



Blink

Arduino, STM32, Raspberry

# Diodo LED

- I diodi LED hanno diversi colori e sopportano tutti una corrente massima di 30 mA. Varia invece la tensione assorbita in base a questa tabella:

Colore	Tensione (V)
Infrarosso	1.3
Rosso	1.8
Verde	1.9
Giallo	2.0
Blu	3.5

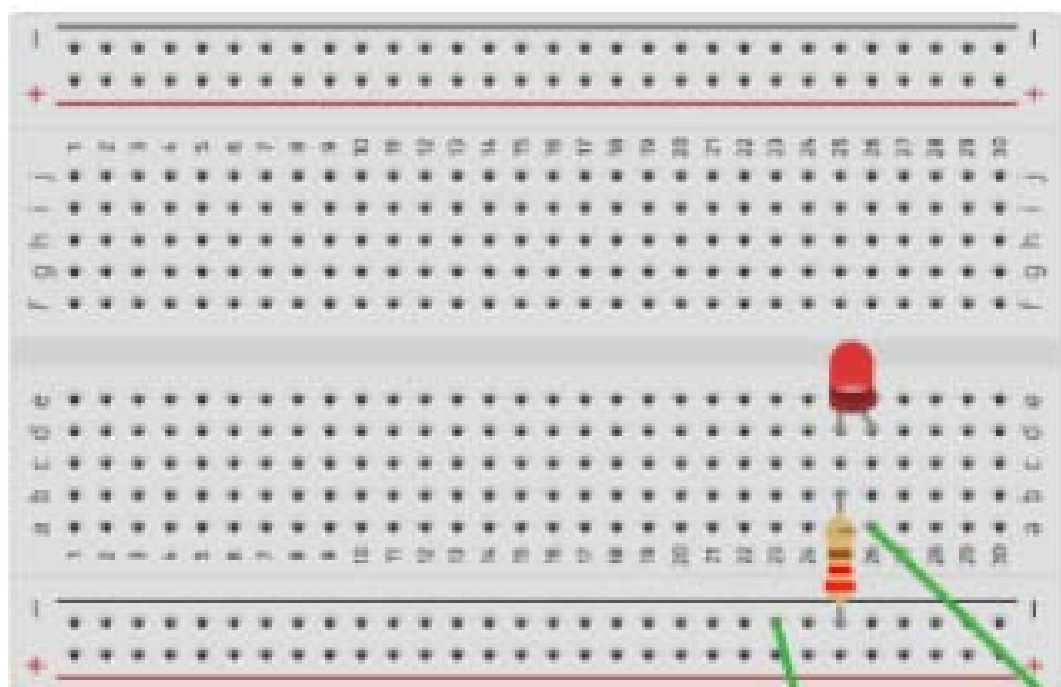
I pin delle tre schede in considerazione erogano una tensione di 5V. Per fare in modo che non ci siano guasti, bisogna introdurre dei resistori in serie e dimensionarle per calcolarne il valore R. Supponiamo che la tensione di ciascun diodo sia 1.8V e la corrente 20 mA:

$$5V - 1.8V = 0.02 * R \rightarrow R = 3.2 / 0.02 = 160 \text{ Ohm}$$

In genere si utilizzano resistori da 220 Ohm

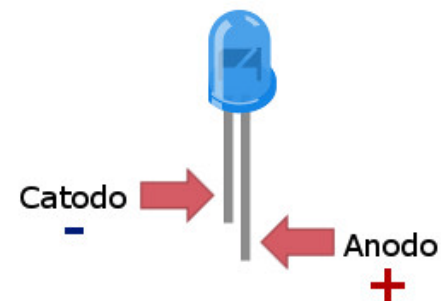
# Circuito base per le tre schede

I diodi led sono elementi circuitali polari. Il pin anodo deve essere collegato sull'alimentazione e il pin catodo sul GND



