

# LED

## Collegare LED RGB a Raspberry Pi

Il pin più lungo del LED è il **common**, che sarà un anodo o un catodo a seconda della tipologia del LED.

### Verificare la tipologia del LED

- Mettere il multmetro in modalità contatto
- Collegare il multmetro al common e a un altro pin qualsiasi
- Se il LED si illumina quando il pin common è collegato al cavo di ground, allora il common è un **catodo**.
- Se il LED si illumina quando il pin common è collegato al cavo positivo, allora il common è un **anodo**

### Forward voltage

Il forward voltage è la quantità di volt che il led assorbe e determina quindi, a parità di resistenza, una corrente maggiore.

**Metodo rapido:** Calcolare la resistenza basandosi sul forward voltage minimo fornito dalle specifiche del led

**Metodo esaustivo:** Determinare manualmente il forward voltage

- Mettere il multmetro in modalità diodo
- Collegare il multmetro al common e a uno dei restanti pin RGB
- Il valore misurato è il forward voltage per quello specifico colore

Esempio

Red  $V_f = 1.8V$

Green  $V_f = 2.8V$

Blue  $V_f = 2.8V$

Il voltaggio di alimentazione che dobbiamo fornire al LED deve essere maggiore di ogni forward voltage di un singolo colore.

Esempio:  $V_{CC} = 3.3V$  (maggiore di 2.8V)

Verificare la corrente tipica sulle specifiche del LED.

Esempio:  $I = 25mA = 0.025A$  (per singolo colore)

### Nota

Il raspberry pi 4 supporta ufficialmente massimo 16mA per pin e 50mA in totale. Occorre quindi abbassare la corrente in modo che  $I < 16mA$  e  $3 \times I < 50mA$

Esempio:  $I = 10\text{mA} = 0.01\text{A}$

Notare che dimezzando la corrente la differenza luminosa percepita è in realtà molto piccola

## Calcolo della resistenza

Dividere la differenza tra VCC e Vf per l'amperaggio stabilito, ottenendo la minima resistenza necessaria.

**Formula:**  $R = (VCC - Vf) / I$

Calcolo:

- Red  $R = (3.3\text{V} - 1.8\text{V}) / 0.01\text{A} = 150\text{ohm}$
- Green  $R = (3.3\text{V} - 2.8\text{V}) / 0.01\text{A} = 50\text{ohm}$
- Blue  $R = (3.3\text{V} - 2.8\text{V}) / 0.01\text{A} = 50\text{ohm}$

## Collegamento al Raspberry

Collegare a ognuno dei tre pin RGB la relativa resistenza calcolata

Attaccare ogni resistenza a un pin GPIO

Se il common del LED è un anodo, collegarlo a 3v3

Se è un catodo, collegarlo a GND

## Installazione software di controllo

Scaricate sul raspberry tutti gli script per led presenti sul git iotaiuto

```
cd $HOME
mkdir led_scripts
cd led_scripts
# posizionare qui i seguenti file:
# led_init.sh, led_lib.py, led.py, led_server.py
chmod +x led_init.sh

cd /etc/systemd/system
# posizionare qui i seguenti file:
# led.service, led-server.service

sudo systemctl daemon-reload
sudo systemctl enable led.service
sudo systemctl enable led-server.service

# riavviare il sistema
```

Spiegazione dei file: *ledlib.py* è il controllore dei led *ledserver.py* è un listener di richieste di controllo dei led *led.py* è un handler per eseguire una richiesta di controllo del led manualmente o tramite

home assistant led\_init.sh è uno script che monitora lo stato della vpn (e eventualmente di altri servizi) e regola il led di conseguenza

## Home assistant script

Modificare il file ~/homeassistant/configuration.yaml aggiungendo:

```
shell_command:
  led_ok: python ~/led_scripts/led.py ok
  led_error: python ~/led_scripts/led.py error
```

Vedere il file led.py per ulteriori comandi

Restartare home assistant Ora dalla GUI è possibile aggiungere due script che chiamano i servizi appena creati

From:  
<https://wiki.csgalileo.org/> - **Galileo Labs**



Permanent link:  
<https://wiki.csgalileo.org/projects/iotaiuto/led?rev=1640337001>

Last update: **2021/12/24 10:10**