Una striscia di 120 LED NeoPixel, alimentata come si deve

Due metri di LED indirizzabili con la FireBeetle 2 ESP32-E: alimentatore a parte, corrente ai due capi, condensatore, resistenza sul dato e traslatore di livello. Poi una cometa e un arcobaleno.

Guida online, sempre aggiornata

A cura di **Ada Intelisano** · redazione tecnica di informaticait.it

IN BREVE

- 120 LED tutti bianchi possono chiedere fino a 7,2 A (conto di Adafruit): con un alimentatore da 5 V e 4 A la luminosità si limita a 96 su 255.
- La corrente entra ai due capi, così nessun LED è a più di 1 m da un punto di alimentazione; i dati entrano solo dall'inizio.
- Condensatore da 1000 µF sull'alimentazione, resistenza da 470 Ω sul Din vicino ai LED, masse tutte in comune.
- La FireBeetle lavora a 3,3 V: il Pixel Shifter porta il segnale dei dati a 5 V.

Tempo: un pomeriggio | Livello: si saldano tre fili | Prima: guide n. 21 e 48

Nella guida n. 48 dodici LED indirizzabili si accendevano dal pin 5V di Arduino. Una striscia vera è un'altra storia: 120 LED in due metri, a bianco pieno, chiedono molta più corrente di quella che dà una porta USB, e lungo la striscia la tensione cala. Qui la colleghiamo come si deve: alimentatore a parte, corrente portata ai due capi, condensatore, resistenza sul filo dei dati e un traslatore di livello, perché la FireBeetle 2 ESP32-E lavora a 3,3 V. Alla fine due effetti. Le regole generali sono nell'articolo «LED indirizzabili NeoPixel: alimentazione e collegamenti» di «Notizie e consigli»; qui le mettiamo in pratica.

1 Il materiale

- la **striscia DFRobot FIT0613**: 120 LED RGB indirizzabili in 2 m, cioè 60 al metro, a 5 V, larga 10 mm, con l'adesivo sul retro;
- la **FireBeetle 2 ESP32-E** con i pin saldati e il suo cavo USB-C;
- l'**Adafruit Pixel Shifter**, che porta il segnale dei dati da 3,3 a 5 V, tutto a morsetti a vite;
- un **alimentatore da 5 V e 4 A** con spinotto 5,5 × 2,1 mm e l'**adattatore** Adafruit da presa DC a morsetti;
- un **condensatore elettrolitico da 1000 µF**, da 6,3 V o più (noi lo usiamo da 10 V), e una **resistenza da 470 Ω**;
- due **morsetti a leva** da tre fili, 4 m di **cavo rosso e nero** da 0,75 mm², tre cavetti femmina-maschio, guaina termorestringente, saldatore e tester.

Parola nuova: iniezione di corrente
Portare l'alimentazione in più punti della striscia, non solo all'inizio. I dati entrano sempre e soltanto dal primo LED; la corrente, scrive Adafruit, può arrivare da qualunque parte.

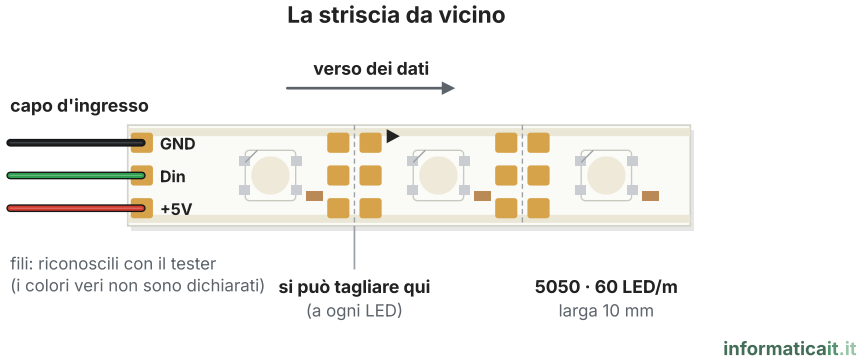
2 Prima il conto della corrente

Il conto di Adafruit	Per LED	120 LED
Caso peggiore: tutti bianchi alla massima luce	60 mA	**7,2 A**
Regola per le animazioni normali	20 mA	2,4 A
Caso peggiore con `setBrightness(96)` (calcolo nostro)	–	circa 2,7 A

Il conto è quello della guida di Adafruit: numero di LED per 20, o per 60 se vuoi stare sicuro, diviso 1000, e hai gli ampere. DFRobot per questa striscia dichiara un consumo che non coincide con il conto di Adafruit: noi usiamo quello più prudente.