catodo

anodo



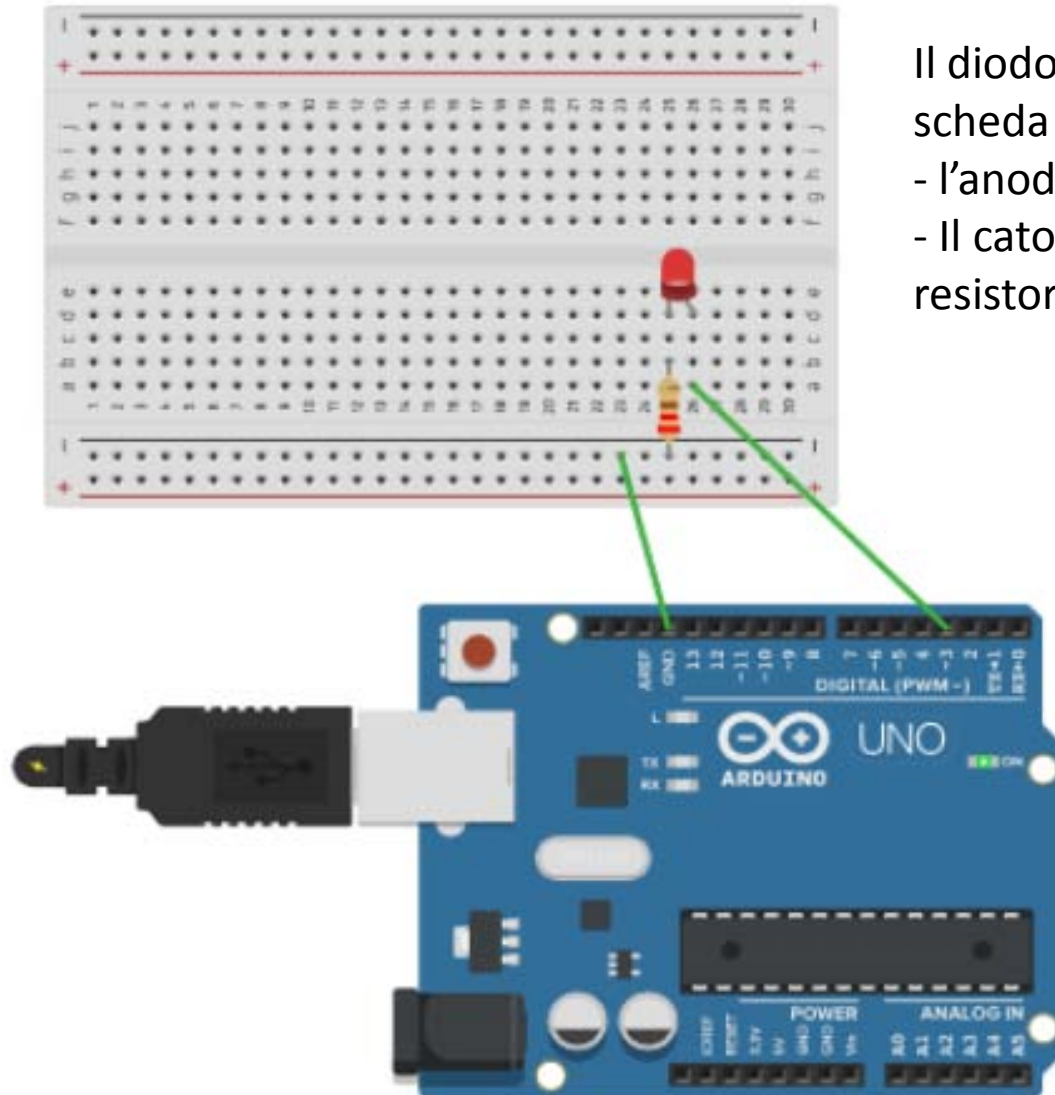
# Blink scheda Arduino UNO

The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.13". The menu bar includes "File", "Modifica", "Sketch", "Strumenti", and "Aiuto". The toolbar contains icons for saving, running, and uploading. The main text area displays the following C++ code for a Blink sketch:

```
Blink $  
  
int led=3; //dichiarazione grandezza globale  
//il pin 3 è denominato led  
void setup() {  
  pinMode(led,OUTPUT); //il pin led è di output  
}  
  
void loop() {  
  digitalWrite(led, HIGH); il pin led è a livello logico alto  
  delay(3000); rimane a livello logico alto per 3000 millisecondi  
  digitalWrite(led,LOW); il pin led è a livello logico basso  
  delay(2000); rimane a livello logico basso per 2000 millisecondi  
}
```

At the bottom, a status bar shows "Salvataggio completato" and a message: "Trovata libreria non valida in C:\Program Files (x86)\Arduino\libraries\ITEADLIB\_Arduino\_Nextion: Missing".

# Circuito Blink Arduino UNO



Il diodo led viene collegato alla scheda Arduino nel seguente modo:

- l'anodo sul pin 2
- Il catodo sul GND attraverso il resistore da 220 Ohm

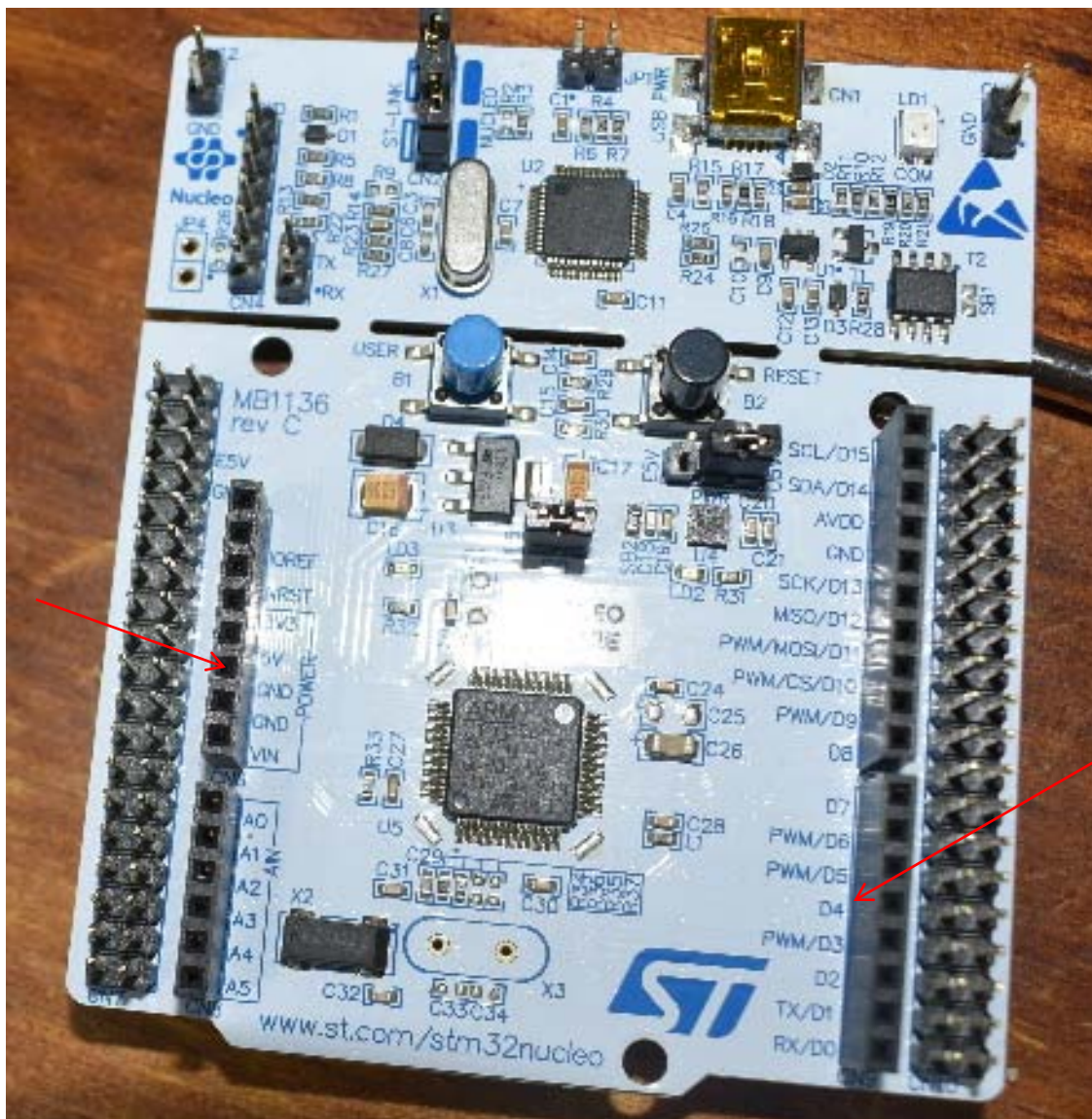
# Blink STm32

The screenshot shows the Mbed IDE interface. The top bar displays the URL `ide.mbed.com/compiler/#nav:/Nucleo_blink_led/main.cpp` and the version `1.10.25.0`. The main workspace shows a project named `Nucleo_blink_led` with a file `main.cpp` open. The code in `main.cpp` is as follows:

```
1 #include "mbed.h"
2 DigitalOut myled(D4); // dichiarazione variabile globale
3
4 int main() {
5     while(1) {
6         myled = 1; // il pin myled è ON
7         wait(2); // 2 sec
8         myled = 0; // il pin myled è OFF
9         wait(1); // 1 sec
10    }
11 }
12
```

The board selection dropdown menu is set to `NUCLEO-F401RE`. A red arrow points to this dropdown, with the text `Scheda selezionata` next to it. The bottom of the IDE shows the compile output for the program, with 0 errors, warnings, and infos.

