

2019

Manuale d'utilizzo

INTRODUZIONE ALL'UTILIZZO DELLE LIBRERIE
PAOLO WEISHAAPT, CARLO PEZZOTTI

Indice

Indice	1
Manuale d'uso delle librerie.....	2
Premessa:	3
Installazione della scheda Digispark	3
Uso della scheda Digispark con l'IDE	7
Primo test	8
Led RGB:	9
Come utilizzare il Led RGB	9
Come utilizzare la libreria del Led RGB:	10
Push button:	11
Come utilizzare un pulsante:.....	11
Come utilizzare la libreria del pulsante:.....	11
Led:	12
Come utilizzare un Led:	12
Come utilizzare la libreria del Led:	12
Potenziometro:.....	13
Come utilizzare un potenziometro:.....	13
Come utilizzare la libreria del potenziometro:	13
Piezo Buzzer:.....	14
Come utilizzare un Piezo Buzzer:	14
Come utilizzare la libreria del Piezo Buzzer:	14
Sitografia.....	15

Manuale d'uso delle librerie

L'utilizzo di queste librerie è raccomandato a delle persone che si stanno avvicinando allo sviluppo sulla piattaforma Arduino e hanno bisogno di una guida nei primi passi della programmazione. Al loro interno ci sono anche tre codici di esempio di utilizzo della libreria per ogni attuatore elencato:

Guida alle librerie

1. Led RGB
2. Push Button
3. Led
4. Potenzziometro
5. Piezo buzzer

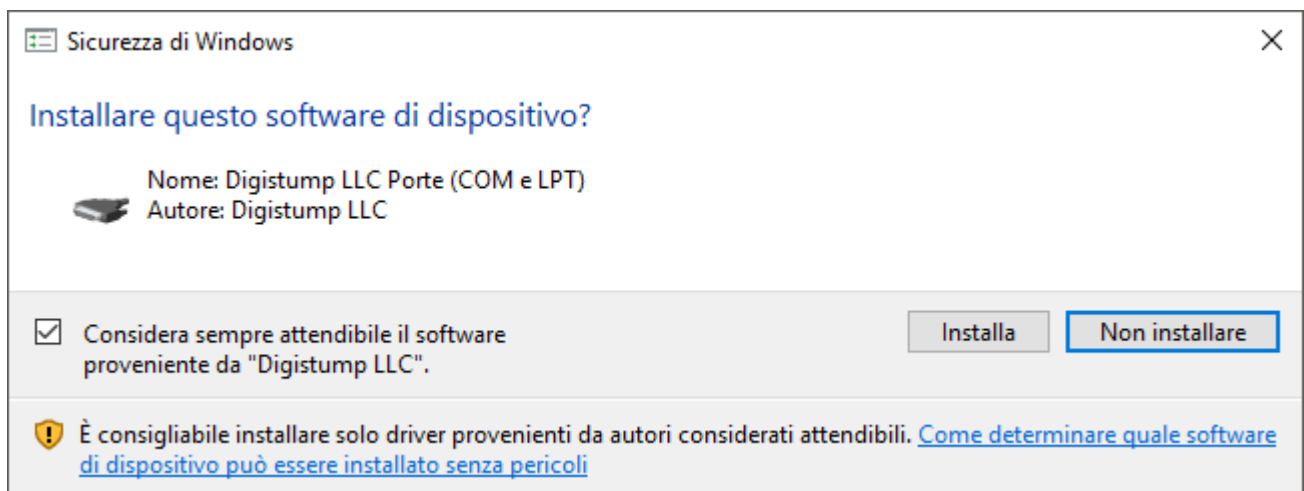
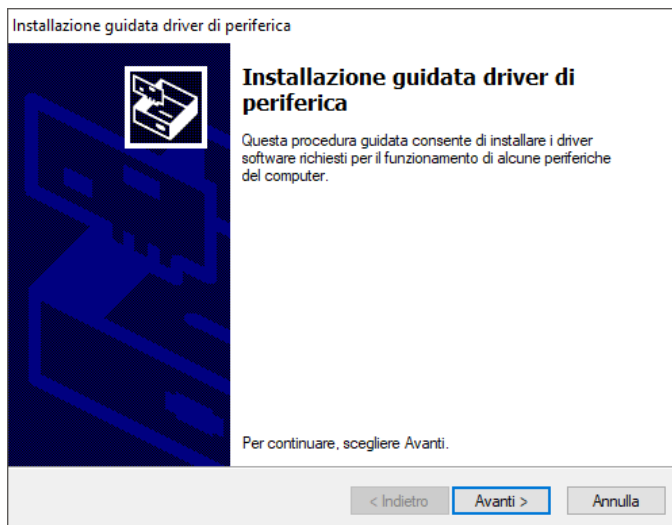
Premessa:

