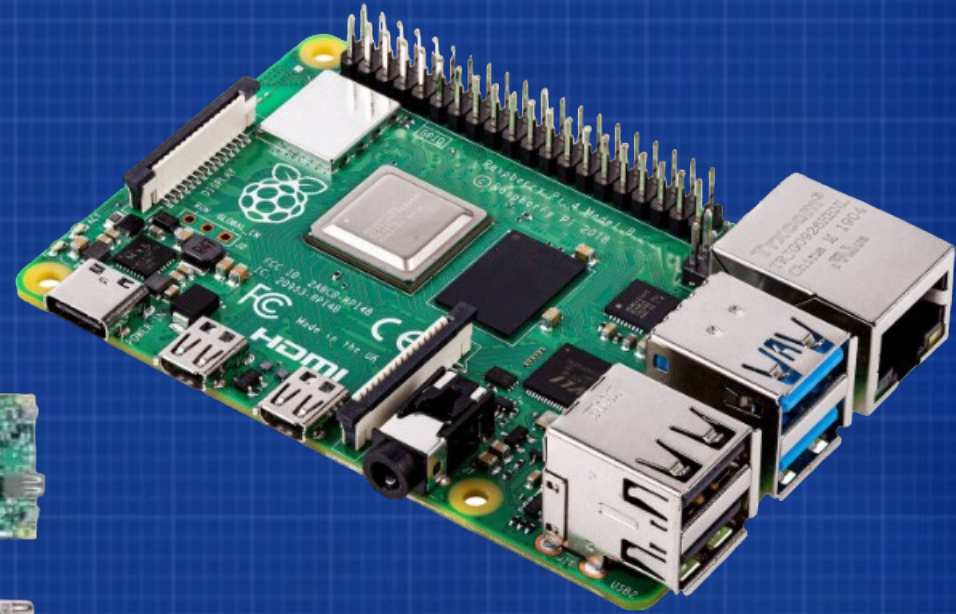


LE RASPBERRY Pi



Pi Zero et ZeroW



Pi 3B,3B+,4,5



Pi Pico

Le Raspberry Pi

Affectation des broches du 40 pins

26 pins GPIO
Programmables
en entrée ou en sortie
Numerique uniquement

Tension d'entrée et de sortie :
3,3 V

courant max par GPIO : **16mA**

Total courants GPIOs : **50mA**

Pull up/down programmable



Peripherals	GPIO	Particle	Pin #		Pin #	Particle	GPIO	Peripherals	
3.3V			1	X	X	2	5V		
I2C	GPIO2	SDA	3	X	X	4	5V		
	GPIO3	SCL	5	X	X	6	GND		
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND		
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND		
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI
GND			25	X	X	26	CE1	GPIO7	(chip enable)
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND		
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND		
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O
GND			39	X	X	40	D16/A7	GPIO21	Digital I/O

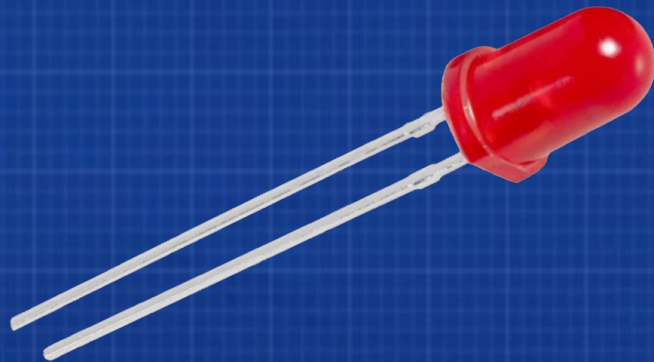
Le Raspberry Pi

Ce que l'on va faire dans les sessions qui viennent Électronique - programmation

- #1 Allumer/éteindre une LED en python
- #2 Allumer/éteindre une LED commander par un bouton poussoir
- #3 Piloter un composant de puissance avec les GPIO
- #4 Gérer différentes entrées et protéger notre Raspi
- #5 Programmer du son
- #6 connecter/gérer un détecteur de présence
- #7 Gérer un moteur actionneur linéaire
- #8 Gérer un moteur pas à pas
- #9 Gérer des effets lumineux
- #10 Créer une interface graphique
- #11 Préparer les automatismes pour le festival

Le Raspberry Pi

1 Allumer/éteindre une LED en python



Le Raspberry Pi

CALCUL DE LA RESISTANCE EN SERIE AVEC LA DIODE

Diode rouge standard tension de seuil 1,6 à 2V
Courant max 20 mA
on part sur 1,8V et 10mA

$$U = R \times I \longrightarrow R = U / I$$

$$U_{GPIO} = U_{diode} + U_{résistance} \longrightarrow U_{résistance} = U_{GPIO} - U_{diode}$$
$$1,5v = 3.3v - 1.7v$$

