

Università degli Studi della Campania Luigi Vanvitelli - Dipartimento di Ingegneria

Laurea Triennale in Ingegneria Elettronica e Informatica

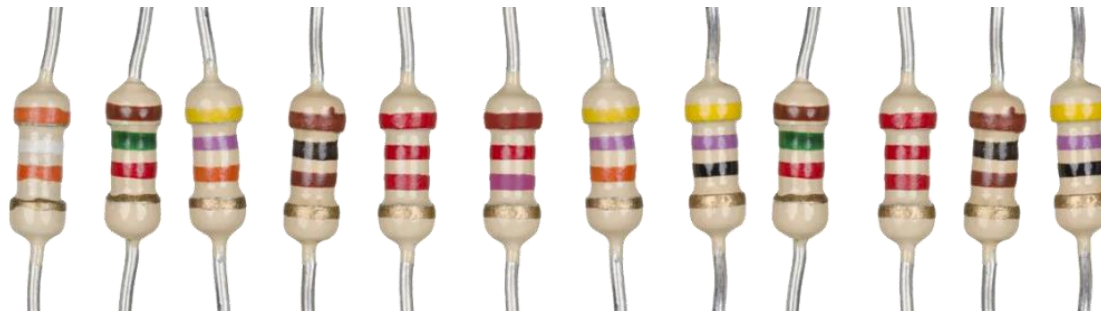
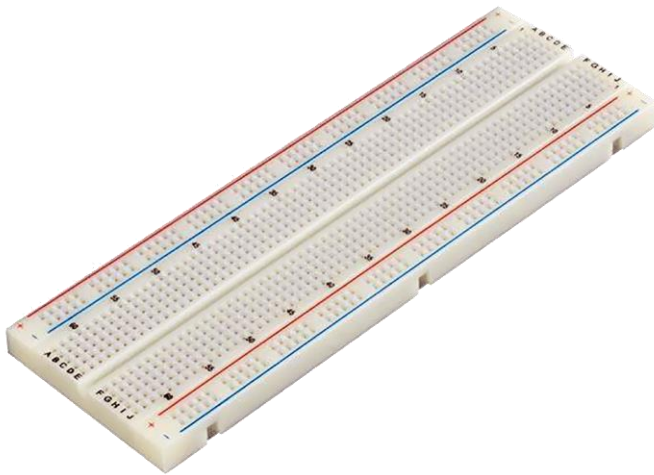
**Laboratorio di Sviluppo di Applicazioni per IoT
a.a. 2023-2024**

Esempi di Base

Docente: Carlo Mazzocca
e-mail: carlo.mazzocca@unibo.it

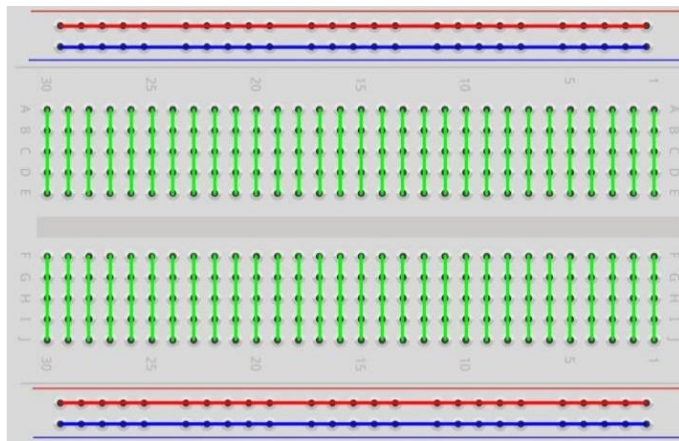
Componenti Utili

- **Breadboard, resistori e jumper** sono componenti fondamentali nello sviluppo di applicazioni IoT



Breadboard

- Una breadboard è una piastra forata in cui è possibile inserire componenti elettronici senza saldatura
- Consente di creare e testare circuiti rapidamente
- Le linee di alimentazione si trovano lungo i bordi
- I ranghi e le colonne dei fori sono collegati elettricamente



