

Commander le GPIO du Raspberry Pi en mode texte

En utilisant les commandes ci-dessus, écrivez un script faisant clignoter une LED connectée au GPIO 20. Ouvrez l'éditeur de texte nano :

```
pi@raspberrypi ~ $ nano led_clignote.sh
```

Saisissez le script destiné à faire clignoter une LED branchée sur le GPIO20.

```
#!/bin/bash
# Script led_clignote.sh
# Clignotement d'une LED en ligne de commande
# Ce script ne comporte pas de test destiné à l'interrompre
# Il s'arrête en appuyant sur CTRL C

# Effacer l'écran
clear

# Rendre le répertoire /sys/class/gpio actif
cd /sys/class/gpio

# Créer l'accès au port GPIO 20
# Ne rien faire s'il existe déjà
if [ -d "gpio20" ]; then
    echo "gpio20 existe déjà"
else
    echo "gpio20 : Création"
    echo 20 > export
fi

# Rendre le répertoire gpio20 actif
cd gpio20/

# Configurer le port GPIO 20 en sortie
echo out > direction

# Eteindre la LED
echo 0 > value

# Initialiser la variable x à 1
x="1"

# Boucle while : cette boucle s'exécute tant que
# la condition est vérifiée
# ici la condition est toujours vraie :
# la boucle ne s'interrompt jamais
# Il faudra en sortir avec un CTRL C

while [ $x -gt 0 ]
# Le bloc inclus entre do et done est exécuté
# par la boucle while

# Début du bloc d'instructions
do
    # L'instruction echo envoie la valeur 1 dans le fichier value
    # Ceci a pour effet d'allumer la LED
    echo 1 > value

    # Un message sur l'écran informe l'utilisateur
    # de l'allumage de la LED
    echo "LED allumée"
```

```
# La commande sleep suspend l'exécution du programme
# pendant la durée spécifiée (en secondes)

sleep 1

# L'instruction echo envoie la valeur 0 dans le fichier value
# Ceci a pour effet d'éteindre la LED
echo 0 > value

# Un message sur l'écran informe l'utilisateur
# de l'allumage de la LED
echo "LED éteinte"

# La commande sleep suspend l'exécution du programme
# pendant la durée spécifiée (en secondes)
sleep 1

# Fin du bloc d'instructions
done
```