

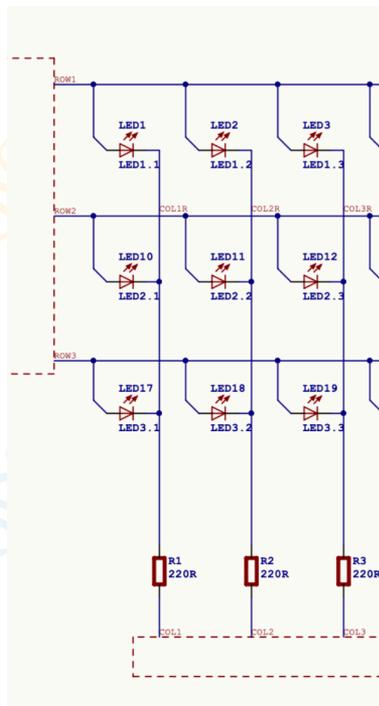
# microbit display

struttura del display, tempi di commutazione

## Il display a LED nella versione 1

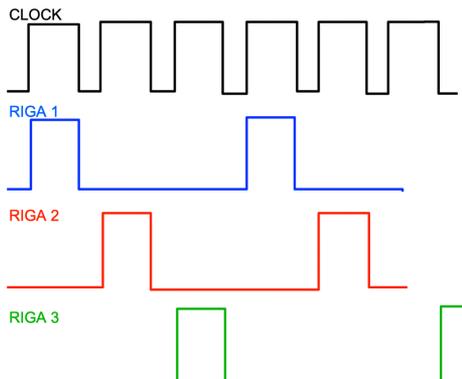
Prima di procedere occorre sapere che i 25 LED disposti in un quadrato 5x5 della **versione 1** di microbit sono comandati da un sistema a multiplex di 3 righe per 9 colonne.

L'accensione di un singolo LED avviene solo quando la sua riga si trova collegata alla tensione +3V e la sua colonna si trova collegata a GND, altrimenti il LED è spento.



Ogni riga invia un impulso positivo che occupa un terzo del periodo a disposizione per il comando dei LED.

Il segnale di clock del display viene ripartito sulle tre righe in modo che ogni riga sia attiva solo un terzo del tempo:



Quando l'impulso della riga 1 è a 3V tutti i diodi con l'anodo collegato alla riga 1 possono accendersi ma si accendono solo quelli che hanno anche il catodo collegato a GND tramite il segnale di colonna.

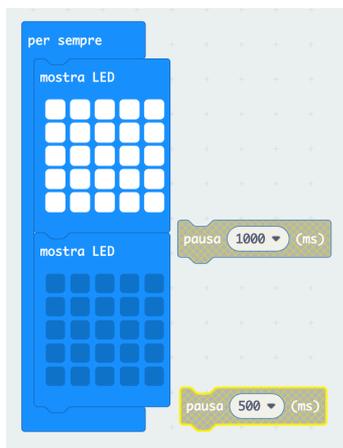
A ciascuna riga sono collegati gli anodi di 9 diodi che hanno il catodo connesso a ciascuna della nove colonne.

A ciascuna colonna sono collegati i catodi di tre diodi.

Vedi lo [schema dei collegamenti](#) dei LED di microbit v1.5.

## Test del display con MS-editor

Con MS\_editor viene programmato un accendi/spegni di tutti i LED del display ripetuto in continuazione (fig. 1)



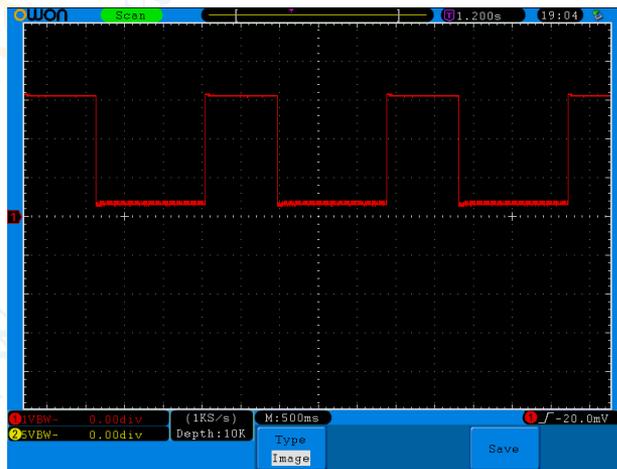
### Con le pause

In un primo esperimento sono stati introdotti i due ritardi con i quali ci aspetta un display interamente acceso per 1 secondo ed un display spento per mezzo secondo, in tutto per una durata di un secondo e mezzo.