L'alimentatore da 4 A basta per le animazioni, non per due metri di bianco alla massima luce. Per questo negli sketch la luminosità è fissata con `setBrightness(96)`: su 255 è il 38 %, e anche nel caso peggiore si resta intorno a 2,7 A (calcolo nostro, 7,2 × 96 ÷ 255, supponendo che la corrente scenda in proporzione). Se un giorno vuoi più luce, prendi un alimentatore con più ampere: per Adafruit «ampere in più vanno bene, volt in più no».

3 Prepara la striscia

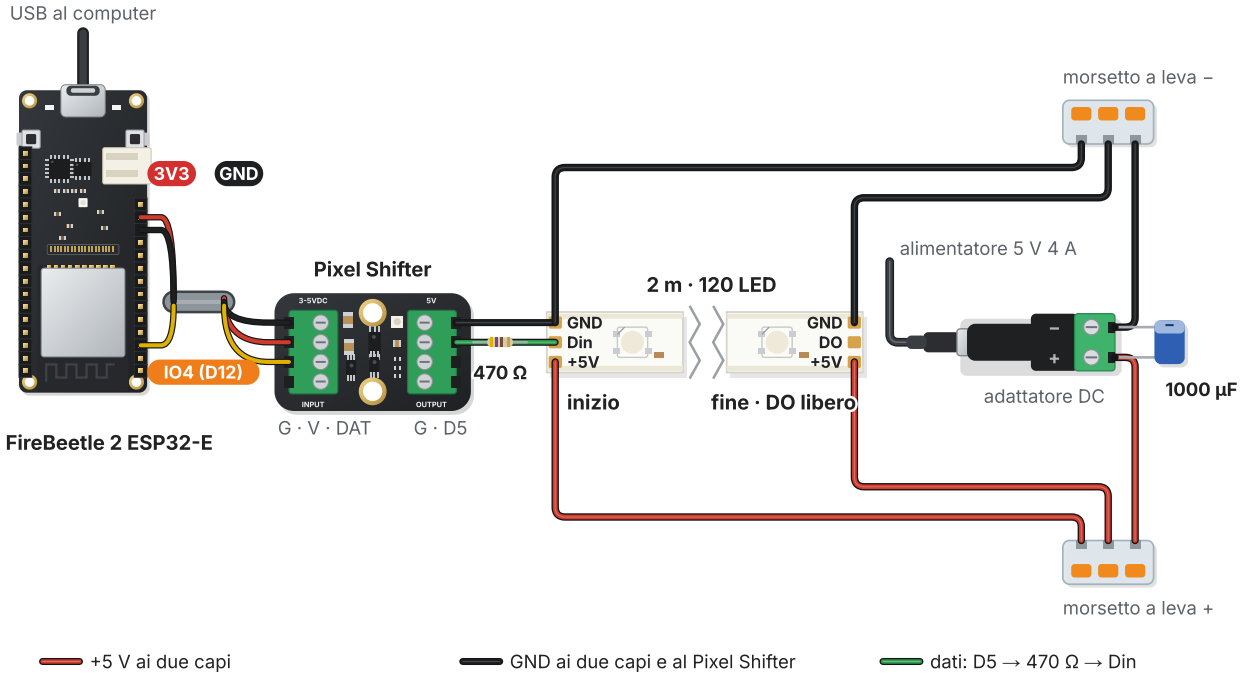


- **Trova l'inizio.** Adafruit avverte che il connettore non indica sempre il capo d'ingresso: valgono i triangoli stampati, che puntano nel verso dei dati. L'inizio è il capo con **Din**.
- **Riconosci i fili.** A ogni capo DFRobot monta un connettore a tre vie e due fili sciolti, ma non scrive quale filo è quale. Con il tester in continuità (guida n. 8) cerca il filo che arriva alla piazzola +5V, quello di Din (o DO) e quello di GND, e segnali con un pezzo di nastro. Adafruit lo ripete: fidati sempre delle scritte stampate sulla striscia, non dei colori.
- **All'inizio** taglia il connettore lasciando i fili lunghi qualche centimetro: Din e GND vanno al Pixel Shifter, il +5V del connettore lo isoli con la guaina. I due fili sciolti portano l'alimentazione.
- **Alla fine** servono solo i due fili sciolti, +5V e GND. DO resta libero: in fondo alla striscia non serve niente.

