



ELETTRONICA E MICROPYTHON

LEZIONE 1

GRUPPO :

Aiuto multimetro

Posizione del selettore	Grandezza misurata	Come si collega	
V = (Volt)	Tensione continua	In parallelo al circuito 	
A = (Ampere)	Corrente continua	In serie al circuito 	
Ω (Ohm)	Resistenza	Senza alimentazione	

Esperimento 1

Una pila	Due pile in serie	Due pile in parallelo	Portabatterie acceso	Portabatterie spento
Misura tensione	Misura tensione	Misura tensione	Misura tensione	Misura tensione
V	V	V	V	V



ELETTRONICA E MICROPYTHON

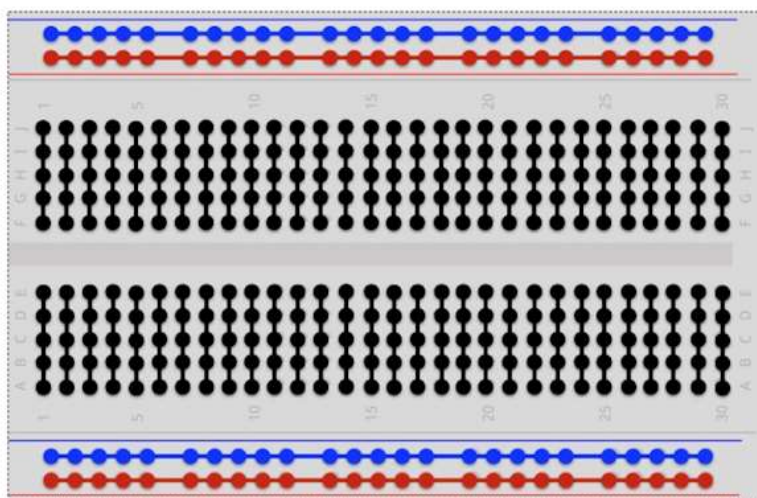
LEZIONE 2

GRUPPO :

Aiuto multimetro

Posizione del selettore	Grandezza misurata	Come si collega	
V = (Volt)	Tensione continua	In parallelo al circuito	
A = (Ampere)	Corrente continua	In serie al circuito	
Ω (Ohm)	Resistenza	Senza alimentazione In parallelo	

Aiuto breadboard



Le rette rappresentano le connessioni elettriche fra i vari fori della breadboard



ELETTRONICA E MICROPYTHON

LEZIONE 2

GRUPPO :

Esperimento 1 – motore

Inserire graffette nella breadboard (1 e 2), poi infilare l'asse della bobina all'interno delle graffette (3), poi i magneti (4) e in fine i fili delle batterie nelle file verticali della breadboard (5).

	<p>5. COLLEGARE BATTERIE - E +</p> <p>3. INFILARE LE ESTREMITA' DELLA BOBINA DENTRO LE GRAFFETTE</p> <p>4. MAGNETI (ALMENO 3)</p> <p>2. GRAFFETTA +</p> <p>1. GRAFFETTA -</p>		
<p>RISULTATO DA OTTENERE</p>			
<p>Segnare con la X il senso di rotazione osservato della bobina guardandola come sulla foto di sopra</p>			

Descrivete un modo per cambiare verso di rotazione e verificatelo

.....

.....

.....

.....

.....



ELETTRONICA E MICROPYTHON

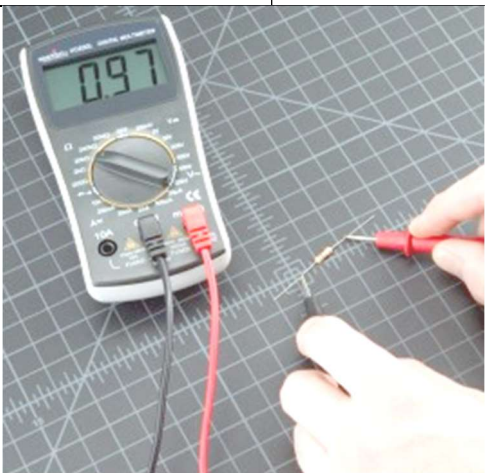
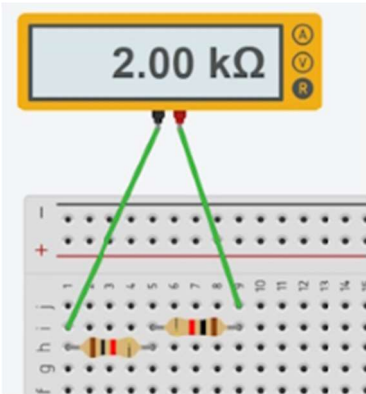
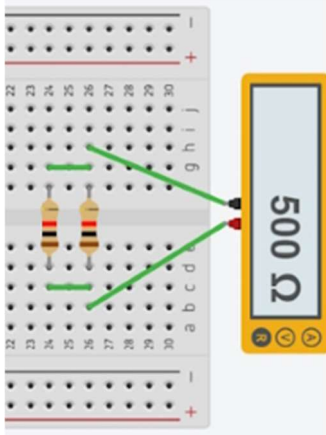
LEZIONE 2

