

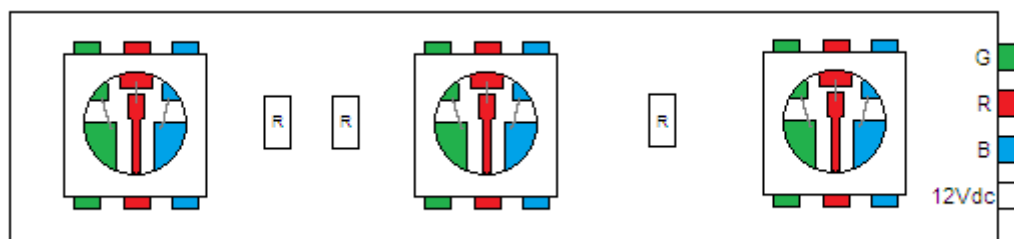
Come usare Arduino per pilotare strip led RGB

La tecnologia led si è ormai evoluta ed inizia ad essere presente in modo pronunciato in molti ambienti, dalla casa agli uffici fino ad arrivare all'illuminazione pubblica. I vantaggi in termini di risparmio energetico sono enormi e la qualità della luce emessa sta raggiungendo un comfort visivo soddisfacente.

Anche nell'arredamento troviamo soluzioni carine che vanno dalla lampada da salotto multicolore alle luci di cortesia nei piani cottura oppure alle barre led impermeabili poste nel box doccia.

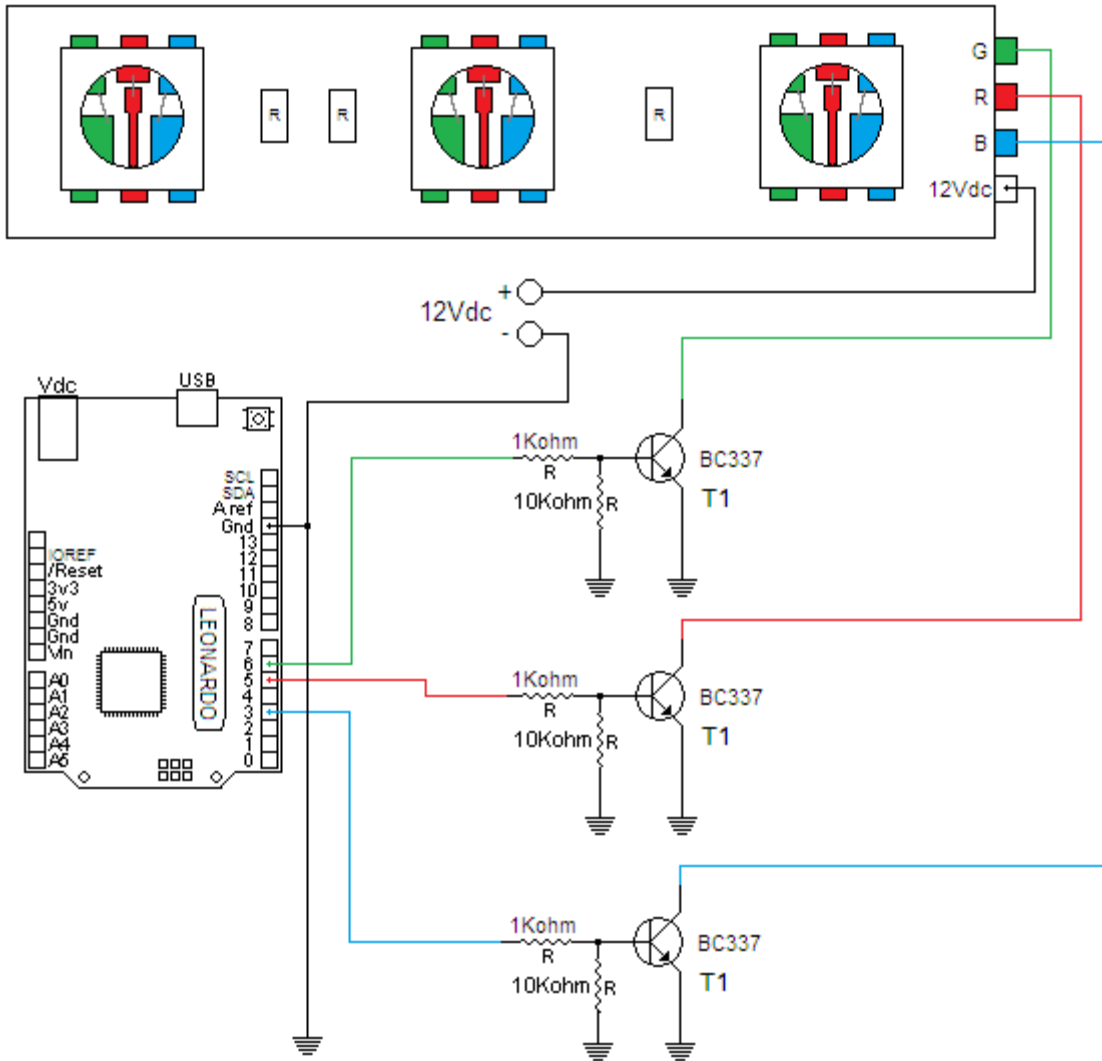
Vista la nostra passione per l'auto-costruzione possiamo realizzare un dispositivo basato su Arduino per pilotare una barra led RGB. Per questo tutorial ho usato una Arduino Leonardo e uno spezzone di [10 cm di una strip led RGB da un metro](#). Questo modello viene alimentato a 12Vdc e un modulo assorbe, con tutti i led accesi circa 55mA. (quindi un metro assorbe circa 550mA).