I est identique pour la diode et la résistance 10 mA

$$R = U_{résistance} / I$$

$$R = 1.5v / 0.01mA$$

$$R = 150 \text{ ohm}$$

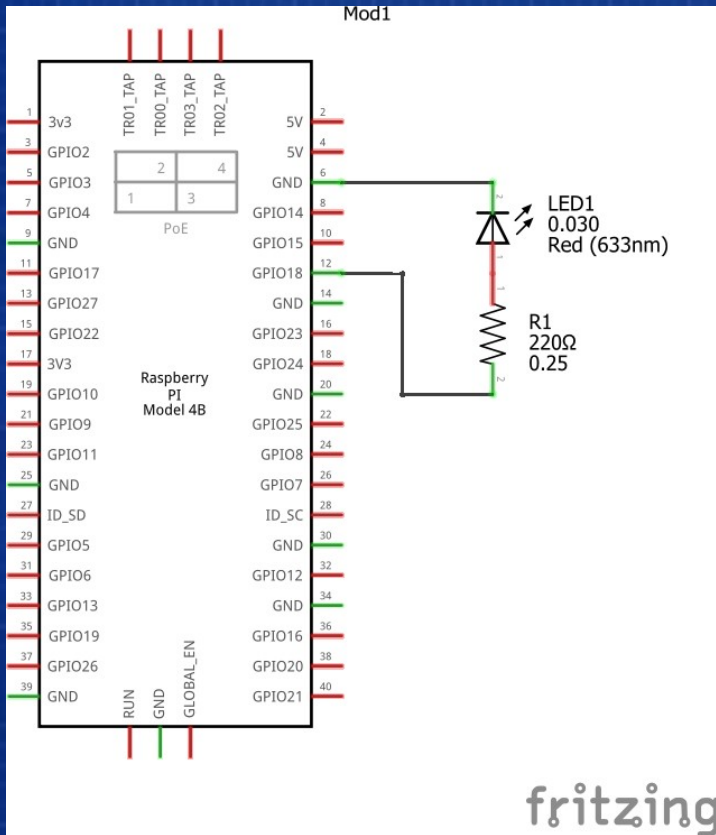
