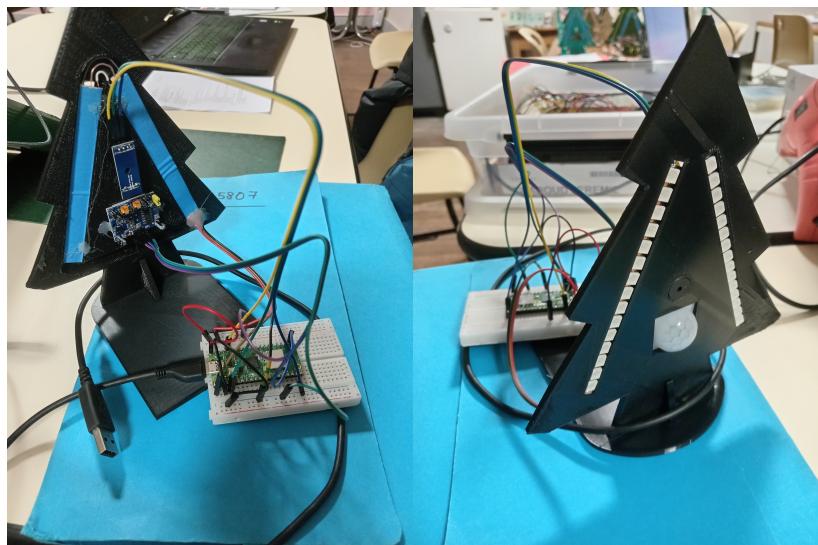


Sapin de Noël



Main Exemple LEDs « Fill »	2
Main Exemple LEDs « Set pixels »	3
Main Exemple LEDs « Liste Chenillard »	4
Main Exemple LEDs « Dictionnaire »	5
Main Exemple LEDs « Soft Timer ».....	6
Main Exemple Detecteur Infra Rouge PIR	7
Main Exemple Buzzer 1 « Stop».....	8
Main Exemple Buzzer 2 « Jingle Bells ».....	9
Main Exemple Buzzer 3 « Tempo »	10
Main Exemple Potentiomètre 1.....	12
Main Exemple Potentiomètre 2.....	13
Main Exemple LCD I2C	14

Main Exemple LEDs « Fill »

```
1  from neopixel import Neopixel
2  import time
3
4  # DECLARATIONS:
5  # -----
6
7  # declaration du ruban de leds (Pin 0)
8  NUMLED =30
9  leds = Neopixel(NUMLED, 0, 0, "GRB")
10
11 # INITIALISATIONS:
12 # -----
13
14 # on definit la luminosite des leds
15 leds.brightness(15)
16 # on eteint toute les leds
17 leds.clear()
18 # on affiche
19 leds.show()
20
21 # BOUCLE PRINCIPALE:
22 # -----
23
24 while True:
25     # on allume toutes les leds du sapin en rouge
26     leds.fill((255,0,0))
27     leds.show()
28     time.sleep(1)
29     # on allume toutes les leds du sapin en vert
30     leds.fill((0,255,0))
31     leds.show()
32     time.sleep(1)
33     # on allume toutes les leds du sapin en bleu
34     leds.fill((0,0,255))
35     leds.show()
36     time.sleep(1)
37
38
```

Main Exemple LEDs «Set pixels »

```
1  from neopixel import Neopixel
2  import time
3
4  # DECLARATIONS:
5  # -----
6
7  # declaration du ruban de leds (Pin 0)
8  NUMLED = 30
9  leds = Neopixel(NUMLED, 0, 0, "GRB")
10
11 # declaration des couleurs (Rouge,Vert,Bleu)
12 couleur_rouge = (255,0,0)
13 couleur_vert = (0,255,0)
14 couleur_bleu = (0,0,255)
15
16 # INITIALISATIONS:
17 # -----
18
19 # on definit la luminosite des leds
20 leds.brightness(15)
21 # on eteint toute les leds
22 leds.clear()
23 # on affiche
24 leds.show()
25
26 # BOUCLE PRINCIPALE:
27 # -----
28
29 while True:
30     # on allume tour a tour les leds du sapin en rouge
31     for i in range (0,NUMLED):
32         leds.set_pixel(i,couleur_rouge)
33         leds.show()
34         time.sleep(0.1)
35     time.sleep(1)
36     # on allume tour a tour les leds du sapin en vert
37     for i in range (0,NUMLED):
38         leds.set_pixel(i,couleur_vert)
39         leds.show()
40         time.sleep(0.1)
41     time.sleep(1)
42     # on allume tour a tour les leds du sapin en bleu
43     for i in range (0,NUMLED):
44         leds.set_pixel(i,couleur_bleu)
45         leds.show()
46         time.sleep(0.1)
```

