

## INTRODUZIONE AL PROGETTO:

Il progetto si propone di mettere in campo le competenze e le abilità acquisite durante la prima parte del corso "Didattica creativa con Micro:bit e Arduino".  
L'idea è nata dall'esigenza di dover creare un piccolo progetto con la scheda "Micro:bit" per quanto concerne la sua programmazione ed alcuni componenti disponibili in laboratorio.

## DESCRIZIONE TECNICA:

Il programma elaborato tramite <https://makecode.microbit.org/#editor> permette alla scheda "Micro:bit" di eseguire i comandi stabiliti nella fase di programmazione secondo le seguenti istruzioni:

AVVIO: melodia riprodotta dal passive buzzer

TASTO A: accensione dei led

TASTO B: luminosità a diversi livelli (basso, medio e alto)

POTENZIOMETRO: variazione della velocità nel movimento dei led

TASTI A+B: spegnimento dei led

## ELENCO COMPONENTI UTILIZZATE:

- scheda microbit
- breadboard
- adattatore scheda
- striscia dritta 8 led
- striscia circolare 8 led
- potenziometro
- passive buzzer
- 5 cavi maschio-femmina
- 8 cavi maschio-maschio

## LINK DI RIFERIMENTO:

Video sul funzionamento:

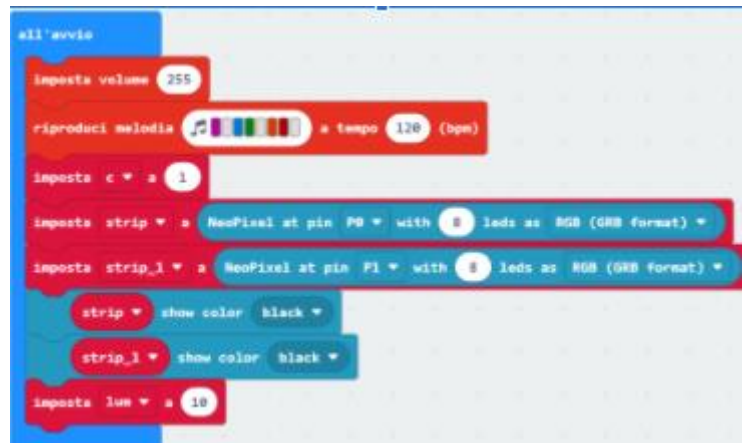
<https://youtu.be/cH9I7m2JW9A>



## CODICE MAKECODE:

### - IMPOSTAZIONI INIZIALI:

A seguito della melodia riprodotta dal passive buzzer, segnale di avvio del programma, sono stati impostati i valori da 1 e a 10 per le variabili "c" e "lum". Successivamente le due strisce led vengono attivate e colorate di nero affinché queste risultino sicuramente spente.