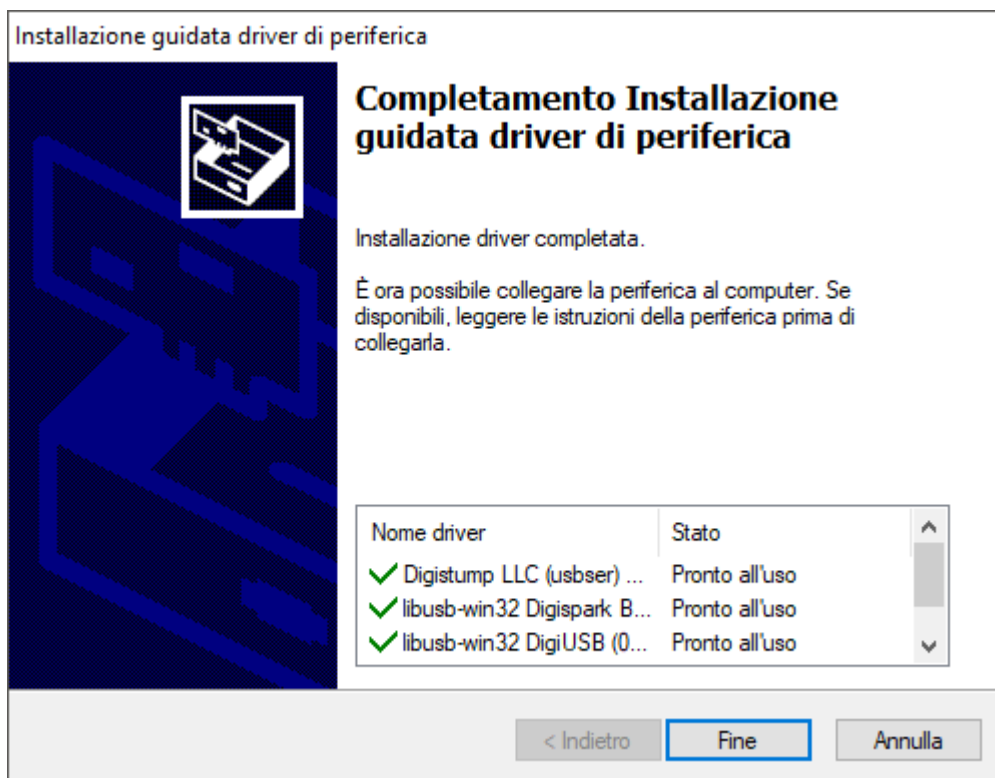
Installazione della scheda Digispark

Dopo questa breve introduzione vediamo come muovere i primi passi con questa piccola scheda.

La procedura è leggermente diversa a quella tipica per le altre schede Arduino.

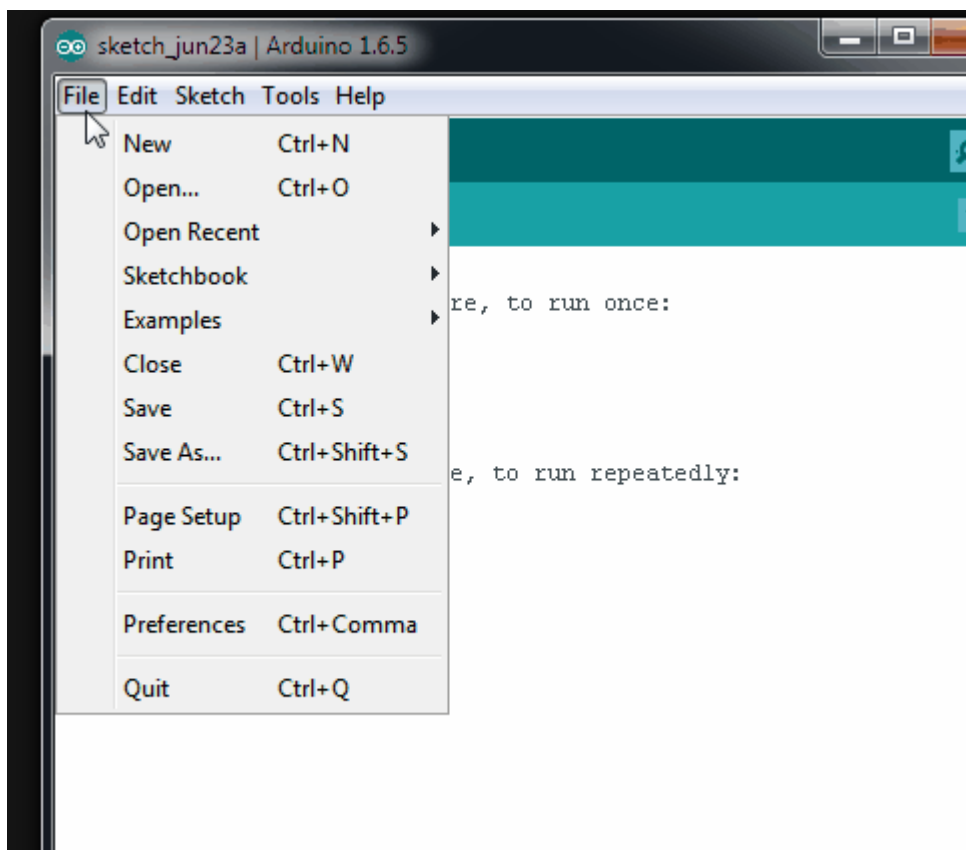
- Sarà quindi necessario scaricare e installare manualmente i driver per la scheda [Digispark](#), Scaricare, decomprimere ed eseguire "**Install Drivers**" (su sistemi a 32 bit) o "DPIInst64" (su sistemi a 64 bit). I file del driver si trovano [in questo link](#).



