

## 7 GÉRER LES LED

Maintenant notre capteur de température et notre diagramme à barres LED sont en place. Associons-les pour afficher la température sur le graphique à barres LED. Dans IDLE, créez un nouveau fichier, entrez le code de ListingTemp3.py et enregistrez-le. En haut, nous importons les classes MCP3008 et LEDBarGraph de GPIO Zero, ainsi que la fonction de sommeil de la bibliothèque de temps. Nous définissons ensuite une

fonction pour convertir les lectures du capteur de température en degrés Celsius. Nous attribuons la variable `adc` à la chaîne 7 du MCP3008 et au graphique aux broches GPIO de nos DEL, ce qui permet de définir PWM sur `true`.

## 8 AJOUTER DES VARIABLES

Enfin, dans notre boucle `for`, nous ajoutons une variable `bars` pour déterminer le nombre de LED allumées dans le graphique à barres. Dans cet exemple, nous avons divisé la température par 40, ce qui correspond à la température maximale sur île de France, donc, si elle atteint 40° C, toutes les LED doivent s'allumer complètement. Naturellement, vous pouvez ajuster ce chiffre en fonction du climat de votre propre localisation. Lorsque vous êtes prêt, exécutez le code et voyez ces voyants s'allumer pour afficher la température actuelle.

### CODE PYTHON

#### ListingTemp1.py

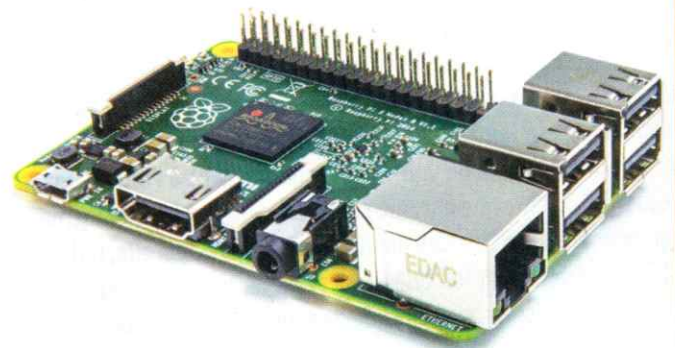
```
from gpiozero import MCP3008
from time import sleep
def convert_temp(gen):
    for value in gen:
        yield (value * 3.3 - 0.5) * 100
adc = MCP3008(channel=7)
for temp in convert_temp(adc.values):
    print(«The temperature is», temp, «C») sleep(1) c
```

#### ListingTemp2.py

```
from gpiozero import LEDBarGraph
from time import
sleep graph = LEDBarGraph (26, 19, 13, 6, 5, pwm=True)
```

```
graph.value = 1/10 sleep(1)
```

```
graph.value = 3/10 sleep(1)
```



```
graph.value = -3/10 sleep(1)
```

```
graph.value = 9/10 sleep(1)
```

```
graph.value = 95/100 sleep(1)
```

```
graph.value = 0
```

#### ListingTemp3.py

```
from gpiozero import MCP3008, LEDBarGraph
from time import sleep
def convert_temp(gen): for value in gen: yield (value * 3.3 -
0.5) * 100
adc = MCP3008(channel=7)
graph = LEDBarGraph (26, 19, 13, 6, 5, pwm=True) for temp in
convert_temp(adc.values):
    bars = temp / 40
    graph.value = bars
    sleep(1)
```