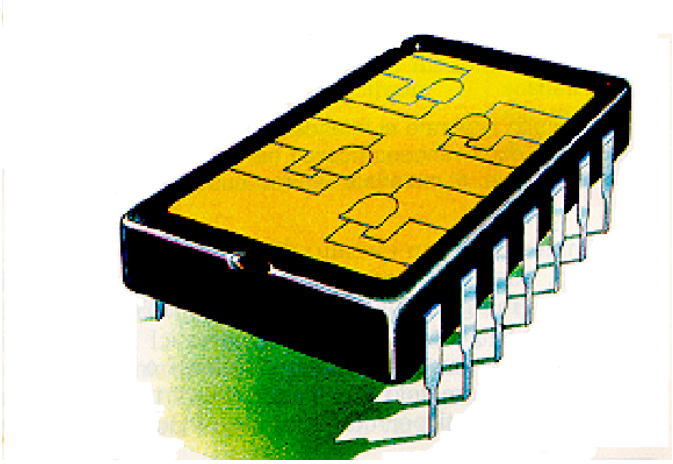
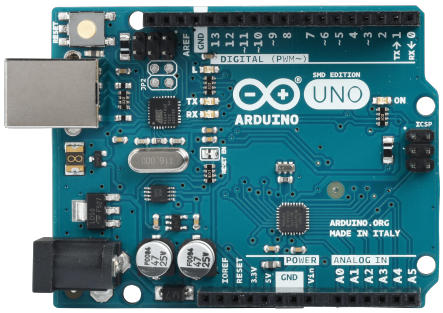
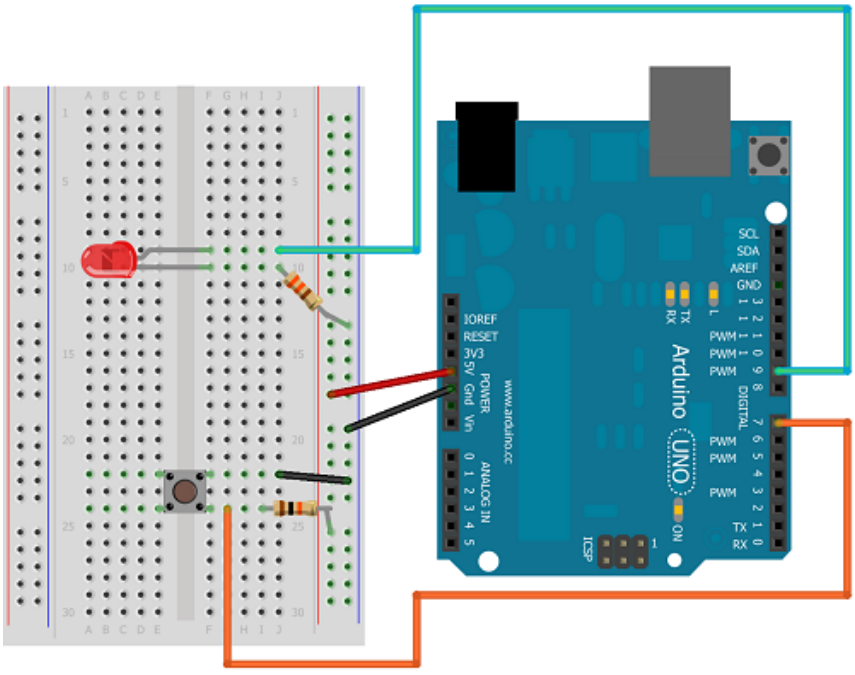
Lez. N. 03 Accendere e spegnere un Led con un push button







Materiali

Arduino
1 Led + 1 push button
1 resistenze da 330 Ω + 1 da 10 K Ω
Jumper vari

Tanta pazienza!

Introduzione

Con il seguente circuito ci si propone di far lampeggiare 1 Led con l'uso di un push button. L'uso dei pulsanti è un altro modo per rendere interattivo il tuo circuito e dare un controllo all'utente sul comportamento dei componenti.