GRUPPO :

Esperimento 2 – misura resistenza

Valore misurato : Ω

Esperimento 3 – più resistenze

Resistenza 1 da sola	Resistenza 2 da sola	Due resistenze uguali in serie	Due resistenze uguali in parallelo
			
Misura R1	Misura R2	Misura R3	Misura R4
..... Ω Ω Ω Ω

Esperimento 4 – Python calcolatore

Lanciare Thonny e eseguire nel REPL le istruzioni seguenti

Istruzione	Risposta di Python
<code>1+1</code>
<code>3*10**15</code>



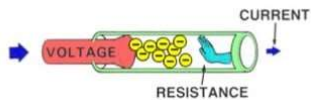
ELETTRONICA E MICROPYTHON

LEZIONE 3

GRUPPO :

Aiuto leggi di ohm e board

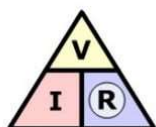
TRIANGOLO



$$V = I \times R$$



$$I = \frac{V}{R}$$



$$R = \frac{V}{I}$$

In python:

```
def ohm_u(r, i):
    return r * i
```

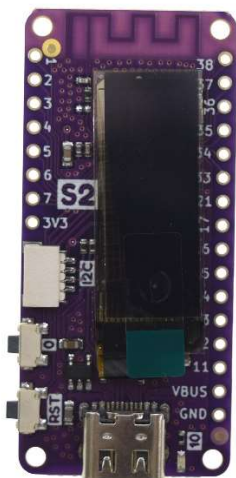
```
print(ohm_u(100, 0.1))
```

```
def ohm_i(u, r):
    return u / r
```

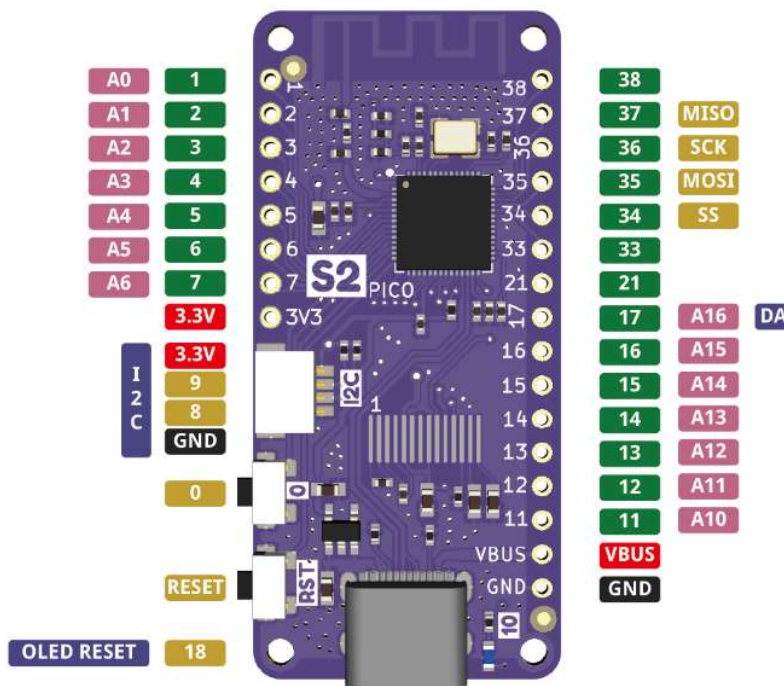
```
print(ohm_i(4, 2000))
```

```
def ohm_r(u, i):
    return u / i
```

```
print(ohm_r(4, 0.001))
```