- L'esempio nel wiki DFRobot alimenta la striscia dal 5V di una Arduino UNO e conta 180 LED: non seguirlo.

4 I collegamenti

I collegamenti: dati da un capo, alimentazione dai due capi



informaticait.it

- **Dati.** IO4 (D12) della FireBeetle va a **DAT** del Pixel Shifter; **D5** va al Din della striscia passando per la resistenza da 470 Ω. Adafruit vuole la resistenza dal lato dei LED: saldala sul filo Din, vicino alla striscia, e coprila con la guaina. Tieni corto il filo dei dati: per Adafruit uno o due metri di solito vanno bene.
- **Pixel Shifter.** V al **3V3** e G al **GND** della FireBeetle: Adafruit chiede di alimentarlo alla tensione logica della scheda (accetta da 3 a 5 V) e i 5 V per i dati se li fa da solo. In uscita non c'è un +5 V: la G d'uscita va al GND della striscia, ed è così che le masse diventano una sola.
- **Alimentazione.** Il + e il − dell'adattatore vanno ai due morsetti a leva; da ogni morsetto partono un filo per l'inizio e uno per la fine della striscia, rosso sul +5V e nero sul GND. Per il tratto lungo usa il cavo da 0,75 mm² (proposta nostra: Adafruit non indica una sezione).
- **Condensatore.** I suoi due reofori vanno negli stessi morsetti dell'adattatore, insieme ai fili: la fascia chiara (−) sul −.

5 Accendi nell'ordine giusto

1. Prima di collegare la striscia infila l'alimentatore e misura con il tester sui morsetti dell'adattatore: sul **+** devono esserci +5 V. L'adattatore dà per scontato il positivo al

centro dello spinotto, e Qoltec non dichiara la polarità.
2. Stacca la spina e collega tutto: prima i GND, poi il +5 V, per ultimo i dati. È l'ordine che indica Adafruit.
3. Dai corrente alla striscia **prima** della FireBeetle: altrimenti i LED provano ad alimentarsi dal filo dei dati.
4. Solo adesso collega la FireBeetle al computer con il cavo USB.

6 La prima prova

```
// Prova della striscia: i 120 LED si accendono uno alla volta, poi tutti bianchi.
// Sul Monitor seriale (115200) la corrente stimata con la regola di Adafruit.
#include <Adafruit_NeoPixel.h>

const int PIN_DATI = 4;      // IO4 = D12, va al DAT del Pixel Shifter
const int NUM_LED = 120;     // 2 metri da 60 LED al metro
const int LUMINOSITA = 96;   // su 255: tiene la corrente sotto i 4 A (vedi la guida)

Adafruit_NeoPixel striscia(NUM_LED, PIN_DATI, NEO_GRB + NEO_KHZ800);

// Adafruit conta fino a 60 mA per LED a bianco pieno (rosso + verde + blu): qui li
// dividiamo in parti uguali, 20 mA per colore al valore 255 (stima nostra).
// getPixels() dà i valori che partono davvero, già ridotti da setBrightness.
float stimaAmpere() {
  uint8_t *valori = striscia.getPixels();
  long somma = 0;
  for (int i = 0; i < NUM_LED * 3; i++) {
    somma += valori[i];
  }
  return somma * 0.020 / 255.0;
}

void riempi(uint32_t colore) {   // un LED alla volta, dal primo all'ultimo
  for (int i = 0; i < NUM_LED; i++) {
    striscia.setPixelColor(i, colore);
    striscia.show();
    delay(15);
  }
  Serial.printf("Corrente stimata: %.2f A\n", stimaAmpere());
}

void setup() {
  Serial.begin(115200);
  striscia.begin();
  striscia.setBrightness(LUMINOSITA);   // una volta sola, qui nel setup
  striscia.clear();
  striscia.show();
}

void loop() {
  riempi(striscia.Color(255, 0, 0));       // rosso
  riempi(striscia.Color(0, 255, 0));       // verde
  riempi(striscia.Color(0, 0, 255));       // blu
```

```
  riempi(striscia.Color(255, 255, 255));   // bianco: il caso che chiede più corrente
  delay(5000);                             // i LED in fondo restano bianchi?
  striscia.clear();
  striscia.show();
  delay(1000);
}
```

