



# Raspberry Pi (5) - gestion des INPUT -

## Gestion des INPUT 3 : bouton BISTABLE

Nom :	Appréciation :	Note :
Prénom :		
Classe :		
Date :		

Objectif :	durée : 4h
Utilité :	

Matériel : plaque labdec – composants électroniques

Prérequis : Connexion à distance avec SSH, commande GPIO

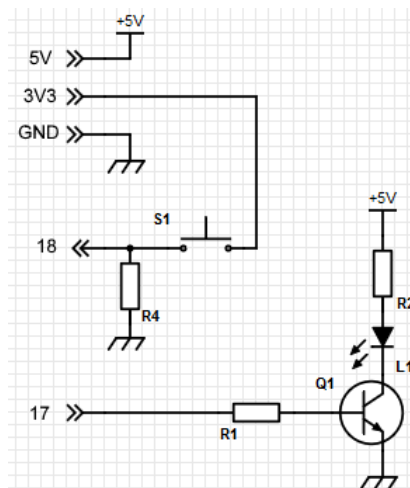
Compétences et savoirs principalement visées :  
C2-1, C2-2 (page 3a), C3-2, C3-3 (page 3b à 6)

Travail à réaliser :

-  
-  
-

Schéma du système :

Pi B+ GPIO Ref		
3.3V	●	5V
2	●	5V
3	●	GND
4	●	14
GND	●	15
17	●	18
27	●	GND
22	●	23
3.3V	●	24
10	●	GND
9	●	25
11	●	8
GND	●	7
IDSD	●	IDSC
5	●	GND
6	●	12
13	●	GND
19	●	16
26	●	20
GND	●	21

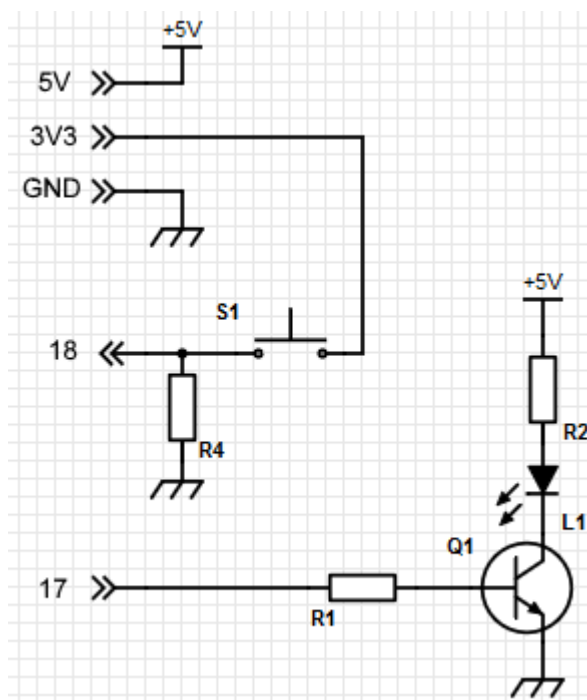


- Réalise le montage suivant sur la plaque Labdec

### AVERTISSEMENT

**Attention** : en cas d'erreur de branchement, ton Raspberry Pi risque d'être **détruit** !!! Ne mets pas le circuit sous tension **avant** que le professeur l'ai vérifié.

Pi B+ GPIO Ref		
3.3V	●	5V
2	●	5V
3	●	GND
4	●	14
GND	●	15
17	●	18
27	●	GND
22	●	23
3.3V	●	24
10	●	GND
9	●	25
11	●	8
GND	●	7
IDSD	●	IDSC
5	●	GND
6	●	12
13	●	GND
19	●	16
26	●	20
GND	●	21