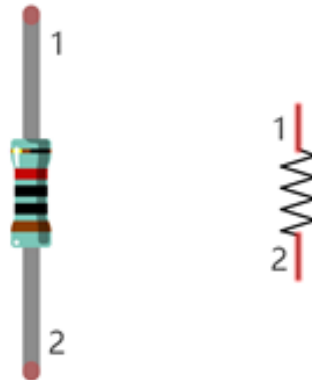
Jumper

- Un jumper è un connettore che collega componenti elettronici sulla breadboard
- Possono maschio-maschio, maschio-femmina e femmina-femmina
- Rigidi o flessibili



Resistore

- Un resistore è un componente elettronico passivo che limita o regola il flusso di corrente in un circuito elettronico
- Viene utilizzato per prevenire danni ai componenti
- I resistori sono contrassegnati con una serie di strisce colorate, dove ognuna rappresenta un valore numerico
- **L'unità di misura della resistenza è l'Ohm (Ω)**

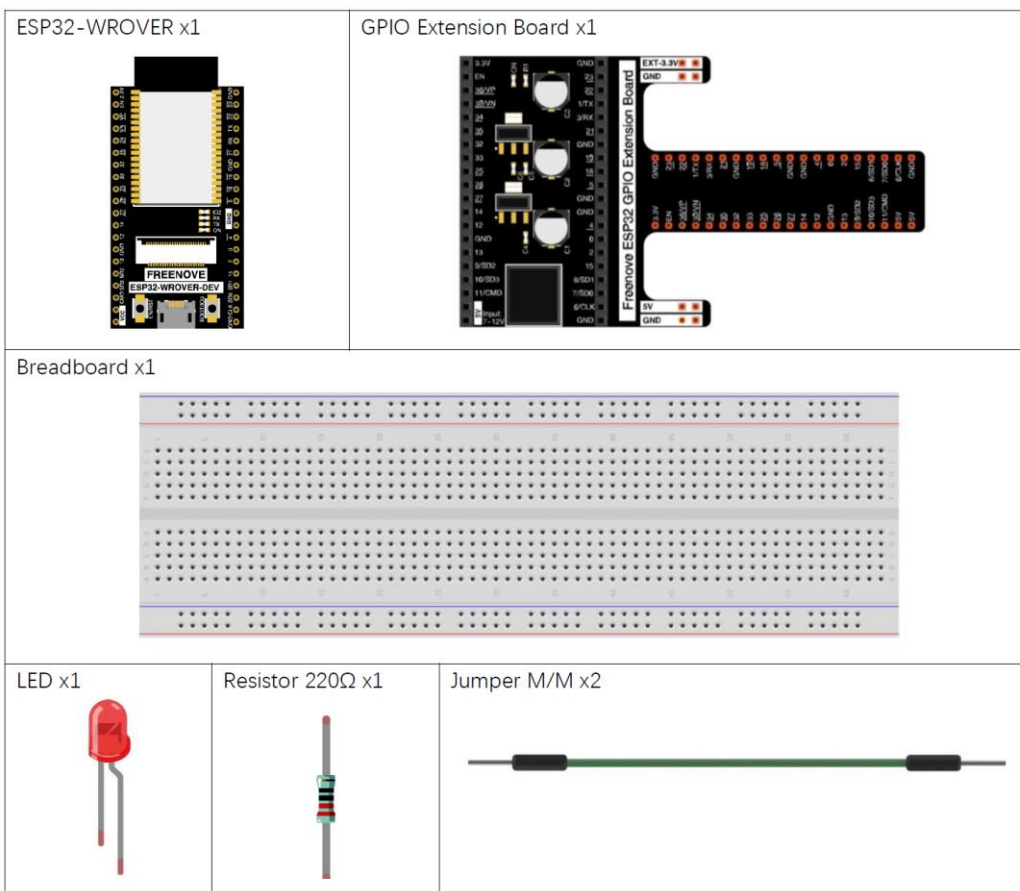


Esempio Calcolo Corrente

- La corrente massima che un led rosso può sopportare è di 20mA
- Per la legge di Ohm: $corrente = \frac{tensione}{resistenza}$
- Considerando un'alimentazione di 3.3V, ci basterà utilizzare una resistenza di $220\Omega = 0.22k\Omega$
- Infatti $\rightarrow corrente = \frac{3.3V}{0.22k\Omega} = 15mA$

