morsetto centrale si collega ad una delle due tensioni di riferimento. Tale interruttore è collegato direttamente ad un LED (una semplice lampada che funziona con una tensione di 1,5 Volt). Una resistenza in serie al LED è indispensabile al fine di regolare la tensione e la corrente presente sulla lampada evitando di danneggiarla.

Collegamento Circuitale:



Codice:

Non serve codice.

Osservazioni:

1. Il circuito non è completo, divertiti a completarlo.

Pulsante come Interruttore

Obiettivo: Utilizzo di un pulsante come interruttore per l'accensione di un led (stato ON/OFF)

Componenti elettronici:

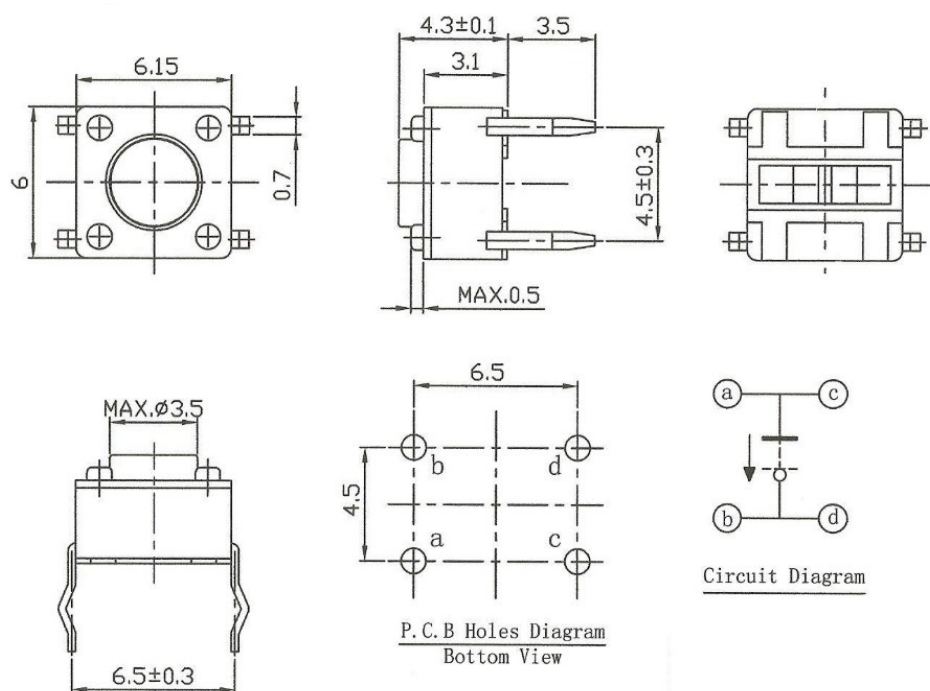
- Arduino UNO
- Breadboard
- 1 Led
- 1 Pulsante
- 1 Resistenza (100 Ohm)
- 1 Resistenza (1k0hm)

Pre-Requisiti

[LED e Pulsante](#)

Teoria: Da un punto di vista pratico mantenere una luce accesa tenendo premuto il pulsante (vedi lezione pre-requisiti) non è proprio il massimo della vita. Se da un punto di vista elettronico il componente tipicamente utilizzato per gestire lo stato ON/OFF di una lampada è l'interruttore, attraverso una corretta gestione del software di controllo è possibile "trasformare un pulsante in interruttore". Per questo motivo,