BOARD PICO S2



SCHEMA CON NUMERO PINS



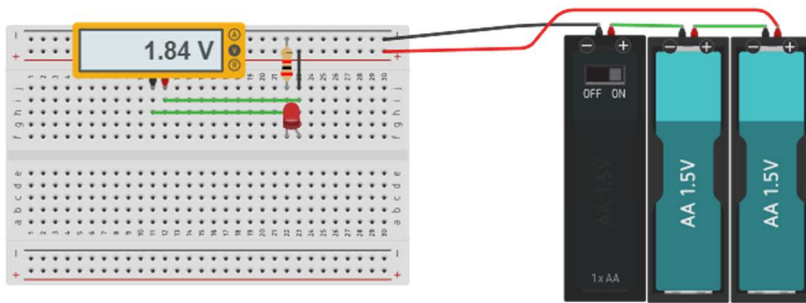
ELETTRONICA E MICROPYTHON

LEZIONE 3

GRUPPO :

Esperimento 1 – LED e misura di tensione

Rappresentazione



Schema elettrico



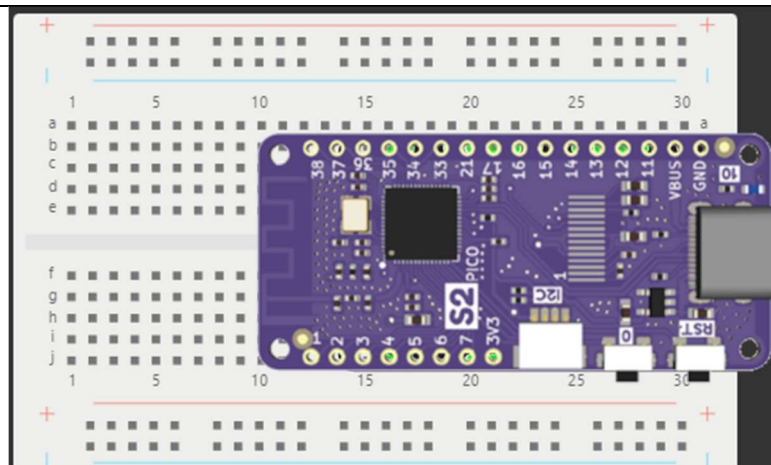
Risultati sperimentali

Colore del diodo ?	Tensione attorno al diodo?	U1 = V
A Tensione attorno alla resistenza?	U2 = V	Tensione delle batterie?	U3 = V
A cosa è uguale U1 + U2 ?	Abbiamo misurato il giorno 1 la tensione batterie	

Calcolo della corrente applicando la legge di Ohm alla resistenza: A

Esperimento 2 – Board e Micropython

Circuito



Istruzioni

Collegare il cavo USB alla board

Collegare Thonny alla board

Eeguire nel REPL di Thonny

```
Pin(10).on()
```

```
Pin(10).off()
```





ELETTRONICA E MICROPYTHON

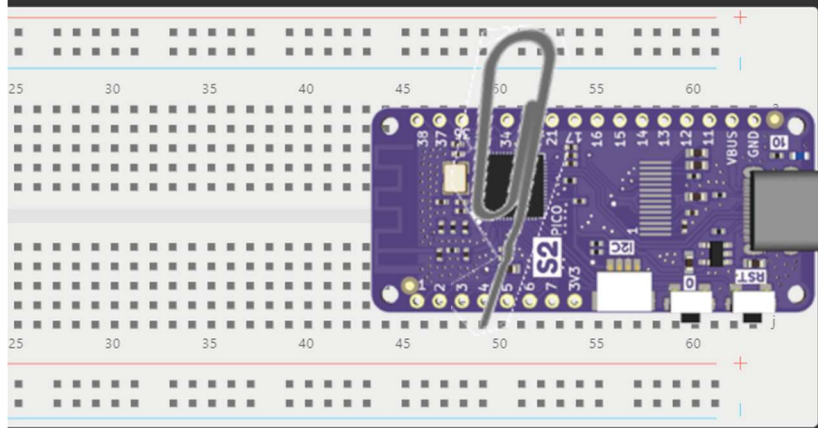
LEZIONE 4

GRUPPO :

Esperimento 1 – Pilotare lo schermo SSD

Circuito	Istruzioni
	<p>Collegare il cavo USB alla board</p> <p>Collegare Thonny alla board</p> <p>Aprire main.py sulla board</p> <pre>display.text('Ciao <nome>', 40, 12, 1) display.show()</pre> <p>Salvare e RESET</p>

Esperimento 2 – TouchPAD

Circuito	Istruzioni di collegamento
	<p>Collegare la graffetta aperta al pin 4 della board usando la fila verticale della breadboard</p> <p>Thonny / main.py</p> <pre>tp = TouchPad(Pin(4)) while True: print(tp.read()) sleep_ms(250)</pre>

Quale valore soglia possiamo usare per riconoscere il tocco ?