Un elemento RGB è composto da 3 LED distinti (Rosso, Verde, Blu) disposti in modo da poter interagire tra loro per formare diversi colori complementari.



Il modo più semplice per accendere i led è di collegare uno dei pin (B,R,G) verso massa; non è necessario utilizzare delle resistenze per limitare la corrente.

Vediamo ora come usare Arduino per accendere e spegnere singolarmente i vari led. Il circuito da realizzare è il seguente e si basa su tre transistor PNP tipo BC337.



Le basi dei tre transistors sono collegate ai pin 3, 5 e 6 non a caso poiché su questi pin è disponibile anche il segnale PWM che utilizzeremo per andare a pilotare i led RGB in modo migliore.

Codice di esempio:

```

01 void setup()
02 {
03     //led Rosso
04     pinMode(5, OUTPUT);
05     //led verde
06     pinMode(6, OUTPUT);
07     //led blu
08     pinMode(3, OUTPUT);
09
10     //init seriale
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     //se arrivano dati...
16     if (Serial.available())
17     {
18         //leggi il carattere in arrivo
19         byte data = Serial.read();
20
21         if(data == 'q') //accende il led blu
22             digitalWrite(3, HIGH);
23     }
24 }

```

```

19     else if (data == 'w') //spegne il led blu
20         digitalWrite(3, LOW);
21
22     if(data == 'a') //accende il led verde
23         digitalWrite(5, HIGH);
24     else if (data == 's') //spegne il led verde
25         digitalWrite(5, LOW);
26
27     if(data == 'z') //accende il led rosso
28         digitalWrite(6, HIGH);
29     else if (data == 'x') //spegne il led rosso
30         digitalWrite(6, LOW);
31 }
32 }
33
34
35
36

```

Il codice precedente permette di accendere o spegnere i led inviando tramite serial monitor i caratteri corrispondenti.

Per un maggior controllo dei led RGB possiamo utilizzare il segnale PWM. In questo modo abbiamo la possibilità di creare dissolvenze per creare combinazioni di colori visivamente più interessanti.

Ad esempio il codice seguente permette di accendere e spegnere il led BLU in modo graduale tramite una dissolvenza

Codice di esempio:

```

01
02
03 void setup()
04 {
05     //led Rosso
06     pinMode(5, OUTPUT);
07     //led verde
08     pinMode(6, OUTPUT);
09     //led blu
10     pinMode(3, OUTPUT);
11
12     //init seriale
13     Serial.begin(9600);
14 }
15
16 void loop()
17 {
18     //Aumenta il duty cycle da 100% a 0%
19     for(byte sfuma = 0; sfuma < 255; sfuma++)
20     {
21         analogWrite(3, sfuma);
22         delay(40);
23     }
24
25     //diminuisce il duty cycle da 100% a 0%
26     for(byte sfuma = 255; sfuma > 0; sfuma--)
27     {
28         analogWrite(3, sfuma);
29         delay(40);
30     }
31 }
32
33
34
35
36

```

Se volessi miscelare i colori potrei scrivere qualcosa del genere

Codice di esempio:

```
01
02
03
04 void setup()
05 {
06   //led Rosso
07   pinMode(5, OUTPUT);
08   //led verde
09   pinMode(6, OUTPUT);
10   //led blu
11   pinMode(3, OUTPUT);
12
13   //init seriale
14   Serial.begin(9600);
15 }
16
17 void loop()
18 {
19   //accendo il led rosso
20   digitalWrite(6, HIGH);
21
22   //Aumenta il duty cycle da 100% a 0%
23   for(byte sfuma = 0; sfuma < 255; sfuma++)
24   {
25     analogWrite(3, sfuma);
26     delay(40);
27   }
28
29   //spengo il rosso
30   digitalWrite(6, LOW);
31   //accendo il verde
32   digitalWrite(5, HIGH);
33
34   //diminuisce il duty cycle da 100% a 0%
35   for(byte sfuma = 255; sfuma > 0; sfuma--)
36   {
37     analogWrite(3, sfuma);
38     delay(40);
39   }
40 }
```

Le combinazioni che possiamo creare sono innumerevoli, potete provare a sperimentare creando effetti luminosi che possono abbellire i vostri ambienti.