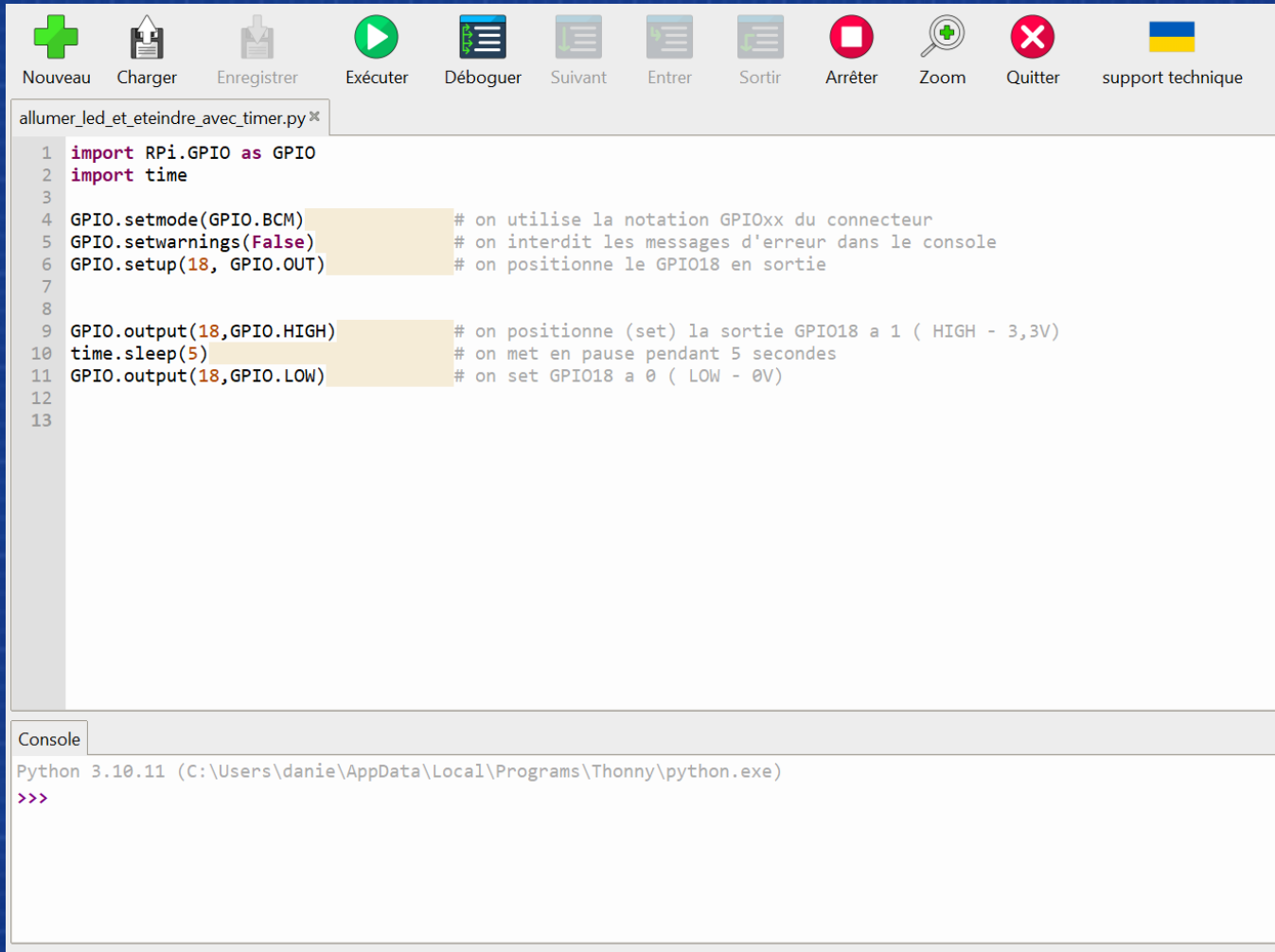


The screenshot shows the Thonny Python IDE interface. At the top, there is a menu bar with icons and labels: Nouveau, Charger, Enregistrer, Exécuter, Débuguer, Suivant, Entrer, Sortir, Arrêter, Zoom, Quitter, and support technique. Below the menu bar is a text editor window titled 'allumer_led_et_eteindre.py'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