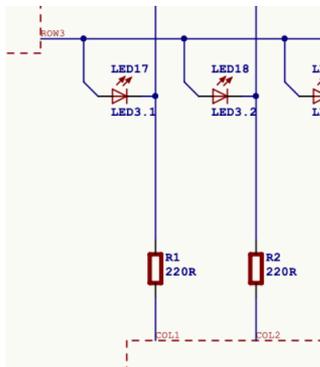
Osservare dieci periodi di oscillazione significa cronometrare 15 secondi.

In pratica, se si usa un cronometro si rileva che ci vogliono 23,3 secondi circa per completare 10 periodi per cui un periodo dura 2,3 s e non 1,5 s come sembrava corretto aspettarsi, ci sono 0,8 s di troppo.

L'oscillogramma, che ha il valore alto quando i LED sono spenti ed il valore basso quando i LED sono accesi, conferma l'osservazione:



La misura è stata fatta con l'oscilloscopio collegato al pin 4 che, secondo lo [schema dei collegamenti](#) dei LED di microbit v1.5, riporta il valore della colonna 2.



Quando la tensione di riga 3 è alta e la tensione di colonna 2 è bassa, il LED 18 si accende.



La fig. 2 riporta alcuni dati del l'oscillogramma da cui risulta che il valore +D alto, quando il LED è spento, dura 0,930 s mentre il valore basso -D dura 1,410 s.

La durata totale del ciclo è di  $0,930 + 1,410 = 2,340$  s che sono 0,840 secondi in più di quelli inseriti con i ritardi.

Il risultato corrisponde alla misurazione fatta col cronometro.

Evidentemente ci sono ritardi nella esecuzione del codice che pure è ridotto al minimo: la ripetizione del ciclo, richiede tempo.

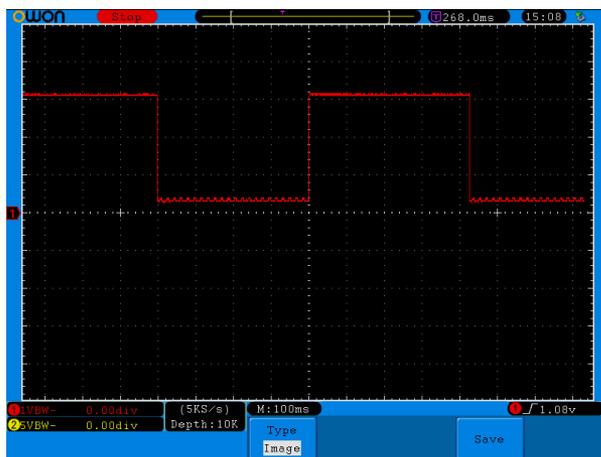
Dal riquadro risulta anche che la tensione alta dell'impulso è 3,120V mentre la tensione della parte bassa non è zero ma 0,320V.

### Senza pause

Per prova si eliminano i due ritardi: il lampeggio è più rapido ma non quanto ci si poteva aspettare.

Si contano 10 cicli e risulta un tempo di 8,20 secondi quindi un ciclo dura 8,2 s anche in assenza di ritardi.

L'oscillogramma che si ottiene è:



che riporta i dati di durata



$$+D = 0,426 \text{ s}$$

$$-D = 0,402 \text{ s}$$

per un totale di 0,828 s che corrisponde alla differenza riscontrata nel caso precedente. Senza inserire ritardi potevamo giustamente aspettarci una esecuzione di ciascun ciclo rapidissima ma non è così.

Questo ritardo prodotto da microbit per realizzare il ciclo sembra eccessivo e bisogna tenerne conto quando si realizzano progetti che devono tenere conto dei tempi di esecuzione.

Per esempio non si può realizzare un cronometro utilizzando i ritardi.