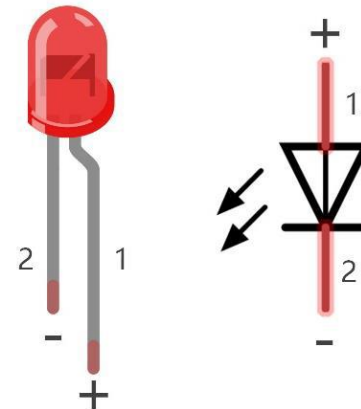
Blink LED Esterno

- Proviamo ad utilizzare breadboard, jumper e resistenze per fare accendere e spegnere un LED esterno



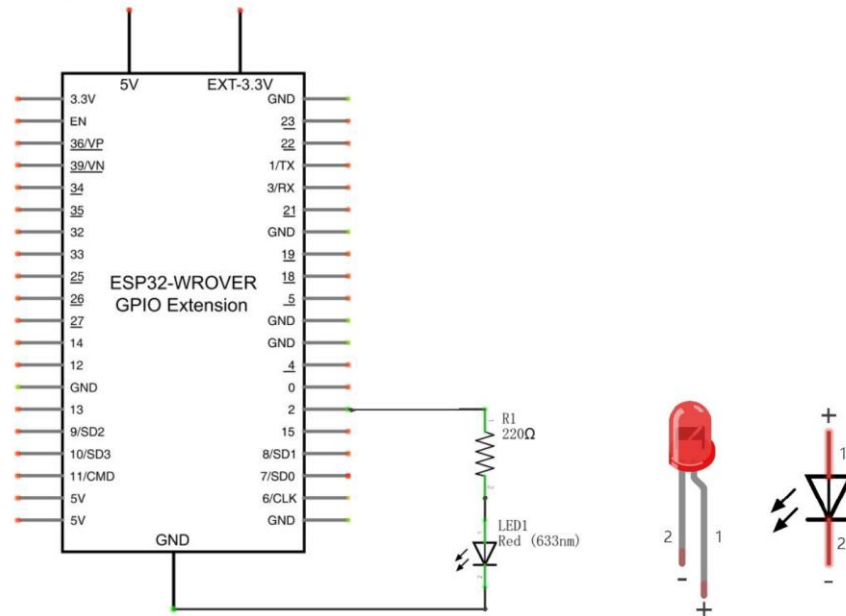
LED

- Un LED è un diodo, per cui funziona solo se la corrente fluisce nella giusta direzione
- In particolare, il pin più lungo deve essere connesso all'uscita positiva di corrente, mentre quello più corto deve essere connesso a quella negativa, anche detta **ground (GND)**
- Un voltaggio troppo alto può danneggiare il LED, rendendolo inutilizzabile

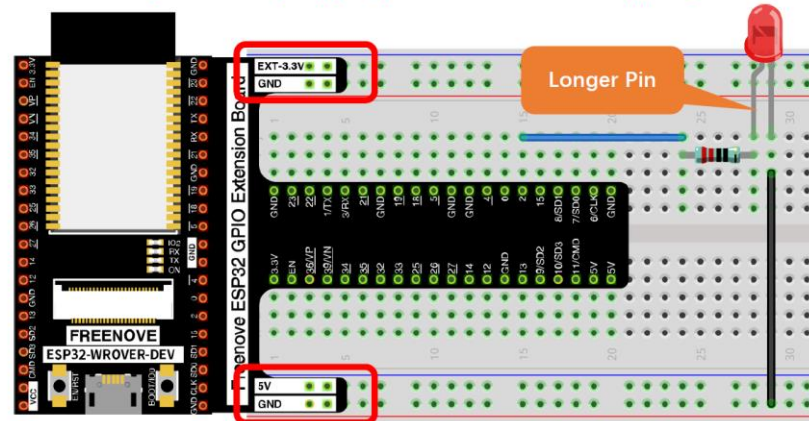


Circuito

Schematic diagram



Hardware connection. If you need any support, please contact us via: support@freenove.com



