



Led RGB e Potenzimetri

Basso
 Rosso 5° da sx 5V
 nero 6° da sx Gnd
 verde A4
 Bianco A5

Alto:ù

rosso pin 9
 verde 10
 blu 11

Nello schema in alto puoi vedere come collegare un led RGB a catodo comune, due potenziometri ed arduino.

Ho volutamente utilizzato due potenziometri in quanto nei led RGB la caduta di tensione ai capi del diodo colore non è la stessa per tutti e tre i colori, in particolare nel tipo di led da me acquistato il led rosso ha una caduta di 1,9 – 2,2 v mentre i led verde e blue presentano una caduta di tensione di 3,3 – 3,5 v.

Se il tuo led dovesse avere differenti cadute di tensione rispetto ai colori potrai modificare lo sketch per adattarlo alle tue esigenze.

Ho protetto il led con un unica resistenza da 470Ω sufficiente sia se alimenti il circuito a 5v sia se lo alimenti a 12v. Ti basta un unica resistenza collegata tra polo negativo e catodo essendo quest'ultimo comune a tutti i diodi led incassati nel contenitore plastico da 5mm.

Ed ecco lo sketch:

```

01/*
02 * Sketch Led RGB e potenziometri
03 *
04 * @author: Alfieri Mauro
05 * @see:   http://www.mauroalfieri.it
06 *
07 * @licenze: Creative common
08 */
09
10int ledR = 9;
11int ledG = 10;
12int ledB = 11;
13
14int potA = A5;
15int potB = A4;
16
17void setup()
18{
19  pinMode(ledR,OUTPUT);
20  pinMode(ledG,OUTPUT);
21  pinMode(ledB,OUTPUT);
22
23  Serial.begin(9600);
24}
25
26void loop()
27{
28  int valA = map(analogRead(potA),0,1023,0,255);
29  int valB = map(analogRead(potB),0,1023,0,255);
30
31  Serial.print(" Potenzometro A: ");
32  Serial.print( valA );
33  Serial.print(" Potenzometro B: ");
34  Serial.println( valB );
35
36  analogWrite(ledR,valA);
37  analogWrite(ledG,valB);
38  analogWrite(ledB,valB);
39}

```

linee 10-12: definisci 3 variabili di tipo integer per indicare i pin di OUTPUT con cui piloterai i 3 colori del led distintamente;

linee 14-15: definisci altre due variabili di tipo integer per indicare i pin di INPUT analogico A5 e A4 a cui sono collegati i due potenziometri;

linee 19-21: indica ad arduino che i tre pin assegnati alle variabili ledR, ledG e ledB sono da utilizzare come OUTPUT;

linea 23: imposta la comunicazione seriale tra Arduino e il tuo computer o mac a 9600 baud, userai questa connessione per vedere a video i valori letti dai potenziometri e tradotti in scala 0-255 per i led;

linee 28-29: sono il cuore dello sketch o almeno la prima parte del cuore 😊 , con il comando analogRead(pin), dove il pin glielo passi attraverso una variabile definita alle linee 14 o 15, leggi il valore dal potenziometro. Come sai il valore

letto varia da 0 a 1023, ossia 1024 step, tuttavia al led questa informazioni sarebbe superflua perchè il valore in OUTPUT possibile sui pin PWM di arduino è 0-255, se non facessi nulla otterresti che nel primo quarto, circa, di rotazione il led varia la sua luminosità e dal valore 255 in poi resterebbe sempre alla massima intensità luminosa. Per correggere questa differenza tra INPUT ed OUTPUT arduino ti mette a disposizione la funzione `map(valore,minInput,maxInput,minOutput,maxOutput)` con cui traslare, più o meno in modo lineare, i valori letti su un pin analogico dalla scala 0-1023 alla scala 0-255;

potresti fare questa proporzione con dei calcoli e crearti tu una funzione, ma arduino ti semplifica la vita e te ne da una già fatta e collaudata;

linee 31-34: printano sul monitor seriale i valori provenienti dai potenziometri A e B traslati dalla funzione `map()`;

linee 36-38: finalmente il duro lavoro è concluso, scherzi a parte, con la funzione `analogWrite()` scrivi il valore del potenziometro A sul led Rosso ed il valore del potenziometro B sui led Green e Blu.

Se non ti è chiaro perchè utilizzo la funzione `analogWrite()` se ho collegato il led alle uscite digitali PWM puoi leggere questo mio articolo che ti chiarirà le idee: [Programmare con Arduino – Pin Digitali e Analogici](#).

Buon divertimento