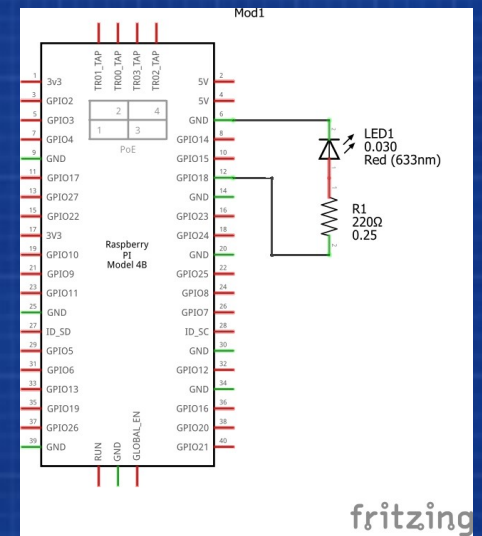
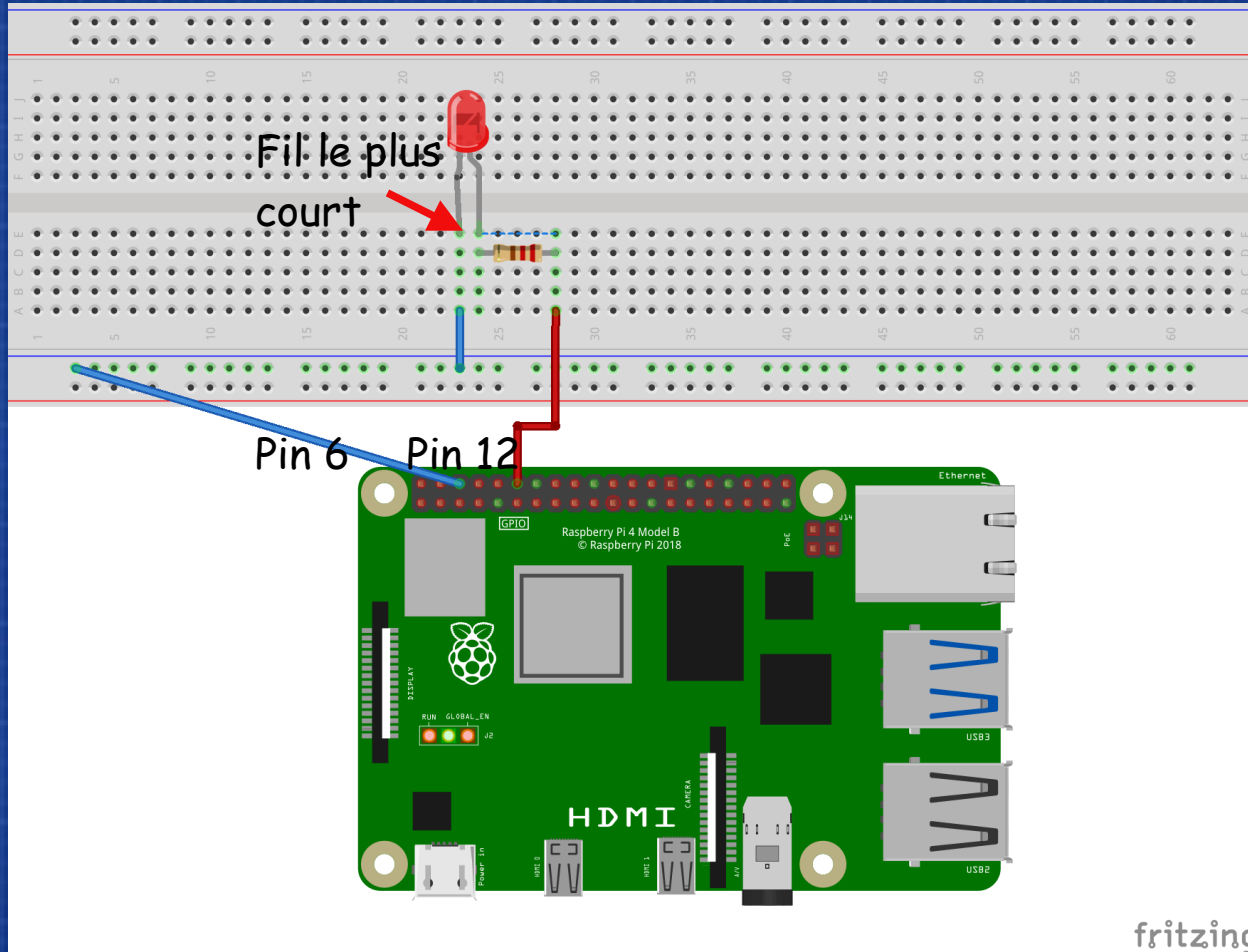
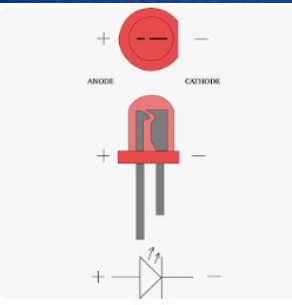
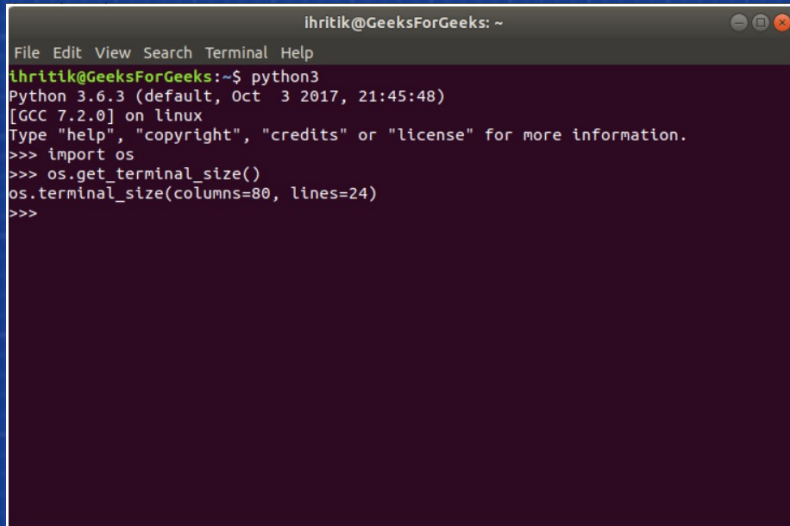