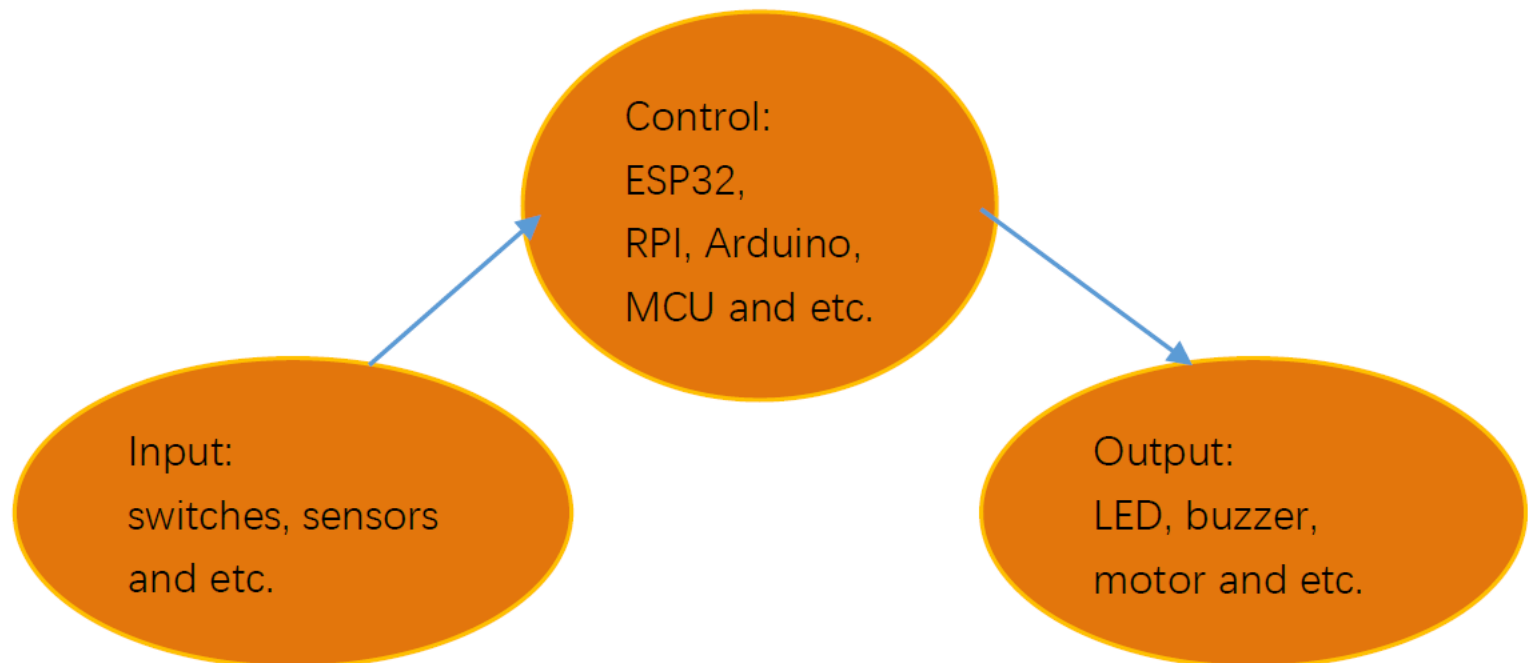
Don't rotate ESP32-WROVER 180° for connection.

Blink LED - Sketch

```
1  #define PIN_LED 2
2  // the setup function runs once when you press reset or power the board
3  void setup() {
4      // initialize digital pin LED_BUILTIN as an output.
5      pinMode(PIN_LED, OUTPUT);
6  }
7
8  // the loop function runs over and over again forever
9  void loop() {
10     digitalWrite(PIN_LED, HIGH); // turn the LED on (HIGH is the voltage level)
11     delay(1000);                  // wait for a second
12     digitalWrite(PIN_LED, LOW);  // turn the LED off by making the voltage LOW
13     delay(1000);                  // wait for a second
14 }
```

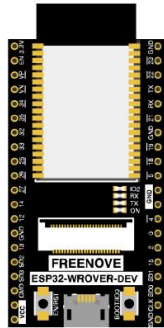
Pulsante & LED

- Un'applicazione si compone solitamente di tre parti fondamentali: **INPUT**, **OUTPUT** e **CONTROL**
- Utilizziamo un pulsante per controllare lo stato del LED: quando il pulsante è premuto il LED è acceso, quando viene rilasciato il LED si spegne

