

**GUIDA ARDUINO DIGISPARK**

**Potenziometro + Led RGB**

Mattia Ruberto & Matteo Ghilardini

SOMMARIO

[SOMMARIO 2](#_Toc532385039)

[Scopo 3](#_Toc532385040)

[Componenti 3](#_Toc532385041)

[Arduino Digispark 3](#_Toc532385042)

[Potenziometro 3](#_Toc532385043)

[Led RGB 4](#_Toc532385044)

[Schema logico 4](#_Toc532385045)

[Software & Libreria 5](#_Toc532385046)

[Utilizzo 5](#_Toc532385047)

[Hardware 5](#_Toc532385048)

[Software 5](#_Toc532385049)

Scopo

Lo scopo di questa guida è illustrare il funzionamento del circuito in modo che sia facilmente comprensibile anche agli utenti più inesperti. Illustreremo perciò ogni componente utilizzato e il funzionamento di essi singolarmente, così anche per il prodotto globale.

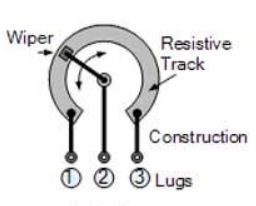
Componenti

## Arduino Digispark

Arduino Digispark, così come tutti gli altri componenti della famiglia Arduino, è una scheda elettronica dotata di un microcontrollore. La funzionalità principale di Arduino è quella di realizzare in maniera pressoché semplice dei dispositivi di controllo oppure degli automatismi (specialmente nel caso di Arduino Digispark). Uno dei punti di forza di Arduino è la sua convenienza economica dal momento che le schede programmabili hanno prezzi veramente bassi (per Digispark meno di 5 CHF) e inoltre il software e il linguaggio di programmazione utilizzato sono Open Source (ossia gratis).

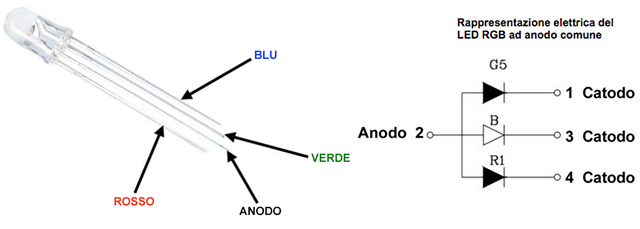
Per collegare elementi esterni alle schede si utilizzando dei pin che possono venir saldati sulle apposite interfacce. L’alimentazione (ossia il +) è indicata da “5V”, mentre la terra (ossia il -) è indicata da “GND”.

## Potenziometro

Il potenziometro è una sorta di resistenza che però può essere modificata, ossia può essere gestita la sua resistenza elettrica. Nell’elettronica “semplice” viene utilizzato per modificare delle frequenze o per modificare la luminosità dei led, mentre nell’elettronica di Arduino i suoi utilizzi aumentano a dismisura grazie ad un semplice comando utilizzato molto spesso: *map(value, fromLow, fromHigh, toLow, toHigh).* Tramite un analogRead() il potenziometro ritorna un valore fra 0 e 1023 perciò molte volte può essere utile magari rimappare l’intervallo fra magari 0 e 100 per ricevere la percentuale di rotazione (se utilizziamo un potenziometro rotativo), in questo caso usiamo il comando *map(analogRead(pinPotenziometro),0,1023,0,100)* e otterremo il valore percentuale di quanto è stato rotato il potenziometro (sempre se utilizziamo un potenziometro rotativo).