Main Exemple LEDs « Liste Chenillard »

```
1  from neopixel import Neopixel
2  import time
3
4  # DECLARATIONS:
5  # -----
6
7  # declaration du ruban de leds (Pin 0)
8  NUMLED = 30
9  leds = Neopixel(NUMLED, 0, 0, "GRB")
10
11 # declaration des couleurs (Rouge,Vert,Bleu)
12 couleur_rouge = (255,0,0)
13 couleur_vert = (0,255,0)
14 couleur_bleu = (0,0,255)
15
16 # declaration du chenillard (num led, couleur)
17 chenillard = [(0,couleur_rouge),(1,couleur_vert),(2,couleur_bleu),(3,couleur_rouge),(4,couleur_vert),(5,couleur_bleu),
18             (6,couleur_rouge),(7,couleur_vert),(8,couleur_bleu),(9,couleur_rouge),(10,couleur_vert),(11,couleur_bleu),
19             (12,couleur_rouge),(13,couleur_vert),(14,couleur_bleu),(15,couleur_rouge),(16,couleur_vert),(17,couleur_bleu),
20             (18,couleur_rouge),(19,couleur_vert),(20,couleur_bleu),(21,couleur_rouge),(22,couleur_vert),(23,couleur_bleu),
21             (24,couleur_rouge),(25,couleur_vert),(26,couleur_bleu),(27,couleur_rouge),(28,couleur_vert),(29,couleur_bleu)]
22
23 # INITIALISATIONS:
24 # -----
25
26 # on definit la luminosite des leds
27 leds.brightness(15)
28 # on eteint toute les leds
29 leds.clear()
30 # on affiche
31 leds.show()
32
33 # BOUCLE PRINCIPALE:
34 # -----
35
36 while True:
37     # on allume tour a tour les leds du sapin selon le chenillard
38     for element in chenillard:    # pour tous les elements de la liste chenillard
39         numero_led = element[0]  # on recupere le numero de la led = premiere valeur
40         couleur_led = element[1] # on recupere la couleur de cette led = seconde valeur
41         # on programme la led
42         leds.set_pixel(numero_led,couleur_led)
43         leds.show()
44         time.sleep(0.1)
45
```

Main Exemple LEDs « Dictionnaire »

```
1  from neopixel import Neopixel
2  import time
3
4  # DECLARATIONS:
5  # -----
6
7  # declaration du ruban de leds (Pin 0)
8  NUMLED = 30
9  leds = Neopixel(NUMLED, 0, 0, "GRB")
10
11 # on place les couleurs dans un dictionnaire
12 colors = {"noir":(0,0,0),"blanc":(255,255,255),"vert":(0,255,0),"rouge":(255,0,0),"bleu":(0,0,255)}
13
14 # declaration du chenillard (num led, couleur)
15 chenillard = [(0,colors["rouge"]),(1,colors["vert"]),(2,colors["bleu"]),(3,colors["rouge"]),(4,colors["vert"]),
16 |   (5,colors["bleu"]),(6,colors["rouge"]),(7,colors["vert"]),(8,colors["bleu"]),(9,colors["rouge"]),
17 |   (10,colors["vert"]),(11,colors["bleu"]),(12,colors["rouge"]),(13,colors["vert"]),(14,colors["bleu"]),
18 |   (15,colors["rouge"]),(16,colors["vert"]),(17,colors["bleu"]),(18,colors["rouge"]),
19 |   (19,colors["vert"]),(20,colors["bleu"]),(21,colors["rouge"]),(22,colors["vert"]),
20 |   (23,colors["bleu"]),(24,colors["rouge"]),(25,colors["vert"]),(26,colors["bleu"]),
21 |   (27,colors["rouge"]),(28,colors["vert"]),(29,colors["bleu"])]
22
23 # INITIALISATIONS:
24 # -----
25
26 # on definit la luminosite des leds
27 leds.brightness(15)
28 # on eteint toute les leds
29 leds.clear()
30 # on affiche
31 leds.show()
32
33 # BOUCLE PRINCIPALE:
34 # -----
35 while True:
36     # on allume tour a tour les leds du sapin selon le chenillard
37     for element in chenillard:    # pour tous les elements de la liste chenillard
38         numero_led = element[0]  # on recuper le numero de la led = premiere valeur
39         couleur_led = element[1] # on recuper la couleur de cette led = seconde valeur
40         # on programme la led
41         leds.set_pixel(numero_led,couleur_led)
42         leds.show()
43         time.sleep(0.1)
44
45
```

