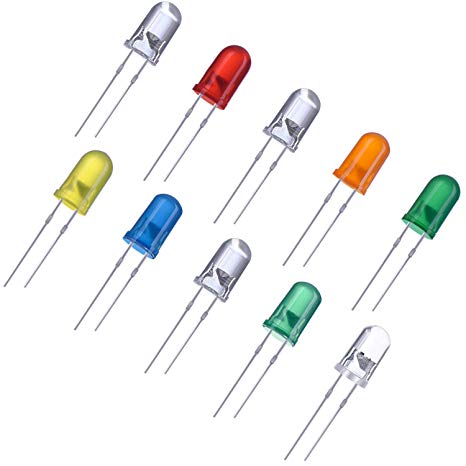


**GUIDA ARDUINO DIGISPARK**

**Pulsante + Led**



Mattia Ruberto & Matteo Ghilardini

SOMMARIO

[SOMMARIO 2](#_Toc536014718)

[Scopo 3](#_Toc536014719)

[Componenti 3](#_Toc536014720)

[Arduino Digispark 3](#_Toc536014721)

[Pulsante 4](#_Toc536014722)

[Led RGB 4](#_Toc536014723)

[Schema Elettrico 5](#_Toc536014724)

[Librerie 5](#_Toc536014725)

[Libreria Led 5](#_Toc536014726)

[Utilizzo 5](#_Toc536014727)

[Hardware 6](#_Toc536014728)

[Software (ogni codice dovrà essere mostrato) 6](#_Toc536014729)

Scopo

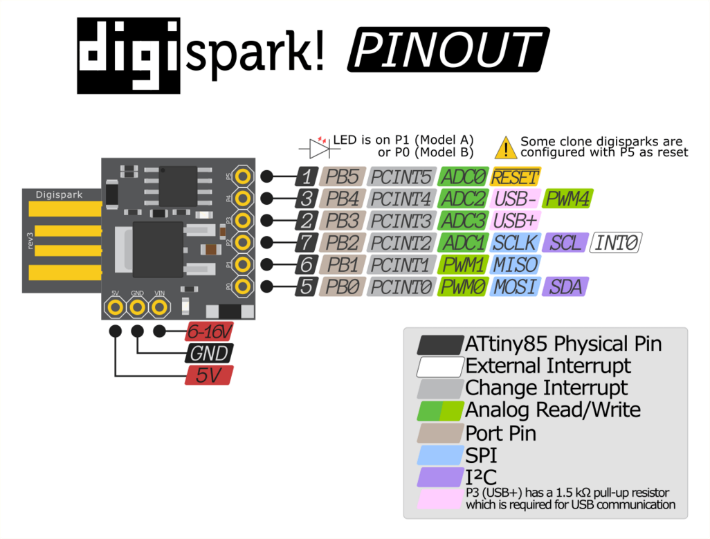
Lo scopo di questa guida è illustrare il funzionamento del circuito in modo che sia facilmente comprensibile anche agli utenti più inesperti. Illustreremo perciò ogni componente utilizzato e il funzionamento di essi singolarmente, così anche per il prodotto globale.

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”, mentre gli altri pin (da P0 a P5) possono assumere diverse funzionalità seguendo il seguente modello:



Per poter utilizzare il software di Arduino col Digispark sono necessari alcuni accorgimenti, per poter installare le schede è necessaria una connessione a internet (preferibilmente senza proxy):

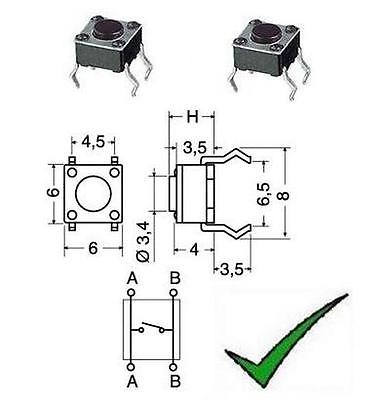
* Nelle impostazioni di arduino (File🡪Impostazioni), nel campo “URL aggiuntive per il Gestore schede:” inserire l’URL <http://digistump.com/package_digistump_index.json>;
* Riavviare il software (se procedendo qualcosa non va, riavviare il pc);
* Cliccando “Gestore schede” (Strumenti 🡪 Scheda) verrà aperta una schermata nella quale è presente una barra di ricerca, scriveteci “Digistump” e verrà mostrata una possibilità come quella da immagine:

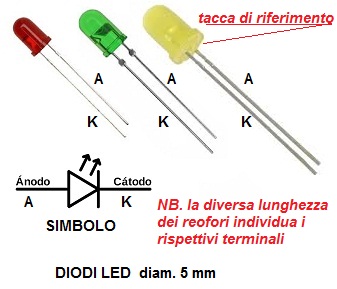
nell’angolo in basso a detra di questa sarà presente il pulsante Installa (premerlo);

* Riavviare il software (se procedendo qualcosa non va, riavviare il pc);
* Nella selezione delle schede cercare e selezionare “Digispark (Default – 16.5 MHz)”;
* Per quanto riguarda la selezione della porta (COM…) dipende dal vostro computer e da quale porta usb utilizzerete per inserire il digispark.

## Pulsante

Un pulsante si comporta come se fosse un cavo che viene collegato e scollegato. La funzione corrispondente al fatto che è collegato, sarebbe quando viene premuto il pulsante, mentre quando viene rilasciato il circuito viene aperto (e quindi scollegato).

Esistono numerosi tipi diversi di pulsanti, ma quelli più comuni e più utilizzati sono quelli a 4 pin come quello mostrato nelle foto. I pin sono collegati a coppie e perciò per collegarli in modo da rilevare la pressione del pulsante bisogna seguire lo schema a sinistra (collegare un polo A, con un polo B).

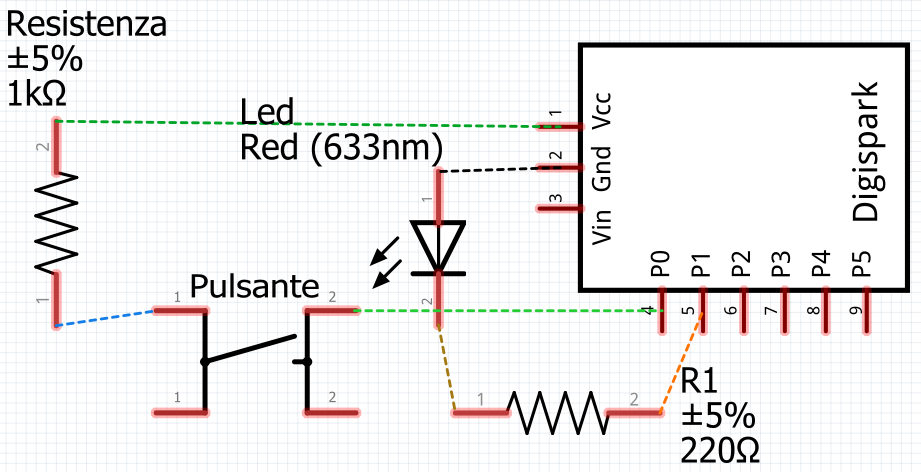
I pulsanti non hanno nessun circuito sensibile al loro interno, quindi non è necessario prestare attenzione ai voltaggi che gli vengono impressi o alla loro polarità. Questo perché, come già detto, i pulsanti sono esattamente come se fossero due cavi che vengono collegati e scollegati a seconda del fatto che sia stato premuto o meno il pulsante.

## Led

Led non possono essere collegati direttamente al polo positivo o negativo della corrente 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.

Per distinguere il pin positivo e quello negativo è sufficiente guardare la lunghezza del suddetto pin e la posizione della tacca di riferimento (Vedi immagine a fianco). Il pin più lungo rappresenta il polo positivo, quindi il più corto quello negativo. Il polo positivo è identificabile anche dalla presenza della tacca.