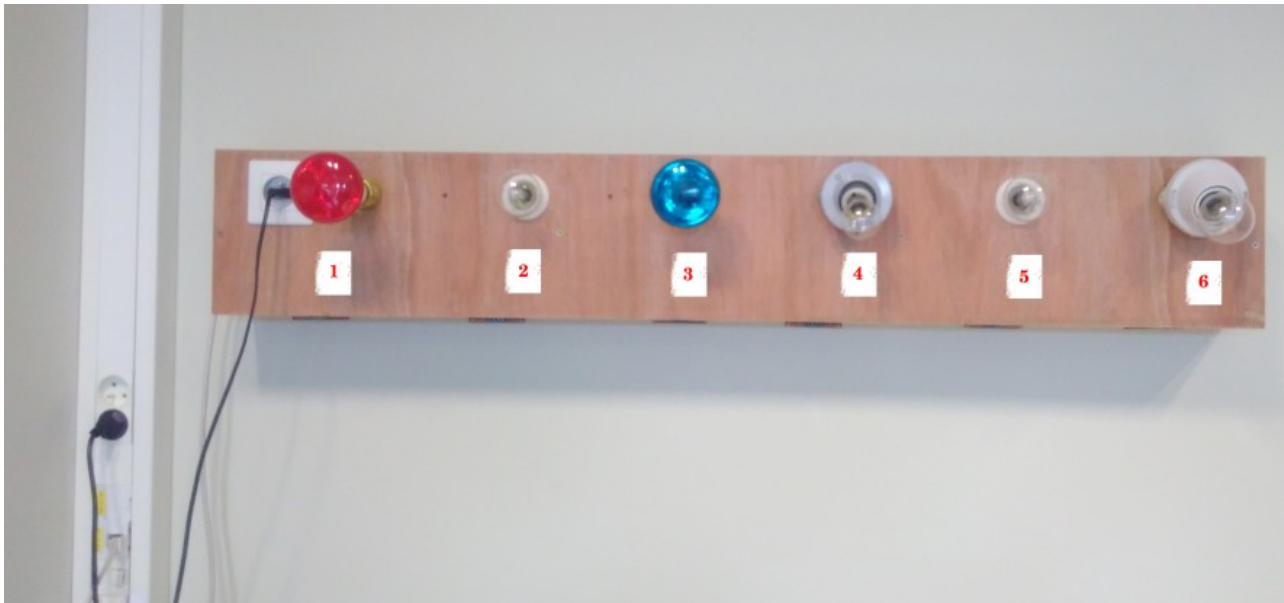


### AVERTISSEMENT

**Attention** : Il ne faudra surtout jamais mettre le signal 18 en mode OUT avec ce circuit ; en cas d'appui sur S1, ton Raspberry Pi serait immédiatement **détruit** !!!

## Câblage du système d'éclairage

Dans cette réalisation, tu ne vas plus commander des LED, mais les lampes d'un système d'éclairage réel, fonctionnant sur le réseau EDF en 220V~



### 1 - Présentation

L'éclairage est alimenté par le réseau EDF en 220V~ ; Voici le schéma de l'installation pour une ampoule :

Notez les 2 zones de ce circuit :

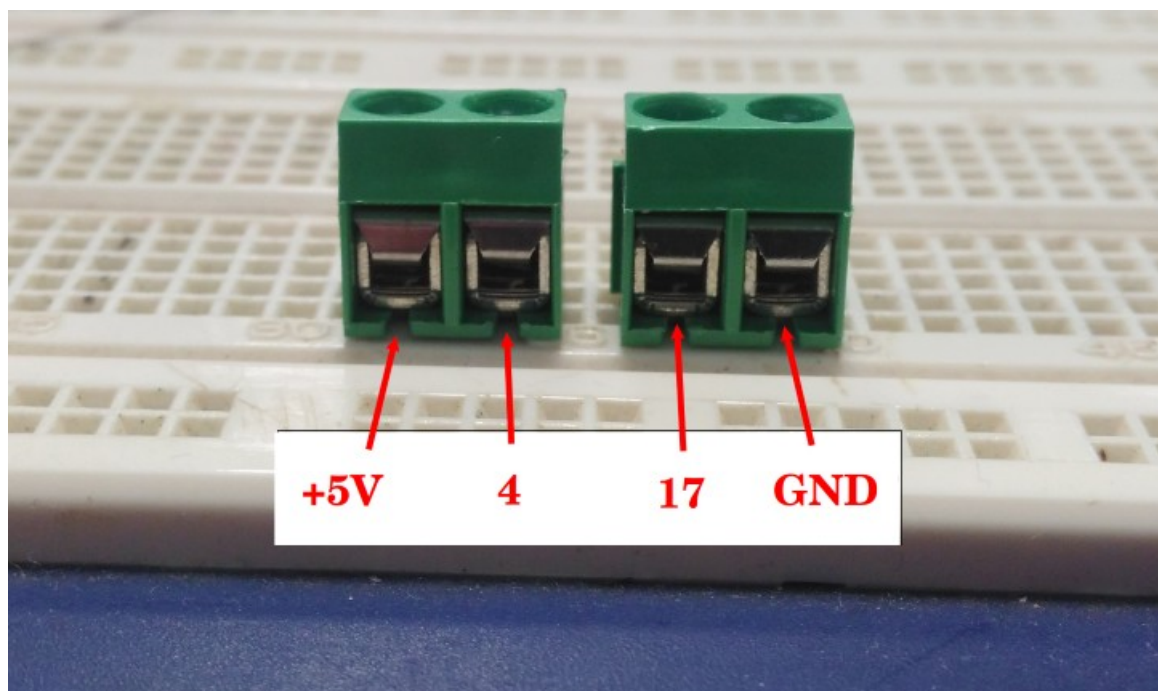
- la zone de puissance en 220V~ (c'est la zone « dangereuse »)
- la zone de commande en 5V=