Main Exemple LEDs « Soft Timer »

```
1  ##  
2  ## Sample Code to Animate an LED Strip With a Timer  
3  ##  
4  
5  from machine import Pin, Timer  
6  from utime import sleep  
7  from neopixel import Neopixel  
8  
9  NEOPIXEL_PIN = 0  
10 NUMBER_PIXELS = 24  
11 leds = Neopixel(NUMBER_PIXELS,NEOPIXEL_PIN, 0, "GRB")  
12  
13 # on definit la luminosite des leds  
14 leds.brightness(15)  
15 # on eteint toute les leds  
16 leds.clear()  
17 # on affiche  
18 leds.show()  
19  
20 # create an uninitialized timer object  
21 myTimer = Timer()  
22  
23 counter = 0  
24 # a callback function. Use this with a timer that triggers 20 times a second  
25 def move_pixel(myTimer):  
26     global counter  
27     for i in range(0, NUMBER_PIXELS):  
28         if i == counter:  
29             leds.set_pixel(i,(10,0,0))  
30         else:  
31             leds.set_pixel(i,(0,0,10))  
32         leds.show()  
33         counter += 1  
34     if counter > NUMBER_PIXELS:  
35         counter = 0  
36  
37 # initialize the timer object to tick 20 times per second (50 milliseconds)  
38 myTimer.init(period=50, mode=Timer.PERIODIC, callback=move_pixel)  
39  
40 while True:  
41     print('Just sleeping here. The timer is doing all the work flashing the LED...', counter)  
42     sleep(5) # sleep for five seconds
```

Main Exemple Detecteur Infra Rouge PIR

```
1  from machine import Pin
2  from neopixel import Neopixel
3  import time
4
5  # DECLARATIONS:
6  # -----
7
8  # declaration du detecteur infra rouge
9  pir_sensor = Pin(16, Pin.IN)
10 # declaration du ruban de leds (Pin 0)
11 NUMLED = 30
12 leds = Neopixel(NUMLED, 0, 0, "GRB")
13
14 # INITIALISATIONS:
15 # -----
16
17 # on definit la luminosite des leds
18 leds.brightness(15)
19 # on eteint toute les leds
20 leds.clear()
21 # on affiche
22 leds.show()
23
24 # BOUCLE PRINCIPALE:
25 # -----
26
27 while True:
28     time.sleep(1)
29     reading = pir_sensor.value()
30     print(reading)
31     #si quelquechose a ete detecte
32     if reading == 1:
33         # on allume toutes les leds du sapin en rouge
34         leds.fill((255,0,0))
35         leds.show()
36         time.sleep(1)
37     #si rien n'a ete detecte
38     else:
39         # on eteint les leds du sapin
40         leds.clear() # equivalent a: leds.fill((0,0,0))
41         leds.show()
42         time.sleep(2)
```