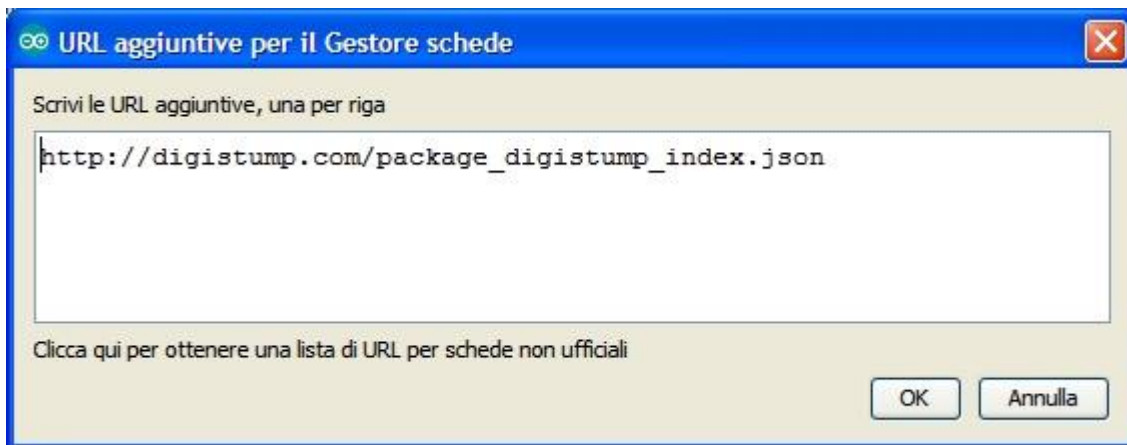


Nota: Attenzione, se inserite la scheda in una porta USB quando non richiesto dall'IDE, la scheda non sarà riconosciuta, ma questo è normale.

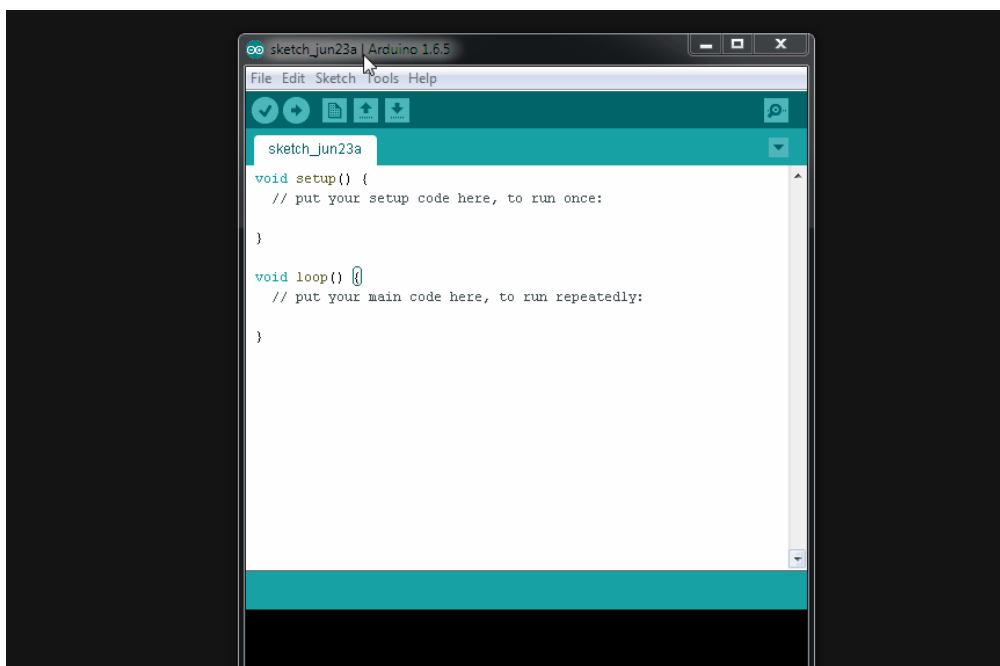
A questo punto, supponendo che l'IDE sia già installato (il test è stato fatto con la versione 1.8.3) andare al menu "**File**" e selezionare "**Impostazioni**"



