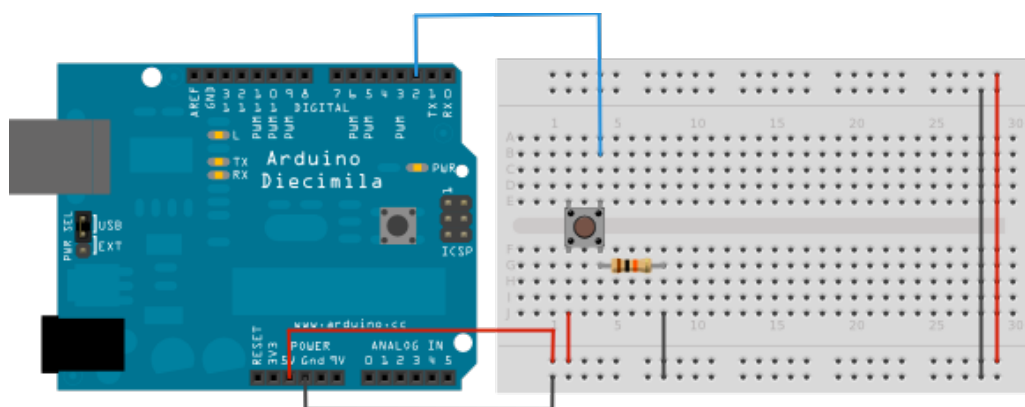
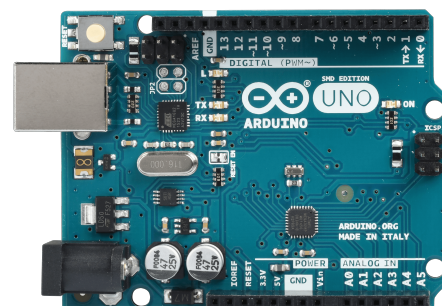
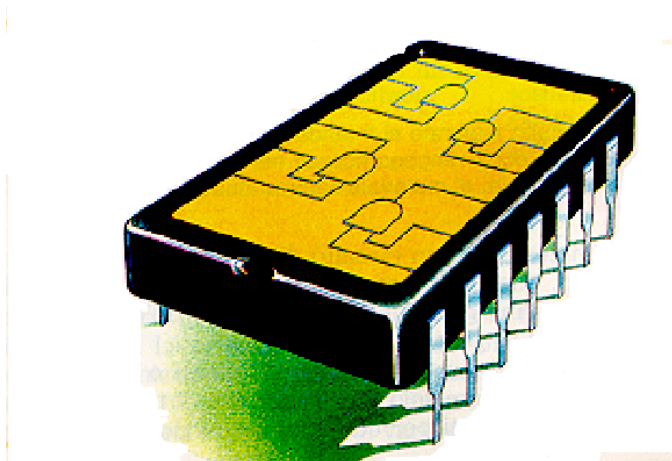


---

## Lez. N. 04 Button e monitor seriale

---



- Scheda Arduino o Genuino
- Un interruttore, un pulsante o un interruttore momentaneo
- Resistenza da 10k ohm
- cavi di aggancio (jumper)
- breadboard (tagliere)

Tanta pazienza!