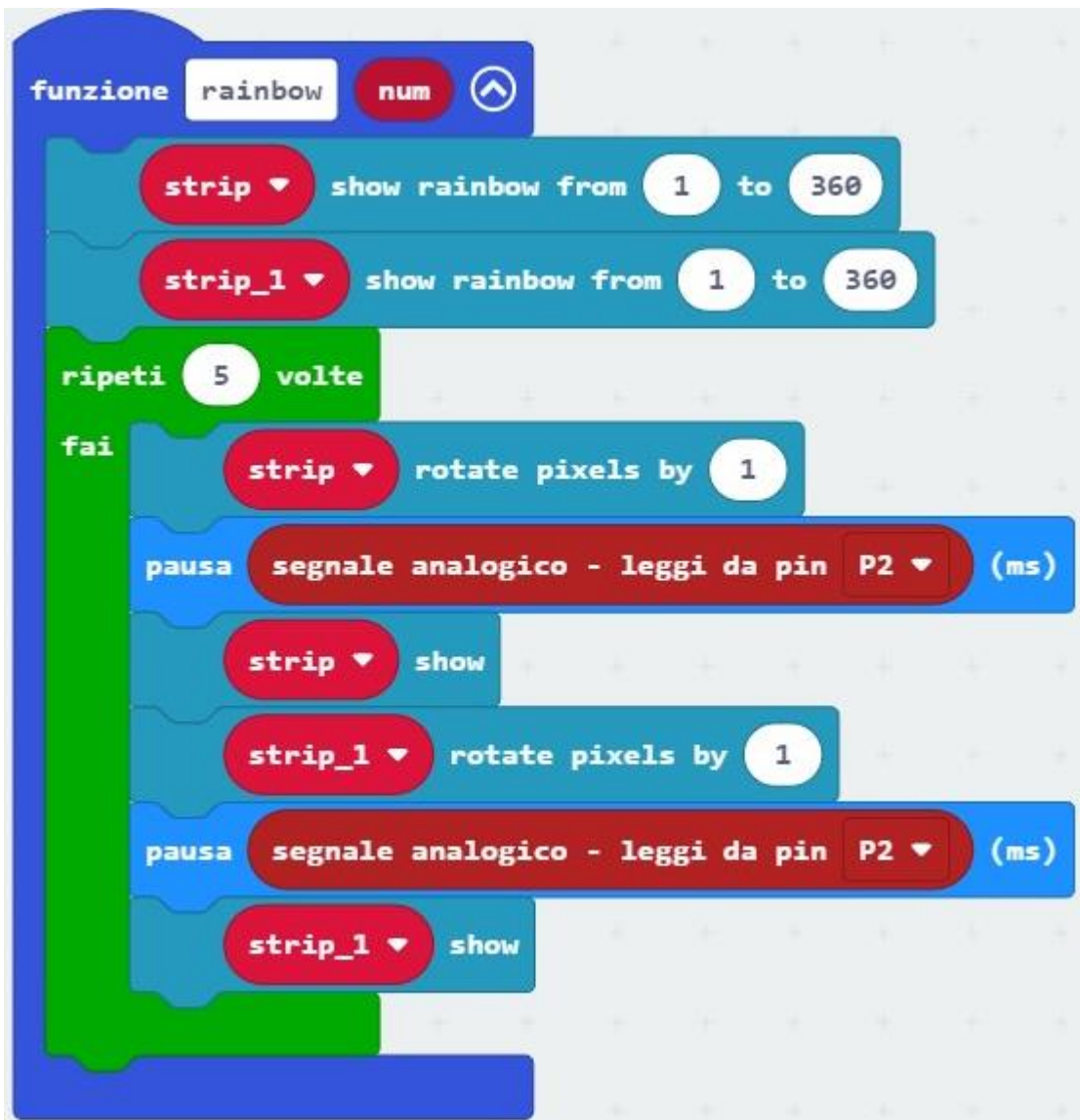


In questo blocco si sono create le impostazioni di accensione le quali garantiscono il funzionamento grazie alla variabile "c". Come dimostrato in precedenza la variabile "c" assume valore 1 e fino a quando il valore resterà invariato i led risultano disattivati. Quando si pigia il tasto "A", sito sulla scheda "Micro:bit" il valore di "c" muterà poiché sarà portato a 0 e chiamerà la funzione "rainbow". Premendo i tasti "A" e "B" contemporaneamente il valore di "c" tornerà ad essere pari ad 1 e i led si spegneranno.



## - FUNZIONE RAINBOW:

In questa sezione ci si è occupati di far accendere i led. Si è iniziato mostrando gli otto colori per poi, attraverso un ciclo, spostare di un led ogni colore creando così il disegno ritmico di colori in movimento. Tra una variazione e l'altra è stata programmata una pausa pari al segnale analogico del potenziometro: così che si possa regolare la velocità con cui i colori eseguono il movimento.



```
funzione rainbow num
  strip show rainbow from 1 to 360
  strip_1 show rainbow from 1 to 360
  ripeti 5 volte
  fai
    strip rotate pixels by 1
    pausa segnale analogico - leggi da pin P2 (ms)
    strip show
    strip_1 rotate pixels by 1
    pausa segnale analogico - leggi da pin P2 (ms)
    strip_1 show
```

The image shows a Scratch code block for a function named 'rainbow' that takes a parameter 'num'. The function starts by showing a rainbow on two LED strips, 'strip' and 'strip\_1', from pixel 1 to 360. It then enters a loop that repeats 5 times. Inside the loop, it performs the following steps: rotates the pixels on 'strip' by 1, pauses for a duration determined by an analog signal from pin P2, shows the updated 'strip', rotates the pixels on 'strip\_1' by 1, pauses again for the same duration, and finally shows the updated 'strip\_1'. The code is written in a block-based programming style with various colored blocks for different actions.

## - PULSANTE B:

In questa sezione ci si è occupati della luminosità delle due matrici led tramite la variabile "lum". Quando la variabile presenta un valore minore o uguale a 255 (valore massimo di luminosità) e pigiando il tasto "B" la luminosità delle strisce crescerà progressivamente di 50 in 50 finché non si giungerà al valore limite. Una volta giunti a 255 pigiando ancora una volta il tasto "B" la luminosità delle strisce tornerà al valore di base 10.