Main Exemple Buzzer 1 « Stop»

```
1  from machine import Pin
2  from buzzer import Buzzer
3  import time
4
5
6  # declaration du buzzer (Pin2, alimentation en 3.3V)
7  buz = Buzzer(2)
8  #buz.stop()
9
10 # fonction qui va arreter le buzzer
11 def buzzer_stop(pin):
12     buz_stop = Pin(pin, Pin.OUT)
13     buz_stop.value(1)
14
15     time.sleep(1)
16     buzzer_stop(2)
17
18 while True:
19     pass
20
21
```

Main Exemple Buzzer 2 « Jingle Bells »

```
1  from machine import Pin
2  from buzzer import Buzzer
3  from time import sleep
4
5  # frequences des notes
6  freq_notes = {"do":1046,"do_":1109,
7  "re":1175,"re_":1245,
8  "mi":1318,
9  "fa":1397,"fa_":1480,
10 "so":1568,"so_":1661,
11 "la":1760,"la_":1864,
12 "si":1967}
13
14
15  # declaration du buzzer (Pin2, alimentation en 3.3V)
16  buz = Buzzer(2)
17  buz.stop()
18
19  # fonction qui va jouer une note
20  def joue_note(val_note,time):
21      global buz
22      buz.set_freq(freq_notes[val_note])
23      buz.start()
24      sleep(time)
25      buz.stop()
26
27  #Jingle Bells
28  notes = [("mi",0.25),("mi",0.25),("mi",0.5),
29  ("mi",0.25),("mi",0.25),("mi",0.5),
30  ("mi",0.25),("so",0.25),("do",0.25),("re",0.25),
31  ("mi",1),
32  ("fa",0.25),("fa",0.25),("fa",0.5),
33  ("mi",0.25),("mi",0.25),("mi",0.5),
34  ("re",0.25),("re",0.25),("re",0.25),("mi",0.25),
35  ("re",0.5),("so",0.5),
36  ("mi",0.25),("mi",0.25),("mi",0.5),
37  ("mi",0.25),("mi",0.25),("mi",0.5),
38  ("mi",0.25),("so",0.25),("do",0.25),("re",0.25),
39  ("mi",1),
40  ("fa",0.25),("fa",0.25),("fa",0.5),
41  ("mi",0.25),("mi",0.25),("mi",0.5),
42  ("so",0.25),("fa",0.25),("mi",0.25),("re",0.25),
43  ("do",1)]
44
45  while True:
46      for note in notes:
47          joue_note(note[0],note[1])
48          sleep(0.01)
49
```