Ora accendiamo il LED interno quando la graffetta è toccata

<pre>tp = TouchPad(Pin(4)) SOGLIA = ... while True: if tp.read() > SOGLIA: Pin(10).on() else: Pin(10).off() sleep_ms(250)</pre>	<p>Thonny/main.py</p> <p>Sostituire i tre puntini con il valore scelto sopra</p>
--	---



ELETTRONICA E MICROPYTHON

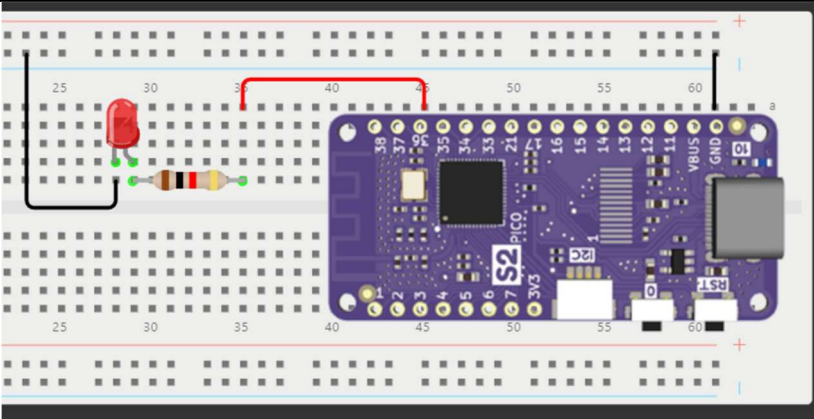
LEZIONE 4

GRUPPO :

Ora accendiamo il LED interno e un messaggio quando la graffetta è toccata

<pre>tp = TouchPad(Pin(4)) SOGLIA = ... while True: if tp.read() > SOGLIA: Pin(10).on() display.text('Ciao!', 40, 12, 1) display.show() else: Pin(10).off() display.text('Arrivederci!', 40, 12, 1) display.show() sleep_ms(250)</pre>	<p>Thonny/main.py</p> <p>Sostituire i tre puntini con il valore scelto sopra</p>
---	--

Esperimento 3 – Lampeggi LED esterno

Circuito	Istruzioni di collegamento
	<p>Collegare un cavo fra il pin 36 e una resistenza di almeno 1kΩ</p> <p>Collegare la resistenza all'anodo del diodo</p> <p>Collegare il catodo del diodo alla fila MENO orizzontale della breadboard</p> <p>Collegare il GND della board alla fila MENO orizzontale della breadboard</p>

Lampeggio sincronizzato LED esterno e LED integrato ogni secondo	Lampeggio alternato LED esterno e LED integrato ogni secondo
<p>Thonny / main.py</p> <pre>led1 = Pin(36, Pin.OUT) led2 = Pin(10, Pin.OUT) while True: led1.on() led2.on() sleep(1) led1.off() led2.off() sleep(1)</pre>	<p>Thonny / main.py</p> <pre>led1 = Pin(36, Pin.OUT) led2 = Pin(10, Pin.OUT) while True: led1.on() led2.off() sleep(1) led1.off() led2.on() sleep(1)</pre>



ELETTRONICA E MICROPYTHON

LEZIONE 4

GRUPPO :

--	--

Esperimento 4 – Programmazione asincrona

Circuito	Istruzioni di collegamento
	Stesso circuito dell'esperimento 3
<p>Lampeggio con periodi diversi (asincrono)</p> <pre>import asyncio led1 = Pin(36, Pin.OUT) led2 = Pin(10, Pin.OUT) async def blink_1(): global led1 while True: led1.value(not(led1.value())) await asyncio.sleep_ms(2000) async def blink_2(): global led2 while True: led2.value(not(led2.value())) await asyncio.sleep_ms(1000) def main(): t1 = asyncio.create_task(blink_1) t2 = asyncio.create_task(blink_2) asyncio.gather(t1, t2)</pre> <p>(Thonny/main.py)</p>	<p>Approccio classico</p> <pre>led1 = Pin(36, Pin.OUT) led2 = Pin(10, Pin.OUT) while True: led1.on() led2.off() sleep(1) led1.on() led2.on() sleep(1) led1.off() led2.off() sleep(1) led1.off() led2.on() sleep(1)</pre> <p>(Thonny/main.py)</p>

Fin'ora sembra un po' la stessa cosa, ma più complessa.