Schéma diode Rpi

LED seule



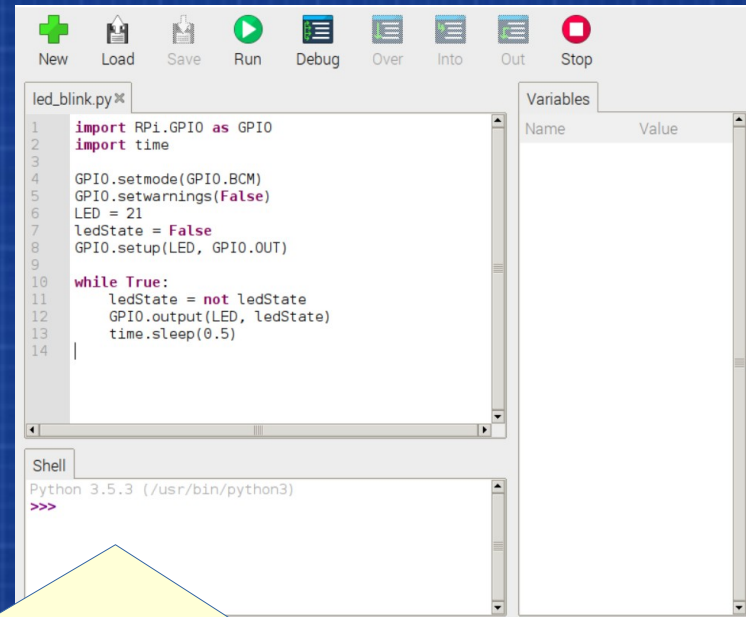
TERMINAL (Shell)



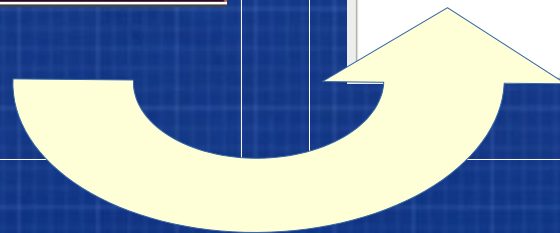
```
ihritik@GeeksForGeeks: ~  
File Edit View Search Terminal Help  
ihritik@GeeksForGeeks:~$ python3  
Python 3.6.3 (default, Oct 3 2017, 21:45:48)  
[GCC 7.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import os  
>>> os.get_terminal_size()  
os.terminal_size(columns=80, lines=24)  
>>>
```

IDE

(Integrated Development Environment)



```
led_blink.py%  
1 import RPi.GPIO as GPIO  
2 import time  
3  
4 GPIO.setmode(GPIO.BCM)  
5 GPIO.setwarnings(False)  
6 LED = 21  
7 ledState = False  
8 GPIO.setup(LED, GPIO.OUT)  
9  
10 while True:  
11     ledState = not ledState  
12     GPIO.output(LED, ledState)  
13     time.sleep(0.5)  
14 |  
  
Shell  
Python 3.5.3 (/usr/bin/python3)  
>>>
```



Structures

(lignes de commandes,
fonctions)

Actions :

Print, input, lst.....

Conditions :

While, if, for, etc.....