# Scheda STM32F401



GND  
Il catodo è  
collegato  
su uno dei  
GND

D4  
L'anodo viene collegato  
sul pin D4

# Blink Raspberry pi3

## Piedinatura

Raspberry Pi2 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)	DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)	(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 1  
26/01/2014

<http://www.element14.com>

Raspberry Pi 3 GPIO Header

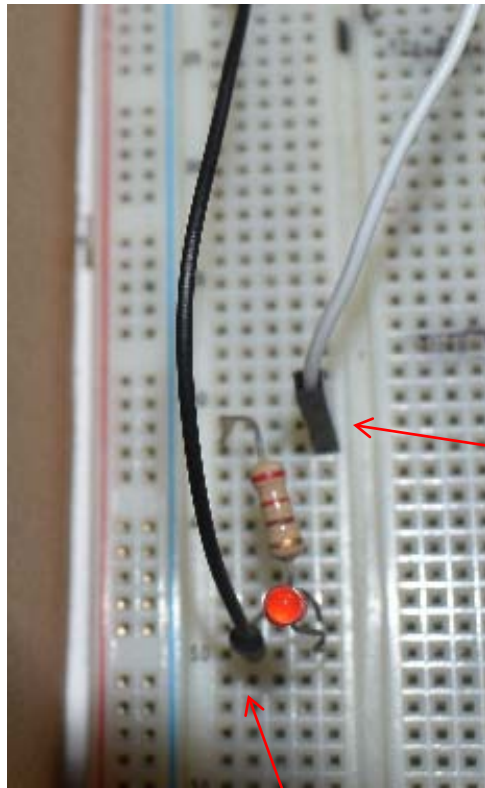
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)	DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)	(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 2  
29/02/2016

[www.element14.com/RaspberryPi](http://www.element14.com/RaspberryPi)