## Test con Python

Si ripetono entrambi i tentativi programmando microbit con [Python editor for microbit](#):

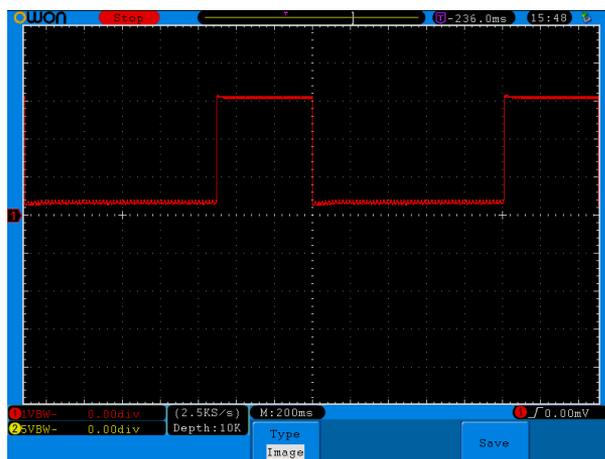
```
from microbit import *

sleep(2000)
while True:
display.show(Image('00000:'
                    '00000:'
                    '00000:'
                    '00000'))
    sleep(500)
display.show(Image('99999:'
                    '99999:'
                    '99999:'
                    '99999'))
    sleep(1000)
```

I risultati sono interessanti.

Con 10 cicli si cronometrano 15,05 secondi che significa una durata del ciclo di 1,5 s.

Gli oscillogrammi sono questi:



Con la durata dei LED spenti  $+D = 0,504$  e dei LED accesi  $-D = 1,008$  s per una durata complessiva del ciclo di  $1,512$  s si trova un valore della durata molto più vicino a quanto ci si poteva aspettare.

Questa versione di Python permette di definire i tempi con maggiore precisione.

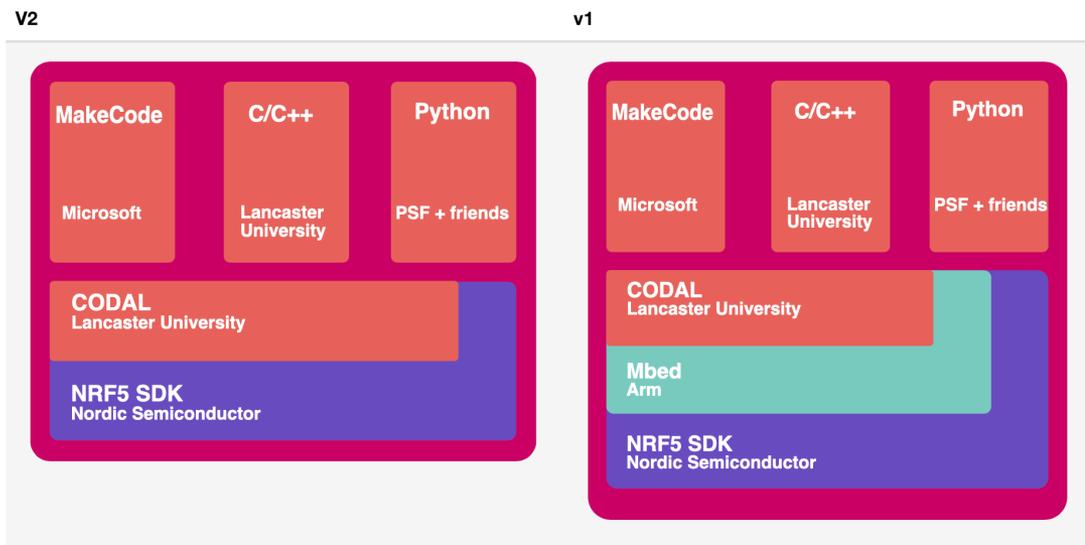
### Perché?

In termini semplificati, la spiegazione di questa differenza sta nel fatto che tra il codice scritto con MS-editor ed il processore della scheda microbit si “infilta” un interprete

messo a punto dalla Lancaster University grazie al quale diventa facile programmare in diversi linguaggi molto accessibili per la loro semplicità d'uso.

Il codice scritto in MS-editor deve essere tutte le volte tradotto in codice macchina.

Programmando con Python si salta l'interprete intermediario **run time** per cui viene prodotto un codice scritto nel linguaggio Mbed che è un linguaggio molto prossimo al codice macchina.



Giovanni Mastropaolo  
www.mastropaolo.n