Main Exemple Buzzer 3 « Tempo »

```
1  from machine import Pin
2  from buzzer import Buzzer
3  from time import sleep
4
5  tempo = 200
6
7  # temps des note
8  D = 0.25 # double croche
9  C = 0.5 # croche
10 N = 1 # noir
11 N_C = 1.5
12 B = 2 # blanche
13 R = 4 # ronde
14
15
16
17 # frequence note
18 freq_notes = {"do_2":130.81,"do#_2":138.59,
19         "re_2":146.83,"re#_2":155.56,
20         "mi_2":164.81,
21         "fa_2":174.61,"fa#_2":185.00,
22         "sol_2":196,"sol#_2":207.65,
23         "la_2":220,"la#_2":233.08,
24         "si_2":246.94,
25         "do_3":261.63,"do#_3":277.18,
26         "re_3":293.66,"re#_3":311.13,
27         "mi_3":329.63,
28         "fa_3":349.23,"fa#_3":369.99,
29         "sol_3":392,"sol#_3":415.30,
30         "la_3":440,"la#_3":466.16,
31         "si_3":493.88,
32         "do_4":523.25,"do#_4":554.37,
33         "re_4":587.33,"re#_4":622.25,
34         "mi_4":659.26,
35         "fa_4":698.46,"fa#_4":739.99,
36         "sol_4":783.99,"sol#_4":830.61,
37         "la_4":880,"la#_4":932.33,
38         "si_4":987.77,
39         "do_5":1046.50,"do#_5":1108.73,
40         "re_5":1174.66,"re#_5":1244.51,
41         "mi_5":1318.51,
42         "fa_5":1396.91,"fa#_5":1479.98,
43         "sol_5":1567.98,"sol#_5":1661.22,
44         "la_5":1760.00,"la#_5":1864.66,
45         "si_5":1967.53}
46
47 #
48 buz = Buzzer(2)
49 buz.stop()
50
51 #
52 def joue_note(val_note,time):
53     global buz
54     buz.set_freq(freq_notes[val_note])
55     buz.start()
56     sleep((60/tempo)*time)
57     buz.stop()
58
```

```

59      #
60      jingle = [ ("mi_3",N), ("mi_3",N), ("mi_3",B),
61          ("mi_3",N), ("mi_3",N), ("mi_3",B),
62          ("mi_3",N), ("sol_3",N), ("do_3",N), ("re_3",N),
63          ("mi_3",R),
64          ("fa_3",N), ("fa_3",N), ("fa_3",B),
65          ("mi_3",N), ("mi_3",N), ("mi_3",B),
66          ("re_3",N), ("re_3",N), ("re_3",N), ("mi_3",N),
67          ("re_3",B), ("sol_3",B),
68          ("mi_3",N), ("mi_3",N), ("mi_3",B),
69          ("mi_3",N), ("mi_3",N), ("mi_3",B),
70          ("mi_3",N), ("sol_3",N), ("do_3",N), ("re_3",N),
71          ("mi_3",R),
72          ("fa_3",N), ("fa_3",N), ("fa_3",B),
73          ("mi_3",N), ("mi_3",N), ("mi_3",B),
74          ("sol_3",N), ("fa_3",N), ("mi_3",N), ("re_3",N),
75          ("do_3",R)]
76
77      tetris = [("re_3",N), ("si_2",C), ("do_3",C), ("re_3",N), ("do_3",C), ("si_2",C),
78          ("la_2",N,C), ("do_3",C), ("mi_3",N), ("re_3",C), ("do_3",C),
79          ("si_2",N), ("si_2",C), ("do_3",C), ("re_3",N), ("mi_3",N),
80          ("do_3",N), ("la_2",N), ("la_2",B), # ligne 1
81          ("re_3",N), ("re_3",C), ("fa_3",C), ("la_3",N), ("sol_3",C), ("fa_3",C),
82          ("mi_3",N,C), ("do_3",C), ("mi_3",N), ("re_3",C), ("do_3",C),
83          ("si_2",N), ("si_2",C), ("do_3",C), ("re_3",N), ("mi_3",N),
84          ("do_3",N), ("la_2",N), ("la_2",B), # ligne 2
85          ("re_3",N), ("re_3",C), ("fa_3",C), ("la_3",N), ("sol_3",C), ("fa_3",C),
86          ("mi_3",N,C), ("do_3",C), ("mi_3",N), ("re_3",C), ("do_3",C),
87          ("si_2",N), ("si_2",C), ("do_3",C), ("re_3",N), ("mi_3",N),
88          ("do_3",N), ("la_2",N), ("la_2",B), # ligne 3
89          ("mi_3",N), ("do_3",N), ("re_3",N), ("si_2",N),
90          ("do_3",N), ("la_2",N), ("sol_2",N), ("si_2",N),
91          ("mi_3",N), ("do_3",N), ("re_3",N), ("si_2",N),
92          ("do_3",C), ("mi_3",C), ("la_3",C), ("sol_3",C), ("mi_3",B), # ligne 4
93          ("mi_3",N), ("do_3",N), ("re_3",N), ("si_2",N),
94          ("do_3",N), ("la_2",N), ("sol_2",N), ("si_2",N),
95          ("mi_3",N), ("do_3",N), ("re_3",N), ("si_2",N),
96          ("la_3",R)]
97
98      #notes = jingle
99      notes = tetris
100
101     for note in notes:
102         joue_note(note[0],note[1])
103         sleep(0.01)
104