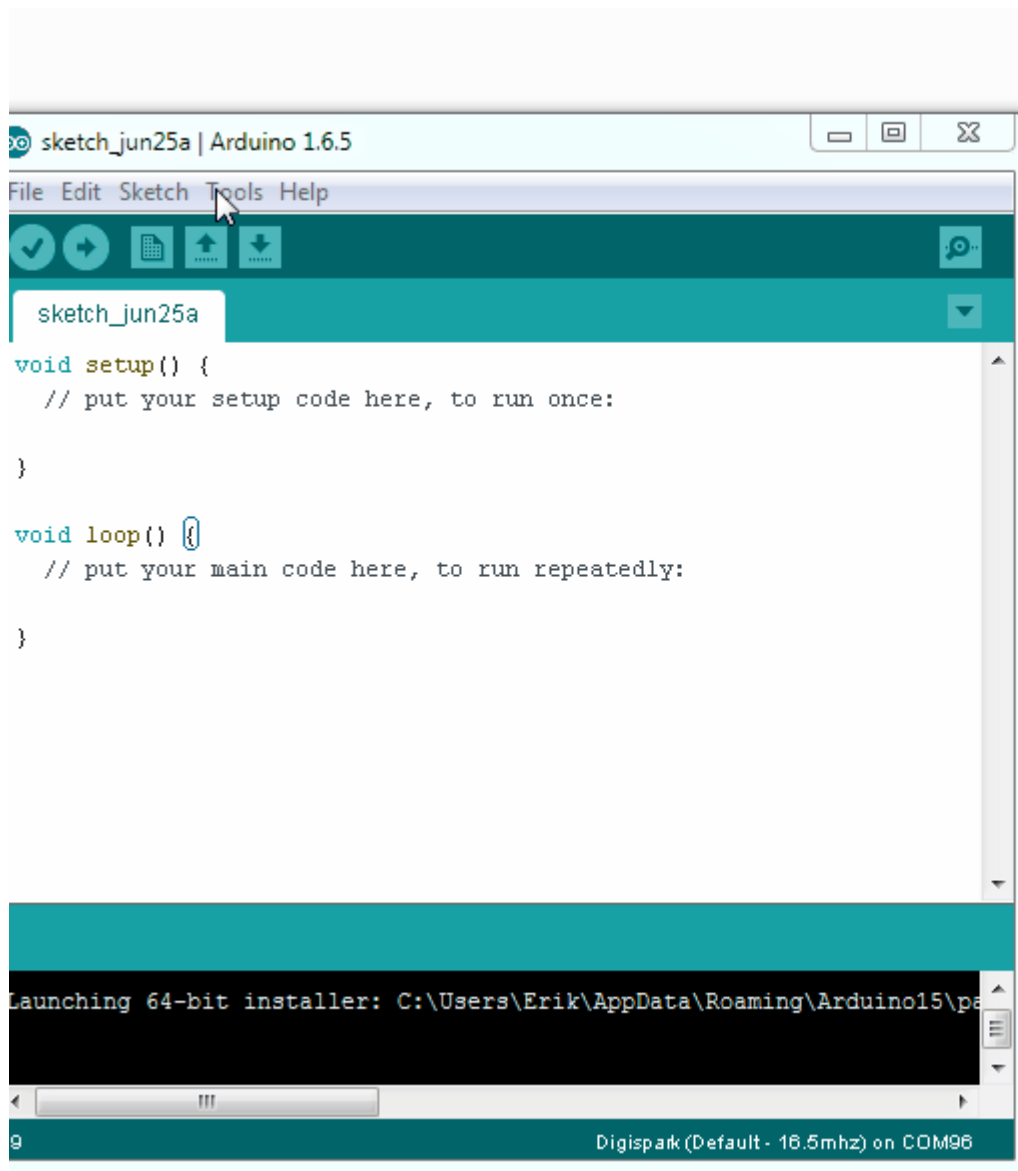
Nella casella con l'indicazione "**URL aggiuntive per il Gestore schede**" immettere: "**http://digistump.com/package_digistump_index.json**" e fare clic su OK.



Andare ora al menu "**Strumenti**" e poi nel sottomenu "**Gestore schede**": dopo una breve attesa, dal menu a discesa **Tipo** selezionare "**Fornito da terzi**"
Selezionare il pacchetto "**Digitump AVR Boards**" e fare clic sul pulsante "**Installa**" e fare clic su OK



- Verrà visualizzato il progresso di download nella barra di fondo della finestra "Gestore schede", quando sarà completata verrà visualizzato "**INSTALLED**" accanto a quella voce dell'elenco.
- Con l'installazione completa, chiudere la finestra "Gestore schede" e selezionare **Digispark** dal menu Strumenti → Schede. "**Digispark (Default - 16.5mhz)**"