Dans ce TP vous interviendrez uniquement sur la zone de commande ; **cela signifie-t-il qu'il n'y a pas de danger ? Non !** Car certaines connexions reliées au 220V se trouvent très proches du bornier de connexion de la zone de commande ; Or, si ta main se trouve à **moins de 30 cm** d'une zone dangereuse, on considère que tu es **au « voisinage » de la tension**, et cela nécessite de posséder l'habilitation électrique B1V minimum, ainsi que des précautions comme :

La consignation électrique consiste à couper le courant avant d'intervenir et à veiller à ce qu'il reste coupé

Dans tous les cas, vous interviendrez sur le système d'éclairage obligatoirement en binôme ; **l'un des deux sera chargé de couper le courant et de s'assurer que personne ne le rebranche pendant l'opération.**

## 2 – Connexion au niveau de la plaque LabDec



Tu utiliseras 2 doubles borniers à vis, et 4 fils du Rpi reliés à ces 4 connexions, comme indiqué sur la photo

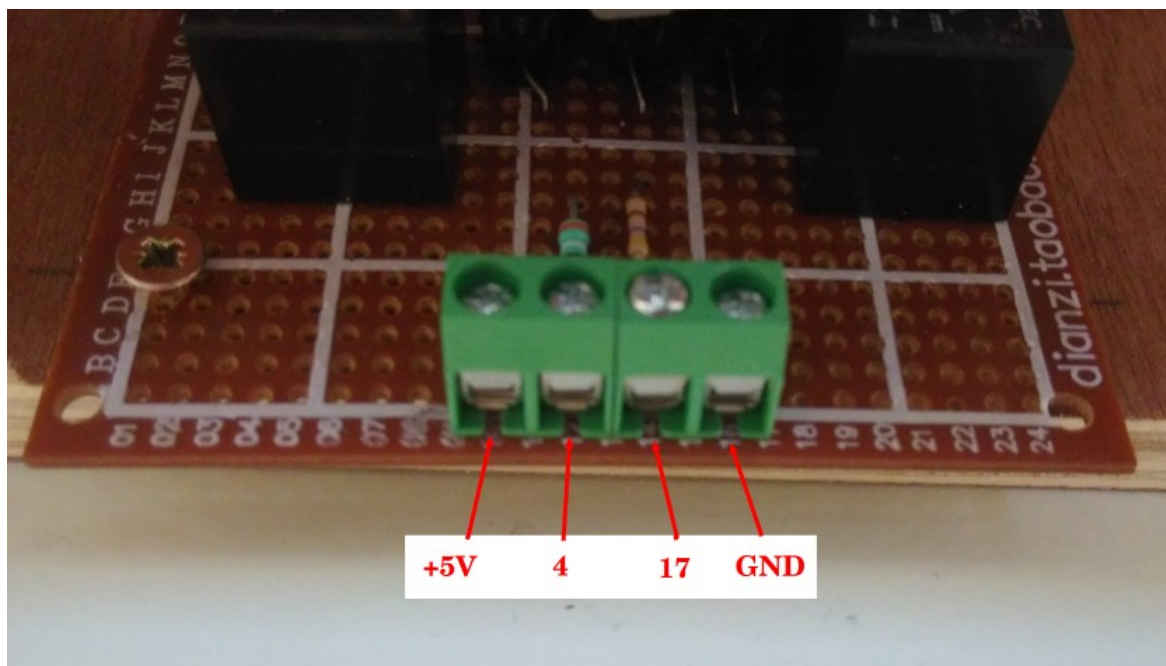
Ensuite tu visseras le câble qui va courir jusqu'au système d'éclairage ; le câble doit être inséré dans la goulotte et ne pas être coupé trop court (laisser 30-40cm de plus de chaque côté)

Repère bien la couleur du fil utilisé sur chacune des 4 connexions :

Connexion	Couleur du fil
+5V	
4	
17	
GND	

Photo du bornier câblé

### 3 – Connexion au système d'éclairage



Recopie ici le tableau de la page précédente =>

Connexion	Couleur du fil
+5V	
4	
17	
GND	

Avant toute intervention sur le système d'éclairage, débranche la prise 220V pour couper le courant !!