Schema Elettrico



Librerie

*Tutte le librerie realizzate per questo progetto sono state realizzate nel linguaggio di C++, come d’altronde anche il software di Arduino.*

Tutte le nostre librerie sono composte da un’interfaccia (chiamata nel linguaggio specifico di “C” *Header*, ed è un file con estensione “.h”) e la libreria in sé (con estensione “.cpp”) che estende l’interfaccia.

Sia l’Header, che la libreria devono includere l’interfaccia “Arduino.h”.

## Libreria Led

L’Header racchiude 5 attributi privati:

* int led: indica il pin di Arduino che darà corrente al led;
* int state\_led: rappresenta lo stato del led (acceso/spento);
* bool lastButtonState:

La libreria in sé è leggermente più complessa. All’inizio del codice dobbiamo anche in questo caso includere delle interfacce, in questo caso abbiamo sia quella base (“Arduino.h”), sia quella che abbiamo creato prima: “LedRGB.h”.

Ora parliamo dei metodi della libreria:

* **setLedPin(int, int, int)** richiede come parametri gli indici dei pin di ogni colore del led RGB. Al suo interno il metodo setta semplicemente i parametri passati con gli attributi della classe.
* **setColor(int, int, int)** richiede come parametri dei valori da 0 a 255 corrispondenti all’intensità del colore corrispondente. In seguito rappresenta il colore ricevuto sul led RGB.
* **setRed(int)** richiede come parametro il valore da 0 a 255 corrispondente all’intensità di rosso. In seguito rappresenta il valore ottenuto sul led RGB.
* **setGreen(int)** richiede come parametro il valore da 0 a 255 corrispondente all’intensità di verde. In seguito rappresenta il valore ottenuto sul led RGB.
* **setBlue(int)** richiede come parametro il valore da 0 a 255 corrispondente all’intensità di blu. In seguito rappresenta il valore ottenuto sul led RGB.

Per quanto riguarda la libreria per il pulsante abbiamo concluso con il committente che non era necessario crearla dal momento che non sarebbe sensato creare un metodo contenete un metodo già esistente nel ambiente di Arduino.

Utilizzo

# Hardware

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

* Arduino Digispark;
* Pulsante;
* Led (di un qualunque colore);

Per costruire il circuito dobbiamo fissare il pulsante in modo che le 2 coppie di pin non siano in contatto fra loro. Il led allo stesso modo può essere montato in qualunque modo a patto che i 2 pin non siano connessi.

Per gestire la corrente nel circuito abbiamo bisogno di 2 resistenze

* 220Ω per il led: deve essere collegata in serie fra il collegamento al Digispark (P1) e il polo positivo del led (quello più lungo).
* 10K Ω per il pull-Up o pull-Down del pulsante: deve essere collegata fra il pin del pulsante diagonalmente opposto a quello al quale è collegato il pin di lettura del Digispark (P0) e, a seconda del fatto che viene utilizzata per fare pull-Up o pull-Down, rispettivamente al +5V o al GND.

**Fare attenzione alle piste delle board per evitare cortocircuiti**  ***(In caso di dubbio, eccoti un’immagine che mostra com’è fatta una breadboard di Arduino al suo interno)***

In caso di dubbio su come collegare il pulsante, consulta la documentazione del pulsante a pagina 4.

# Software (ogni codice dovrà essere mostrato)

Per ogni programma eseguito utilizzando una qualunque libreria esterna, come prima cosa è necessario inserire il comando *#include <Interfaccia.h>* sostituendo la parola “Interfaccia” col nome dell’interfaccia che andremo ad estendere.

Come seconda cosa dobbiamo istanziare un’oggetto del tipo della libreria, ossia una variabile di tipo “Classe utilizzata”. Ad esempio, per la libreria del led RGB, il comando sarà:  
*LedRGB myLedRGB;*

A questo punto possiamo cominciare a programmare normalmente.

1. Gamma di colori definita con if dall’utente e per ognuno stampa un colore;
2. Un ciclo di rotazione passa tutta la gamma