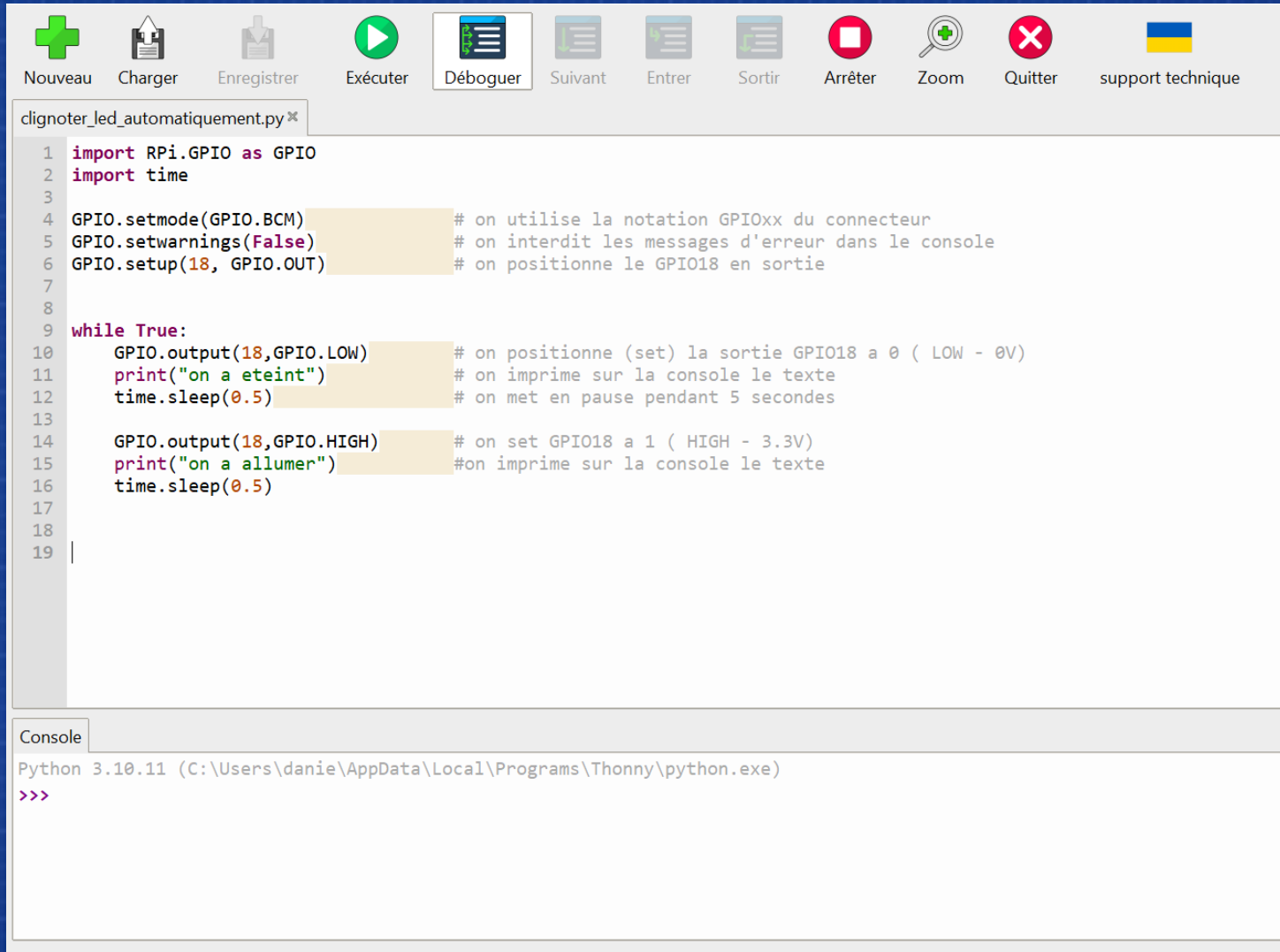
```
Python 3.10.11 (C:\Users\danie\AppData\Local\Programs\Thonny\python.exe)
>>>
```



