

## Compteur binaire de 0 à 3 avec 2 LED

Cahier des charges : Le programme attend un appui sur le bouton poussoir et affiche les valeurs binaires 0, 1, 2 et 3 grâce à 2 LED. La LED de poids fort est connectée au GPIO20 et la LED de poids faible est connectée au GPIO23.

Ouvrez l'éditeur de texte nano :

```
pi@raspberrypi ~ $ nano cde_LED_06.sh
```

Saisissez le script destiné à réaliser un compteur binaire

```
#!/bin/bash
# Script cde_LED_06.sh
# Lire l'état d'un bouton poussoir
# Et commander un compteur binaire
# LED poids faible en GPIO23
# LED poids fort en GPIO20
# 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

##### Configuration port GPIO du bouton poussoir

# Créer l'accès au port GPIO 16
# Pour lire l'état du bouton poussoir
# Ne rien faire s'il existe déjà
if [ -d "gpio16" ]; then
    echo "gpio16 existe déjà"
else
    echo "gpio16 : Création"
    echo 16 > export
fi

# Rendre le répertoire gpio16 actif
cd gpio16/

# Configurer le port GPIO 16 en entrée
# Normalement il y est par défaut... mais on ne sait pas ce qui a pu se
passer avant
echo in > direction

##### Configuration port GPIO de la LED1

# revenir dans le dossier /sys/class/gpio
cd ..
```

```
# Créer 1 accès au port GPIO 20
# Pour commander la LED
# 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 connectée au GPIO20
echo 0 > /sys/class/gpio/gpio20/value

##### Configuration port GPIO de la LED2

# revenir dans le dossier /sys/class/gpio
cd ..

# Créer 1 accès au port GPIO 23
# Pour commander la LED
# Ne rien faire s il existe déjà
if [ -d "gpio23" ]; then
    echo "gpio23 existe déjà"
else
    echo "gpio23 : Création"
    echo 23 > export
fi

# Rendre le répertoire gpio23 actif
cd gpio23/

# Configurer le port GPIO 23 en sortie
echo out > direction

# Eteindre la LED connectée au GPIO23
echo 0 > /sys/class/gpio/gpio23/value

##### Boucle de scrutation du port GPIO 16

# Initialiser la variable x à 1 -> pour une boucle infinie
x="1"

# Initialiser la variable compteur à 0
compteur="0"

# L instruction cat affiche le contenu d un fichier à l écran
```

```

# Le fichier value contient l'état du bouton poussoir
z="$(cat /sys/class/gpio/gpio16/value)"

# 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
# c est la boucle principale

# Début du bloc d'instructions
do

# Allumer les LED en fonction de la valeur du compteur
case $compteur in
0)
echo 0 > /sys/class/gpio/gpio20/value
echo 0 > /sys/class/gpio/gpio23/value ;;

1)
echo 0 > /sys/class/gpio/gpio20/value
echo 1 > /sys/class/gpio/gpio23/value ;;

2)
echo 1 > /sys/class/gpio/gpio20/value
echo 0 > /sys/class/gpio/gpio23/value ;;

3)
echo 1 > /sys/class/gpio/gpio20/value
echo 1 > /sys/class/gpio/gpio23/value ;;
esac

# Attendre un appui/relâchement du bouton poussoir
# Lire l'état du bouton poussoir
z="$(cat /sys/class/gpio/gpio16/value)"

# Boucler tant que le bouton poussoir n'est pas appuyé
while [ $z -eq 0 ]
# Le bloc inclus entre do et done est exécuté
# par la boucle while

# Début du bloc d'instructions
do
# On lit l'état du bouton poussoir
z="$(cat /sys/class/gpio/gpio16/value)"
# le bouton poussoir a été appuyé, on peut continuer
done

# Boucler tant que le bouton poussoir n'est pas relâché

```

```
while [ $z -eq 1 ]
do
# on lit l etat du bouton poussoir
z="$(cat /sys/class/gpio/gpio16/value)"
# le bouton poussoir est relâché, on peut continuer
done

# Incrémenter le compteur
compteur=$((compteur+1))

# Vérifier le compteur
if [ "$compteur" -eq "4" ]; then
    let "compteur = 0"
fi

echo "Compteur : " $compteur

# Fin du bloc d instructions
done
```