`stimaAmpere()` somma i valori di rosso, verde e blu che partono davvero verso la striscia, già ridotti da `setBrightness`, e li trasforma in ampere con il conto di Adafruit. È una stima, utile per non sforare.

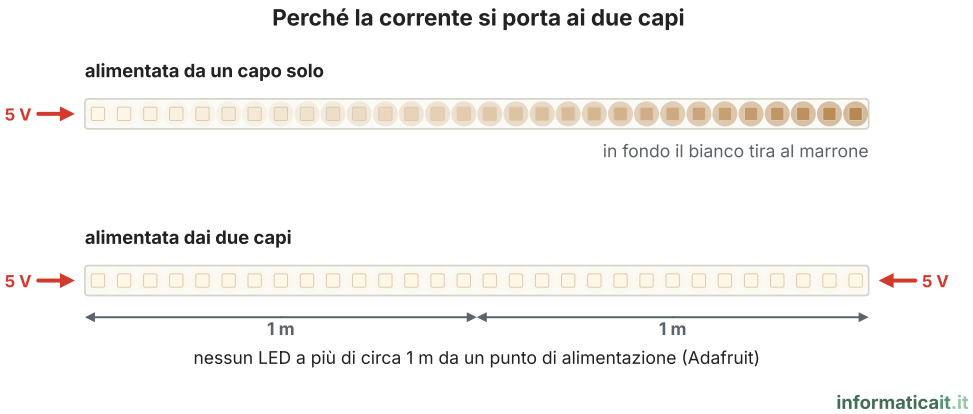
✓ Cosa devi vedere

I LED si accendono uno alla volta fino in fondo: rossi, verdi, blu, poi bianchi. Sul Monitor seriale, a 115200 baud, la stima della corrente: 0,90 A con un colore solo, 2,71 A con il bianco.

Se non funziona

- Tutto spento: guarda il piccolo LED sul Pixel Shifter, che sta sul filo dei dati. Se si accende, i dati arrivano: controlla il Din (hai preso il capo giusto?) e il GND tra Pixel Shifter e striscia.
- Il primo LED va e gli altri fanno colori a caso: manca la massa in comune, o il filo dei dati è troppo lungo.
- Rosso e verde scambiati: nella riga della striscia prova `NEO_RGB` al posto di `NEO_GRB`.
- Con il bianco gli ultimi LED tirano al giallo o al marrone: i fili della fine non portano corrente. Controlla i morsetti a leva.

7 Perché ai due capi



Il rame della striscia ha una sua resistenza: più la corrente viaggia, più la tensione cala. Adafruit lo descrive così: i LED più lontani diventano marroncini, perché il blu e il verde vogliono più tensione del rosso. La regola che dà è tenere ogni LED entro circa 1 m da un punto di alimentazione. Con due metri, alimentando i due capi, il LED più lontano è proprio a metà, a un metro.

Per vederlo, con il bianco della prova stacca dal morsetto a leva i due fili della fine e confronta il primo e l'ultimo LED, poi ricollegali.