Connectes les 4 fils en respectant les couleurs choisies ; ne serres pas trop fort les borniers sinon ils risquent de casser !!

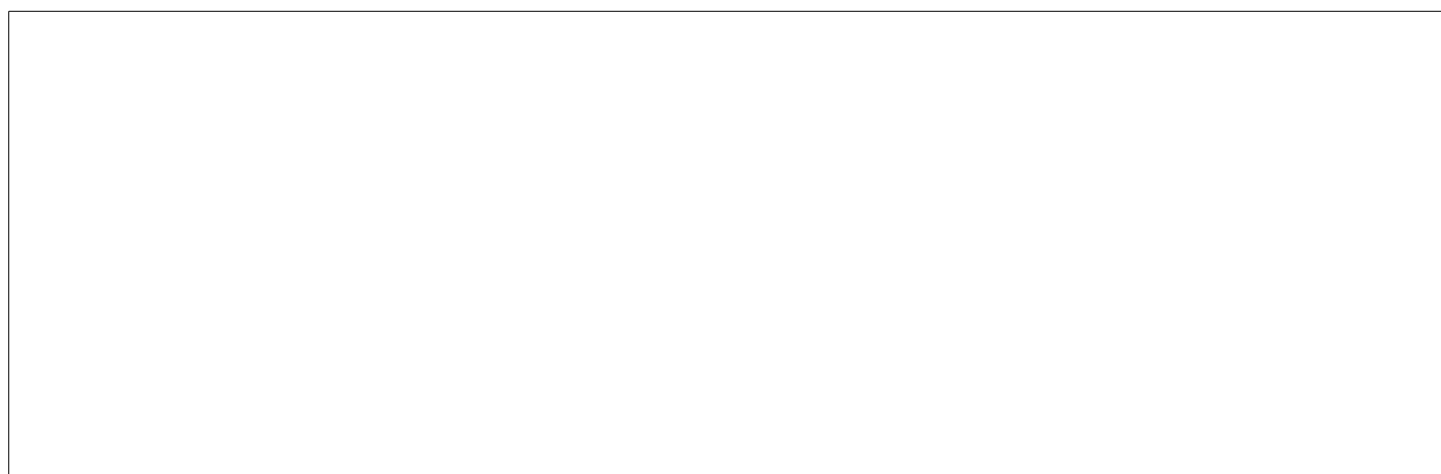


Photo du bornier câblé

## Vérification de fonctionnement du système

Avant de passer à la programmation, il faut vérifier que le câblage est correct, au niveau du système d'éclairage et au niveau du Raspberry Pi

- ◆ Refais une vérification visuelle du câblage (les couleurs sont là pour t'aider)
  - la vérification visuelle est-elle satisfaisante ? \_\_\_\_\_
- ◆ Vérifie ensuite que le montage fonctionne en effectuant la série de commande :

Commande	Résultat attendu *	Résultat obtenu
gpio -g mode 4 out		#
gpio -g mode 17 out		#
gpio -g write 4 1		
gpio -g write 4 0		
gpio -g write 17 1		
gpio -g write 17 0		

\* remplir cette case avant de tester la commande

# vérifie en faisant : « gpio readall »

→ Les résultats sont-ils conformes ? (sinon essaie d'en découvrir la cause)

## Bouton BISTABLE

l'appui sur le bouton BISTABLE provoque l'allumage de la LED; elle s'éteint quand on appui à nouveau sur ce bouton.

- ◆ A partir de WinSCP, crée le fichier «**bistable .sh** » dans le dossier « **/root** »

### Fichier bistable.sh

```
#!/bin/bash

gpio -g mode 17 out
gpio -g mode 18 in
b=0
# la variable $etat prendra les valeurs 0 ou 1 pour "allumer" ou "éteindre"
# la led est initialement éteinte
etat=0
gpio -g write 17 $etat

while [ 1==1 ]
do

# attendre que le bouton S1 soit poussé
while [ $b -eq 0 ]
do
    b=$(gpio -g read 18)
done

# inverser la valeur de $etat
if [ $etat -eq 0 ]
then
    etat=1
else
    etat=0
fi

# éteindre ou allumer la led selon la valeur de $etat
gpio -g write 17 $etat

# attendre que le bouton S1 soit relâché
while [ $b -eq 1 ]
do
    b=$(gpio -g read 18)
done

sleep 0.5

done
```

◆ Dans un terminal Putty connecté au Raspberry Pi, tapes les commandes suivantes :

```
cd /root
```

```
chmod +x bistable.sh
```

◆ Lances ton script avec la commande : `./ bistable .sh` , comment le bouton fonctionne-t-il ?