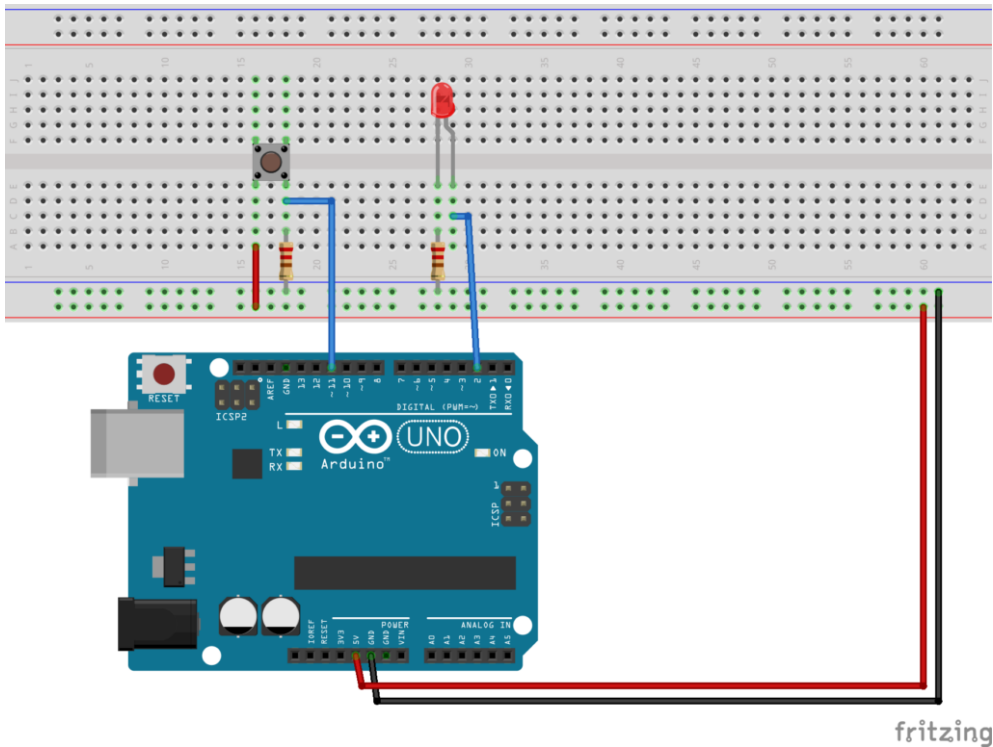
agendo direttamente sul software e lasciando invariato l'hardware proposto nell'esperienza [LED e Pulsante](#), è possibile trasformare il sistema in questione in un controllo ON/OFF dove il pulsante è utilizzato come interruttore.



Datasheet Pulsante

Anche se in questa esperienza il pulsante viene utilizzato come interruttore è importante ribadire che l'impiego di resistenze di Pull-up o Pull-down sono necessarie per un corretto funzionamento del circuito. Maggiori informazioni sulle resistenze di Pull-up o Pull-down possono essere trovate [qui](#).

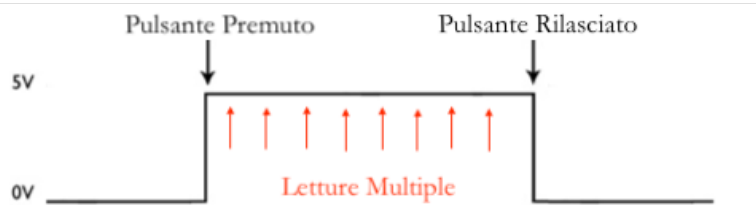
Collegamento Circuitale:



Collegamento Circuitale

Codice: Il codice utilizzato per trasformare il pulsante in interruttore si basa sull'impiego di due differenti variabili globali (create fuori dalle funzioni loop e setup, pertanto visibili in tutto il codice). Queste variabili sono:

- **valButtonOld:** Memorizza la lettura precedente effettuata sul pulsante. Attraverso questa variabile è possibile evitare letture multiple. E' importante considerare che essendo il clock di Arduino più elevato del tempo di reazione umano, premendo il pulsante anche per un istante brevissimo il controllore eseguirà più letture con stato del pulsante HIGH eseguendo in seguito il codice corrispondente. Per questo motivo si preferisce leggere lo stato di transizione da basso ad alto piuttosto che lo stato alto di un pulsante.



Esempio di lettura multipla

- **ledState:** Memorizza lo stato del led (se acceso o spento). Premuto il pulsante, nel caso in cui il led sia acceso questo viene spento viceversa nel caso in cui il led sia spento questo viene acceso. Per questo motivo è indispensabile utilizzare una variabile per controllare lo stato del pulsante.

Personalizzazioni:

L'impiego della variabile `valButtonOld`, utilizzata per evitare letture multiple, può essere evitato aggiungendo un delay dopo la lettura del pulsante. Si provi ad implementare il circuito ed il relativo codice.