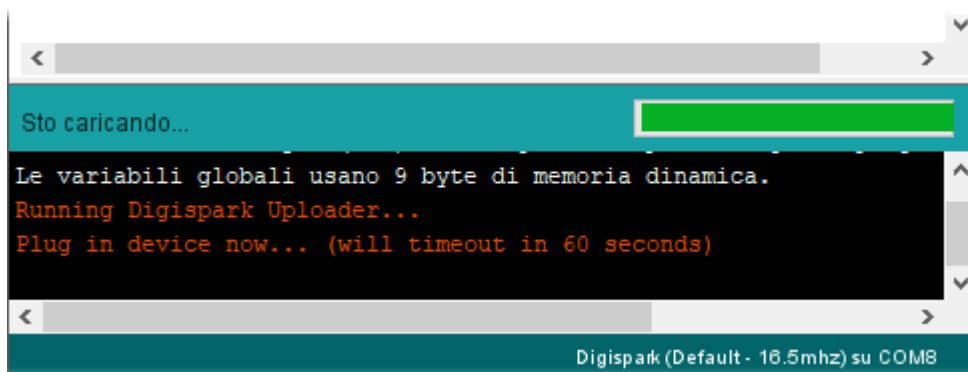


- a questo punto l'installazione può considerarsi completa

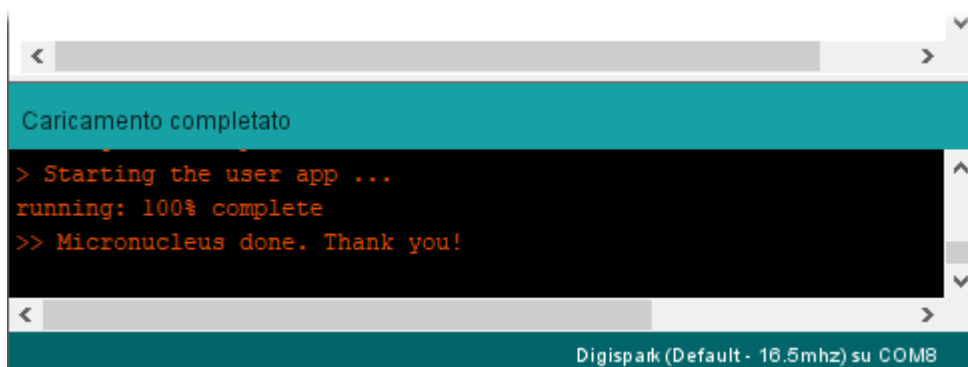
Uso della scheda Digispark con l'IDE

La scheda Digispark funziona un po' diversamente da altri prodotti compatibili con Arduino. La programmazione segue una procedura diversa.

- Verificare che dal menu Strumenti sia selezionata la Scheda → Digispark (default- 16.5Mhz), non è necessario modificare la voce programmatore.
- Scrivere un codice, aprire un codice precedentemente salvato o aprire un esempio Digispark.
- **Non è necessario collegare il Digispark prima di richiamare il caricamento**
- Premere il pulsante di caricamento. Dopo la compilazione nella casella di stato inferiore sarà richiesto di collegare il vostro Digispark - a questo punto è necessario collegarlo - oppure scollegarlo e ricollegarlo.



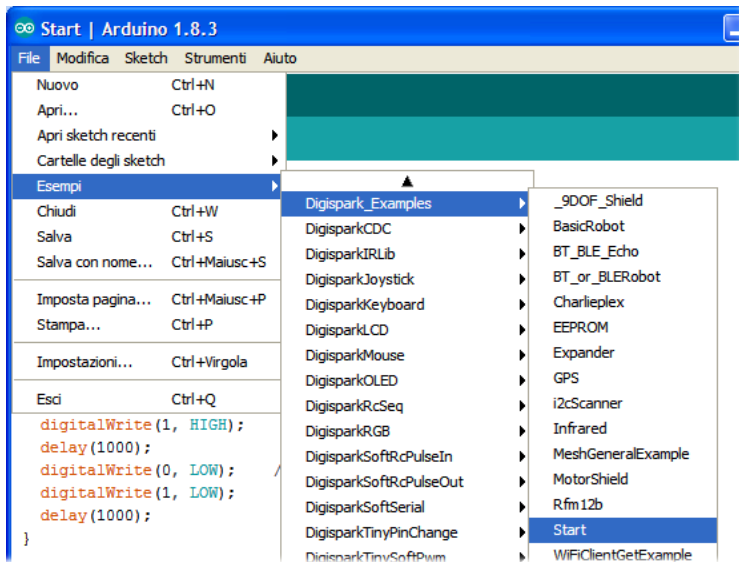
- Si vedrà il progresso del caricamento e al termine la scritta "**Caricamento terminato**", il codice sarà eseguito immediatamente sul Digispark.



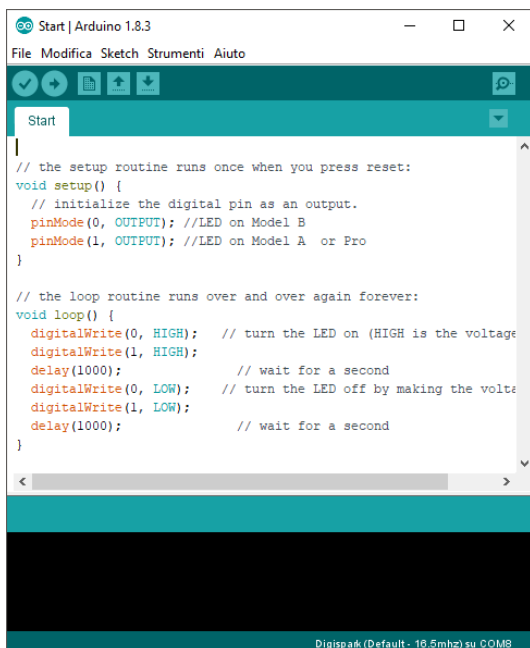
- Se si scollega il Digispark e si inserisce nuovamente o si collega ad un'altra sorgente di alimentazione, si verificherà un ritardo di 5 secondi prima che il codice programmato sia eseguito. Questo ritardo è dovuto al controllo di Digispark per verificare se si sta cercando di programmarlo.