◆ **BONUS** (*pour les élèves le plus rapides !!*) Modifie le script pour que la LED clignote 2 fois avant de s'allumer, ou de s'éteindre...

**Info** : utilise une durée de 0.2s pour l'état allumé/éteint

Recopie ici ton script modifié :



## Bouton BISTABLE : exemple d'un bouton Play/Pause

Le fonctionnement doit être conforme à ce qui suit :

- l'appui sur le BP provoque la lecture (« Play ») d'un fichier mp3 et en même temps l'allumage de la LED
- Si on appuie à nouveau sur le BP pendant la lecture, celle-ci se met en « Pause » et la LED se met à clignoter
- Quand la lecture est terminée la LED s'éteint

- ◆ Débranche les 4 fils qui vont vers le système d'éclairage
- ◆ Installe le logiciel omxplayer : **apt-get install omxplayer**
- ◆ Télécharge le fichier « mp3 » : wget [http://10.3.14.254/eleve/multimedia/Cruel\\_Summer.mp3](http://10.3.14.254/eleve/multimedia/Cruel_Summer.mp3)
- ◆ A partir de WinSCP, crée le fichier « **mp3clignote.sh** » dans le dossier « **/root** », après l'avoir complété et fait vérifier par le professeur

### Fichier mp3clignote.sh

```
#!/bin/bash

delai=$1

# le signal 17 doit être dans le bon mode
gpio -g mode [REDACTED]

# boucle infinie
while true
do
    # allumer la LED et attendre pendant $delai
    [REDACTED]

    # éteindre la LED et attendre pendant $delai
    [REDACTED]
done
```

- ◆ A partir de WinSCP, crée le fichier « **mp3.sh** » dans le dossier « **/root** », après l'avoir complété et fait vérifier par le professeur

#### Fichier mp3.sh

```
#!/bin/bash
killall omxplayer.bin
killall omxplayer
killall mp3clignote.sh
if [ -e /root/omxplayer.pipe ]; then
    rm /root/omxplayer.pipe
fi
mkfifo /root/omxplayer.pipe

#mettre les signaux 17 et 18 dans le bon mode ("in" ou "out")
gpio -g mode 17 [REDACTED]
gpio -g mode 18 [REDACTED]
b=0
# la LED doit être éteinte
[REDACTED]
# on attend que le bouton play soit appuyé
while [ $b -eq 0 ]
do
    b=$(gpio -g read 18)
done
# attendre que le bouton play soit relâché
while [ $b -eq 1 ]
do
    b=$(gpio -g read 18)
done
# allumer la LED
[REDACTED]
# lancer la lecture
omxplayer -o local "/root/cruel_summer.mp3" < /root/omxplayer.pipe &
echo -n . > /root/omxplayer.pipe &
echo -n "1" > /root/omxplayer.pipe &
on=1
sleep 1

#
while true
do
    # vérifier si le bouton play/pause est poussé
    # en même temps vérifier si la chanson est terminée
    while [ $b -eq 0 ]
    do
        b=$(gpio -g read 18)
        #la lecture est-elle terminée ?
        # si encours == " cela signifie que la lecture est terminée
        encours=`ps ax | grep omxplayer.bin | grep -v grep`
```

```

if [[ $encours == " " ]]; then
  # éteindre la LED
  [redacted]
  # quitter le script
  exit 0
fi
# on met le script en sommeil pendant 100ms pour ne pas surcharger le CPU
sleep 0.1
done

# attendre que le bouton play/pause soit relaché
while [ $b -eq 1 ]
do
  b=$(gpio -g read 18)
done

# mettre en pause ou reprendre la lecture ?
if [ $on -eq 0 ]
then
  on=1
  killall mp3clignote.sh
  # allumer la LED
  [redacted]
  echo -n "p" > /root/omxplayer.pipe &
  echo "Reprise de la lecture..."
else
  on=0
  # on lance mp3clignote avec le paramètre 0.2 (pour 0.2 secondes)
  [redacted] &
  echo -n "p" > /root/omxplayer.pipe &
  echo "Lecture en pause..."
fi
done
exit 0

```

◆ **BONUS 2** (*pour les élèves super rapides !!*) Télécharge le morceau « Pink\_Floyd.mp3 » et modifie le script de façon à ce qu'il soit joué à la place de « Cruel\_Summer.mp3 »

**Recopie ici les lignes modifiées :**

◆ **BONUS 3** (*pour les élèves super rapides !!*) Écris un script « fpush.sh » pour que la LED clignote rapidement quand on appuie sur le bouton et qu'elle s'éteigne quand on le relâche

Recopie ici ton script modifié :