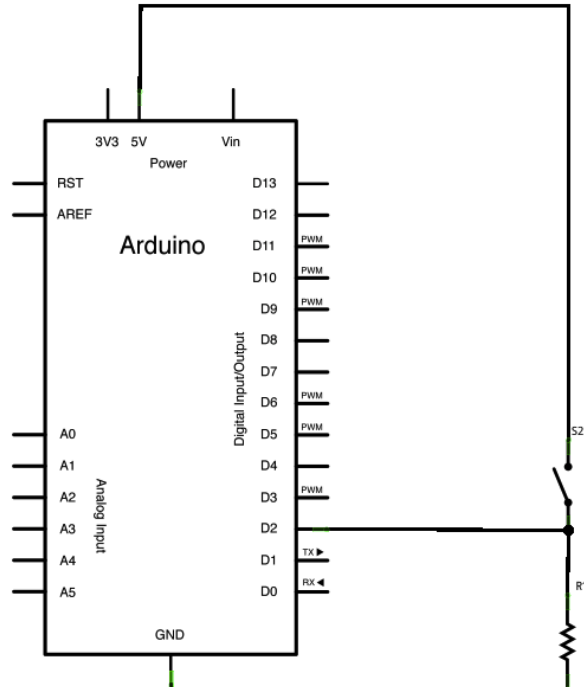
## Circuito

Collegare tre fili alla scheda. I primi due, rosso e nero, si collegano alle due lunghe file verticali sul lato della breadboard per fornire l'accesso all'alimentazione e alla massa a 5 volt. Il terzo filo va dal pin digitale 2 a una gamba del pulsante. La stessa gamba del pulsante si collega tramite una resistenza di pull-down (qui 10k ohm) a terra. L'altra parte del pulsante si collega all'alimentazione a 5 volt.

Pulsanti o interruttori collegano due punti in un circuito quando li si preme. Quando il pulsante è aperto (non compresso) non vi è alcuna connessione tra i due piedini del pulsante, quindi il pin è collegato a massa (attraverso il resistore di pull-down) e legge come LOW, o 0. Quando il pulsante è chiuso (premuto), crea una connessione tra le sue due gambe, collegando il pin a 5 volt, in modo che il pin sia ALTO o 1.

Se si scollega il pin I / O digitale da tutto, il LED potrebbe lampeggiare in modo irregolare. Questo perché l'input è "flottante" - cioè, non ha una connessione solida a tensione o massa, e restituirà a caso HIGH o LOW. Ecco perché è necessario un resistore di pull-down nel circuito.

## Schema elettrico



## Codice

---

Nel programma qui sotto, la prima cosa che farai nella funzione di configurazione è di iniziare le comunicazioni seriali, a **9600 baud** (bit di dati al secondo), tra la tua scheda e il tuo computer con la linea:

**Serial.begin(9600);**

Quindi, inizializza il pin digitale 2, il pin che leggerà l'output dal tuo pulsante, come input: **pinMode(2, INPUT);**

Ora che la configurazione è stata completata, spostati nel ciclo principale del tuo codice. Quando si preme il pulsante, 5 volt passeranno liberamente attraverso il circuito e, quando non viene premuto, il pin di ingresso sarà collegato a massa attraverso la resistenza da 10k ohm. Questo è un input digitale, il che significa che lo switch può essere solo in uno stato on (visto dal tuo Arduino come "1" o **HIGH**) o in uno stato off (visto dal tuo Arduino come uno "0" o **LOW**), senza niente in mezzo.

La prima cosa che devi fare nel ciclo principale del tuo programma è stabilire una variabile per contenere le informazioni provenienti dal tuo switch. Poiché le informazioni provenienti dallo switch saranno "1" o "0", puoi utilizzare un [int tipo di dati](#). Chiama questa variabile **sensorValue** e impostala su qualsiasi cosa venga letta sul pin digitale 2. Puoi realizzare tutto questo con una sola riga di codice:

**int sensorValue = digitalRead(2);**

Una volta che il bord ha letto l'input, fallo stampare questa informazione sul computer come valore decimale. Puoi farlo con il comando [Serial.println\(\)](#) nella nostra ultima riga di codice: **Serial.println(sensorValue);**

Ora, quando apri il tuo monitor seriale nel software Arduino (IDE), vedrai un flusso di "0" se il tuo switch è aperto, o "1" se il tuo switch è chiuso.

```
/* DigitalReadSerial
Legge un ingresso digitale sul pin 2, stampa il risultato sul monitor
seriale
Questo codice di esempio è di pubblico dominio.
http://www.arduino.cc/en/Tutorial/DigitalReadSerial
*/

// il pin digitale 2 ha un pulsante collegato ad esso. Dagli un nome:
int pushButton = 2 ;

// la routine di installazione viene eseguita una volta quando si preme
reset:
void setup ( ) {
  // inizializza la comunicazione seriale a 9600 bit al secondo:
  Serial.begin ( 9600 ) ;
  // rendere il pin del pulsante un input:
  pinMode (pushButton , INPUT ) ;
}
```

```
// la routine ciclo corre più e più volte per sempre:
void loop ( ) {
    // vuoto ad anello. leggere il pin di ingresso:
    int buttonState = digitalRead (pushButton) ;
    // stampa lo stato del pulsante:
    delay(100);
    Serial.println ( buttonState ) ;
    delay ( 1000 ) ;          // ritardo tra le letture per la stabilità
}
```