Variables

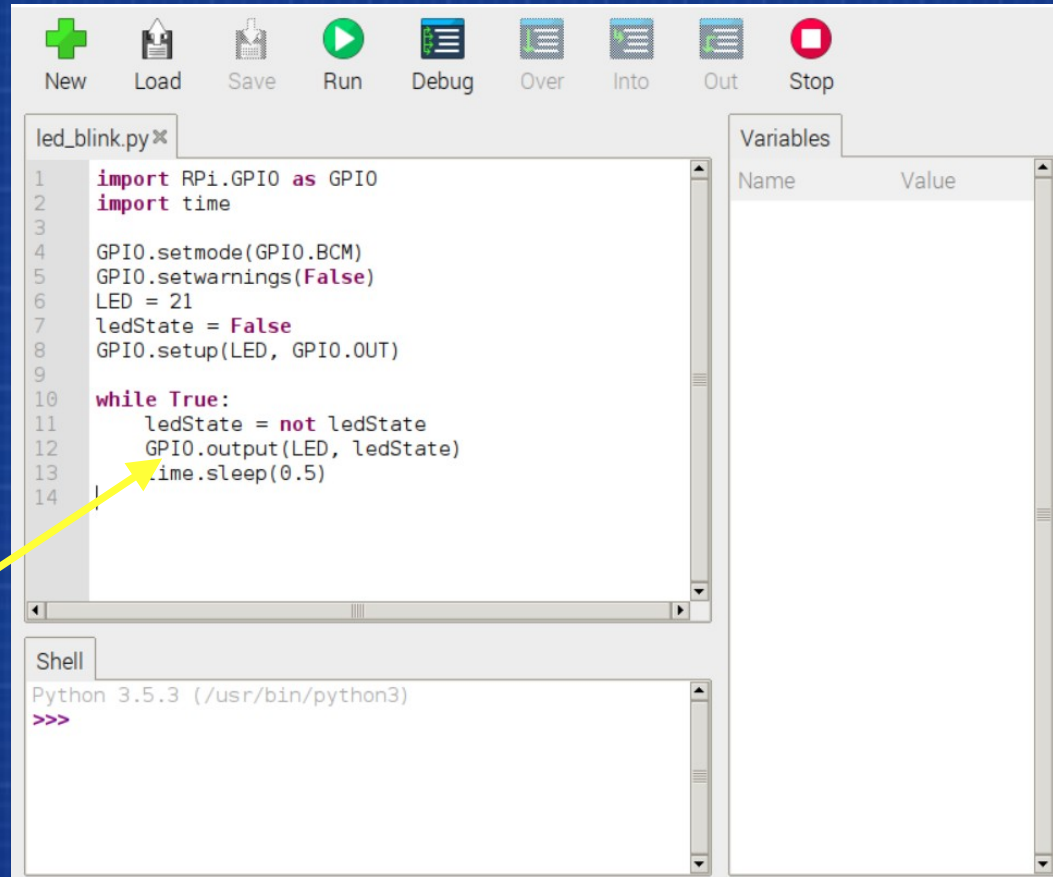
```
ma_variable = xxxxx
```

Conteneurs

List, dictionnaire, string,
tuple ,etc....

Imports

Time, Gpio, sys, math



The screenshot shows a Python IDE with a toolbar at the top containing icons for New, Load, Save, Run, Debug, Over, Into, Out, and Stop. The main editor window displays a file named 'led_blink.py' with the following code:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 LED = 21
7 ledState = False
8 GPIO.setup(LED, GPIO.OUT)
9
10 while True:
11     ledState = not ledState
12     GPIO.output(LED, ledState)
13     time.sleep(0.5)
14
```

A yellow arrow points to the indentation of the while loop body (lines 11-13). To the right of the editor is a 'Variables' panel with columns for 'Name' and 'Value'. At the bottom is a 'Shell' panel showing the Python 3.5.3 prompt.

Indentation
(décalage)

Etapes pour configurer et gérer les GPIOs :

- importer la bibliothèque GPIO
- définir le mode de numérotation des broches GPIO (BOARD ou BCM)
- définir le mode de fonctionnement de la broche (en entrée ou en sortie)
- activer la fonction (en sortie : etat haut/bas, en entrée : acquisition de l'état d'entrée)

importer la bibliothèque `GPIO`

```
Import RPi.GPIO as GPIO # on importe la bibliothèque GPIO et l'on créer un objet GPIO
```

définir le mode de numérotation des broches `GPIO`

```
GPIO.setmode(GPIO.BCM) #on définit la numérotation employée pour les broches
```

BCM : nom du `GPIO`

BOARD : position de la broche sur le board