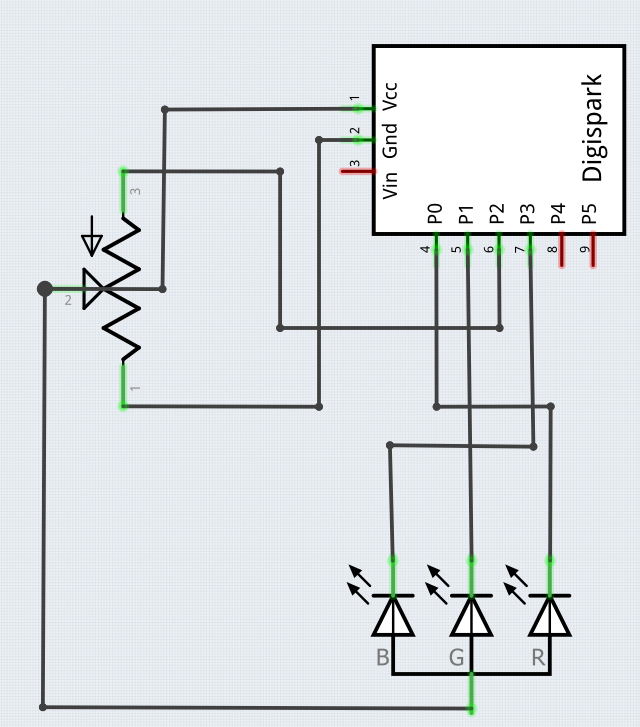
I potenziometri hanno un multiplo di 3 pin (il più comune e quello che usiamo noi ne ha infatti 3) dove i 2 laterali vengono collegati al “+” e al “-“, mentre quello centrale ritorna il valore desiderato. Come nelle resistenze normali, anche i potenziometri non hanno polarità perciò è indifferente quale dei pin esterni inseriamo nel “+” o rispettivamente nel “-“, ma se non ritorna il valore che ci aspettiamo dovremo invertirne il senso.

## Led RGB

Un led RGB è un solo led con al suo interno 3 led rispettivamente di colore rosso, verde e blu. I led (sia quelli RGB che quelli semplici) non possono essere collegati direttamente al polo positivo (o negativo, dipende se anodo o catodo comune. Vedi sotto) perché subirebbero un voltaggio troppo alto rispetto a quello supportato, per questo dobbiamo utilizzare delle resistenze. Il minimo per il led che utilizziamo noi è una resistenza da 330 Ω.

Esistono globalmente 2 tipi di led RGB, quelli con l’anodo (“+”) comune o con catodo (“-“) comune. Nel nostro caso utilizziamo un led anodo comune.

Schema logico



Software & Libreria

In principio, il software è stato realizzato con il software ufficiale di Arduino.

# Software in Arduino

La prima parte di codice è composta dalle variabili che indicano le porte del Digispark dove il valore della variabile corrisponde all’indice della porta (0 per la prima porta, 1 per la seconda, …).   
Abbiamo anche deciso di definire un valore booleano (acceso o spento) per ogni led, con LOW il led corrispondente è acceso.

Nel *setup* inizializziamo tutte le variabili in scrittura (i led). Per tutte utilizziamo il comando *pinMode(<porta>, <OUTPUT/INPUT>)*, essendo in scrittura, dovrà essere *OUTPUT*.

Nel *loop* eseguiamo una lettura del valore del potenziometro e passiamo questo valore come attributo al metodo *getColor (…)*. Alla fine, viene mostrato il colore ottenuto tramite il metodo appena citato grazie al metodo *digitalWrite(<porta>, <valoreBooleano>)*.

Nel metodo *getColor (…)* ri-mappiamo il valore ricevuto da valori 0-1023 a valori 0-60 grazie al metodo *map(<valore>, <valoreMinVecchio>, <valoreMaxVecchio>, <valoreMinNuovo>, <valoreMaxNuovo>)*. Ogni decina corrisponde ad un colore definito da i valori delle variabili booleane relative ai tre colori primari (rosso, verde, blu), in ordine crescente: rosso, giallo, verde, azzurro, blu, viola.

# Libreria in C++

Utilizzo

# Hardware

I componenti da utilizzare per questo progetto sono i 3 citati più e più volte all’interno di questa guida:

* Arduino Digispark;
* Potenziometro (rotativo);
* Led RGB anodo comune;

Per costruire il circuito dobbiamo fissare il potenziometro e il led su di una breadboard (circuito provvisorio) o su una veroboard (circuito definitivo), non è necessario metterli in uno schema preciso a patto che abbia un senso.

* **Fare attenzione alle piste delle board per evitare cortocircuiti** 

Collegare i pin esterni del potenziometro ad alimentazione e GND (è indifferente quale ad uno o all’altro, nel caso in cui i valori non corrispondano a quelli desiderati è sufficiente invertirli), mentre il pin centrale del potenziometro deve essere collegato alla porta P2 del Digispark.

Collegare al pin più lungo del led RGB l’alimentazione (anodo in comune). A tutti gli altri pin colleghiamo delle resistenze da 330 Ω e in serie anche i connettori per portare il segnale alle corrispettive porte del Digispark, nello specifico:

* Il rosso alla porta P0;
* Il verde alla porta P1;
* Il blu alla porta P3;

In caso di dubbio su quale sia il pin corrispondente ad un determinato colore, consulta la documentazione del led RGB a pagina 4.

# Software