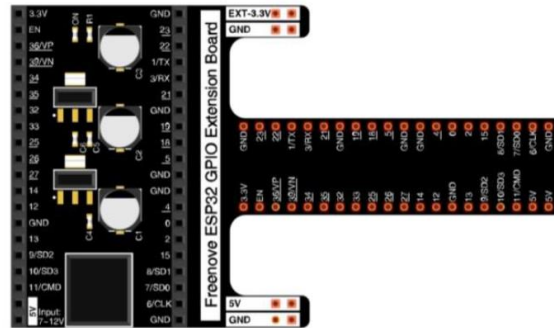


Componenti

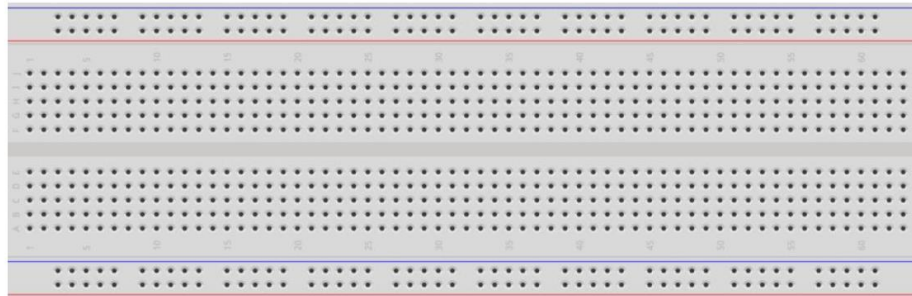
ESP32-WROVER x1



GPIO Extension Board x1



Breadboard x1



Jumper M/M x4



LED x1



Resistor 220Ω x1



Resistor 10kΩ x2

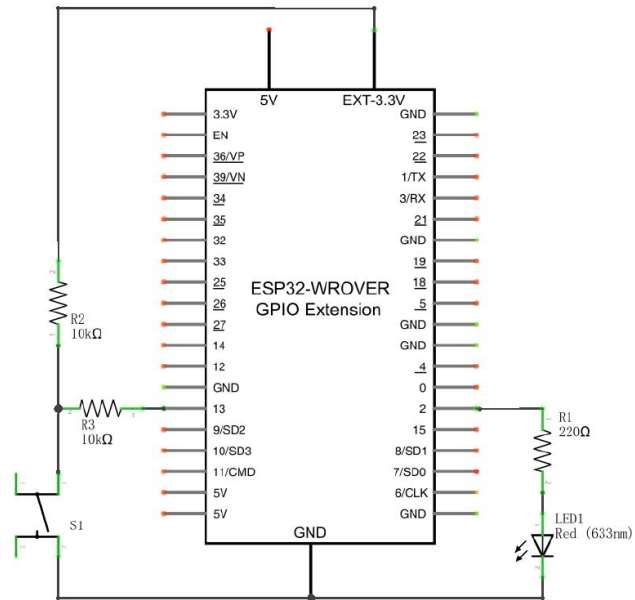


Push button x1

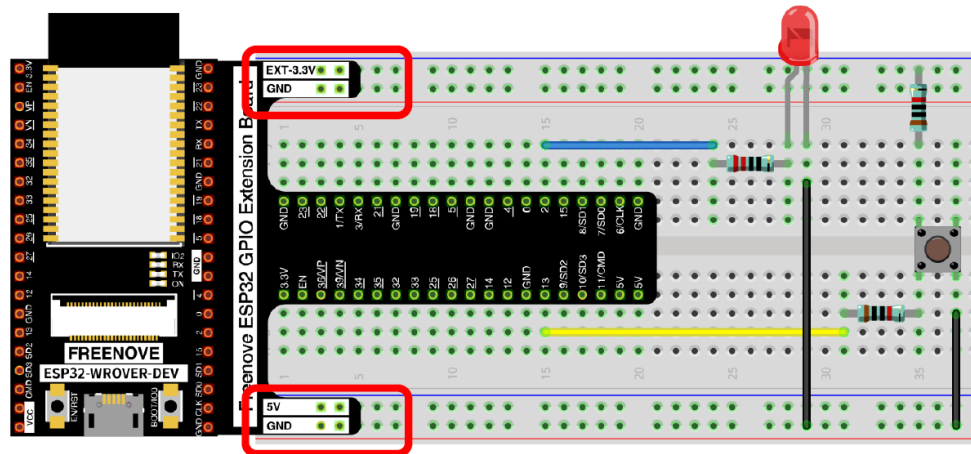


Circuito

Schematic diagram

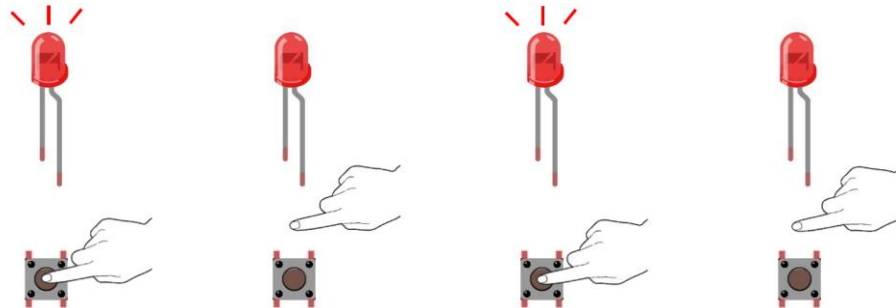


Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



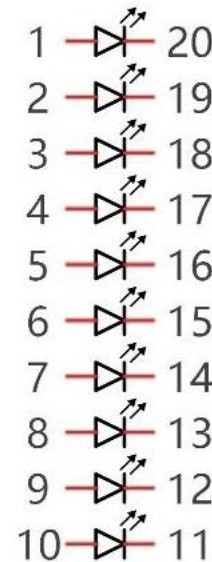
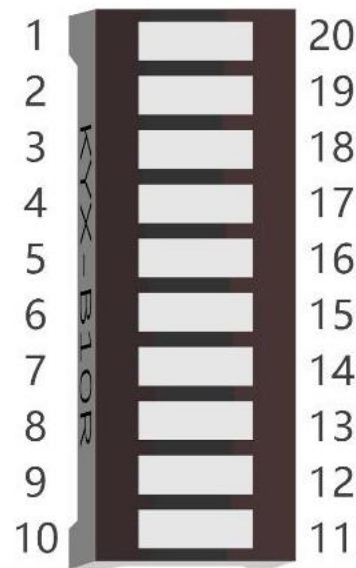
Pulsante & LED - Sketch

```
1  #define PIN_LED    2
2  #define PIN_BUTTON 13
3  // the setup function runs once when you press reset or power the board
4  void setup() {
5      // initialize digital pin PIN_LED as an output.
6      pinMode(PIN_LED, OUTPUT);
7      pinMode(PIN_BUTTON, INPUT);
8  }
9
10 // the loop function runs over and over again forever
11 void loop() {
12     if (digitalRead(PIN_BUTTON) == LOW) {
13         digitalWrite(PIN_LED, HIGH);
14     } else {
15         digitalWrite(PIN_LED, LOW);
16     }
17 }
```



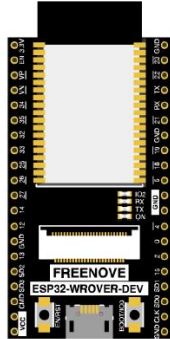
LED a Barre

- Un LED a barre contiene 10 LED integrati, dove ogni led è identificato da una coppia di PIN come per il LED che abbiamo utilizzato precedentemente