Codice

```
int ledPin = 9;
int buttonPin = 7;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // read from the button pin
  int button = digitalRead(buttonPin);
  if (button==HIGH)
  {
    digitalWrite(ledPin,HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

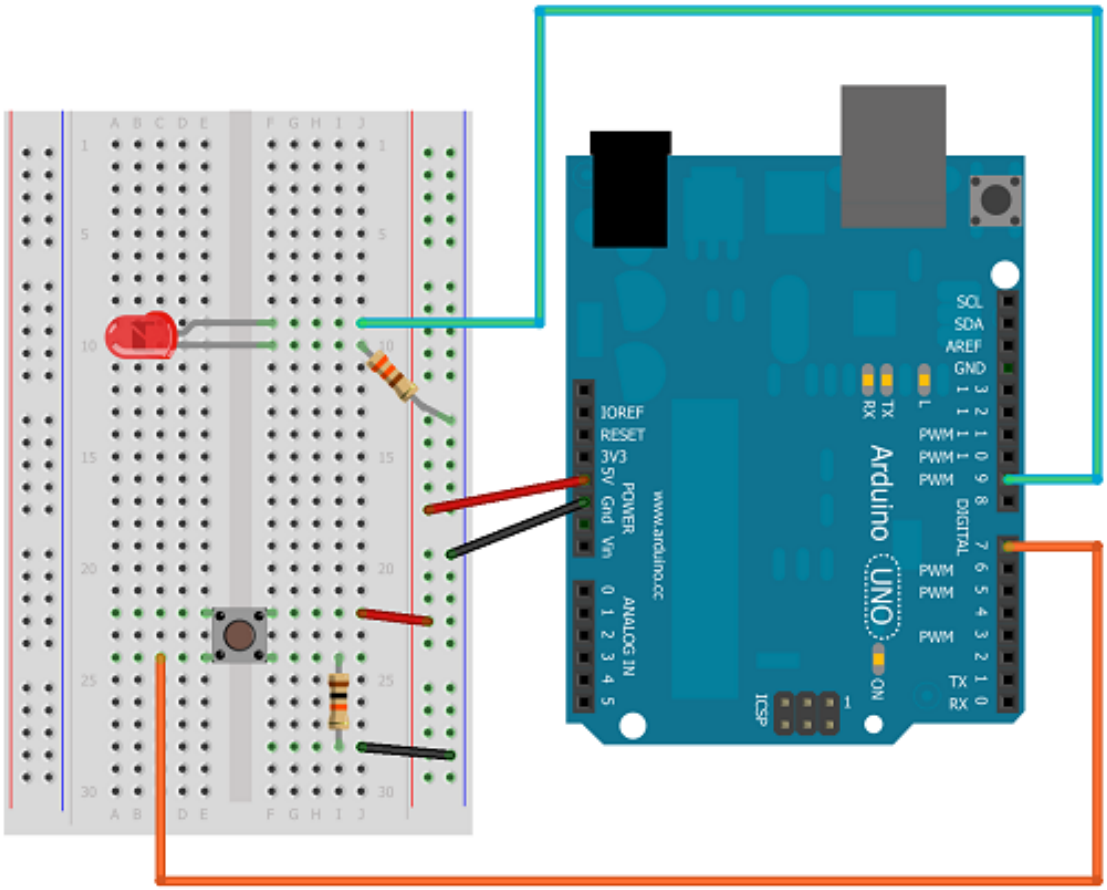
Quando carichi questo programma su Arduino, noterai che il LED si accende. Quando si preme il pulsante, il LED dovrebbe spegnersi. Il doppio equivale (**==**) **nell'istruzione** IF non è un errore - è l'operatore per il confronto nella maggior parte delle lingue della famiglia C. Potresti anche aver notato che il valore che leggiamo dal pulsante, **HIGH** o **LOW** è il valore che stiamo scrivendo sul pin LED. Il codice seguente raggiunge lo stesso effetto senza richiedere l'uso **dell'istruzione IF**.

```
int ledPin = 9;
int buttonPin = 7;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // legge da buttonPin
  int button = digitalRead(buttonPin);
  digitalWrite(ledPin, button);
}
```

Un'altra cosa da notare con questa configurazione è che abbiamo creato un pulsante che spegne la luce quando la premiamo. Ciò significa che il circuito è completo finché non premiamo il pulsante. In altre parole, possiamo dire che il pulsante è **push-to-break** .

Setup alternativo

È possibile impostare il circuito in modo che leggiamo un **LOW** dal pulsante quando non viene premuto. Ciò trasformerebbe il pulsante in un pulsante **push-to-make** . Per fare ciò, cambiamo la disposizione delle connessioni al pulsante.



Ora abbiamo la resistenza collegata a GND e stiamo leggendo da uno dei pin del pulsante che è collegato a GND. Premendo il pulsante si accende la luce se si utilizza il seguente codice,

```
int ledPin = 9;
int buttonPin = 7;
void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
void loop()
```

```
{
   int button = digitalRead(buttonPin);
  digitalWrite(ledPin, button);
}}
```

Registrazione di singole spinte del pulsante

In alcuni progetti, potresti desiderare che qualcosa accada quando viene premuto un pulsante. Spesso però, vuoi solo fare qualcosa una volta quando viene premuto il pulsante. Possiamo far funzionare questi pulsanti in questo modo se cambiamo il modo in cui funziona il codice.

Imposta il circuito come hai fatto all'inizio di questa pagina,

```
int ledPin = 9;
int buttonPin = 7;
int lastButtonState = HIGH;
int ledState = HIGH;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // read from the button pin
  int buttonState = digitalRead(buttonPin);
  // if the button is not in the same state as the last reading
  if (buttonState==LOW && buttonState!=lastButtonState)
  {
    // change the LED state
    if (ledState==HIGH)
    {
      ledState = LOW;
```

```
    }
    else
    {
      ledState = HIGH;
    }
  }
  digitalWrite(ledPin, ledState);
  // store the current button state
  lastButtonState = buttonState;
  // add a delay to avoid multiple presses being registered
  delay(20);
}
```

Abbiamo bisogno di variabili aggiuntive per tenere traccia dello stato del LED (HIGH o LOW, on o off) e in modo che il nostro programma sappia cosa stava succedendo con il pulsante sul loop precedente. Il ritardo alla fine del ciclo è quello di garantire che la pressione del pulsante venga registrata una sola volta. Se rimuovi il ritardo, scoprirai che il LED si accenderà e spegnerà rapidamente. Un approccio simile a questo è usato quando si scrivono giochi per XBox360 usando il controller. Ci sono momenti in cui si desidera lavorare con il pulsante quando viene premuto continuamente, a volte si desidera gestire ogni pulsante premere separatamente.
Un altro approccio a questo è chiamato **Debouncing** . Cerca questo con la parola **Arduino** nella tua ricerca e troverai alcuni esempi di codice che ti aiuteranno. C'è una leggera differenza nel modo in cui viene utilizzato il ritardo che potrebbe funzionare meglio per alcuni progetti.