Primo test

Per testare la scheda e la procedura, si può utilizzare l'esempio che si può trovare passando a **File** → **Esempi** → **Digispark_Examples** → **Start**



Ecco il codice che verrà caricato:



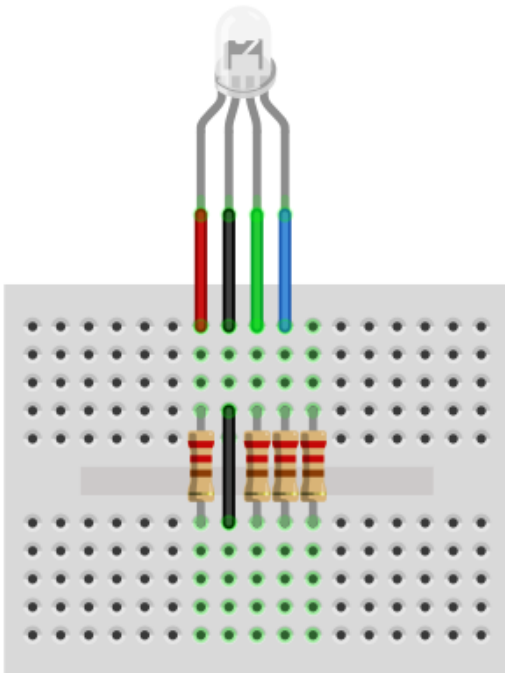
A questo punto inserite la vostra Digispark ed attendete che il caricamento vada a buon fine. Il Led sulla scheda lampeggerà al ritmo di 1Hz.

Led RGB:

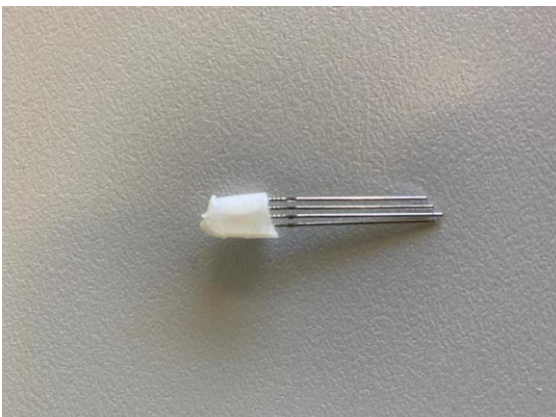
Come utilizzare il Led RGB

Per l'utilizzo di un led RGB bisogna avere come minimo a disposizione 3 pin liberi, e un pin che deve essere collegato alla massa.

Ai pin che controllano il colore del led bisogna attaccarci 3 resistenza da 150 Ohm.



Nell'immagine a sinistra si può vedere come bisogna attaccare il Led RGB. Al primo pin partendo da sinistra bisogna collegare il pin che imposterà il colore rosso al led. Nel secondo bisogna attaccare la massa, ovvero quel pin sull'Arduino con scritto GND oppure "-". Nel terzo bisogna collegare il pin che imposterà il colore verde al led. Nell'ultimo per esclusione si andrà a collegare il pin che imposterà il colore blu al led. Le resistenze che ho utilizzato nello schema sono solo indicative e non rispecchiano il vero colore della resistenza.



Un aiuto per migliorare l'esperienza con il Led RGB è di attaccarci un giro di scotch intorno così da rendere più omogeneo il colore che si andrà a vedere.

Come utilizzare la libreria del Led RGB:

L'accensione del Led RGB può avvenire grazie ad una libreria creata e ideata da me e Paolo Weishaupt.

Per il giusto funzionamento bisogna fare i seguenti passaggi:

- **Accertarsi che la libreria sia stata inclusa nel progetto:**

Facilmente verificabile controllando che all'inizio del programma ci sia scritta una stringa di codice simile a questa:

```
#include "LightLed.h"
```

- **Verificare che la variabile sia stata creata in modo corretto:**

Il modo più facile per dichiarare una variabile è il seguente:

LightLed* [nomeVariabile]; e nel metodo setup istanziarla.

- **Istanziare variabile:**

A questo punto bisognerà istanziarla, quindi recarsi nel metodo setup e istanziare la variabile dichiarata in precedenza nel seguente modo:

```
[nomeVariabile] = new LightLed(pin rosso, pin blu, pin verde);
```

- **Utilizzare il metodo per l'accensione del led:**

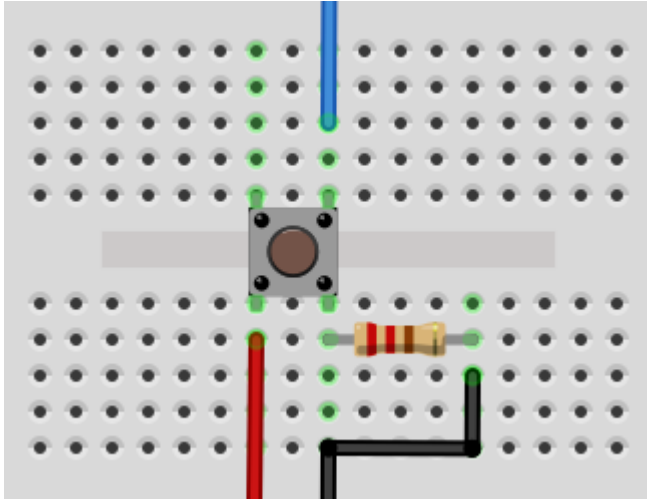
Per accendere il led bisognerà quindi chiamare un semplice metodo facilmente utilizzabile nel seguente modo:

```
[nomeVariabile]->lightLed(255, 0, 0);
```

Il seguente codice colorerà di rosso il led. I valori possono andare da 0 fino a 255.