## PROGRAMMA IN PYTHON:

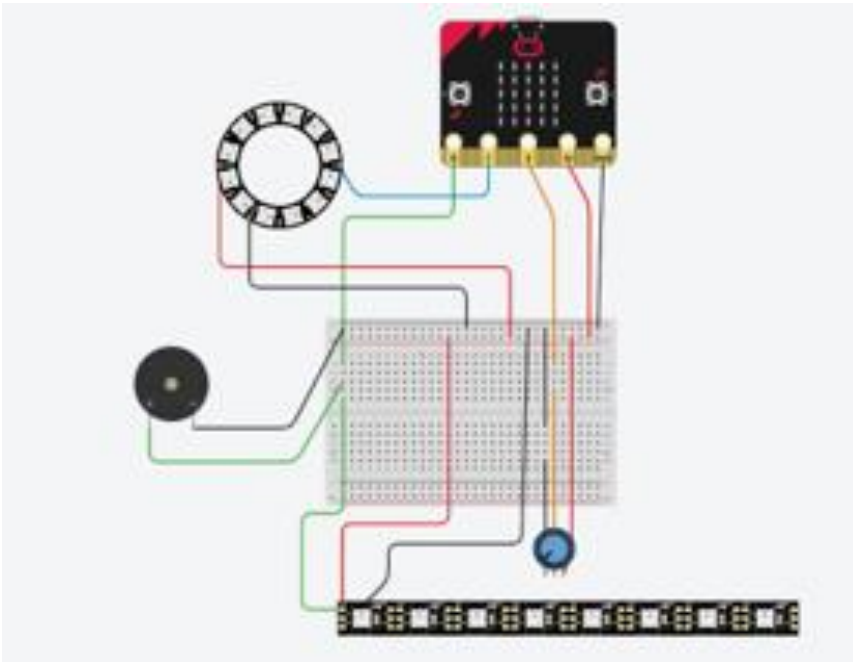
```
def rainbow(num: number):
    strip.show_rainbow(1, 360)
    strip_1.show_rainbow(1, 360)
    for index in range(5):
        strip.rotate(1)
        basic.pause(pins.analog_read_pin(AnalogPin.P2))
        strip.show()
        strip_1.rotate(1)
        basic.pause(pins.analog_read_pin(AnalogPin.P2))
        strip_1.show()

def on_button_pressed_b():
    global lum
    if lum <= 255:
        lum += 50
        strip.set_brightness(lum)
        strip_1.set_brightness(lum)
    else:
        lum = 10
        strip.set_brightness(lum)
        strip_1.set_brightness(lum)
input.on_button_pressed(Button.B, on_button_pressed_b)

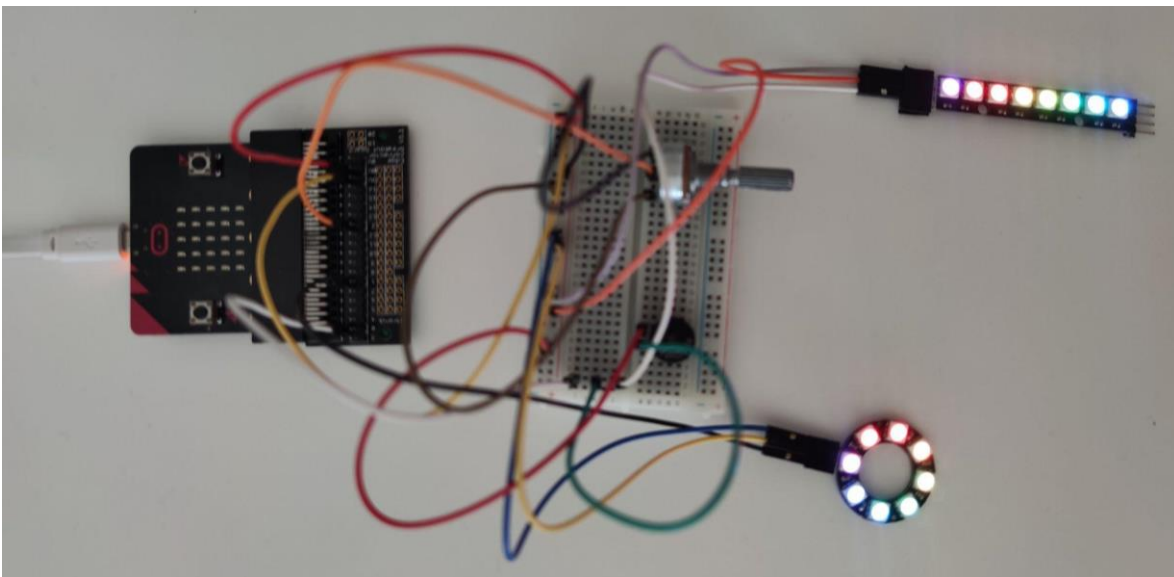
lum = 0
strip_1: neopixel.Strip = None
strip: neopixel.Strip = None
music.set_volume(255)
music.play_melody("C - E G - B C5 - ", 120)
c = 1
strip = neopixel.create(DigitalPin.P0, 8, NeoPixelMode.RGB)
strip_1 = neopixel.create(DigitalPin.P1, 8, NeoPixelMode.RGB)
strip.show_color(neopixel.colors(NeoPixelColors.BLACK))
strip_1.show_color(neopixel.colors(NeoPixelColors.BLACK))
lum = 10

def on_forever():
    global c
    if input.button_is_pressed(Button.A):
        c = 0
    if input.button_is_pressed(Button.AB):
        c = 1
        strip.show_color(neopixel.colors(NeoPixelColors.BLACK))
        strip_1.show_color(neopixel.colors(NeoPixelColors.BLACK))
    if c == 0:
        rainbow(1)
basic.forever(on_forever)
```

## SIMULAZIONE CIRCUITO:



## FOTO DEL PROGETTO:



## CORSO ARDUINO 2022

Alessia Marco, Alessandro Piepoli, Filippo Intini, Alessandro Lippolis, Francesco Gigante, Pierluigi Pallozzi Lavorante