8 Due effetti: la cometa e l'arcobaleno

```
// Due effetti sui 120 LED che si danno il cambio ogni 15 secondi: cometa e arcobaleno.
#include <Adafruit_NeoPixel.h>

const int PIN_DATI = 4;               // IO4 = D12, va al DAT del Pixel Shifter
const int NUM_LED = 120;
const int LUMINOSITA = 96;            // la stessa della prova: corrente sotto i 4 A
const int CODA = 20;                  // quanti LED è lunga la scia della cometa
const unsigned long DURATA = 15000;   // millisecondi per ogni effetto

Adafruit_NeoPixel striscia(NUM_LED, PIN_DATI, NEO_GRB + NEO_KHZ800);

int testa = 0;          // dove si trova la cometa
uint16_t tinta = 0;     // colore della cometa e punto di partenza dell'arcobaleno

// La cometa: la testa piena e dietro la scia, sempre più scura.
void cometa() {
  striscia.clear();
  for (int j = 0; j < CODA; j++) {
    int i = testa - j;
    if (i < 0) {
      i += NUM_LED;                   // la scia continua dall'inizio della striscia
    }
    uint8_t luce = 255 - j * 255 / CODA;
    striscia.setPixelColor(i, striscia.gamma32(striscia.ColorHSV(tinta, 255, luce)));
  }
  striscia.show();
  testa = (testa + 1) % NUM_LED;      // un LED più avanti
  tinta += 256;                       // e un colore un po' diverso
  delay(15);
}

// L'arcobaleno: tutto il giro dei colori steso sui 120 LED, che scorre a ogni passo.
void arcobaleno() {
  for (int i = 0; i < NUM_LED; i++) {
    uint16_t t = tinta + i * 65536L / NUM_LED;
    striscia.setPixelColor(i, striscia.gamma32(striscia.ColorHSV(t)));
  }
  striscia.show();
  tinta += 256;
  delay(10);
}

void setup() {
  striscia.begin();
  striscia.setBrightness(LUMINOSITA);
  striscia.show();
}

void loop() {
  if ((millis() / DURATA) % 2 == 0) {
    cometa();
  } else {
    arcobaleno();
```

```
  }
}
```

- **La cometa**: a ogni passo la striscia si spegne, poi si accendono venti LED, la testa e la scia, ognuno un po' più scuro. Il terzo numero di `ColorHSV` è la luce, da 255 a quasi zero.
- **L'arcobaleno**: il giro dei colori, 65536 tinte, steso su 120 LED; ogni passo lo sposta di 256.
- `gamma32` rende i colori più naturali: Adafruit consiglia di usarla sempre insieme a `ColorHSV`.
- `millis() / DURATA % 2` vale 0 per 15 secondi, poi 1, poi di nuovo 0: è il cambio di effetto, senza fermare il programma.

✓ Cosa devi vedere
La cometa corre lungo i due metri cambiando colore; dopo 15 secondi parte l'arcobaleno che scorre, poi torna la cometa.

Prova tu

- Aggiungi un terzo effetto: il `theaterChase` dell'esempio strandtest della libreria.
- Cambia `LUMINOSITA` nello sketch di prova e guarda la stima: resta sotto i 4 A dell'alimentatore.
- Taglia la striscia su una linea di taglio e aggiorna `NUM_LED`.

✓ Cosa hai imparato

- la corrente si calcola prima di scegliere l'alimentatore;
- i dati entrano dall'inizio, la corrente si porta ai due capi;
- condensatore sull'alimentazione, resistenza sul Din dal lato dei LED, masse in comune;
- con una scheda a 3,3 V i dati passano da un traslatore a 5 V.

Trovi i componenti su informaticait.it

- Striscia LED RGB indirizzabile WS2812, 120 LED, 2 m, 5 V, PCB bianco
- Level shifter Pixel Shifter per LED indirizzabili NeoPixel
- Scheda FireBeetle 2 ESP32-E con pin saldati
- Alimentatore a spina 5 V DC 4 A 20 W con spinotto 5,5×2,1 mm, Qoltec
- Adattatore da presa DC 2,1 mm a morsettiera a vite, centrale positivo
- Morsetti rapidi a leva per giunzione di 3 fili, trasparenti
- Cavo per altoparlanti Goobay 2 × 0,75 mm² in rame puro OFC, rosso/nero

Clicca sul nome per aprire la scheda prodotto. Disponibilità e prezzi aggiornati sul sito.

Fonti: DFRobot, striscia FIT0613: pagina prodotto, wiki e foto · Adafruit, Pixel Shifter (6066): pagina prodotto, guida Learn e file Eagle · Adafruit, «NeoPixel Überguide»; libreria Adafruit NeoPixel 1.15.5 ed esempio strandtest · DFRobot, FireBeetle 2 ESP32-E: wiki e schema; Espressif, arduino-esp32 · Qoltec, alimentatore 50778; Adafruit, adattatore 368 · informaticait.it: articolo «LED indirizzabili NeoPixel: alimentazione e collegamenti»; guide n. 8, 21 e 48