Push button:

Come utilizzare un pulsante:



Nell'immagine a sinistra si può vedere come bisogna attaccare il pulsante al circuito. Il pin rosso sta a significare il "+", quello nero il "-" e quello blu il dato che si andrà a leggere che sarà 0 se il pulsante non sarà premuto o 1 quando lo sarà.

Come utilizzare la libreria del pulsante:

La lettura del pulsante può avvenire grazie ad una libreria creata e ideata da me e Paolo Weishaupt.

Per il giusto funzionamento bisogna fare i seguenti passaggi:

- **Accertarsi che la libreria sia stata inclusa nel progetto:**
Facilmente verificabile che all'inizio del programma ci sia scritta una stringa di codice simile a questa:

```
#include "ButtonState.h"
```
- **Verificare che la variabile sia stata creata in modo corretto:**
Il modo più facile per dichiarare una variabile è il seguente:
`ButtonState* [nomeVariabile];` e nel metodo setup istanziarla.
- **Istanziare variabile:**
A questo punto bisognerà istanziarla, quindi recarsi nel metodo setup e istanziare la variabile dichiarata in precedenza nel seguente modo:

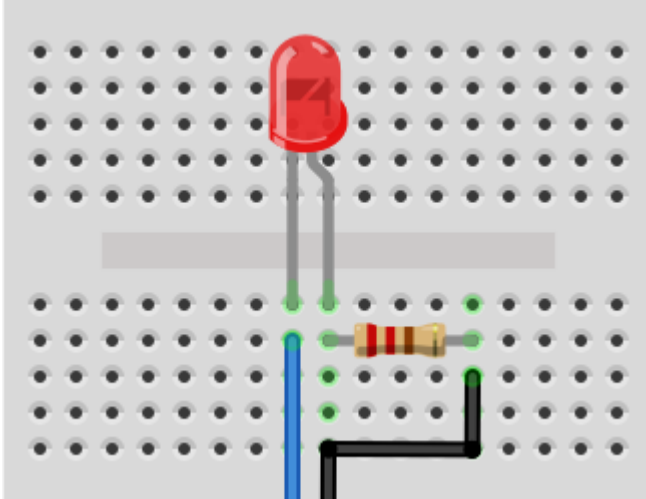
```
[nomeVariabile] = new ButtonState(pin del pulsante);
```
- **Utilizzare il metodo per la lettura del pulsante:**
Per leggere il pulsante bisognerà quindi chiamare un semplice metodo facilmente utilizzabile, per esempio nel seguente modo:

```
[nomeVariabile]->getStateButton()
```


Il seguente codice ritornerà lo stato del pulsante.

Led:

Come utilizzare un Led:



Nell'immagine a sinistra si può vedere come bisogna attaccare il Led al circuito. Il cavo blu sta ad indicare lo stato di scrittura. Il cavo nero invece il "-". La resistenza bisogna metterla per non rischiare di bruciare il led.

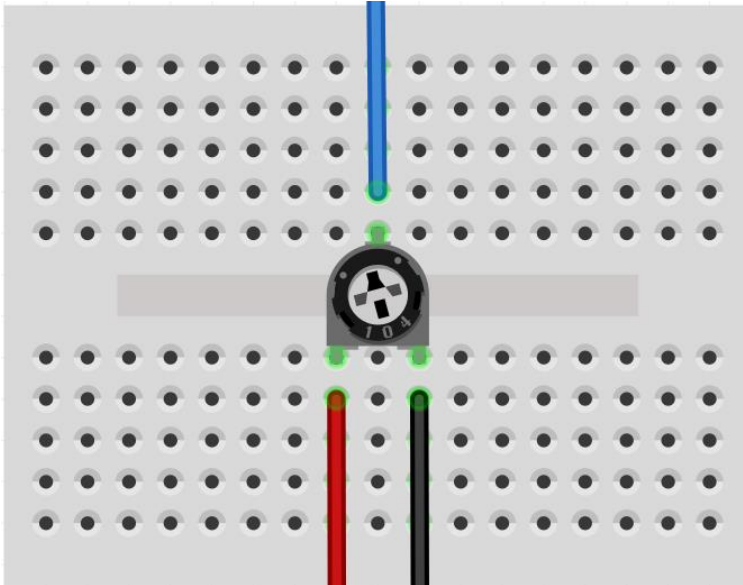
Come utilizzare la libreria del Led:

L'accensione del led può avvenire grazie ad una libreria creata e ideata da me e Paolo Weishaupt. Per il giusto funzionamento bisogna fare i seguenti passaggi:

- **Accertarsi che la libreria sia stata inclusa nel progetto:**
Facilmente verificabile che all'inizio del programma ci sia scritta una stringa di codice simile a questa:
`#include "LightSingleLed.h"`
- **Verificare che la variabile sia stata creata in modo corretto:**
Il modo più facile per dichiarare una variabile è il seguente:
`ButtonState* [nomeVariabile];` e nel metodo setup istanziarla.
- **Istanziare variabile:**
A questo punto bisognerà istanziarla, quindi recarsi nel metodo setup e istanziare la variabile dichiarata in precedenza nel seguente modo:
`led = new LightSingleLed(pin del led);`
- **Utilizzare il metodo per l'accensione del led**
Per accendere il led bisognerà quindi chiamare un semplice metodo facilmente utilizzabile nel seguente modo:
`led->lightSingleLed();`

Potenziometro:

Come utilizzare un potenziometro:



Nell'immagine a sinistra si può vedere come bisogna attaccare il potenziometro al circuito. Il cavo blu sta ad indicare lo stato di lettura. Il cavo nero invece il "-". Il cavo rosso invece sta a rappresentare il "+". La resistenza bisogna metterla per non rischiare di bruciare il led.

Come utilizzare la libreria del potenziometro:

La lettura dello stato del potenziometro si può eseguire grazie alla libreria scritta e ideata da me e Paolo Weishaupt.

- **Accertarsi che la libreria sia stata inclusa nel progetto:**
Facilmente verificabile controllando che all'inizio del programma ci sia scritta una stringa di codice simile a questa:

```
#include "Potenziometro.h"
```
- **Verificare che la variabile sia stata creata in modo corretto:**
Il modo più facile per dichiarare una variabile è il seguente:
`Potenziometro* [nomeVariabile];` e nel metodo setup istanziarla.
- **Istanziare variabile:**
A questo punto bisognerà istanziarla, quindi recarsi nel metodo setup e inizializzare la variabile dichiarata in precedenza nel seguente modo:

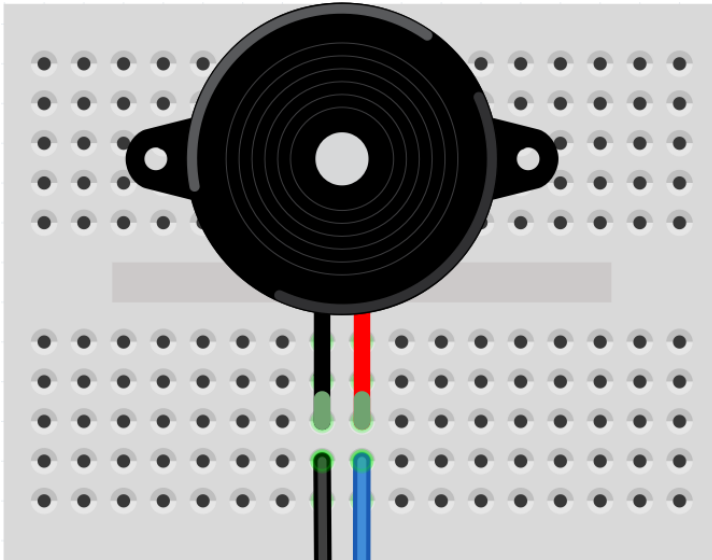
```
potenziometro = new Potenzometro(pin di lettura del potenziometro);
```
- **Utilizzare il metodo per la lettura del potenziometro**
Per leggere lo stato del potenziometro bisognerà quindi chiamare un semplice metodo facilmente utilizzabile nel seguente modo:

```
potenziometro->ottieniValorePotenziometro();
```


Il metodo ritorna un valore di tipo int che va di solito da 0 a 1023, dove 0 è il valore minimo di resistenza che il potenziometro può assumere e 1023 il valore massimo.

Piezo Buzzer:

Come utilizzare un Piezo Buzzer:



Nell'immagine a sinistra si può vedere come bisogna attaccare il Piezo Buzzer al circuito. Il cavo blu sta ad indicare lo stato di scrittura. Il cavo nero invece il “-”.

Come utilizzare la libreria del Piezo Buzzer:

La scrittura dello stato del piezo buzzer si può eseguire grazie alla libreria scritta e ideata da me e Paolo Weishaupt.

- **Accertarsi che la libreria sia stata inclusa nel progetto:**
Facilmente verificabile controllando che all’inizio del programma ci sia scritta una stringa di codice simile a questa:
`#include "Buzzer.h"`
- **Verificare che la variabile sia stata creata in modo corretto:**
Il modo più facile per dichiarare una variabile è il seguente:
`Buzzer* [nomeVariabile];` e nel metodo setup istanziarla.
- **Istanziare variabile:**
A questo punto bisognerà istanziarla, quindi recarsi nel metodo setup e inizializzare la variabile dichiarata in precedenza nel seguente modo:
`buzzer = new Buzzer(pin buzzer);`
- **Utilizzare il metodo per la lettura del potenziometro**
Per far suonare il piezo buzzer bisognerà quindi chiamare un semplice metodo facilmente utilizzabile nel seguente modo:
`buzzer->beep(tempo di suono, tonalità del suono);`

Sitografia

- <https://www.adrirobot.it/arduino/digispark/digispark.htm>