ELETTRONICA E MICROPYTHON

LEZIONE 4

GRUPPO :

Aggiungiamo un contatore di secondi sull'LCD

Approccio asincrono	Approccio classico
<pre>import asyncio led1 = Pin(36, Pin.OUT) led2 = Pin(10, Pin.OUT) contatore = 0 async def blink_1(): global led1 while True: led1.value(not(led1.value())) await asyncio.sleep_ms(2000) async def blink_2(): global led2 while True: led2.value(not(led2.value())) await asyncio.sleep_ms(1000) async def conta(): global contatore while True: contatore += 1 display.text(str(contatore), 40, 12, 1) display.show() await asyncio.sleep_ms(1000) def main(): t1 = asyncio.create_task(blink_1) t2 = asyncio.create_task(blink_2) t3 = asyncio.create_task(conta) asyncio.gather(t1, t2, t3)</pre>	<pre>contatore = 0 led1 = Pin(36, Pin.OUT) led2 = Pin(10, Pin.OUT) while True: led1.on() led2.off() contatore += 1 display.text(str(contatore), 40, 12, 1) display.show() sleep(1) led1.on() led2.on() contatore += 1 display.text(str(contatore), 40, 12, 1) display.show() sleep(1) led1.off() led2.off() contatore += 1 display.text(str(contatore), 40, 12, 1) display.show() sleep(1) led1.off() led2.on() contatore += 1 display.text(str(contatore), 40, 12, 1) display.show() sleep(1)</pre>

Se ora dovessi cambiare il lampeggio del primo led a 0,5 s?

Basterebbe cambiare una riga nel primo caso (asincrono), mentre dovrei riscrivere quasi tutto il secondo programma (sincrono).



ELETTRONICA E MICROPYTHON

LEZIONE 5

GRUPPO :

Aiuto coordinate dato

	<p>Asse X →</p> <p>Asse Y V</p>
	<p>Elisse centrata in posizione (0,0)</p>



ELETTRONICA E MICROPYTHON

LEZIONE 5

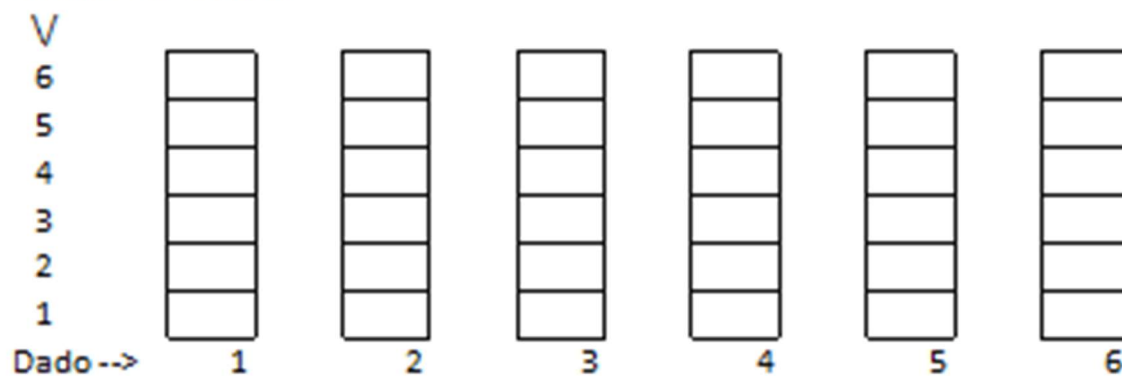
GRUPPO :

Esperimento 1 : generare numeri casuali

Python	Output programma (6 tiri)
<pre>import random print(random.randint(1,6))</pre>	<div>.....</div> <div>.....</div> <div>.....</div> <div>.....</div> <div>.....</div> <div>.....</div>