The screenshot shows the Thonny IDE interface. At the top, there is a toolbar with icons for: Nouveau (New), Charger (Load), Enregistrer (Save), Exécuter (Run), Débugger (Debug), Suivant (Next), Entrer (Enter), Sortir (Exit), Arrêter (Stop), Zoom, Quitter (Quit), and support technique (Technical support). Below the toolbar, the file name 'allumer_led_et_eteindre_avec_timer.py' is displayed. The main editor area contains 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 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
```

At the bottom, the Console window shows the Python version and path: 'Python 3.10.11 (C:\Users\danie\AppData\Local\Programs\Thonny\python.exe)' and the prompt '>>>'.



The image shows 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 displays a Python script named 'clignoter_led_automatiquement.py'. The script is as follows:

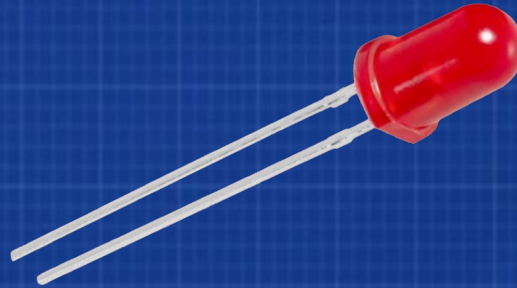
```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM) # on utilise la notation GPIOxx du connecteur
5 GPIO.setwarnings(False) # on interdit les messages d'erreur dans le console
6 GPIO.setup(18, GPIO.OUT) # on positionne le GPIO18 en sortie
7
8
9 while True:
10     GPIO.output(18,GPIO.LOW)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

Below the editor, the 'Console' window shows the following output:

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

Le Raspberry Pi

#2 Allumer/éteindre une LED commander par un bouton poussoir



Programmation d'une entrée

définir le mode de fonctionnement de la broche

```
GPIO.setup(23,GPIO.IN) # on déclare la pin 23 comme une entrée
```

```
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) # on affecte une résistace de pull-down , résistance entre l'entrée du signal et la masse)
```

#Une méthode d'acquisition de l'état de l'entrée :

Créer une variable

```
etat_entre_23 == 0 # variable crée et mise a l'état 0
```