```
GPIO.setwarnings(False) #On désactive les messages d'alerte
```

définir le mode de fonctionnement de la broche

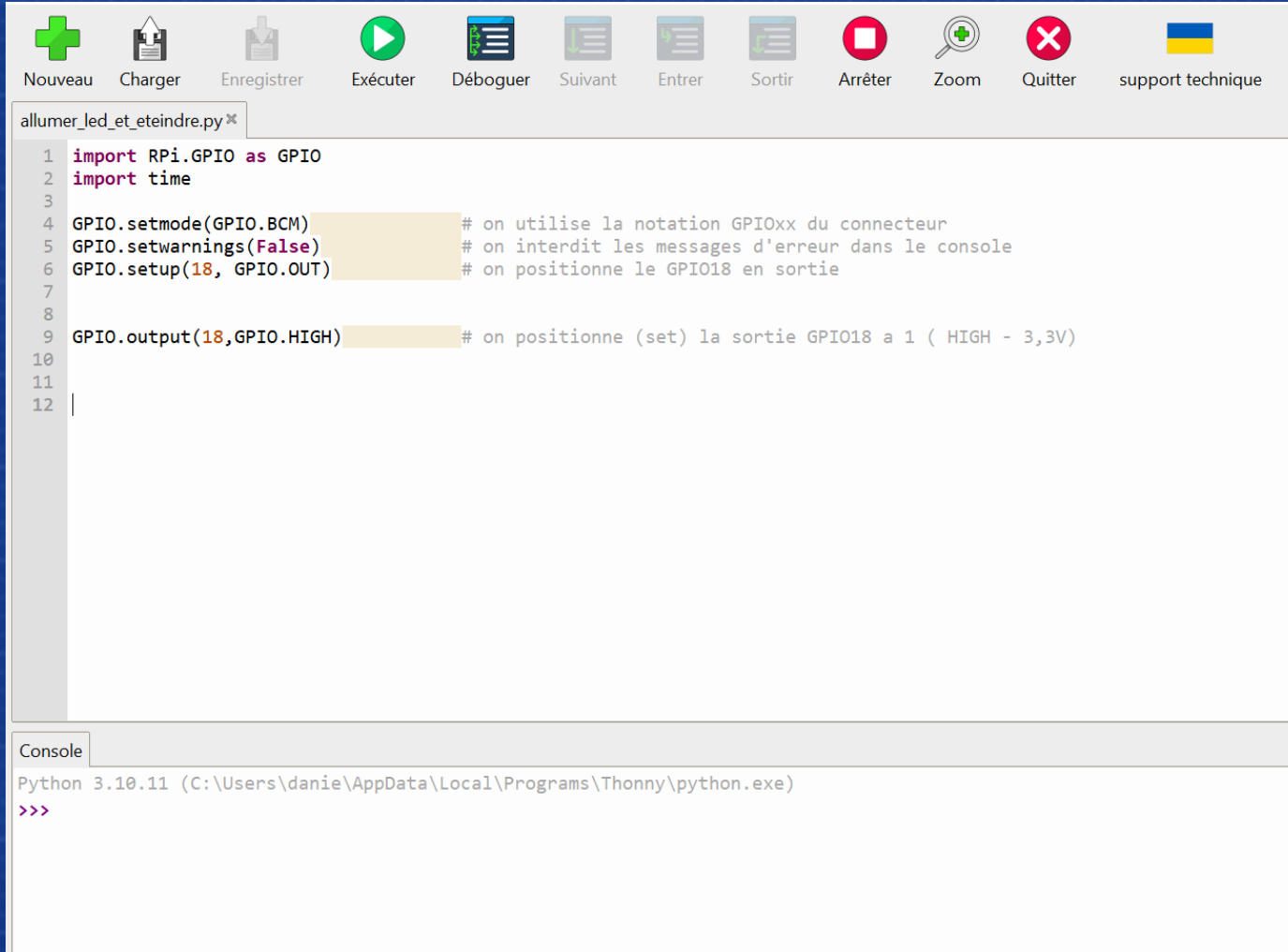
```
GPIO.setup(21,GPIO.OUT) # on déclare la pin 21 comme une sortie #sortie:OUT, entré:IN
```

```
GPIO.output (21.GPIO.LOW) # on met la pin 21 a l'état bas = 0 (0v) #LOW : 0, HIGH : 1
```

```
GPIO.setup(led, GPIO.OUT, initial=0) # permet d'initialiser à 0 ou 1 la sortie
```

activer la fonction