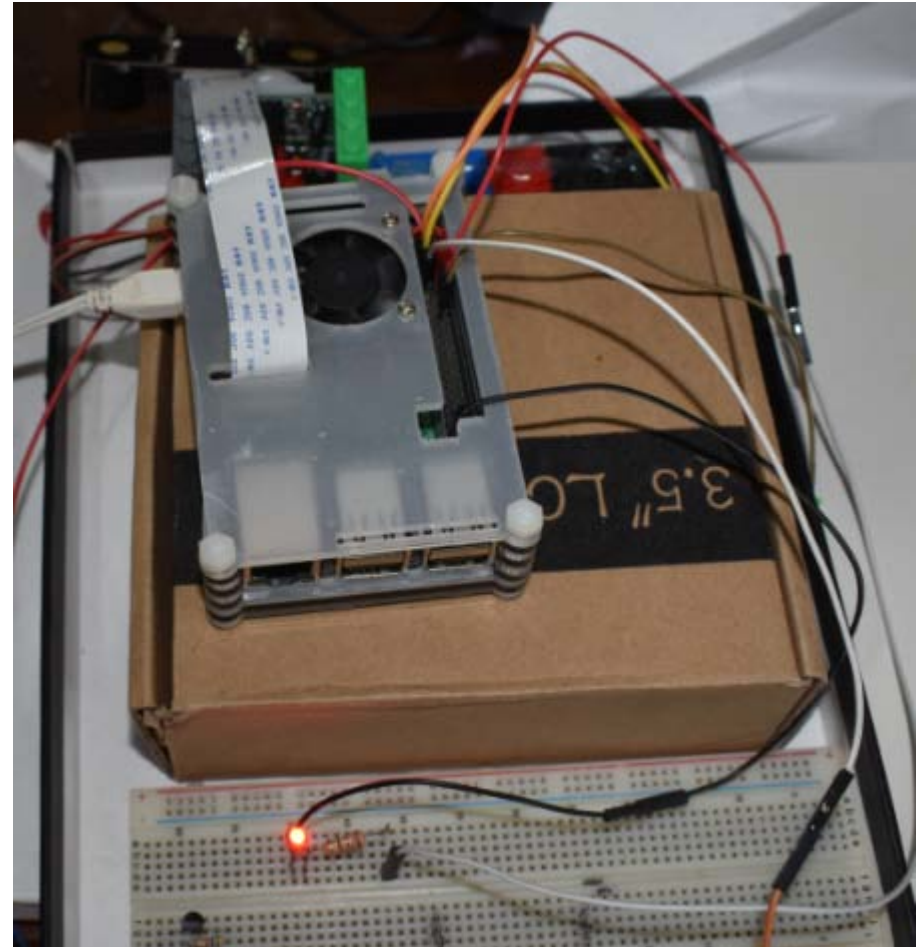


# Circuito Blink Raspberry PI3



Anodo

Catodo



# Comandi di Bash in Raspbian per accendere e spegnere un diodo led

- 1 `gpio -g mode 18 out`
- 2 `gpio -g write 18 1`
- 3 `gpio -g write 18 0`

Spiegazione:

- 1 pone il pin 18 come output
- 2 il pin 18 è a livello logico alto
- 3 il pin 18 è a livello logico basso

# Programma in Python per Blink

```
import RPi.GPIO as gpio #libreria
import time             #libreria
gpio.setwarnings(False) #tutti i pin impegnati vengono #disimpegnati
gpio.setmode(gpio.BCM) #il nome dei pin è quello del #costruttore e
                       non sequenziale
gpio.setup(18,gpio.OUT) #pin 18 OUTPUT (18 come BCM, 12 come
                       #sequenza)
while True:             #fino a che è alimentata la scheda
    gpio.output(18,True) #il led si accende
    time.sleep(2)        #acceso per 2 secondi
    gpio.output(18,False) #il led si spegne
    time.sleep(1)        #spento per 1 secondo
```