```

Main Exemple Potentiomètre 1

```
1  from machine import Pin, ADC
2  from neopixel import Neopixel
3  import time
4
5  # configuration
6  NUMLED = 24#30
7
8  # declaration du potentiometre (ADC 0)
9  potar = ADC(0)
10 # declaration du ruban de leds (Pin 0)
11 leds = Neopixel(NUMLED,0,0,"GRB")
12
13 # fonction qui va lire la valeur du potentiometre
14 def lit_potar():
15     global potar
16     return potar.read_u16()
17
18 # fonction qui va definir la led a allumer en fonction de la valeur du potentiometre
19 def get_led_for( potar_value ):
20     return (((NUMLED-1)*potar_value)/65535)
21
22
23 # definit la luminosite des leds
24 leds.brightness(15)
25 # on eteint toute les leds
26 leds.clear()
27 # on affiche
28 leds.show()
29
30 # initialisation des variables
31 old_val_potar = lit_potar()
32
33 # boucle principale
34 while True:
35     # lit la valeur du potentiometre
36     valeur_potar = lit_potar()
37     # test si la valeur lue est differente de la precedente
38     if valeur_potar != old_val_potar:
39         # affiche la valeur du potentiometre
40         print(valeur_potar)
41         # sauvegarde la nouvelle valeur du potentiometre
42         old_val_potar = valeur_potar
43         # converti la valeur du potentiometre en numero de led
44         numero_led = int(get_led_for( valeur_potar ))
45         # affiche le numero de led
46         print(f"led a allumer : {numero_led}")
47         leds.clear()
48         leds.set_pixel(numero_led,(128,0,120))
49         leds.show()
50
51     time.sleep(0.3)
52
53
```

Main Exemple Potentiomètre 2

```
1  from machine import Pin, ADC
2  from neopixel import Neopixel
3  import time
4
5  # configuration
6  NUMLED = 24#30
7
8  # declaration du potentiometre (ADC 0)
9  potar = ADC(0)
10 # declaration du ruban de leds (Pin 0)
11 leds = Neopixel(NUMLED,0,0,"GRB")
12
13 # fonction qui va lire la valeur du potentiometre
14 def lit_potar():
15     global potar
16     return potar.read_u16()
17
18 # fonction qui va definir la led a allumer en fonction de la valeur du potentiometre
19 def get_led_for( potar_value ):
20     return (((NUMLED-1)*potar_value)/65535)
21
22
23 # definit la luminosite des leds
24 leds.brightness(15)
25 # on eteint toute les leds
26 leds.clear()
27 # on affiche
28 leds.show()
29
30 # initialisation des variables
31 old_val_potar = lit_potar()
32
33 # boucle principale
34 while True:
35     # lit la valeur du potentiometre
36     valeur_potar = lit_potar()
37     # test si la valeur lue est différente de la précédente
38     if valeur_potar != old_val_potar:
39         # affiche la valeur du potentiometre
40         print(valeur_potar)
41         # sauvegarde la nouvelle valeur du potentiometre
42         old_val_potar = valeur_potar
43         # converti la valeur du potentiometre en numero de led
44         numero_led = int(get_led_for( valeur_potar ))
45         # affiche toute les leds jusqu'au numero de led
46         print(f"led a allumer : {numero_led}")
47         leds.clear()
48         for i in range (0,numero_led):
49             leds.set_pixel(i,(128,0,120))
50         leds.show()
51
52     time.sleep(0.3)
53
54
```

Main Exemple LCD I2C

```
1  from machine import Pin, I2C, ADC
2  from time import sleep, sleep_ms
3  from machine_i2c_lcd import I2cLcd
4
5  #Board Setting
6  i2c = I2C(0, scl=Pin(5), sda=Pin(4), freq=400000)
7
8  addr = i2c.scan()[0]
9  # print(hex(addr[0]))
10
11 lcd = I2cLcd(i2c, addr, 2, 16)
12
13 sensor_temp = ADC(4)
14 conversion_factor = 3.3 / (65535)
15
16
17 while True:
18     reading = sensor_temp.read_u16() * conversion_factor
19     temperature = 27 - (reading - 0.706)/0.001721
20     lcd.putstr(" Joyeux Noel! \n")
21     lcd.putstr(" MJC Raspi 2024")
22     #lcd.putstr(" CPU Temp:%.2f" % temperature)
23     sleep(1)
24     lcd.clear()
```