Rappresentare i vostri risultati come un istogramma colorando il numero di volte in cui è uscito ogni risultato

Numero di risultati





ELETTRONICA E MICROPYTHON

LEZIONE 5

GRUPPO :

Esperimento 2: disegnare sullo schermo

Quadrato	<code>display.fill_rect(0, 0, 32, 32, 1)</code>
Cerchio	<code>display.ellipse(16, 16, 4, 4, 0, True)</code> Centro del cerchio : X = 16, Y = 16 Semiasse orizzontale : 4 pixels Semiasse verticale : 4 pixels Colore dell'elisse : 0 Ultimo parametro True per riempire l'elissi
Disegnare cerchi da una lista	<pre># Facciamo una lista di coordinate (tuple) centri = [(16,16),(8,8),(24,24)] for centro in centri: display.ellipse(centro[0], centro[1], 4, 4, 0, True)</pre>

Numero scelto per il gruppo :

Numero	Coordinate dei centri dei puntini
1	(..... ,)
2	(..... ,) (..... ,)
3	(..... ,) (..... ,) (..... ,)
4	(..... ,) (..... ,) (..... ,) (..... ,)
5	(..... ,) (..... ,) (..... ,) (..... ,) (..... ,)
6	(..... ,) (..... ,) (..... ,) (..... ,) (..... ,) (..... ,)



ELETTRONICA E MICROPYTHON

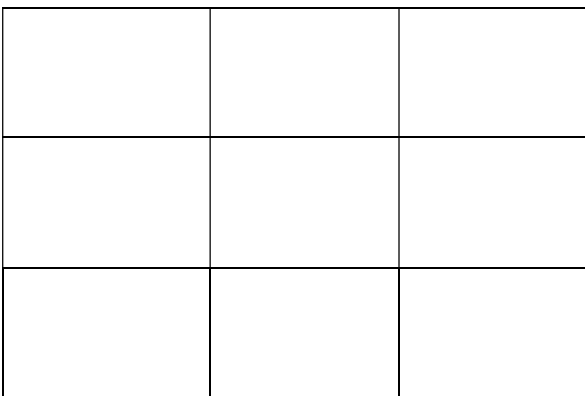
LEZIONE 5

GRUPPO :

Esperimento 3: dato un numero, disegnare il dado corrispondente

```
def disegna_dado(valore_tratto):  
    display.fill(0)  
    display.fill_rect(0, 0, 32, 32, 1)  
  
    dado = { 1: [(16,16)],  
            2: [(8,8), (24,24)],  
            3: [(8,8), (16,16), (24,24)],  
            4: [(8,8), (8,24), (24,8), (24,24)],  
            5: [(8,8), (8,24), (24,8), (24,24), (16,16)],  
            6: [(8,8), (8,24), (24,8), (24,24), (8,16), (24,16)] }  
  
    puntini = dado[valore_tratto]  
    for punto in puntini:  
        display.ellipse(punto[0], punto[1], 4, 4, 0, True)  
  
    display.show()  
  
# Proviamo con un numero  
disegna_dado(6)
```

Disegnare il risultato ottenuto





ELETTRONICA E MICROPYTHON

LEZIONE 5

GRUPPO :

Esperimento 4 : conclusione dado elettronico

	<p>Graffetta su Pin4</p>
<pre>def disegna_dado(display, value, pos = 0): WIDTH = 32 RADIUS = 4 if pos > 0: xmin = (WIDTH + 2) * pos else: xmin = 0 display.fill_rect(xmin, 0, WIDTH, WIDTH, 1) coordinate = { 1: [(16,16)], 2: [(8,8), (24,24)], 3: [(8,8), (16,16), (24,24)], 4: [(8,8), (8,24), (24,8), (24,24)], 5: [(8,8), (8,24), (24,8), (24,24), (16,16)], 6: [(8,8), (8,24), (24,8), (24,24), (8,16), (24,16)] } for center in coordinate[value]: display.ellipse(xmin + center[0], center[1], RADIUS, RADIUS, 0, True) display.show() cpt = 0 led = Pin(10, Pin.OUT) touch = TouchPad(Pin(4)) NUM_DADI = 3 while True: if touch.read() > 10000: display.fill(0) led.value(not led.value()) for num_dado in range(0, NUM_DADI): disegna_dado(display, random.randint(1, 6), num_dado) time.sleep_ms(250) cpt += 1</pre>	