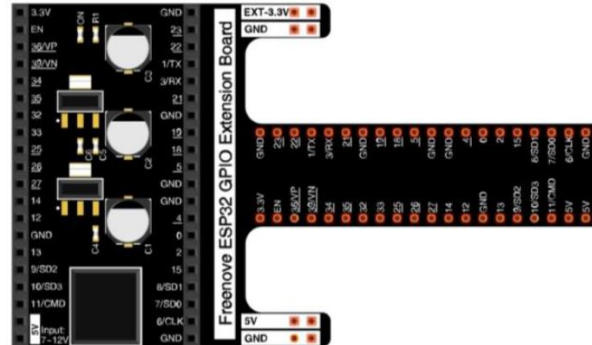


Componenti

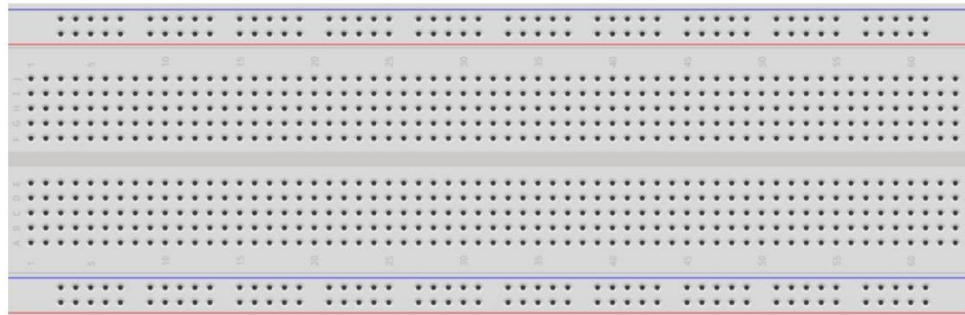
ESP32-WROVER x1



GPIO Extension Board x1



Breadboard x1



Jumper M/M x10



LED bar graph x1

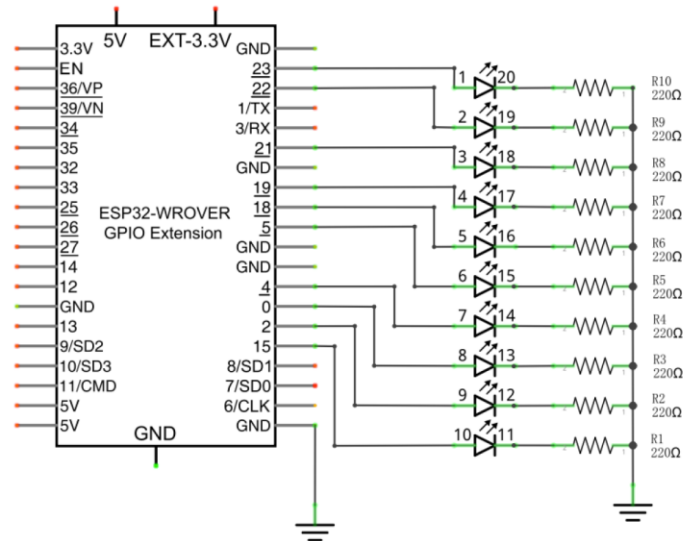


Resistor 220Ω x10

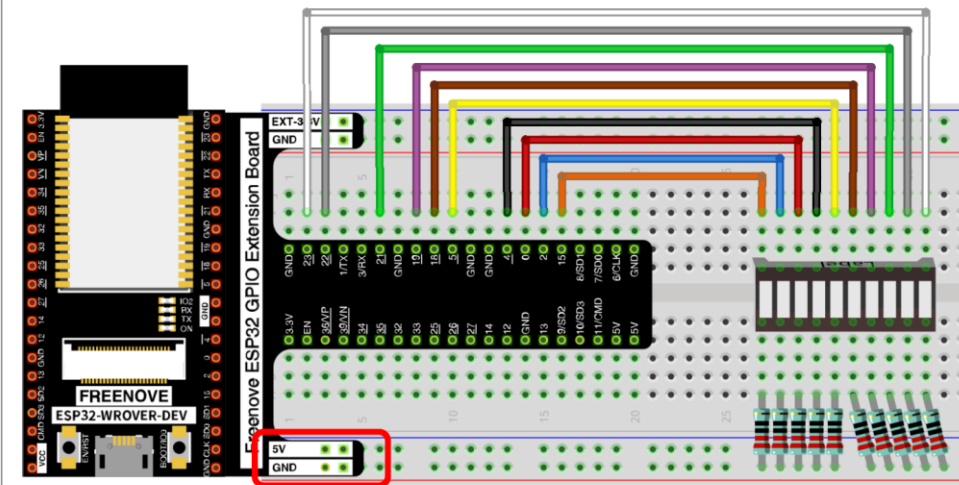


Circuito

Schematic diagram



Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



If LED bar does not work, try to rotate it for 180°. The label is random.

LED a Barre - Sketch

```
1 byte ledPins[] = {15, 2, 0, 4, 5, 18, 19, 21, 22, 23};
2 int ledCounts;
3
4 void setup() {
5     ledCounts = sizeof(ledPins);
6     for (int i = 0; i < ledCounts; i++) {
7         pinMode(ledPins[i], OUTPUT);
8     }
9 }
10
11 void loop() {
12     for (int i = 0; i < ledCounts; i++) {
13         digitalWrite(ledPins[i], HIGH);
14         delay(100);
15         digitalWrite(ledPins[i], LOW);
16     }
17     for (int i = ledCounts - 1; i > -1; i--) {
18         digitalWrite(ledPins[i], HIGH);
19         delay(100);
20         digitalWrite(ledPins[i], LOW);
21     }
22 }
```