```
GPIO.output (21.GPIO.HIGH) #on met la broche 21 a l'état haut = 1 (3,3v)
```

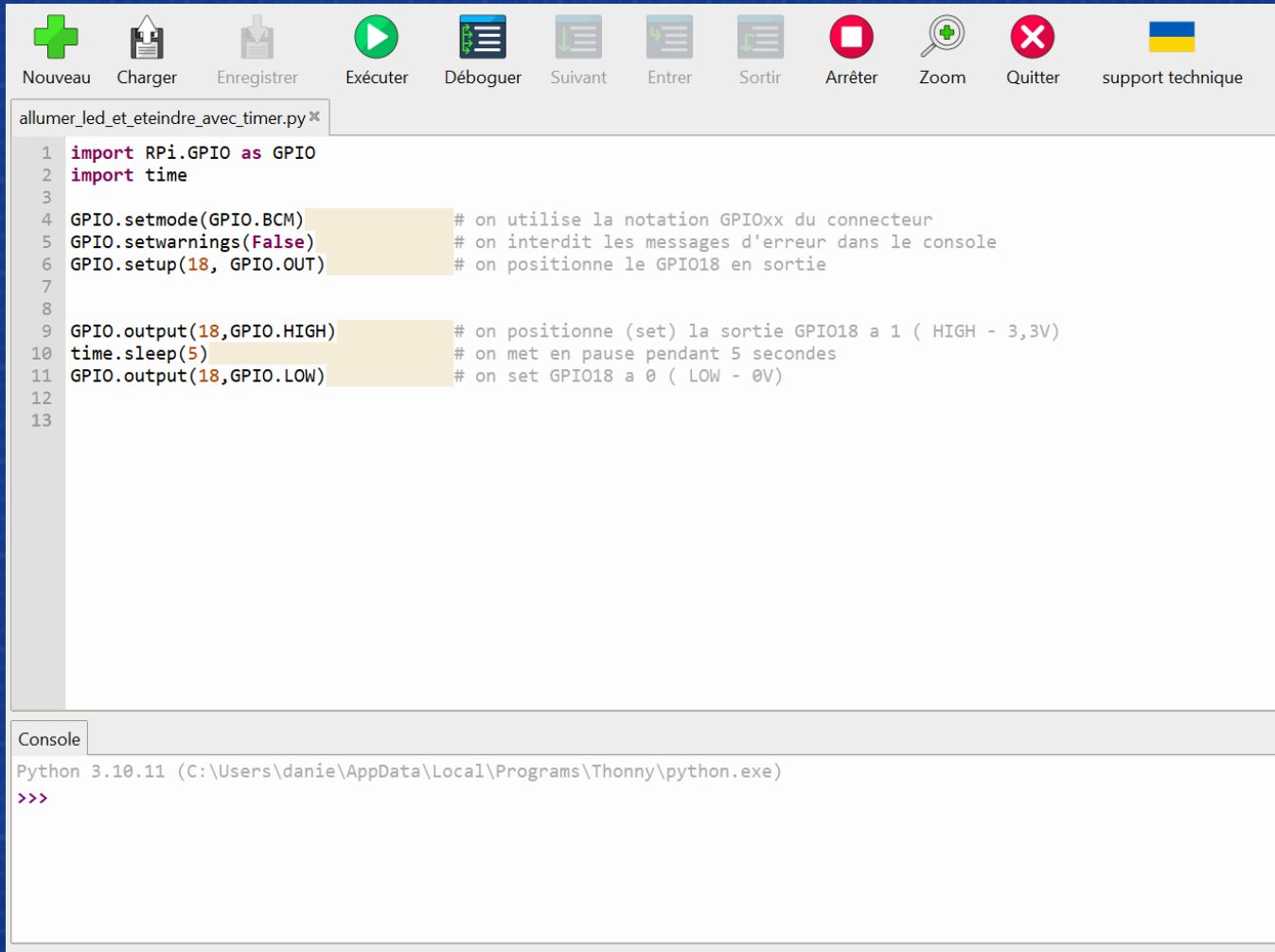


The screenshot shows the Thonny Python IDE interface. At the top, there is a menu bar with icons and labels: Nouveau, Charger, Enregistrer, Exécuter, Débuguer, Suivant, Entrer, Sortir, Arrêter, Zoom, Quitter, and support technique. Below the menu bar is a text editor window titled 'allumer_led_et_eteindre.py' containing the following Python code:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM) # on utilise la notation GPIOxx du connecteur
5 GPIO.setwarnings(False) # on interdit les messages d'erreur dans le console
6 GPIO.setup(18, GPIO.OUT) # on positionne le GPIO18 en sortie
7
8
9 GPIO.output(18,GPIO.HIGH) # on positionne (set) la sortie GPIO18 a 1 ( HIGH - 3,3V)
10
11
12 |
```

At the bottom of the IDE is a console window titled 'Console' showing the Python version and the current prompt:

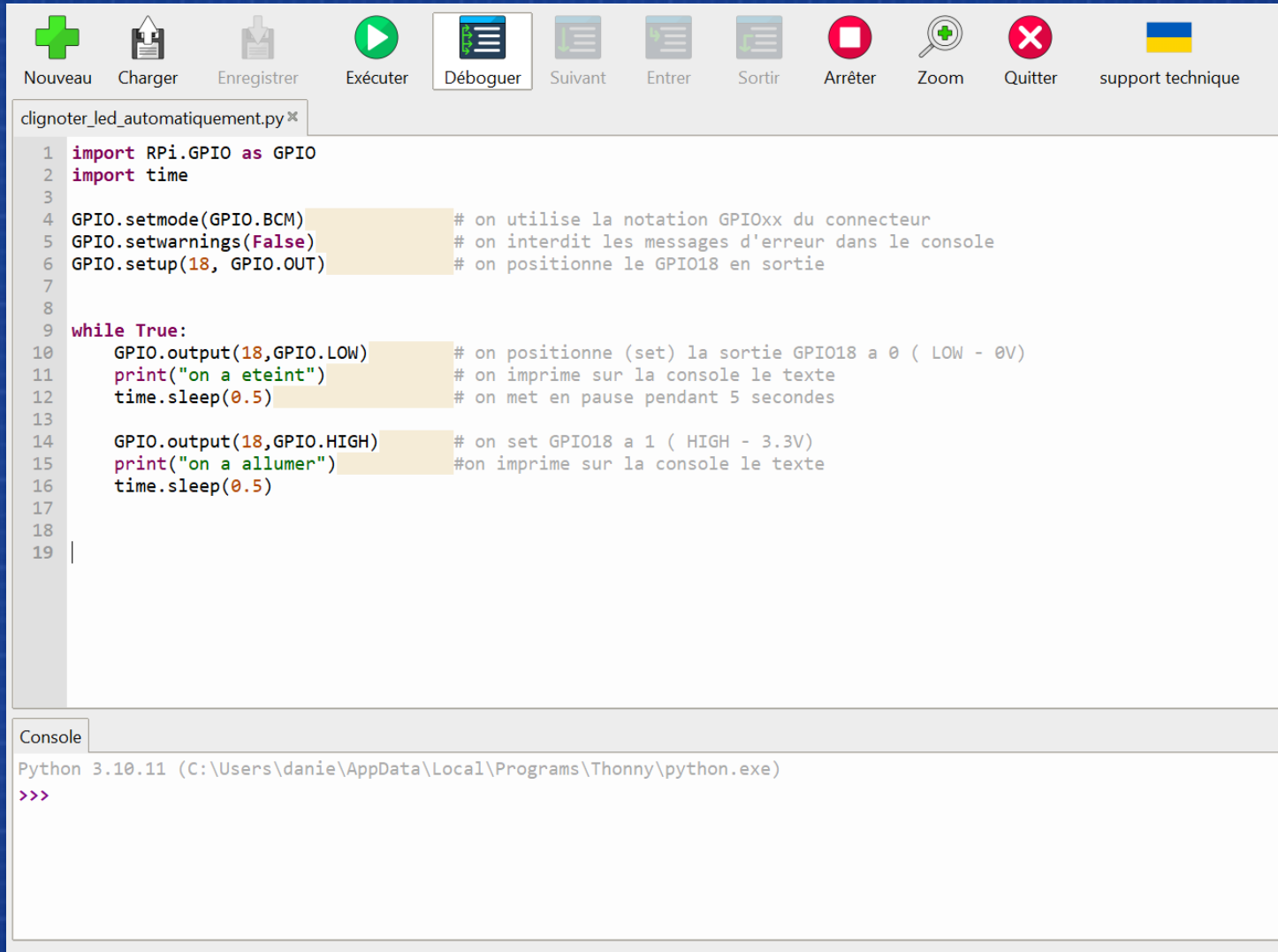
```
Python 3.10.11 (C:\Users\danie\AppData\Local\Programs\Thonny\python.exe)
>>>
```



The screenshot displays the Thonny IDE interface. At the top, there is a toolbar with icons for 'Nouveau', 'Charger', 'Enregistrer', 'Exécuter', 'Débuguer', 'Suivant', 'Entrer', 'Sortir', 'Arrêter', 'Zoom', 'Quitter', and 'support technique'. Below the toolbar, the main editor window shows a Python script named 'allumer_led_et_eteindre_avec_timer.py'. The script contains the following code:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM) # on utilise la notation GPIOxx du connecteur
5 GPIO.setwarnings(False) # on interdit les messages d'erreur dans le console
6 GPIO.setup(18, GPIO.OUT) # on positionne le GPIO18 en sortie
7
8
9 GPIO.output(18,GPIO.HIGH) # on positionne (set) la sortie GPIO18 a 1 ( HIGH - 3,3V)
10 time.sleep(5) # on met en pause pendant 5 secondes
11 GPIO.output(18,GPIO.LOW) # on set GPIO18 a 0 ( LOW - 0V)
12
13
```

Below the editor, the 'Console' window shows the Python version and path: 'Python 3.10.11 (C:\Users\danie\AppData\Local\Programs\Thonny\python.exe)'. The prompt '>>>' is visible, indicating the program has been executed.



The image shows the Thonny IDE interface. The top toolbar contains icons for 'Nouveau', 'Charger', 'Enregistrer', 'Exécuter', 'Débuguer', 'Suivant', 'Entrer', 'Sortir', 'Arrêter', 'Zoom', 'Quitter', and 'support technique'. The main editor window displays a Python script named 'clignoter_led_automatiquement.py' with the following code:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM) # on utilise la notation GPIOxx du connecteur
5 GPIO.setwarnings(False) # on interdit les messages d'erreur dans le console
6 GPIO.setup(18, GPIO.OUT) # on positionne le GPIO18 en sortie
7
8
9 while True:
10     GPIO.output(18,GPIO.LOW) # on positionne (set) la sortie GPIO18 a 0 ( LOW - 0V)
11     print("on a eteint") # on imprime sur la console le texte
12     time.sleep(0.5) # on met en pause pendant 5 secondes
13
14     GPIO.output(18,GPIO.HIGH) # on set GPIO18 a 1 ( HIGH - 3.3V)
15     print("on a allumer") #on imprime sur la console le texte
16     time.sleep(0.5)
17
18
19 |
```

The bottom console window shows the following output:

```
Python 3.10.11 (C:\Users\danie\AppData\Local\Programs\Thonny\python.exe)
>>>
```