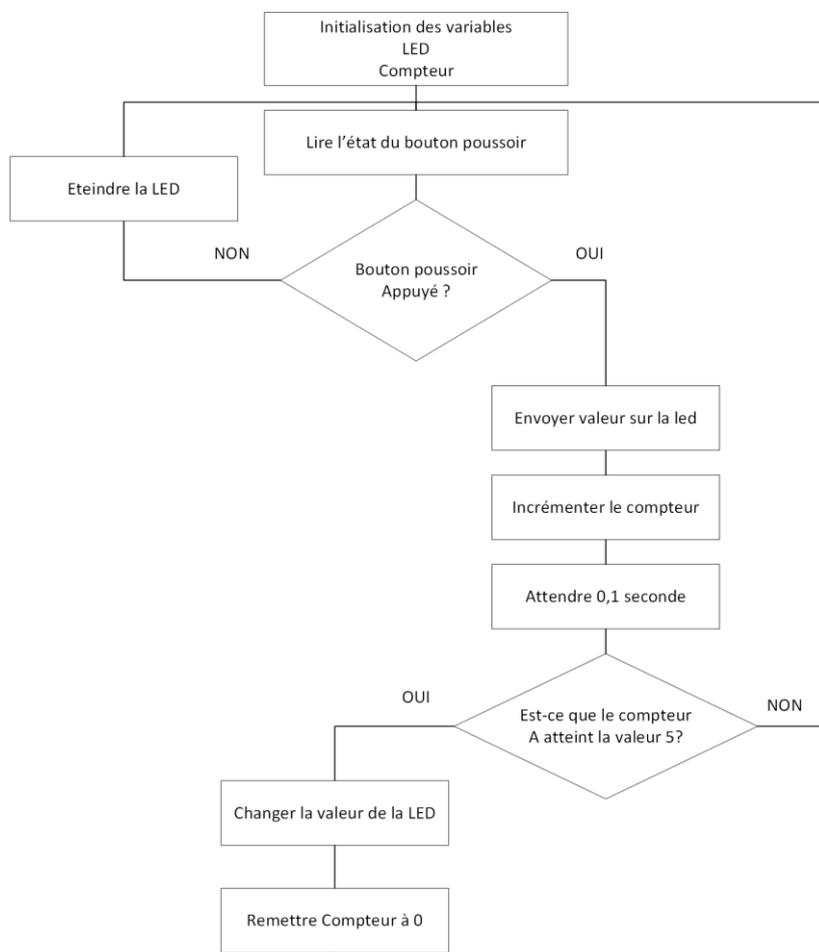


## Commander la LED avec un bouton poussoir

Cahier des charges : Lorsqu'on appuie sur le bouton poussoir, la LED clignote une fois par seconde (0,5 seconde allumée, 0,5 seconde éteinte). Elle clignote tant qu'on appuie sur le bouton poussoir. Elle s'éteint quand on relâche le bouton poussoir.

Organigramme :



Ouvrez l'éditeur de texte nano :

```
pi@raspberrypi ~ $ nano cde_LED_03.sh
```

Saisissez le script destiné à lire l'état d'un bouton poussoir connecté sur le GPIO16 et à commander une LED connectée sur le GPIO 20.

```
#!/bin/bash
# Script cde_LED_03.sh
# Lire l'état d'un bouton poussoir
# Et commander une LED
# 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 LED

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

# Créer l'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
echo 0 > value

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

# Initialiser la variable x à 1 pour la boucle principale
x="1"
# initialiser le compteur
cpt="0"

# initialiser LED allumée
led="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
# c est la boucle principale

# Début du bloc d instructions
do
    #echo "Boucle 1 démarre"
    z="$(cat /sys/class/gpio/gpio16/value)"

    while [ $z -eq 1 ]
    # 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)"

        # on teste si le bouton poussoir est appuyé
        # s il est appuyé la LED clignote, sinon on peut revenir au début
        if [ "$z" -eq "1" ]; then
            # Si le bouton est appuyé on allume la LED
            #echo "le bouton est appuyé"
            echo $led > /sys/class/gpio/gpio20/value
            # On incrémente le compteur de 1/10 eme de secondes
            cpt=$((cpt+1))
            #echo "compteur"
            #echo $cpt
            # on attend 0,1 seconde
            sleep 0.1

            # est-ce que le compteur est arrivé à 5 ?
            if [ "$cpt" -gt "4" ]; then
                # on va changer l état de la LED
                # lire l état de la LED
                led="$(cat /sys/class/gpio/gpio20/value)"
                # inverser l état de la LED (toggle)
                led=$((1-led))
                echo $led > /sys/class/gpio/gpio20/value
                # et remettre le compteur à 0
                let "cpt = 0"

                # avant de finir, on relit l état du bouton poussoir
                z="$(cat /sys/class/gpio/gpio16/value)"
                # echo "boucle clignote"
            else
                # si le compteur n est pas arrivé à 5, on lit simplement l état du
                bouton poussoir
                z="$(cat /sys/class/gpio/gpio16/value)"
                #echo "boucle clignote pas"
            fi

            # Fin du premier if
            fi

        # Fin du bloc d instructions
    done
    # Le bouton n'est plus appuyé, on éteint la LED
    let "led = 0"
    echo $led > /sys/class/gpio/gpio20/value

```

```
# Fin du bloc d instructions  
done
```