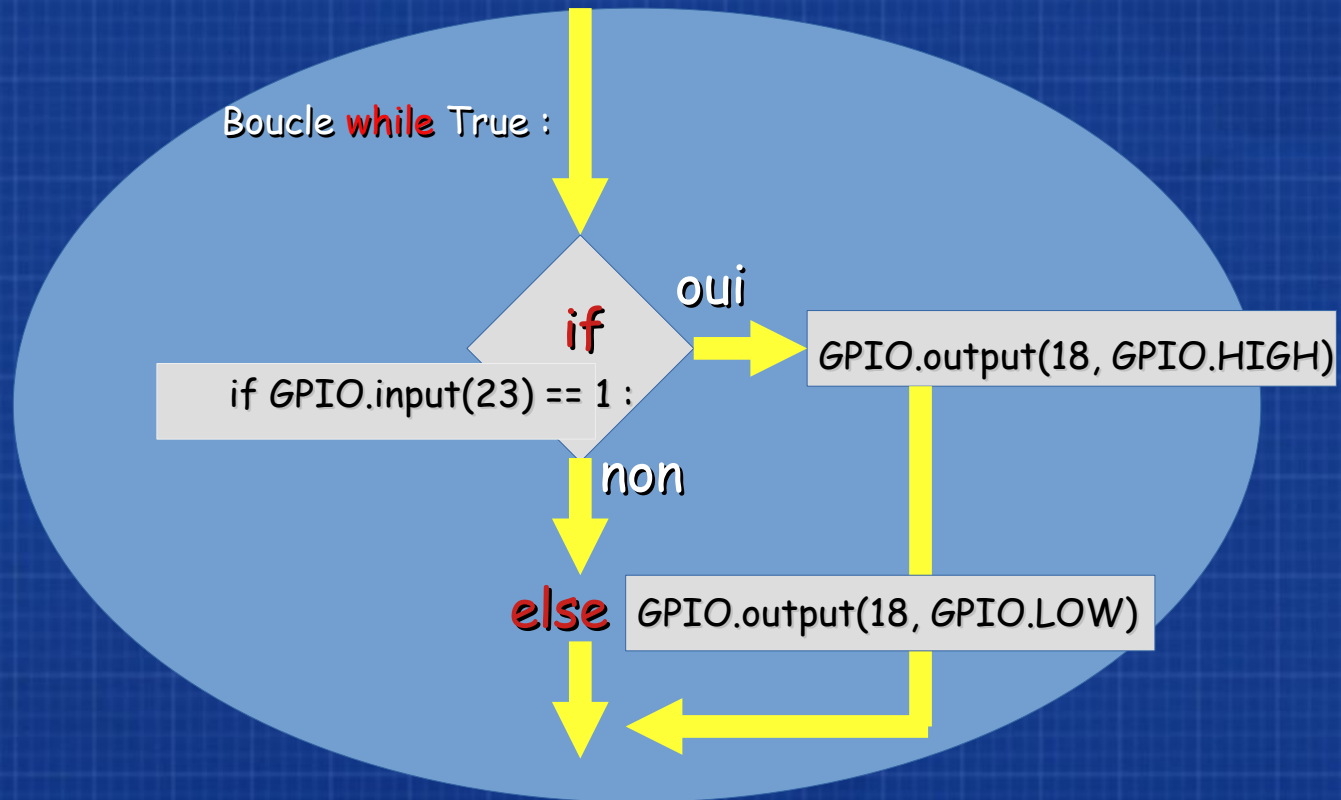
Acquisition de la valeur de l'entée

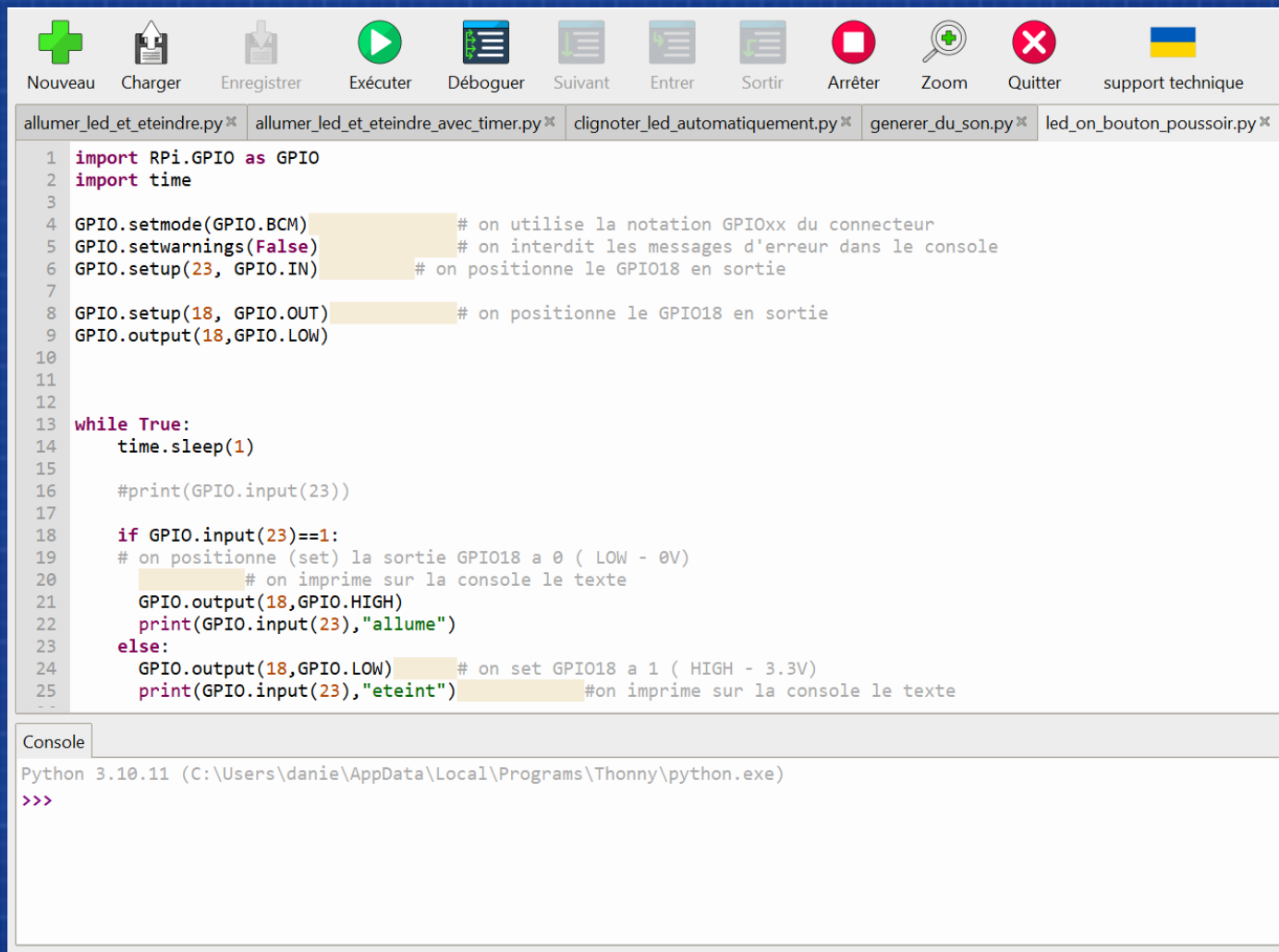
```
GPIO.input(23) == etat_entre_23 # lecture de l'état de l'entrée et enregistrement dans la variable
```

Le Raspberry Pi

Programmation

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18, GPIO.OUT)
GPIO.setup(23, GPIO.IN)
GPIO.output(18, GPIO.LOW)
```





The image shows a Python IDE window with a toolbar at the top and a code editor below. The toolbar contains icons for 'Nouveau', 'Charger', 'Enregistrer', 'Exécuter', 'Débuguer', 'Suivant', 'Entrer', 'Sortir', 'Arrêter', 'Zoom', 'Quitter', and 'support technique'. The code editor displays a Python script for controlling an LED with a push button. The script imports the RPi.GPIO and time modules. It configures GPIO pins 23 and 18. A while loop checks the state of pin 23 and sets pin 18 accordingly, printing the state to the console. The console output shows the Python version and the prompt '>>>'.

```
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(23, GPIO.IN) # on positionne le GPIO18 en sortie
7
8 GPIO.setup(18, GPIO.OUT) # on positionne le GPIO18 en sortie
9 GPIO.output(18,GPIO.LOW)
10
11
12
13 while True:
14     time.sleep(1)
15
16     #print(GPIO.input(23))
17
18     if GPIO.input(23)==1:
19         # on positionne (set) la sortie GPIO18 a 0 ( LOW - 0V)
20         # on imprime sur la console le texte
21         GPIO.output(18,GPIO.HIGH)
22         print(GPIO.input(23),"allume")
23     else:
24         GPIO.output(18,GPIO.LOW) # on set GPIO18 a 1 ( HIGH - 3.3V)
25         print(GPIO.input(23),"eteint") #on imprime sur la console le texte
--
```

Console

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